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 5A: "An ninh của các hệ điều hành họ Microsoft Windows, Sun Solaris và Linux"

HÀ NỘI-2002

Báo cáo kết quả nghiên cứu

AN NINH, AN TOÀN CỦA MẠNG MÁY TÍNH

Quyển 5A: "An ninh của các hệ điều hành họ Microsoft Windows, Sun Solaris và Linux"

Chủ trì nhóm thực hiện:

TS. Nguyễn Nam Hải,
ThS. Đặng Hoà,
TS. Trần Duy Lai

MỤC LỤC

2- Bảo vệ vật lý
2.1- Khóa máy tính
2.2- Bảo vệ BIOS
2.3- Bảo vệ trình nạp khởi động (Boot Loader) LILO
2.4- xlock and vlock
2.5- Phát hiện sự thỏa hiệp an toàn vật lý
3-Bảo vệ cục bộ
3.1-Tạo các tài khoản mới
3.2- An toàn Root
4-An toàn file và hệ thống file
4.1- Thiết lập Umask
4.2- Quyền của file
4.3- Kiểm tra tính toàn vẹn của hệ thống file
5-An toàn mật khẩu và sự mã hóa
5.1- PGP và mật mã khóa công khai
5.2-SSL, S-HTTP, HTTP và S/MIME
5.3- Ứng dụng Linux IPSEC
5.4- ssh và stelnet
5.5 PAM - Pluggable Authetication Modules
5.6-Cryptographic IP Encapsulation (CIPE)
5.7- Kerberos
5.8-Shadow Passwords
5.9- "Crack" và "John the Ripper"
5.10-CFS-Cryptograpic File System và TCFS - Transparent Cryptographic File System
5.11- X11, SVGA và bảo vệ màn hình
6-An toàn nhân
6.1-Các tùy chọn cấu hình nhân có liên quan tới an toàn
6.2-Các thiết bị nhân
7- An toàn mạng
7.1- Bộ lắng nghe gói (packet sniffer)
7.2-Các dịch vụ hệ thống và tcp_wrappers
7.3-Kiểm tra thông tin DNS
7.4-identd
7.5- sendmail, qmail
7.6-Tấn công từ chối dịch vụ
7.7-An toàn NFS (Network File System)
7.8- NIS (Network Information Service) - Dịch vụ thông tin mạng
7.9- Firewalls
7.10- IP Chains - Linux Kernel 2.2.x Firewalling
7.11- VNPs - Virtual Private Networks
8-Các công việc chuẩn bị để bảo vệ hệ thống của bạn

CHƯƠNG 2. LOGIN VÀ XÁC THỰC NGƯỜI DÙNG

1-Đăng nhập - Login

1.1- Trình getty

1.2- Trình login

2- Tài khoản, quản lý tài khoản và xác thực người dùng trên hệ thống

2.1- Tài khoản người dùng

2.2-Mật khẩu - phương pháp mã hoá

2.3- Mật khẩu shadow

2.4- Cracklib và cracklib_dict

3- PAM

3.1- PAM là gì?

3.2- Tổng quan

3.3- Cấu hình cho Linux PAM

3.4- Các module khả dụng

PHẦN I
AN NINH CỦA HỆ ĐIỀU HÀNH LINUX

CHƯƠNG 1. LINUX SECURITY

1- Giới thiệu

Trong chương này chúng tôi đề cập đến những vấn đề bảo mật chung, mà người quản trị hệ thống Linux phải đối mặt với. Nó bao trùm những triết lý phương bảo mật chung, đồng thời đưa ra một số ví dụ về cách thức bảo mật hệ thống của bạn nhằm chống những người xâm phạm hệ thống mà không được phép. Ngoài ra cũng có chỉ dẫn tới một số tài liệu và chương trình có liên quan đến vấn đề bảo mật.

1.1- Tại sao cần bảo mật

Trong khung cảnh thế giới truyền thông dữ liệu, kết nối Internet không quá đắt, sự phát triển của các phần mềm, thì bảo mật trở thành một vấn đề rất quan trọng. Hiện nay vấn đề bảo mật trở thành một yêu cầu cơ bản bởi vì việc tính toán mạng là hoàn toàn chưa được bảo mật. Ví dụ, khi dữ liệu của bạn truyền từ điểm A sang điểm B qua Internet trên đường đi nó có thể phải qua một số điểm khác trên tuyến đó, điều này cho phép các người sử dụng khác có cơ hội để chặn bắt, thay đổi nó. Thậm trí những người dùng trên hệ thống của bạn có thể biến đổi dữ liệu của bạn thành dạng khác mà bạn không mong muốn. Sự truy nhập không được ủy quyền tới hệ thống của bạn có thể được thu bởi kẻ xâm nhập trái phép (intruder) hay là "cracker", những kẻ này sử dụng các kiến thức tiên tiến để giả dạng bạn, đánh cắp những thông tin của bạn hoặc từ chối truy nhập của bạn tới nguồn tài nguyên của bạn.

1.2- Bạn đang cố gắng bảo vệ những gì?

Trước khi bạn cố gắng thực hiện bảo vệ hệ thống của bạn, bạn phải xác định mức đe dọa nào mà bạn cần bảo vệ, những rủi ro nào mà bạn có thể nhận được, và sự nguy hiểm nào mà hệ thống của bạn phải chịu. Bạn nên phân tích hệ thống của bạn để biết những gì bạn cần bảo vệ, tại sao bạn bảo vệ nó, giá trị của nó, và người chịu trách nhiệm về dữ liệu của bạn.

- Sự rủi ro (risk) có thể do người truy nhập trái phép thành công khi cố gắng truy nhập máy tính của bạn. Họ có thể đọc hoặc ghi các tệp, hoặc thực thi các chương trình gây ra thiệt hại không? Họ có thể xóa dữ liệu không? Họ có thể cản trở bạn hoặc công ty bạn làm một số việc quan trọng không? Đừng quên: một người nào đó truy nhập vào tài khoản của bạn, hoặc hệ thống của bạn, có thể giả dạng là bạn.

  Hơn nữa, có một tài khoản không an toàn trên hệ thống của bạn có thể gây nên toàn bộ mạng của bạn bị thỏa hiệp. Nếu bạn cho phép một người dùng đăng nhập sử dụng tệp .rhosts, hoặc sử dụng một dịnh vụ không an toàn như là tftp, như vậy là bạn đã tạo cho người truy nhập trái phép bước chân vào cách cửa hệ thống của bạn. Người truy nhập trái phép có một tài khoản người dùng trên hệ thống của bạn hoặc hệ thống của một người khác, nó có thể được sử dụng để truy nhập tới hệ thống khác hoặc tài khoản khác.

- Đe dọa (threat) là một điển hình của một ai đó với động cơ để đạt được sự truy nhập không được ủy quyền tới mạng hoặc máy tính của bạn. Bạn phải xác định ai mà bạn tin tưởng có quyền truy nhập tới hệ thống của bạn, và mối đe dọa nào mà có thể xảy ra. Có một vài dạng xâm nhập trái phép, bạn nên nhớ các đặc tính khác nhau của chúng khi bạn đang bảo vệ hệ thống của bạn.
- Tò mò (curious) - là một kiểu intruder thích tìm ra các kiểu hệ thống và dữ liệu mà bạn có.
- Độc ác (malicious) - kiểu intruder này xóa trang web của bạn hoặc bắt bạn phải mất nhiêù thời gian, tiền bạc để khôi phục lại dữ liệu đã bị gây thiệt hại bởi anh ta.

1.3- Các phương pháp để bảo vệ site của bạn

Trong chương này sẽ thảo luận các phương pháp khác nhau để bạn có thể bảo vệ các dữ liệu, tài nguyên mà bạn đã vất vả để có: máy móc, dữ liệu, các người dùng, mạng.

An toàn máy chủ

Có lẽ vùng được bảo vệ mà ở đó người quản trị hệ thống tập trung vào nhất đó là bảo vệ máy chủ. Điển hình là bảo đảm chắc chắn hệ thống của bạn là an toàn và hy vọng mọi người khác trên mạng của bạn cũng hành động như vậy. Chọn mật khẩu tốt, bảo vệ các dịch vụ mạng cục bộ của máy chủ, giữ bản ghi tài khoản, nâng cấp các chương trình nói chung là những công việc mà người quản trị phải làm. Mặc dù điều này là rất cần thiết song nó sẽ làm bạn nản lòng một khi mạng của bạn trở nên lớn hơn chỉ một vài máy.

An toàn mang cuc bô

An toàn mạng thì cần thiết như an toàn máy chủ cục bộ. Với hàng trăm, hàng nghìn hoặc thậm trí rất nhiều máy tính trên cùng một mạng thì bạn không thể tin cậy vào mỗi máy tính trong hệ thống máy tính đó là an toàn. Đảm bảo rằng chỉ những người sử dụng được ủy quyền có thể sử dụng mạng của bạn, xây dựng firewalls, sử dụng mật mã mạnh và đảm bảo rằng không có một máy "không tin cậy" nào có trên hệ thống của bạn.

Trong tài liệu này chúng ta sẽ thảo luận về một vài kỹ thuật được sử dụng để bảo vệ site của bạn, hy vọng sẽ chỉ cho bạn một vài cách để ngăn chặn các kẻ xâm nhập trái phép truy nhập tới những gì mà bạn đang bảo vệ.

Bảo vê thông qua những cái ít đươc chú ý đến (obscurity)

Một ví dụ của kiểu bảo vệ này là chuyển một dịch vụ mà được biết là dễ bị nguy hiểm tới một cổng không chuẩn với hy vọng các cracker sẽ không chú ý đến đó và do đó chúng sẽ không bị khai thác. Kiểu bảo vệ này ít an toàn.

2- Bảo vệ vật lý

Tầng đầu tiên để bảo vệ là bảo vệ vật lý trên hệ thống máy tính của bạn. Khi đó thì những người khác không thể truy nhập trực tiếp vào hệ thống máy móc của bạn và bạn có thể bảo vệ được máy của mình.

Mức độ bảo vệ vật lý mà bạn cần áp dụng phụ thuộc vào tình trạng và ngân sách của bạn. Nếu bạn là người sử dụng bình thường (home user), bạn có thể không cần quan tâm nhiều về vấn đề này. Nếu bạn đang ở trong một tổ chức nào đó, thì bạn cần phải để tâm nhiều hơn, nhưng người dùng vẫn phải làm việc được trên máy của mình. Các mục dưới đây sẽ giúp bạn giải quyết vấn đề này, bạn có thể có hoặc không cần thiết bảo mật máy tính của bạn khi bạn không có mặt ở đó.

2.1- Khóa máy tính

Một số vỏ máy (case) của PC loại mới có đặc tính khóa "locking". Thông thường nó là một socket ở mặt trước của vỏ máy, nó cho phép để ở trạng thái khóa hoặc mở. Việc khóa máy tính có thể giúp cho chúng ta ngăn chặn được ai đó ăn trộm máy tính của bạn hoặc là mở case và trực tiếp lấy trộm phần cứng của bạn. Đôi khi điều này cũng hạn chế được ai đó khởi động lại máy tính của bạn từ một đĩa mềm hoặc là từ một ổ đĩa cứng khác.

Các khóa trên từng máy tính thì khác nhau tùy theo sự hỗ trợ của bản mạch chủ (motherboard) và cách thiết kế case. Trên một số máy tính thực hiện theo cách bắt bạn phải phá case để mở case. Một số máy tính khác, chúng không cho phép bạn cắm bàn phím hoặc chuột mới. Hãy kiểm tra các chỉ dẫn bản mạch chủ hoặc case để lấy thêm thông tin. Điều này đôi khi rất thuận lợi, thậm trí với các khóa chất lượng thấp và có thể dễ dàng đánh bại bởi những kẻ tấn công bằng cách bẻ khóa.

Một số máy (hầu hết SPARCs và macs) có một dongle ở phía đằng sau, nếu bạn đưa một cáp qua đó, kẻ tấn công phải cắt nó hoặc bổ case để thâm nhập vào đó. Hãy đưa padlock hoặc combo lock qua nó, đó sẽ là yếu tố làm nản lòng kẻ muốn ăn trộm máy của bạn.

2.2- Bảo vệ BIOS

BIOS là mức thấp nhất của phần mềm mà để cấu hình hoặc thao tác phần cứng dựa trên x86. LILO và các phương pháp khởi động khác của Linux truy nhập tới BIOS để xác định cách khởi động máy của bạn. Các phần cứng khác mà Linux chạy trên nó có những phần mềm tương tự (OpenFirmware trên máy Macs và máy Suns mới, Sun boot PROM,...). Bạn có thể sử dụng BIOS để ngăn chặn những kẻ tấn công khởi động lại máy tính của bạn và thao tác với hệ thống Linux của bạn.

Nhiều BIOS của máy tính cho phép bạn thiết lập mật khẩu khởi động. Điều này không có nghĩa là cung cấp đầy đủ vấn đề bảo mật (BIOS có thể thiết lập lại hoặc xóa đi nếu một người nào đó khi đã mở được máy tính của bạn), nhưng nó có thể là một sự ngăn cản tốt (ví dụ như mất thời gian và để lại dấu vết của sự lục lọi).

Tương tự, trên hệ thống S/Linux (Linux cho các máy có bộ sử lý SPARC(tm)), EFPROM có thể được thiết lập để yêu cầu mật khẩu khởi động. Điều này làm kẻ tấn công mất thời gian.

Một số x86 BIOS cũng cho phép bạn xác định các cách thiết lập bảo mật khác nhau. Kiểm tra BIOS manual hoặc nhìn mỗi lần bạn khởi động hệ thống. Ví dụ, một số BIOS không cho phép khởi động từ ổ đĩa mềm và một số yêu cầu mật khẩu để truy nhập các đặc tính của BIOS.

***Chú ý*:** Nếu bạn có một máy server và bạn đã thiết lập mật khẩu khởi động thì máy của bạn sẽ không thể khởi động được nếu như không có mật khẩu khởi động. Do vậy bạn phải nhớ mật khẩu.

2.3- Bảo vệ trình nạp khởi động (Boot Loader) LILO

Có nhiều cách khởi động Linux khác nhau, các trình nạp khởi động của Linux cũng có thể được thiết lập mật khẩu khởi động. LILO được sử dụng để khởi động Linux, nó quản lý tiến trình khởi động và có thể khởi động các ảnh nhân Linux từ đĩa mềm, đĩa cứng hoặc có thể khởi động các hệ điều hành khác. LILO thì rất quan trọng cho hệ thống Linux do đó ta phải bảo vệ nó. File cấu hình của LILO là file lilo.conf, file này ở trong thư mục /etc. Với file này ta có thể cấu hình và cải thiện vấn đề an toàn của chương trình và hệ thống Linux. Ba tùy chọn quan trọng sau đây sẽ cải thiện vấn đề bảo vệ chương trình LILO.

- Tùy chọn timeout=<time>

Tùy chọn này điều khiển LILO đợi bao lâu (tính bằng giây) cho người dùng lựa chọn hệ điều hành nào trước khi nó khởi động mặc định. Một trong các yêu cầu an toàn của C2 là thiết lập khoảng thời gian này là 0 trừ khi hệ thống khởi động kép.

- Tùy chọn restriced

Tùy chọn "restricted" chỉ được sử dụng cùng với tùy chọn "password". Đảm bảo chắc chắn bạn sử dụng tùy chọn này cho mỗi image.

- Tùy chọn password=<password>

Tùy chọn này yêu cầu người sử dụng vào một mật khẩu khi cố gắng nạp hệ thống Linux trong chế độ đơn (single mode). Mật khẩu luôn luôn là một thứ nhạy cảm, ngoài ra cũng cần đảm bảo file /etc/lilo.conf sao cho không được phép ghi đại trà, nếu không bất kỳ người dùng nào cũng có thể đọc được mật khẩu đó.

Các bước để bảo vệ LILO với file cấu hình lilo.conf:

Bước 1:Sửa đổi lại file cấu hình lilo.conf và thêm vào 3 tùy chọn ở trên. Ví dụ

```
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
timeout=00
Default=linux
```

```
restricted
password=lucpv
image=/boot/vmlinuz-2.2.12-20
label=linux
initrd=/boot/initrd-2.2.12-10.img
root=/dev/sda6
read-only
```

Bước 2: Bởi vì file cấu hình /etc/lilo còn chứa mật khẩu không được mã hóa, do đó file này chỉ nên đọc bởi siêu người dùng (root). Thay đổi quyền truy nhập của file này sử dụng lệnh sau:

```
[root@deep /]# chmod 600 /etc/lilo.conf
```

Bước 3: Cập nhật file cấu hình này để có sự ảnh hưởng. Ta sử dụng lệnh sau:

```
[root@deep /]# /sbin/lilo –v
```

Bước 4: Thiết lập thuộc tính của file này sử dụng lệnh sau:

```
[root@deep /]# chattr +i /etc/lilo.conf
```

Bạn phải nhớ tất cả mật khẩu mà bạn thiết lập. Bạn cũng nên nhớ rằng các mật khẩu này chỉ đơn thuần bảo vệ một số kẻ tấn công, chúng không ngăn chặn được khi có người nào đó khởi động từ một đĩa mềm, và kết gắn phân vùng gốc của bạn. Nếu bạn đang sử dụng bảo mật kết hợp với một boot loader thì bạn có thể không cho phép khởi động đĩa mềm trong BIOS, và thiết lập mật khẩu bảo vệ BIOS.

2.4- xlock and vlock

Bạn có thể khóa console của bạn để ngăn chặn sự lục lọi hoặc nhìn xem bạn đang làm gì. Có 2 chương trình làm việc nàylà: xlock và vlock.

xlock

xlock là một chương trình khóa hiển thị trên X (X display locker). Nó được gộp vào trong bất cứ phân phối nào của Linux. Xem trang man của nó để có thêm thông tin. Thông thường bạn có thể chạy xlock từ bất kỳ xterm trên console của bạn, nó sẽ khóa những gì hiển thị và yêu cầu mật khẩu để mở khóa.

vlock

Là một chương trình nhỏ cho phép bạn khóa một số hoặc tất cả các console ảo trên Linux box. Bạn có thể chỉ khóa console mà bạn đang làm việc hoặc là tất cả. Nếu bạn khóa một console, thì những console khác có thể vào và sử dụng console này, chúng sẽ không thể sử dụng console của bạn đến khi bạn mở khóa nó.

Tất nhiên khóa console của bạn sẽ ngăn chặn một số người tò mò lục lọi công việc của bạn, nhưng sẽ không ngăn chặn được việc họ khởi động lại máy của bạn hoặc phá vỡ công việc của bạn. Nó cũng không thể ngăn chặn được họ truy

nhập vào máy của bạn từ một máy khác trên mạng và khi đó sẽ nảy sinh các vấn đề khác.

2.5- Phát hiện sự thỏa hiệp an toàn vật lý

Vấn đề đầu tiên luôn luôn cần chú ý đó là khi máy tính của bạn khởi động lại. Bởi vì Linux là một hệ điều hành mạnh và ổn định, máy của bạn chỉ nên được khởi động lại khi bạn cần nâng cấp hệ điều hành, lắp đặt, thay thế phần cứng. Nếu máy của bạn được khởi động lại mà bạn không thực hiện các vấn đề đó thì có thể là một dấu hiệu mà kẻ tấn công đã thỏa hiệp hệ thống của bạn. Có nhiều cách để hệ thống của bạn có thể bị thỏa hiệp phụ thuộc vào một kẻ tấn công khởi động lại hoặc tắt máy tính của bạn.

Kiểm tra các dấu hiệu của sự lục lọi trên vỏ máy và các vùng lân cận của máy tính. Mặc dù nhiều kẻ tấn công xóa dấu vết để lại trong nhật ký hệ thống, song ta nên kiểm tra tất cả và chú ý đến bất kỳ sự khác thường nào.

Deamon sys có thể được cấu hình để tự động gửi dữ liệu nhật ký tới một server syslog trung tâm, nhưng dữ liệu trong quá trình gửi thì không được mã hóa do đó sẽ cho phép một kẻ tấn công xem dữ liệu đó khi nó được truyền. Điều này tiết lộ các thông tin về mạng của bạn mà những thông tin này bạn không muốn công khai. Có một vài deamon có sẵn để mã hóa dữ liệu này khi nó được truyền đi.

Một vài công việc kiểm tra trong các nhật ký của bạn:

- Các nhật ký không đầy đủ hoặc ngắn
- Các nhật ký chứa timestamps lạ
- Nhật ký với quyền truy nhập và thành viên sở hữu không đúng
- Bản ghi khởi động hoặc bắt đầu các dịch vụ
- Các nhật ký bị mất
- Các *su* hoặc các đăng nhập từ các địa điểm lạ

3-Bảo vệ cục bộ

Điều chúng ta quan tâm tiếp theo là vấn đề bảo mật trên hệ thống của bạn chống lại sự tấn công của những người dùng cục bộ (local users). Lấy quyền truy nhập một tài khoản người dùng cục bộ là một những việc đầu tiên mà những kẻ tấn công hệ thống thực hiện để khai thác tài khoản người dùng root. Với sự bảo mật lỏng lẻo, họ có thể nâng cấp quyền truy nhập thông thường của họ ngang với quyền truy nhập của người dùng root bằng cách sử dụng những lỗi khác nhau và các dịch vụ cục bộ được thiết lập tồi. Nếu bạn chắc chắn rằng việc bảo mật cục bộ của bạn là tốt, thì đây sẽ là một hàng rào ngăn cản những kẻ tấn công.

Người dùng cục bộ cũng có thể gây ra rất nhiều sự tàn phá hệ thống của bạn, đặc biệt họ biết người mà họ đang tìm hiểu là ai. Cung cấp tài khoản cho người dùng mà bạn không biết hoặc cho người không liên lạc thông tin với bạn là một điều không thể chấp nhận được.

3.1-Tạo các tài khoản mới

Bạn nên chắc chắn rằng bạn cung cấp tài khoản người dùng chỉ với những yêu cầu tối thiểu cho những tác vụ mà họ cần để làm việc. Giả sử, nếu bạn cung cấp cho con trai bạn (10 tuổi) với một tài khoản, bạn có thể chỉ cho quyền truy nhập bộ sử lý word và chương trình vẽ, nhưng không được xóa dữ liệu nếu nó không phải do con bạn tạo ra.

Một số quy tắc khi cho phép quyền truy nhập người dùng trên máy Linux của bạn:

- Cho họ số lượng đặc quyền tối thiểu mà họ cần thiết.
- Phải biết họ đăng nhập hệ thống khi nào và ở đâu.
- Bạn phải chắc chắn xóa những tài khoản không còn giá trị.
- Nên sử dụng cùng một userid (số hiệu người sử dụng) trên các máy tính và các mạng để giảm công việc bảo trì tài khoản và cho phép dễ dàng phân tích dữ liệu nhật ký.
- Việc tạo số hiệu nhóm người dùng là tuyệt đối cấm. Bởi vì tài khoản người dùng có tính thống kê được (accountability), còn tài khoản nhóm thì không

3.2- An toàn Root

Một tài khoản có đầy đủ đặc quyền trên máy của bạn đó là tài khoản người dùng root (superuser). Tài khoản này có các quyền trên toàn bộ máy, nó cũng có thể có quyền trên các máy khác trên hệ thống mạng. Lưu ý rằng, bạn có thể chỉ sử dụng tài khoản người dùng root trong thời gian rất ngắn, với những tác vụ nhất định, và nên chạy hầu hết với người dùng bình thường. Thậm trí với những lỗi rất nhỏ trong khi đăng nhập với người dùng root có thể gây ra rất nhiều vấn đề. Đó là lý do tại sao bạn nên dùng đặc quyền root chỉ trong thời gian rất ngắn, và khi đó thì hệ thống sẽ được an toàn hơn.

Những điều cần tránh khi đăng nhập với tư cách siêu người dùng:

- Khi thực hiện với những câu lệnh phức tạp, thử chạy trước để không phá hủy hệ thống. Đặc biệt những câu lệnh mang tính chất hủy bỏ. Ví dụ, nếu bạn muốn thực hiện câu lệnh rm foo*.bak, thì đầu tiên bạn nên thực hiện lệnh ls foo*.bak và chắc chắn rằng bạn đang xóa những file nào mà bạn muốn.
- Cung cấp cho người dùng thông báo khi sử dụng lệnh rm để hỏi trước khi thực hiện xóa.
- Bạn nên làm việc với một tài khoản người dùng thông thường, chỉ làm việc với tư cách siêu người dùng với những tác vụ đặc biệt, và sau đó phải trở về tài khoản người dùng bình thường ngay.
- Đường dẫn lệnh cho người dùng root là vấn đề rất quan trọng, (thể hiện qua biến môi trường PATH). Nó chỉ ra các thư mục mà trong đó shell tìm kiếm các chương trình để thực hiện. Cố gắng hạn chế đường dẫn lệnh cho người sử dụng root nhiều như có thể. Hơn nữa, không để các thư mục có thể

được ghi trong đường dẫn tìm kiếm của bạn, nếu điều này xảy ra thì sẽ cho phép các kẻ tấn công thay đổi hoặc di chuyển các file nhị phân trong đường dẫn tìm kiếm của bạn, cho phép chúng chạy như là root ở lần tới khi bạn chạy lệnh.

- Không nên sử dụng các dịch vụ từ xa (công cụ rlogin, rsh, rexec) khi đang đăng nhập với tư cách root. Đừng bao giờ tạo tệp .rhosts cho root.
- File /etc/securetty chứa danh sách các đầu cuối (terminals) mà root có thể đăng nhập từ đó. Red Hat Linux ngầm định thiết lập cho các console ảo cục bộ (vtys). Bạn nên thận trọng khi thêm những gì khác trong tệp này. Bạn nên đăng nhập từ xa bằng một tài khoản người dùng bình thường và sau đó su (switch user) vào người dùng root khi bạn muốn (hy vọng là qua ssh hoặc một kênh khác đã được mã hóa), do vậy không cần thiết bạn phải đăng nhập trực tiếp vào người dùng root.
- Bạn chỉ nên là tư cách root chỉ trong thời gian ngắn với những tác vụ đặc biệt. Bởi mọi hoạt động của bạn có thể gây ra rất nhiều kết quả. Hãy nghĩ kỹ khi thực thi một lệnh.

4-An toàn file và hệ thống file

Một vài phút chuẩn bị và lập kế hoạch trước khi đưa hệ thống của bạn vào chế độ trực tuyến (online) có thể giúp bạn bảo vệ hệ thống file và dữ liệu chứa trong đó.

- Không có một lý do nào cho phép các thư mục home của người dùng được phép chạy các chương trình SUID/SGID trên đó. Sử dụng tùy chọn 'nosuid' trong tệp /etc/fstab cho các phân vùng được ghi bởi người dùng khác root. Bạn cũng có thể sử dụng các tùy chọn 'nodev' và 'noexec' trên các phân vùng thư mục home của người dùng, khi đó cấm thực thi các chương trình, và tạo các thiết bị khối và thiết bị kí tự.
- Nếu bạn đang 'export' hệ thống file sử dụng NFS, phải chắc chắn khi cấu hình tệp /etc/exports với hầu hết các hạn chế quyền truy nhập có thể. Điều này có nghĩa là không sử dụng kí tự thay thế (wildcards), không cho phép root truy nhập ghi, và chỉ đọc.
  Ví dụ: /home/tiendq 192.168.2.220(no_root_squash)
- Cấu hình hệ thống file bằng 'umask' để hạn chế các quyền có thể. (Trình bày sau)
- Nếu bạn đang kết gắn kết hệ thống file sử dụng hệ thống file mạng NFS, phải chắc chắn khi cấu hình tệp /etc/exports với các hạn chế phù hợp. Đặc biệt, sử dụng các tùy chọn `nodev', `nosuid', và `noexec'. (*Xem NFS-howto)*
- Thiết lập giới hạn hệ thống file (mặc định không có). Bạn có thể điều khiển giới hạn tài nguyên cho mỗi người dùng, sử dụng PAM module và /etc/pam.d/ limits.conf. Ví dụ, giới hạn cho nhóm người dùng như sau:

```
@users    hard  core   0
@users    hard  nproc  50
@users    hard  rss    5000
```

Giải thích: Cấm không được tạo các file core, giới hạn số tiến trình là 50, và giới hạn không gian bộ nhớ cho mỗi người dùng là 5M.

- Các file /var/log/wtmp và /var/run/utmp chứa các bản ghi đăng nhập của tất cả người dùng trên hệ thống. Phải duy trì tính toàn vẹn của chúng bởi chúng có thể được sử dụng để xác định khi nào và từ đâu một người dùng đã vào hệ thống của bạn. Các file này có quyền là 644 (không ảnh hưởng tới hệ điều hành bình thường).
- Những bit không thể biến đổi (immutable bit) có thể được sử dụng để ngăn chặn hiểm họa xóa hoặc ghi đè một file mà file này cần được bảo vệ. Nó cũng ngăn chặn một người nào đó tạo liên kết biểu tượng (symbolic link) tới tệp đó (symbolic link trở thành nguồn gốc của các cuộc tấn công thực hiện xóa tệp /etc/passwd hoặc /etc/shadow). Xem chattr(1) man page để thêm thông tin về các bit không biến đổi này.
- Các file SUID và SGID trên hệ thống là một rủi ro an toàn tiềm ẩn, và chúng nên được giám sát cẩn thận. Bởi các chương trình này gán các đặc quyền cho người dùng mà đang thực thi chúng, do vậy cần phải bảo đảm rằng các chương trình không an toàn này không được cài đặt. Một cách tấn công ưa dùng của cracker là khai thác chương trình SUID của root, sau đó để một chương trình SUID như là một cửa sau (backdoor) để vào trong lần tiếp theo.

Tìm tất cả các chương trình SUID/SGID trên hệ thống của bạn và giữ dấu vết những gì mà chúng đã làm, bởi vậy bạn phải biết được bất kỳ các thay đổi mà có thể chỉ ra kẻ tấn công tiềm ẩn. Sử dụng câu lệnh dưới đây để tìm tất cả các chương trình SUID/SGID trên hệ thống của bạn:

```
root#  find / -type f  -perm -04000 -o -perm -02000
```

Bạn có thể xóa các quyền SUID hoặc SGID trên các chương trình khả nghi bằng lệnh chmod, sau đó khôi phục lại các thay đổi này nếu bạn cảm thấy cần thiết.

- Các file world-writable (file ghi đại trà), đặc biệt là các file hệ thống có thể là một lỗ hổng an ninh nếu một cracker dành được quyền truy nhập vào hệ thống của bạn và sửa đổi chúng. Hơn thế nữa, các thư mục world-writable là rất nguy hiểm, bởi vì chúng cho phép một cracker thêm hoặc là xóa các tệp mà anh ta muốn. Để chỉ ra tất cả các tệp world-writable trên hệ thống của bạn sử dụng lệnh sau:

```
root# find / -perm -2 ! -type l -ls
```

và chắc chắn rằng bạn biết tại sao các tệp có thể ghi được. Thông thường, một số tệp sẽ là world-writable, bao gồm các tệp trong thư mục /dev, các liên kết tượng trưng, tùy chọn ! -type l không hiển thị các file dạng này trong câu lệnh find trước.

- Các file không được sở hữu cũng có thể để kẻ truy nhập trái phép truy nhập vào hệ thống của bạn. Bạn nên chỉ ra các file mà không thuộc sở hữu của ai trên hệ thống của bạn, hoặc không thuộc một nhóm nào với lệnh:

```
root# find / -nouser -o -nogroup -print
```

- Tìm các file .rhosts là một phần của nhiệm vụ quản trị hệ thống, những file này không nên được cấp quyền trên hệ thống của bạn. Nhớ rằng, một cracker chỉ cần một tài khoản không an toàn để đạt được sự truy nhập tới toàn bộ mạng. Bạn cần chỉ ra tất cả những file .rhosts trên hệ thống bằng lệnh sau:

```
root# find /home -name .rhosts -print
```

- Cuối cùng, trước khi thay đổi các quyền trên bất kỳ file nào, cần đảm bảo chắc chắn rằng bạn hiểu những gì bạn đang làm. Đừng bao giờ thay đổi quyền trên một file bởi vì đó là cách dễ nhất để có mọi thứ. Luôn luôn xác định rằng tại sao file đó lại có quyền này trước khi thay đổi nó.

4.1- Thiết lập Umask

Lệnh umask được sử dụng để xác định mặc định chế độ (quyền) của file được tạo trên hệ thống. Chế độ này là phần bù cơ số 8 của chế độ file mong muốn. Nếu một file được tạo mà không có bất kỳ sự để ý nào tới việc thiết lập quyền truy nhập, thì người dùng có thể tình cờ cho ai đó quyền read hoặc write mà người này không nên có quyền này. Việc thiết lập umask điển hình là 022, 027 và 077 (việc thiết lập này hạn chế hầu hết các quyền truy nhập). Bình thường umask được thiết lập trong /etc/profile, bởi vậy nó áp dụng tới tất cả người dùng trên hệ thống. mask của file được tạo có thể được tính toán bằng cách lấy 777 trừ đi giá trị mong muốn. Nói cách khác, một umask của 777 sẽ khiến các file được tạo mới sẽ không có quyền nào (không read, không write, không execute) đối với bất kỳ ai. Một umask của 666 sẽ khiến các file được tạo mới có một mask là 111. Ví dụ:

```
# Set the user's default umask
umask 033
```

Trong ví dụ này, các thư mục được tạo mới sẽ có quyền truy nhập là 744 (giá trị này thu được bằng cách lấy 777 trừ đi 033), các file được tạo mới sẽ có quyền là 644.

4.2- Quyền của file

Unix và Linux tách biệt điều khiển truy nhập trên file và thư mục theo 3 đặc tính: người sở hữu (owner), nhóm (group) và các người khác (other). Giải thích nhanh về quyền của file và thư mục trên Linux: Quyền truy nhập của file và thư mục là một tập hợp các bit có thể được thiết lập hoặc xóa bỏ để cho phép các kiểu truy nhập nhất định tới file hoặc thư mục đó. Quyền đối với thư mục có thể có nghĩa khác với quyền cùng quyền truy nhập của file. Trên file và thư mục có các kiểu cho phép truy nhập khác nhau đó là:

Read:

- Cho phép xem nội dung của một file
- Cho phép đọc một thư mục

Write:

- Cho phép thêm hoặc thay đổi một file

- Cho phép xóa hoặc di chuyển các file trong một thư mục

Execute:

- Cho phép chạy một chương trình nhị phân hoặc một shell script
- Cho phép tìm kiếm trong một thư mục (kết hợp với quyền read)

Ngoài 3 đặc tính trên thì còn có một số thuộc tính khác đối với file và thư mục để thiết lập sự cho phép của file và thư mục đó là:

Thuộc tính Save Text (đối với thư mục):

"Bit sticky" có một một nghĩa khác khi áp dụng tới thư mục hơn là khi áp dụng tới file. Nếu bit sticky được thiết lập trên một thư mục thì một người sử dụng chỉ có thể xóa các file mà là sở hữu của anh ta hoặc anh ta được gán quyền write trên file đó. Điều này được áp dụng đối với thư mục như /tmp, thư mục này thì được ghi đại trà (world-writable) nhưng ở đó nó không mong muốn cho phép bất kỳ người dùng nào xóa các file trong đó.

Thuộc tính SUID (đối với các file):

Thuộc tính này mô tả việc thiết lập quyền theo số hiệu người dùng (set-user-id) trên file đó. Khi chế độ truy nhập theo số hiệu (ID) người dùng được thiết lập trong nhóm quyền owner và file đó là file có thể thực thi thì tiến trình mà sẽ chạy nó thì được gán quyền truy nhập tới các nguồn tài nguyên hệ thống dựa trên người dùng mà sở hữu file đó. Việc thiết lập sự cho phép theo kiểu này là nguyên nhân của nhiều sự khai thác tràn bộ đệm (buffer overflow).

Thuộc tính SGID (đối với file):

Nếu thiết lập trong các quyền của nhóm (group), thì bit này điều khiển "thiết lập theo số hiệu (id) của nhóm" trạng thái của file. Việc thiết lập này là một cách tương tự như SUID, ngoại trừ nhóm đó thì được ảnh hưởng. File mà được thiết lập theo thuộc tính này phải là file có thể thực thi để có bất kỳ sự ảnh hưởng nào.

Thuộc tính SGID (đối với các thư mục):

Nếu bạn thiết lập bit SGID trên một thư mục (với lệnh chmod g+s) thì các file được tạo trong thư mục đó sẽ có nhóm thuộc nhóm của thư mục này.

4.3- Kiểm tra tính toàn vẹn của hệ thống file

Cách khác để tìm sự tấn công cục bộ trên hệ thống đó là chạy một chương trình kiểm tra tính toàn vẹn như Tripwire, Aide hoặc Osiris. Các chương trình kiểm tra tính toàn vẹn này chạy một số các tổng kiểm tra trên tất cả các file nhị phân và file cấu hình và so sánh chúng với một cơ sở dữ liệu. Bởi vậy bất kỳ sự thay đổi nào trong các file sẽ được đặt cờ.

Quả là một ý tưởng tốt để cài đặt một phần các chương trình này vào đĩa mềm và rồi thiết lập chốt chống ghi của đĩa mềm này. Với điều này các kẻ xâm

nhập trái phép không thể lục lọi các chương trình kiểm tra tính toàn vẹn này hoặc thay đổi cơ sở dữ liệu của nó.

Bạn có thể thêm một mục crontab để chạy các chương trình này từ trong đĩa mềm của bạn vào mỗi tối và bạn có kết quả trong sáng hôm sau như:

```
#set mailto
mailto=kevin
#run Tripwire
15 05 * * * root /usr/local/adm/tcheck/tripwire
```

Các chương trình kiểm tra tính toàn vẹn để xác định các kẻ xâm nhập trái phép trước khi bạn để ý đến chúng. Bởi vì rất nhiều các file thay đổi trên hệ thống, bạn phải cẩn thận với những gì mà cracker hành động và những gì mà chính bạn đang làm.

5-An toàn mật khẩu và sự mã hóa

Một trong hầu hết các đặc điểm bảo mật được sử dụng ngày nay là mật khẩu. Thật là quan trọng cho cả bạn và tất cả các người sử dụng để có các mật khẩu an toàn, không thể dự đoán. Đa số các phân phối Linux gần đây có các chương trình passwd để không cho phép bạn thiết lập các mật khẩu dễ dàng và có thể dự đoán. Đảm bảo chắc chắn các chương trình passwd này thì được cập nhật và có các đặc điểm này.

Thảo luận kỹ về sự mã hóa thì vượt qúa phạm vi của tài liệu này, ở đây chỉ nhằm mục đính là giới thiệu. Ngày nay mã hóa thì rất hữu ích và cần thiết. Các phương pháp mã hóa thì rất đa dạng mỗi phương pháp có đặc tính riêng.

Đa số các hệ Unix (và cả Linux) sử dụng giải thuật mã hóa một chiều gọi là DES (Data Encryption Standard) để mã hóa mật khẩu của bạn. Những mật khẩu được mã hóa này được chứa trong file /etc/passwd hoặc /etc/shadow. Khi bạn đăng nhập mật khẩu mà bạn gõ vào thì nó được mã hóa và được so sánh với các mục trong file mà chứa mật khẩu của bạn. Nếu giống nhau thì bạn được phép truy nhập vào hệ thống. Mặc dù DES là một giải thuật mã hóa hai chiều (bạn có thể mã và giải mã một thông báo với các khóa đúng đã cho), các biến thể mà hầu hết các Unix sử dụng là giải thuật mã hóa một chiều. Điều này có nghĩa rằng không thể khôi phục lại sự mã hóa để có lại mật khẩu từ nội dung của file /etc/passwd (hoặc /etc/shadow).

Các tấn công Brute force như “Crack” hoặc “John the Ripper” thường dự đoán mật khẩu trừ khi mật khẩu của bạn đủ ngẫu nhiên. Modules PAM (xem ở sau) cho phép bạn sử dụng một chương trình mã hóa khác cho mật khẩu của bạn (MD5). Chạy Crack định kỳ trong cơ sở dữ liệu của bạn để tìm ra các mật khẩu không an toàn và rồi thông báo với người dùng có mật khẩu không an toàn này để thay đổi nó.

5.1- PGP và mật mã khóa công khai

Mật mã khóa công khai sử dụng một khóa để mã hóa và một khóa để giải mã. Tuy nhiên mật mã cổ điển sử dụng cùng một khóa để mã hóa và giải mã. Các khóa này phải biết ở cả hai nơi, bởi vậy vấn đề làm sao để truyền các khóa này từ nơi này đến nơi khác được an toàn. Để giảm công việc truyền các khóa mã hóa này an toàn, khóa công khai sử dụng hai khóa riêng biệt: khóa công khai và khóa bí mật. Khóa công khai của mỗi một người thì bất kỳ ai cũng có để mã hóa, trong khi đó mỗi người giữ một khóa bí mật riêng của mình để giải mã thông báo đó.

PGP (Pretty Good Privacy) là một hỗ trợ nổi tiếng trên Linux. Phiên bản 2.6.2 và 5.0 được biết là làm việc tốt. Các số nguyên tố (primer) tốt của PGP và như thế nào để sử dụng nó bạn có thể xem ở PGP FAQ: http://www.rsa.com/service/export/faq/55faq.cgi. Hãy chắc chắn phiên bản mà được áp dụng vào đất nước bạn. Do luật hạn chế xuất khẩu của chính phủ Mỹ, mật mã mạnh thì bị ngăn cấm đưa ra ngoài đất nước này. Việc điều khiển xuất khẩu của Mỹ bây giờ được quản lý bởi EAR, trước đó chúng được quản lý bởi ITAR.

5.2-SSL, S-HTTP, HTTP và S/MIME

Thường người dùng thắc mắc về sự khác nhau giữa an toàn và các giao thức mã hóa, như thế nào để sử dụng nó. Trong mục này sẽ giải thích ngắn gọn về mỗi giao thức và nơi tìm thấy thông tin về nó.

- SLL - Secure Sockets Layer là một phương pháp mã hóa được phát triển bởi Netscape để bảo vệ trên mạng Internet. Nó hỗ trợ vài giao thức mã hóa khác nhau và cung cấp xác thực khách và chủ. SSL hoạt động ở tầng mạng, nó tạo một kênh mã hóa an toàn cho dữ liệu và có thể mã hóa nhiều kiểu dữ liệu. Bạn có thể tìm thấy nhiều thông tin về nó ở: http://www.consensus.com/security/ssl-talk-faq.html.
- S-HTTP - là một giao thức khác cung cấp dịch vụ bảo mật thông qua Internet. Nó được thiết kế để cung cấp tính tin cẩn, xác thực, tính toàn vẹn và sự không từ chối (non-repudiability) trong đó hỗ trợ nhiều cơ chế quản lý khóa và nhiều giải thuật mã hóa thông qua tùy chọn giữa các tổ chức có liên quan trong mỗi phiên giao dịch. S-HTTP hạn chế tới những phần mềm mà đang thực thi nó, và nó giải mã mỗi thông báo.
- S/MIME: - S/MIME (Secure Multipurpose Internet Mail Extension) là một chuẩn mã hóa được sử dụng để mã hóa thư điện tử và các dạng thông báo khác trên Internet. Nó là một chuẩn mở được phát triển bởi RSA. Để có thông tin nhiều hơn về S/MIME có thể tìm ở: http://home.netscape.com/assist/security/smime/ overview.html.

5.3- Ứng dụng Linux IPSEC

Cùng với CIPE và các dạng khác của mã hóa dữ liệu thì còn có một vài ứng dụng khác của IPSEC cho Linux. IPSEC là một cố gắng lớn của IETF để tạo sự truyền thông mã hóa an toàn ở tầng mạng IP, nó cũng cung cấp xác thực, tính toàn vẹn, điều khiển truy nhập và sự tin cẩn. Để có thông tin về IPSEC và Internet bạn có thể tìm ở http://www.

5.4- ssh và stelnet

Ssh và stelnet là tập hợp các chương trình mà cho phép bạn đăng nhập tới các hệ thống từ xa và có kết nối được mã hóa.

openssh là một bộ các chương trình được sử dụng như là một sự thay thế cho rlogin, rsh và rcp. Nó sử dụng mật mã khóa công khai để mã hóa truyền thông giữa hai máy chủ, và cũng để xác thực người dùng. Nó có thể được sử dụng để đăng nhập an toàn tới một máy chủ từ xa hoặc sao chép dữ liệu giữa các máy chủ, trong khi đó nó ngăn chặn các cuộc tấn công chung cuộc và đánh lừa DNS. Openssh sẽ thực hiện việc nén dữ liệu trên các kết nối của bạn và bảo vệ truyền thông X11 giữa các máy chủ.

Hiện tại có vài ứng dụng ssh. Các ứng dụng thương mại cũ có thể tìm ở http://www .datafellows.com.

Ứng dụng Openssh thì dựa trên một phiên bản gần đây của ssh datafellows và đã được được sửa đổi lại để không thuộc bất kỳ trong bằng sáng chế nào. Openssh thì miễn phí và đặt dưới bằng sáng chế BSD. Nó có thể tìm ở: http://www.openssh.com.

SSLeay là một ứng dụng miễn phí của giao thức Secure Sockets Layer của Netscape, nó được phát triển bởi Eric Young. Nó bao gồm vài ứng dụng như Secure telnet, một mô dudule cho Apache, vài cơ sở dữ liệu và cùng với một vài giải thuật bao gồm DES, IDEA Và Blowfish.

Sử dụng những thư viện này, một thay thế secure telnet đã được tạo để thực hiện mã hóa trên một kết nối telnet. Không như SSH, stelnet sử dụng SSL. Bạn có thể tìm Secure telnet và Secure FTP ở http://www.psy.uq.oz.au/~ftp/Crypto/.

5.5 PAM - Pluggable Authetication Modules.

Các phiên bản mới hơn của phân phối Red Hat Linux có một lược đồ xác thực thống nhất được gọi là "PAM". PAM cho phép bạn thay đổi phương pháp xác thực và yêu cầu (on the fly), nó thâu tóm tất cả các phương pháp xác thực cục bộ mà không phải biên dịch lại bất kỳ một chương trình thực thi nào. Cấu hình PAM thì vượt quá khuân khổ của tài liệu này, để có thông tin nhiều hơn về PAM vạn có thể tìm ở **http://www.kernel.org/pub/linux/libs/pam/index.html**.

Một vài công việc bạn có thể thực hiện với PAM là:

- Sử dụng phương pháp mã hóa khác DES cho các mật khẩu của bạn. (Tạo khó khăn hơn để phá mật khẩu bằng phương pháp vét cạn (brute-force)).
- Thiết lập hạn chế tài nguyên trên tất cả các người dùng bởi vậy họ không thể thực hiện việc tấn công từ chối dịch vụ.
- Cho phép mật khẩu shadow (xem dưới).

- Cho phép các người dùng cụ thể đăng nhập chỉ ở thời gian cụ thể từ một địa điểm xác định.

Với một vài giờ cài đặt và cấu hình hệ thống của bạn, bạn có thể ngăn chặn nhiều cuộc tấn công trước khi chúng xảy ra. Ví dụ, sử dụng PAM không cho phép sử dụng rộng rãi các file .rhosts trong thư mục riêng của người dùng bằng cách thêm các dòng sau tới file /etc/pam.d/rlogin:

```
#  Disable rsh//rlogin/rexec for users
login auth required pam_rhosts_auth.so no_rhosts
```

5.6-Cryptographic IP Encapsulation (CIPE)

Mục đích chính của phần mềm này là cung cấp một tiện ích để bảo vệ (chống lại việc thu trộm, bao gồm phân tích đường truyền, giả mạo thông báo) sự kết nối các mạng con thông qua một mạng gói không an toàn như Internet. CIPE mã hóa dữ liệu ở tầng mạng. Việc truyền các gói giữa các máy chủ trên mạng được mã hóa. Bộ mã hóa được đặt ở gần trình điều khiển mà để gửi và nhận các gói.

Không giống như SSH (SSH mã hóa dữ liệu ở tầng socket). Một sự kết nối logic giữa các chương trình chạy trên các máy chủ khác nhau được mã hóa. CIPE có thể được sử dụng trong đường hầm, nhằm mục đích tạo ra mạng riêng ảo (Virtual Private Network). Mã hóa ở tầng thấp có ưu điểm là nó có thể làm các công việc một cách trong suốt giữa hai mạng được kết nối trong VNP mà không với bất kỳ một thay đổi nào tới phần mềm ứng dụng. Để có thêm thông tin về CIPE bạn có thể tìm ở http://www.inka.de/~bigred/devel /cipe.html.

5.7- Kerberos

Kerberos là một hệ thống xác thực được phát triển bởi đề án Athena ở MIT. Khi một người dùng đăng nhập, Kerberos xác thực người dùng đó (sử dụng một mật khẩu) và cung cấp cho người dùng đó một cách để chứng minh nhận dạng của anh ta tới các server và host trong mạng.

Tiếp theo sự xác thực này được sử dụng bởi các chương trình như rlogin để cho phép người dùng đăng nhập tới các host khác mà không với một mật khẩu (trong vị trí của file .rhosts). Phương pháp xác thực này cũng được sử dụng bởi hệ thống thư nhằm mục đích đảm bảo rằng các thư này thì được chuyển tới đúng người nhận, nó cũng đảm bảo rằng người gửi là người mà người nhận yêu cầu.

Kerberos và các chương trình khác đi kèm với nó ngăn chặn các người dùng khỏi đánh lừa hệ thống khi nó tin tưởng rằng họ không là một ai khác. Không may, cài đặt Kerberos thì khá phức tạp, yêu cầu thay đổi hoặc thay thế một số các chương trình chuẩn. Bạn có thể tìm nhiều thông tin hơn về Kerberos ở **http://nii.isi.edu/info/kerberos/**.

5.8-Shadow Passwords

Shadow Passwords là một phương pháp giữ bí mật thông tin mật khẩu được mã hóa của bạn khỏi các người dùng bình thường. Các phiên bản gần đây của cả Red Hat và Debian Linux sử dụng shadow passwords là mặc định. Nhưng trên các hệ thống khác, các mật khẩu được mã hóa thì được chứa trong file /etc/passwd để cho tất cả mọi người có thể đọc. Bất kỳ ai chạy các chương trình dự đoán mật khẩu trên các hệ thống này thì có thể xác định những gì mà chúng có. Trái lại, shadow passwords lưu các mật khẩu mã hóa ở trong file /etc/shadow, file này thì chỉ người dùng có đặc quyền thì mới có thể đọc được. Nhằm mục đích sử dụng shadow password, bạn cần đảm bảo chắc chắn rằng tất cả các tiện ích truy nhập tới thông tin mật khẩu thì được biên dịch lại để hỗ trợ chúng. Ngoài ra PAM cho phép bạn chỉ chạy trong một module shadow; nó không yêu cầu biên dich lại các chương trình thực thi. Bạn có thể xem tại liệu Shadow-Password HOWTO để có thêm thông tin nếu cần thiết, thông tin này cũng có sẵn ở http://metalab.unc.edu/LDP/HOWTO/Shadow-Password -HOWTO.html.

5.9- "*Crack*" *và* "*John the Ripper*"

Nếu có một vài lý do mà chương trình passwd không bắt buộc các mật khẩu khó dự đoán thì bạn có thể chạy một chương trình phá mật khẩu và đảm bảo rằng mật khẩu của người dùng thì an toàn.

Các chương trình phá mật khẩu làm việc trên một ý tưởng đơn giản: chúng thử mọi từ trong một từ điển, và rồi thay đổi trên các từ này, mã hóa mỗi từ và kiểm tra từ được mã hóa này so sánh với mật khẩu đã được mã hóa của bạn. Nếu chúng giống nhau thì mật khẩu của bạn đã bị phá.

Có một số chương trình phá mật khẩu, nhưng hai chương trình nổi tiếng trong số này đó là "Crack" và "John the Ripper" (http://www.false.com/security/john/index.html). Các chương trình này thì chiếm nhiều thời gian của cpu.

5.10 -CFS - Cryptograpic File System và TCFS - Transparent Cryptographic File System

CFS là một cách mã hóa toàn bộ cây thư mục và cho phép người dùng lưu những file được mã hóa này trên chúng. CFS sử dụng một NFS server chạy trên máy cục bộ. CFS thì có sẵn ở http://www.zedz.net/redhat/. Để có thêm thông tin bạn có thể tìm ở **ftp://ftp.research.att.com/dist/mab/**.

TCFS cải tiến từ CFS bằng cách thêm vào nhiều sự tích hợp với hệ thống file, bởi vậy nó thì trong suốt với người dùng mà hệ thống file đó được mã hóa. Để có thêm thông tin bạn có thể tìm ở http://edu-gw.dia.unisa.it/tcfs/.

5.11- X11, SVGA và bảo vệ màn hình.

X11

Thật là quan trọng cho bạn để bảo vệ màn hình đồ họa của bạn ngăn chặn

các kẻ tấn công thu trộm mật khẩu của bạn khi bạn gõ chúng, đọc tài liệu hoặc thông tin mà bạn đang đọc trên màn hình, hoặc thậm trí sử dụng một kẽ hở an ninh để có được quyền truy nhập root. Chạy các ứng dụng X từ xa trên một mạng cũng có thể dẫn đến nguy hiểm, nó cho phép các bộ lắng nghe (sniffer) xem tất cả các tương tác với hệ thống từ xa.

X có một số cơ chế điều khiển truy nhập. Cơ chế đơn giản nhất là dựa trên host: bạn sử dụng xhost để xác định các host nào được cho phép truy nhập tới màn hình của bạn. Cơ chế này thì không an toàn ở tất cả, bởi vì nếu một ai đó có quyền truy nhập tới máy của bạn thì họ có thể xhost + máy của họ và có sự truy nhập một cách dễ dàng. Ngoài ra nếu bạn cho phép truy nhập từ một máy không tin cậy thì bất kỳ ai cũng có thể thỏa hiệp màn hình của bạn.

Khi sử dụng xdm (X Display Manager) để đăng nhập thì bạn có một phương pháp truy nhập tốt hơn: MIT-MAGIC-COOKIE-1. Một "cookie" 128-bit được sinh ra và được chứa trong file .Xauthority. Nếu bạn cần cho phép một máy từ xa truy nhập tới màn hình của bạn thì bạn có thể sử dụng lệnh xauth và những thông tin trong file .Xauthority để cung cấp quyền truy nhập tới chỉ kết nối đó. Xem Remote-X-Apps mini-howto ở địa chỉ http://metalab.unc.edu/LDP/HOWTO/mini/Remote-X-Apps .html.

SVGA

Các chương trình SVGAlib là SUID-root điển hình nhằm mục đích truy nhập tới tất cả phần cứng video của máy của bạn. Điều này thì rất nguy hiểm. Nếu chúng hỏng thì bạn cần khởi động lại máy để khôi phục lại console thích hợp. Đảm bảo chắc chắn bất kỳ chương trình SVGA mà bạn đang chạy thì xác thực, ít nhất thì tin cậy. Thậm trí tốt hơn là không chạy chúng.

GGI - Đề án giao diên đồ hoa chung

Đề án Linux GGI (Generic Graphic Interface project) cố gắng giải quyết vài vấn đề với các giao diện video trên Linux. GGI sẽ xóa một thành phần nhỏ của mã video trong nhân Linux và rồi điều khiển truy nhập tới hệ thống video. Điều này có nghĩa là GGI sẽ có thể khôi phục lại console của bạn ở bất kỳ thời gian nào tới một trạng thái tốt. Ngoài ra chúng sẽ cho phép một khóa an toàn, bởi vậy bạn có thể chắc chắn rằng không có chương trình đăng nhập Trojan horse đang chạy trên console của bạn. Xem ở địa chỉ http:// synergy.caltech.edu/ ~ggi/ đề có thêm thông tin.

6-An toàn nhân

Mục này liệt kê các tùy chọn cấu hình nhân có liên quan tới an toàn. Để hiểu rõ về chúng làm gì và như thế nào để sử dụng chúng, bạn có thể đọc ở mục 7 trong tài liệu Linux Security HOWTO.

Khi nhân điều khiển mạng máy tính, thì rất quan trọng để bảo đảm nó an

toàn và không bị thỏa hiệp. Để ngăn chặn một vài sự tấn công trên mạng thì bạn nên nhập nhật phiên bản nhân hiện hành. Bạn tìm nhân mới ở **ftp://ftp.kernel.org**.

6.1-Các tùy chọn cấu hình nhân có liên quan tới an toàn

- CONFIG_FIREWALL
- CONFIG_IP_FORWARD
- CONFIG_SYN_COOKIES
- CONFIG_IP_FIREWALL
- CONFIG_IP_FIREWALL_VERBOSE
- CONFIG_IP_NOSR
- CONFIG_IP_MASQUERADE
- CONFIG_IP_MASQUERADE_ICMP
- CONFIG_IP_TRANSPARENT_PROXY
- CONFIG_IP_ALWAYS_DEFRAG
- CONFIG_NCPFS_PACKET_SIGNING
- CONFIG_IP_FIREWALL_NETLINK

6.2-Các thiết bị nhân

Có vài thiết bị khối và thiết bị kí tự có sẵn trên Linux mà giúp bạn bảo vệ hệ thống. Hai thiết bị mà nhân cung cấp là /dev/random và /dev/urandom cung cấp dữ liệu ngẫu nhiên (random data) ở bất kỳ thời gian nào.

Cả /dev/random và /dev/urandom nên an toàn để sử dụng trong việc sinh các khóa PGP, thách thức của ssh và các ứng dụng khác mà ở đó các số ngẫu nhiên bảo vệ được yêu cầu. Các kẻ tấn công không thể dự đoán các số kế tiếp khi cho bất kỳ một dãy số khởi đầu nào từ các nguồn tài nguyên này.

Sự khác nhau giữa hai thiết bị này là /dev/random chạy sinh ra các byte ngẫu nhiên. /dev/random là entropy chất lượng cao, được sinh ra theo phương pháp ngắt thời gian. /dev/urandom thì tương tự, nhưng khi dự trữ của entropy thấp thì nó sẽ trở lại hàm hash mã hóa mạnh của những gì nó có. Điều này thì không an toàn, nhưng nó đủ cho hầu hết các ứng dụng.

Bạn có thể đọc các thiết bị này sử dụng lệnh như ví dụ sau:

```
root# head -c 6 /dev/urandom | mimecode
```

lệnh này sẽ in ra tám ký tự ngẫu nhiên trên console, phù hợp cho sinh mật khẩu. Bạn có thể tìm mimencode trong gói metamail. Xem trong /usr/src/drivers/char/random.c biết sự mô tả giải thuật.

7- An toàn mạng

An toàn mạng ngày càng quan trọng hơn khi mọi người mất nhiều thời gian để kết nối. Sự thỏa hiệp an toàn mạng thì dễ dàng hơn thỏa hiệp vật lý hoặc thỏa

hiệp an toàn cục bộ. Có một vài công cụ tốt để giúp đỡ vấn đề an toàn mạng, và nhiều trong số chúng có quan hệ với phân phối của Linux.

7.1- Bộ lắng nghe gói (packet sniffer)

Một trong những cách chung nhất các kẻ xâm nhập trái phép có được sự truy nhập tới nhiều hệ thống trên mạng của bạn đó là bởi dùng một bộ lắng nghe gói trên một máy host đã bị thỏa hiệp rồi. "Sniffer" chỉ lắng nghe trên cổng Ethernet các vấn đề như passwd, login và su trong luồng gói và ghi đường truyền sau đó. Với cách này, các kẻ xâm nhập trái phép có các mật khẩu của hệ thống mà không phải cố gắng phá vỡ nó. Các mật khẩu ở dạng rõ thì rất nguy hiểm bởi sự tấn công kiểu này.

Trong thời gian gần đây, các kẻ xâm nhập trái phép thậm trí không cần thỏa hiệp một hệ thống để thực hiện sự tấn công này: chúng có thể mang một máy tính xách tay (laptop) hoặc một PC và kết nối nó vào mạng của bạn.

Sử dụng ssh hoặc các phương pháp mã hóa khác để ngăn cản sự tấn công này. Các chương trình như APOP cho POP cũng ngăn cản đươc sự tấn công kiểu này.

7.2-Các dịch vụ hệ thống và tcp_wrappers

Trước khi bạn đặt hệ thống Linux trên bất kỳ một mạng nào thì điều đầu tiên là xem những dịch vụ nào mà bạn cần đưa ra. Các dịch vụ mà bạn không cần sử dụng thì nên được xóa bỏ để bạn khỏi lo lắng về nó và các kẻ tấn công có ít cơ hội hơn để tìm một kẽ hở an toàn.

Có một số cách để loại bỏ các dịch vụ dưới Linux. Bạn có thể xem trong file /etc/ inetd.conf để thấy những dịch vụ nào thì đang được cung cấp bởi inetd. Xóa bỏ bất kỳ dịch vụ nào mà bạn không cần bằng cách thêm dấu # ở đầu dòng đó và rồi gửi tiến trình inetd một SIGHUP.

Ngoài ra bạn có thể xóa các dịch vụ trong file /etc/services (thay vì ghi chú nó ở đầu dòng). Điều này có nghĩa rằng các client cục bộ sẽ không thể tìm được các dịch vụ này. Thường không có phiền toái gì khi xóa các dịch vụ khỏi /etc/services, bởi vì nó không cung cấp thêm sự bảo vệ nào.

Sau đây là một vài dịch vụ mà bạn cần xóa bỏ là:

- ftp
- telnet (hoặc ssh)
- mail, như pop-3 hoặc imap
- identd

Nếu bạn biết bạn sẽ không sử dụng một vài gói cụ thể, thì bạn có thể xóa nó toàn bộ, sử dụng lệnh rpm -e <tên gói> của RPM để xóa toàn bộ gói.

Bạn nên kiểm tra thư mục /etc/rc.d/rc[0-9].d để xem liệu có bất kỳ server nào được bắt đầu trong thư mục này thì không cần thiết. Những file trong thư mục này thì là những liên kết tượng trưng tới những file trong thư mục /etc/rc.d/init.d. Đặt lại tên file trong thư mục init.d để xóa bỏ tất cả các liên kết biểu tượng tới những file trong rc.d hoặc thay đổi tên file tương ứng với dịch vụ mà bạn muốn xóa bỏ.

Đa số các phân phối Linux có tcp_wrappers "chọc thủng" tất cả các dịch vụ TCP. Một tcp_wrapper (tcpd) được gọi từ inetd thay vì server thực sự. tcpd kiểm tra host mà đang yêu cầu dịch vụ này và chạy server thực sự hoặc từ chối truy nhập từ host đó. tcpd cho phép bạn hạn chế truy nhập tới các dịch vụ TCP. Bạn nên tạo một /etc/hosts.allow và thêm trong thư mục này các host mà cần có truy nhập tới các dịch vụ của máy bạn. Nếu bạn là một người dùng quay số bình thường thì bạn nên từ chối tất cả. tcpd cũng ghi lại các cố gắng truy nhập tới các dịch vụ bị thất bại bởi vậy điều này cảnh báo bạn nếu bạn bị tấn công. Nếu bạn thêm các dịch vụ mới, thì bạn nên cấu hình chúng để sử dụng tcp_wrappers nếu các dịch vụ này dựa trên TCP. Nhớ rằng tcp_wrappers chỉ bảo vệ các dịch vụ được chạy từ inetd và một vài dịch vụ lựa chọn khác.

7.3-Kiểm tra thông tin DNS

Nâng cấp và nhập nhật thông tin DNS về tất cả các host trên mạng thì có thể giúp bạn tăng khả năng an toàn. Nếu một host không được ủy quyền kết nối tới mạng của bạn thì bạn có thể nhận ra nó bởi thiếu một mục trong DNS của nó. Nhiều dịch vụ có thể được cấu hình để không chấp nhận các kết nối từ các host mà không có các mục DNS hợp lý.

7.4-identd

identd là một chương trình nhỏ mà chạy inetd server của bạn. Nó giữ dấu vết mà người dùng nào đang chạy dịch vụ TCP gì, và rồi trả lời những thông tin này khi bất kỳ ai yêu cầu nó. Bạn nên cho phép chạy chương trình này. Nhiều người không hiểu sự hữu ích của identd và xóa bỏ nó hoặc ngăn chặn tất cả các site yêu cầu nó.

7.5- sendmail, qmail

Một trong hầu hết các dịch vụ quan trọng bạn có thể cung cấp là một mail server. Không may thay, dịch vụ này thì nguy hiểm cho sự tấn công bởi vì một số nhiệm vụ mà nó phải thực hiện và đặc quyền nó cần.

Sendmail có một lịch sử rất dài về khai thác an ninh, bởi vậy nếu bạn sử dụng senmail thì bạn nên nhập nhật các phiên bản hiện hành của nó. Nhớ rằng senmail chạy không phải cho mục đích gửi thư. Nếu bạn là người sử dụng bình thường bạn nên xóa bỏ toàn bộ senmail và sử dụng mai client để gửi thư.

qmail là dịch vụ có chức năng hoàn toàn như senmail nhưng nó được thiết kế an toàn hơn, ổn định và nhanh hơn.

7.6-Tấn công từ chối dịch vụ

Một tấn công từ chối dịch vụ (denial of service - DoS) là một nơi mà kẻ tấn công cố gắng tạo nên một vài nguồn tài nguyên quá bận để trả lời các yêu cầu hợp lệ, hoặc để từ chối các người dùng hợp pháp truy nhập tới máy của bạn.

Tấn công từ chối dịch vụ đang được tăng lên trong những năm gần đây. Một vài dạng tấn công thông dụng của kiểu tấn công này là: SYN Flooding, Pentium "F00F" Bug, Ping Flooding...Để biết kỹ về các dạng tấn công này bạn tìm ở http://www.rootshell. com.

7.7-An toàn NFS (Network File System)

NFS là một giao thức chia sẻ file được sử dụng rộng rãi. Nó cho phép các server chạy nfsd và mountd để gắn kết toàn bộ hệ thống file tới các máy khác sử dụng sự hỗ trợ hệ thống file NFS được xây dựng trong nhân của các máy đó. mountd giữ dấu vết của hệ thống file được gắn kết trong thư mục /etc/mtab và có thể xem chúng bằng lệnh showmount. Nếu bạn phải sử dụng NFS thì chắc chắn bạn gắn kết tới chỉ những máy mà bạn thực cần. Không gắn kết toàn bộ thư mục gốc. Để có thông tin nhiều hơn về NFS bạn có thể tìm ở http:// metalab.unc.edu/mdw/HOWTO/NFS-HOWTO.html

7.8- NIS (Network Information Service) - Dịch vụ thông tin mạng

NIS là một phương pháp phân phối thông tin tới một nhóm các máy. NIS chủ (master) giữ các bảng thông tin và biến đổi chúng thành các file ánh xạ NIS. Những file ánh xạ này được phân phát trên khắp mạng, cho phép máy NIS khách (client) có thông tin đăng nhập, mật khẩu, thư mục riêng và thông shell (tất cả các thông tin này trong một file chuẩn /etc/passwd). Điều này cho phép người dùng thay đổi mật khẩu của họ và tạo ra ảnh hưởng trên tất cả các máy trong vùng NIS.

NIS thì không an toàn. Bất kỳ ai có thể dự đoán tên vùng NIS của bạn thì có thể có một bản sao file passwd và sử dụng "crack" hoặc "John the Ripper" để phá các mật khẩu của người dùng. Ngoài ra nó có thể đánh lừa NIS và thực hiện các mánh khóe hiểm ác. Nếu bạn sử dụng NIS thì bạn phải có hiểu biết về mối nguy hiểm đó. Để có thông tin thêm về NIS bạn có thể tìm ở http://metalab.unc.edu/mdw/HOWTO/NIS-HOWTO.html.

7.9- Firewalls

Firewalls là một phương pháp điều khiển thông tin nào thì được phép vào và ra từ mạng cục bộ. Có một số kiểu firewalls và phương pháp thiết lập chúng. Các máy Linux tạo firewalls khá tốt. Mã firewalls có thể được xây dựng trong nhân 2.0 hoặc cao hơn. Công cụ ipfwadm cho nhân 2.0 và ipchains cho nhân 2.2 cho phép bạn thay đổi các dạng đường truyền (traffic) mạng.

Firewall là một kỹ thuật rất hữu ích và quan trọng để bảo vệ mạng của bạn. Tuy nhiên không nên nghĩ rằng vì bạn đã có firewalls mà bạn không cần bảo vệ các máy ở sau nó. Điều này là một lỗi tai họa. Để có thông tin nhiều hơn về firewalls và Linux bạn có thể tìm ở http://metalab.unc.edu/mdw/HOWTO/Firewall-HOWTO. html. Để có thông tin về ipfwadm (công cụ để bạn thay đổi thiết lập trên firewalls) bạn có thể tìm ở **http://www.xos.nl/linux/ipfwadm/**

7.10- IP Chains - Linux Kernel 2.2.x Firewalling

Linux IP Firewalling Chains là một sự nâng cấp tới mã Linux firewalling cho nhân 2.2. Nó có nhiều đặc điểm hơn so với ứng dụng trước, bao gồm:

- Thao tác với gói mềm dẻo hơn
- Tài khoản phức tạp hơn
- Chính sách đơn giản thay đổi tự động
- Fragments có thể được ngăn chặn, từ chối...
- Ghi lại các gói nghi ngờ
- Có thể quản lý các giao thức khác ngoài các giao thức ICMP/TCP/UDP

Để có thêm thông tin về IP Chains bạn có thể tìm ở http://www.rustcorp.com/linux /ipchais/HOWTO.html.

7.11- VNPs - Virtual Private Networks

VPN là một cách để thiết lập một mạng "ảo" trên đỉnh một vài mạng đã tồn tại rồi. Mạng ảo này thường được mã hóa và chuyển đến đường truyền chỉ tới và từ một vài thực thể được biết mà đã được gắn với mạng này. VNP thường được sử dụng để kết nối với một ai đang làm việc ở nhà trên mạng Internet công cộng tới một mạng của công ty bên trong.

Có một vài giải pháp Linux VNP có sẵn ở:

- vnpd. Xem ở http://sunsite.auc.dk/vpnd/.
- Free S/Wan, có sẵn ở http://www.xs4all.nl/~freeswan/
- ssh có thể được sử dụng để xây dựng một VNP. Xem VNP mini-howto
- vps (virtual private server) ở http://www.strongcrypto.com.

8-Các công việc chuẩn bị để bảo vệ hệ thống của bạn (trước khi đặt nó vào trực tuyến).

Sau tất cả các mục trên thì bạn có thể kiểm tra hệ thống của bạn và xác định nó thì có khả năng an toàn. Tuy nhiên, có một vài công việc mà bạn nên thực hiện bây giờ nhằm mục đích chuẩn bị đối phó với một sự xâm nhập trái phép:

- Lưu trữ đầy đủ dữ liệu máy của bạn.
- Chọn lịch lưu trữ tốt.
- Lưu trữ file cơ sở dữ liệu của RPM hoặc Debian, các file cơ sở dữ liệu của RPM được chứa trong thư mục /var/lib/rpm.
- Giữ dấu vết của dữ liệu tài khoản hệ thống.

- Áp dụng tất cả các nhập nhật mới vào của hệ thống.

CHƯƠNG 2
LOGIN VÀ XÁC THỰC NGƯỜI DÙNG

Phần này chúng tôi mô tả chi tiết về quá trình đăng nhập (từ khi hiện dấu nhắc login cho tới khi xác thực xong - hệ thống đưa ra dấu nhắc shell), phương pháp xác thực người dùng, cách quản lý người dùng trên hệ thống Linux.

1-Đăng nhập - Login

Quá trình đăng nhập hệ thống được thực hiện bởi ba chương trình là init, getty và login. Trình init khởi tạo tập các tiến trình khác nhau tuỳ theo mức chạy (runlevel). Sau đó nó gọi chương trình getty và trao điều khiển cho chương trình này. Có thể mô tả tổng quát quá trình đăng nhập như hình vẽ dưới.

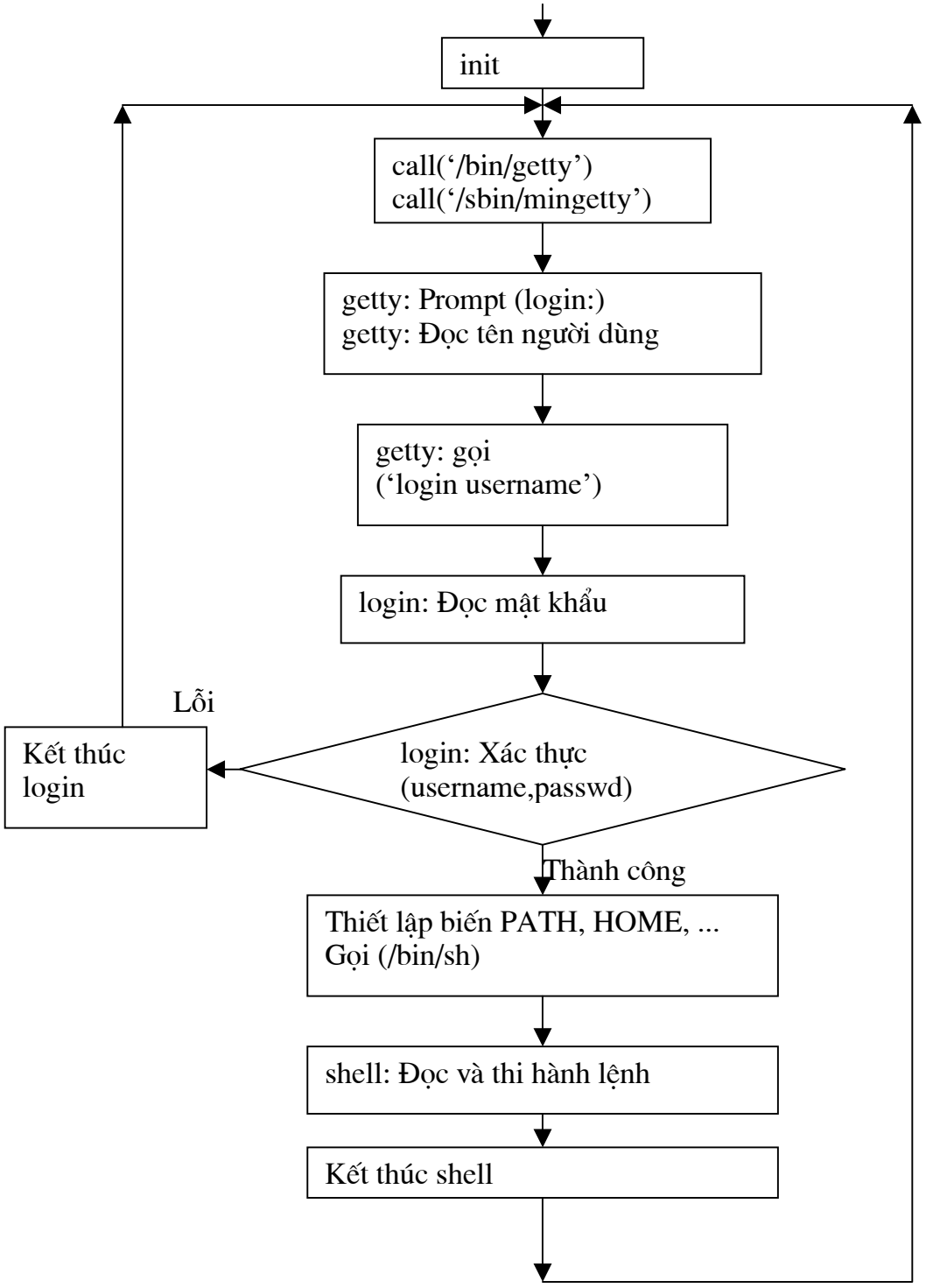
1.1- Trình getty

Getty là chương trình cho phép bạn đăng nhập bằng thiết bị nối tiếp chẳng hạn như "virtual terminal", text terminal hoặc modem. Chương trình getty và login sẽ thực hiện kiểm tra, xác thực và cho phép người dùng đăng nhập (log in) hệ thống.

Trình getty có nhiệm vụ sau:

- Mở tuyến (line) tty và thiết lập chế độ cho chúng.
- In dấu nhắc login, và lấy tên của người dùng.
- Khởi động tiến trình login cho người dùng.

Cụ thể: đầu tiên getty mở tuyến (line) để đọc và viết, và cấm bộ đệm vào ra chuẩn. Sau khi khởi tạo, line sẽ được đóng lại và mở lại. Tại thời điểm này, line được mở ở chế độ khối. Tiếp theo, getty đưa ra dòng login banner (thường được đọc từ file /etc/issue) và đưa ra dấu đăng nhập. Cuối cùng getty đọc tên đăng nhập của người dùng và gọi trình login với tham số là tên người dùng. Trong khi đọc tên, getty cố gắng tạo tốc độ terminal cho phù hợp với hệ thống để sử dụng, và cũng thiết lập các tham số cho terminal. Getty quét file gettydefs để tìm đề mục phù hợp. Nếu không có tốc độ được đưa vào, nó sẽ lấy đề mục đầu tiên trong file /etc/gettydefs. Trong trường hợp file /etc/gettydefs không thể truy cập được, thì đề mục ngầm định đã compiled-in được sử dụng.



QUÁ TRÌNH ĐĂNG NHẬP HỆ THỐNG

Khi ta vào chế độ đơn người dùng (mức chạy 1, S hoặc s) hệ thống sẽ không yêu cầu ta phải xác thực - đưa luôn dấu nhắc cho hệ vỏ shell. Đây có lẽ cũng là một kẽ hở trong vấn đề bảo mật hệ thống Linux. Không những thế ở mức chạy này,

người dùng cũng có các đặc quyền như người dùng root thông thường. Tất nhiên ta có thể hạn chế bằng cách bỏ mức chạy 1, được thiết lập trong file /etc/inittab.

Có nhiều chương trình getty khả dụng trên hệ thống Linux: mgetty (smart modem getty) được thiết kế để khởi tạo modem, vboxgetty (isdn voice box getty) sử dụng cho hệ thống isdn, agetty (chương trình trong bản Debian, có thể sử dụng cho console ảo, terminal, và modem), mingetty (trình getty tối thiểu được thiết kế để quản lý các console ảo), ... Trên hệ thống Linux thông thường, sử dụng trình mingetty (viết tắt của chữ minimal getty). Không giống như trình agetty, mingetty không thể sử dụng cho các line nối tiếp (serial line).

1.2- Trình login

Login được sử dụng khi đăng nhập vào hệ thống. Nó cũng có thể được sử dụng để chuyển từ người dùng này sang người dùng khác ở bất cứ thời điểm nào. Nếu không có tham số (tên người dùng) đi kèm, login sẽ nhắc nhập tên người dùng để đăng nhập vào hệ thống.

Login được trình getty gọi với tham số là tên người dùng. Quá trình thực hiện của trình login được mô tả như sau: Nếu người dùng không là root và tồn tại file /etc/nologin, thì nội dung của file này sẽ được in ra màn hình, login bị ngắt và người dùng không được đăng nhập hệ thống. Đây là một cách để bảo vệ login khi chuẩn bị tắt hệ thống của người quản trị. Nếu người dùng là root, thì tên đăng nhập phải được nhập trên console có tên trong file /etc/securetty. Các lỗi đăng nhập đều được ghi bởi syslog trong thư mục /var/log/. Sau khi kiểm tra xong các điều kiện trên, login sẽ yêu cầu mật khẩu và thực hiện kiểm tra mật khẩu cho tên người dùng. Quá trình kiểm tra tên tài khoản, mật khẩu được gọi quá trình xác thực người dùng trên hệ thống. Vấn đề xác thực người dùng trên hệ thống được trình bày chi tiết ở phần sau.

Giả sử rằng quá trình kiểm tra, xác thực người dùng trên hệ thống tiến hành thành công - trình login cho phép người dùng được đăng nhập vào hệ thống. Login sẽ tiếp tục thực hiện công việc sau:

- Nếu tồn tại file .hushlogin, thì login không thực hiện việc kiểm tra thư (mail) và in ra thời gian đăng nhập cuối và thông báo trong ngày.
- Nếu không có file .hushlogin, nhưng tồn tại file /var/log/lastlog thì thời gian đăng nhập cuối cùng sẽ được in ra màn hình và thời gian đăng nhập hiện tại lại được ghi vào đó.
- Nếu login không tìm thấy file .hushlogin, một thông điệp sẽ được in và tiến hành kiểm tra file trùng với tên người dùng trong thư mục /var/spool/mail/. Một thông điệp sẽ được in ra màn hình, nếu như file này có độ dài khác 0. Khi này, shell của người dùng (thiết lập trong file /etc/passwd) được khởi động. Nếu không có shell nào được chỉ ra cho người dùng trong file /etc/passwd, thì mặc định /bin/sh sẽ được gọi. Và nếu không có thư mục chủ nào xác định trong file /etc/passwd, thì thư mục gốc (/) được sử dụng.

- Tiếp theo login tiến hành thiết lập số định danh người dùng UID và GID của tty hiện đang đăng nhập, các biến môi trường cho TERM (terminal). Sau đó thiết lập các biến môi trường HOME, PATH, SHELL, TERM, MAIL và LOGNAME. Biến PATH ngầm định được thiết lập là /usr/local/bin:/bin:/usr/bin: cho người dùng thông thường, và /sbin:/bin:/usr/sbin:/usr/bin cho người dùng root.

/etc/securetty: file này chứa danh sách tên các thiết bị tty - console mà người dùng root được phép đăng nhập. Mỗi dòng tương ứng với một đề mục là tên của thiết bị tty, không có /dev/ chỉ ra ở trước. Nếu file này không tồn tại, người dùng root sẽ được phép đăng nhập trên bất kỳ console (tty) nào.

/etc/login.def: file này thuộc gói shadow, cung cấp một số thiết lập thêm về tuổi thọ mật khẩu, độ dài tối thiểu cho mật khẩu...

2- Tài khoản, quản lý tài khoản và xác thực người dùng trên hệ thống.

2.1- Tài khoản người dùng

Tất cả mọi người muốn sử dụng hệ thống đều phải có một tài khoản. Tài khoản này gồm hai phần: tên người dùng (username) và mật khẩu (password). Tên người dùng còn được gọi là tên tài khoản (account name) hay tên định danh - để hệ thống biết được bạn là ai. Mật khẩu được dùng để xác thực (authenticator), chứng minh với hệ điều hành là tài khoản đúng của bạn.

Hệ thống sẽ được chia thành các nhóm người dùng, mỗi nhóm được xác định một số quyền nhất định khác nhau. Trong đó có một số tài khoản, nhóm đặc biệt có hầu hết các quyền thao tác trên hệ thống, đó là người dùng root và siêu người dùng. Cơ sở dữ liệu cho các tài khoản trên hệ thống được lưu trữ trên 2 file dữ liệu quan trọng; /etc/passwd cho các tài khoản người dùng, /etc/group cho nhóm người dùng trên hệ thống.

a. File /etc/passwd

Linux sử dụng file /etc/passwd để chứa danh sách tất cả tài khoản người dùng trên hệ thống; ID người dùng, ID nhóm, thư mục chủ, shell,v.v... Thông thường nó cũng chứa mật khẩu đã mã hoá cho mỗi tài khoản. Thông thường file này chỉ có quyền đọc (trừ root); nhiều chương trình (chẳng hạn ls) sử dụng file này để ánh xạ ID người dùng với tên người dùng. Dữ liệu trong tệp tin này được ghi theo định dạng sau:

account:password:UID:GID:GECOS:directory:shell

Trong đó:

account: Tên người dùng trên hệ thống, không được chứa ký tự hoa.
password: Mật khẩu đã hoá, hoặc ký tự *, !.
UID: Số định danh cho người dùng này.
GID: Số định danh nhóm mà người dùng này trực thuộc.
GECOS: Trường này là tuỳ ý, dùng để ghi thông tin thêm cho người dùng; tên

đầy đủ hoặc dòng chú thích.
directory: Thư mục chủ cho tài khoản.
shell: chương trình shell được dùng sau khi đăng nhập. Nếu trường này để trống, login sẽ sử dụng ngầm định là chương trình /bin/sh.

Ví dụ:

```
root:ml5jVpx5RdoiI:0:0:root:/root:/bin/bash
toannq:.myDtwzWuzKuY:500:500::/home/toannq:/bin/bash
thai:4mxoEQQwVU6cQ:502:507::/home/thai:/bin/bash
```

b. File /etc/group

File này chứa cơ sở dữ liệu của tất cả các nhóm người dùng trong hệ thống và tương ứng với mỗi nhóm là số định danh nhóm GID. Định dạng của file này cũng tương tự như định dạng được sử dụng trong file /etc/passwd.

<group>:<password>:<gid>:<members>

Trong đó:

<group> là tên nhóm.
<password> chứa mật khẩu đã mã hoá cho nhóm.
<gid> số định danh cho nhóm người dùng.
<members> các thành viên của nhóm người dùng.

Ví dụ:

```
root:/AZkxFwZvDXZQ:0:root
toannq:1XA/Tq4uqjGnI:500:
thaith:suUZ2ViM6ut5k:507:
```

Hai file này được sử dụng bởi rất nhiều chương trình hệ thống. Khi đăng nhập trình login sẽ đọc thông tin về người dùng, mật khẩu, shell, thư mục chủ,... từ hai file này để quyết định các bước sử lý với từng người dùng. Các ứng dụng khác có thể sử dụng hai file này để ánh xạ giữa từ uid sang tên người dùng. Mọi người trên hệ thống đều có thể đọc được hai file này và có thể lấy trường mật khẩu đã mã hoá của tất cả mọi tài khoản từ 2 tệp này.

Trên quan điểm bảo mật hệ thống, hai tệp tin /etc/passwd và /etc/group là hai file quan trọng bậc nhất; Nếu ta có thể thay đổi nội dung của file này, thì ta có thể thay đổi mật khẩu của bất kỳ người dùng nào, hoặc có thể tạo một tài khoản siêu người với các đặc quyền siêu người dùng.

2.2-Mật khẩu - phương pháp mã hoá

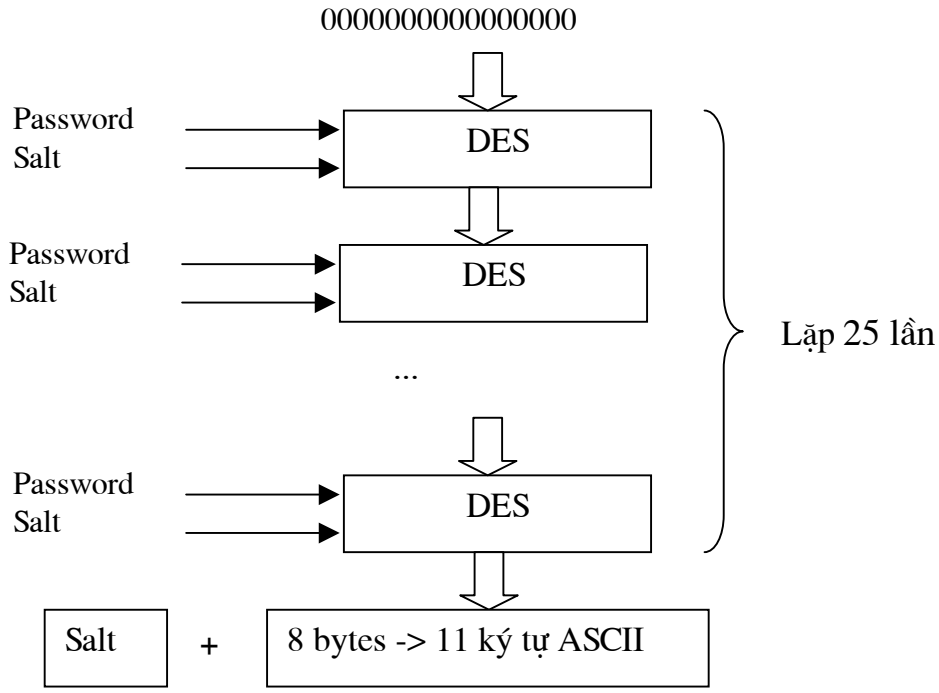
Mật khẩu của tất cả người dùng trên hệ thống phải được lưu trữ trên một file CSDL (cụ thể là /etc/passwd và /etc/group). Để tránh các truy cập bất hợp pháp hoặc tấn công vào hệ thống, file CSDL lưu trữ các mật khẩu người dùng trên hệ thống phải được bảo vệ một cách rất cẩn thận; cả về mặt vật lý cũng như độ phức tạp đối với kẻ tấn công. Trên hệ thống Unix cũng như Linux hiện nay, việc này được thực hiện bằng cách dùng hàm được coi là một chiều, mã hoá các mật khẩu người dùng trước khi lưu giữ nó lên file. Ngoài ra, khi đọc mật khẩu người dùng, hệ thống không hiển thị số ký tự trên màn hình. Điều này phần nào cũng làm tăng

độ bí mật của mật khẩu đối với một kẻ tò mò nào đó. Thư viện glibc (với Linux) đã cung cấp hàm mã hoá crypt() được coi là một chiều dựa trên Thuật toán Chuẩn mã dữ liệu DES và Thuật toán hàm băm MD5.

+ Hàm crypt(): hàm này được khai báo như sau:

```
char * crypt (const char * key, const char *salt)
```

Hàm này mã hoá mật khẩu, dựa trên thuật toán Chuẩn mã dữ liệu DES - 8 vòng. Nó lấy mật khẩu của người dùng làm khoá, để mã với khối rõ (64 bits) không (zero). Kết quả là 64 bit bản mã lại được mã lại với mật khẩu của người dùng; tiến trình này được lặp lại 25 lần. 64 bit mã cuối cùng được kết hợp với giá trị salt (để tạo ra 4096 khả năng có thể khác), sau đó nó được "chuyển" thành 11 ký tự dạng mã ASCII (chỉ cần 6 bit cho một ký tự trong tập [a-zA-Z0-9. /]).



Sơ đồ mã hoá của hàm crypt()

Tham số salt thực hiện 2 việc: thứ nhất, nó được dùng để chọn thuật toán sử dụng để mã hoá: dựa trên MD5 hay DES. Thứ hai: nó làm kẻ tấn công vất vả hơn trong việc dò tìm mật khẩu.

Phương pháp mã hoá mật khẩu này được tóm tắt trong sơ đồ mã hoá của hàm crypt ở trên.

Thuật toán mã hoá sử dụng thuật toán DES (8 vòng) có sửa đổi một chút như sau: salt là một số 12 bits (từ 0 tới 4095) dùng để thay đổi kết quả đầu ra của hàm DES - tránh trùng nhau trong 25 vòng lặp, bản rõ vào là 8 bytes không (zero). Khi thiết lập mật khẩu giá trị salt được lấy là ngẫu nhiên được kết hợp với khoá 8 bytes đầu vào (mật khẩu), chương trình sẽ tạo thành một dãy gồm 4096 khoá con khác. Kết quả cuối cùng là 8 bytes đã mã hoá được chuyển thành xâu 11 ký tự ghép với 2 ký tự biểu diễn giá trị của salt và được lưu vào trường mật khẩu đã mã hoá của file /etc/passwd. Rõ ràng, cách mã hoá theo thuật toán này có sự hạn chế là chỉ 8 ký tự đầu (của mật khẩu) có ý nghĩa trong việc xác thực.

+ Thuật toán mã hoá MD5:

Đây là một thuật toán được bổ sung cho hàm crypt, nhằm khắc phục điểm yếu của thuật toán mã hoá truyền thống của Unix.

Với thuật toán mã hoá dựa trên MD5, salt là 1 xâu có độ dài 8 ký tự. Giá trị salt ghi trong file /etc/passwd, là các ký tự trong tập [./0-9a-zA-Z], được đánh dấu bắt đầu bằng xâu '1', kết thúc bằng một ký tự '$' khác. Như vậy, khi sử dụng mã hoá bằng thuật toán MD5, trường mật khẩu trong file /etc/passwd sẽ có độ dài tổng cộng là 34 ký tự (bao gồm 3 ký tự 1, 8 ký tự cho giá trị của salt, ký tự $ - kết thúc salt, 22 ký tự mật khẩu đã mã hoá).

Mã hoá dựa trên thuật toán MD5 có sẵn trong thư viện glibc. Với thuật toán mã hoá này không giới hạn về độ dài mật khẩu, do đó tính bảo mật cao hơn rất nhiều so với mã hoá dựa trên DES. Do đó, nó được dùng nhiều hơn so với thuật toán DES.

Khi thiết lập mật khẩu người dùng, giá trị cho salt được khởi tạo một cách ngẫu nhiên. Source code của hàm crypt() và md5 là một phần trong gói glibc. Ta có thể thay thế chương trình mã hoá mật khẩu này bằng một chương trình, thuật toán khác. Vấn đề này còn được tiếp tục nghiên cứu và phát triển trong một vài năm tới.

Ví dụ: một đề mục trong file /etc/passwd, mật khẩu được mã hoá bằng MD5

```
root:$1$Myq352Lp$X0lHLaWEykdzTfw63YcQy/:0:0:root:/root:/bin/bash
```

+ Cách xác thực người dùng:

Khi người dùng chọn một mật khẩu, mật khẩu này sẽ được mã hoá bằng một giá trị được tạo một cách ngẫu nhiên (gọi là 'salt'). Giá trị 'salt' sẽ được ghi cùng với mật khẩu đã được mã hoá (hai ký tự đầu tiên trong 13 ký tự đối với mã hoá theo DES và là 8 ký tự đầu - bắt đầu từ xâu '1' tới ký tự '$' khác trong trường mật khẩu khi sử dụng thuật toán MD5).

Ví dụ: một đề mục trong file /etc/passwd, mật khẩu được mã hoá bằng DES.

```
username:Npge08pfz4wuk:503:100:Full Name:/home/username:/bin/sh
```

Khi người dùng đăng nhập vào hệ thống và gõ mật khẩu, giá trị salt sẽ được lấy ra từ trường mật khẩu tương ứng với người dùng đó trong file /etc/passwd. Hàm crypt() sẽ tiến hành kiểm tra xem 3 ký tự đầu tiên có là '1' không, nếu có thì 8 ký tự tiếp theo là giá trị salt của MD5 và gọi hàm mã hoá MD5, nếu không thì 2 ký tự đầu là giá trị salt được sử dụng cho mã hoá DES. Hệ thống sẽ thực hiện mã hoá với mật khẩu người dùng nhập vào với giá trị salt đó và so sánh hai mật khẩu đã mã hoá này với nhau. Nếu trùng khớp, người dùng đã được xác thực và được phép đăng nhập vào hệ thống.

2.3- Mật khẩu shadow

+ Tại sao phải dùng mật khẩu shadow?

File CSDL mật khẩu (/etc/passwd và /etc/group) được thiết lập quyền chỉ đọc bởi các người dùng khác trên hệ thống. Do đó tất cả mọi người dùng (không phải root) trên hệ thống đều có thể đọc được toàn bộ dữ liệu trong file này. Mặc dù rất khó tính toán để khôi phục lại một mật khẩu đã mã hoá về dạng gốc, song không phải là không thể với kẻ tấn công. Kẻ tấn công có thể sử dụng phương pháp tấn công từ điển rất có hiệu quả; mã các từ thông dụng sử dụng 4096 giá trị salt có thể với mỗi từ đó. Khi đó, kẻ tấn công đã có một bảng thống kê các bản rõ, mật khẩu tương ứng với từng giá trị salt cụ thể. Chúng so sánh các mật khẩu đã mã hoá trong file /etc/passwd với các bảng CSDL của chúng. Nếu có một sự trùng hợp nào đó, thì chúng đã có một tài khoản để đăng nhập hệ thống, thậm trí là một tài khoản có các đặc quyền của siêu người dùng.

Một phương pháp khác không cần đến không gian đĩa để thực hiện tấn công từ điển, hiện nay có nhiều chương trình crack có thể phá được ít nhất một vài mật khẩu trên hệ thống với số người dùng đủ lớn nào đó. Chúng tôi đã chạy thử nghiệm chương trình crack mật khẩu john với một file passwd có 8 user và một file từ điển khoảng 44000 từ. Với các mật khẩu dễ và thông dụng, chương trình sẽ tìm ra ngay sau một thời gian ngắn. Ta có thể tìm được rất nhiều các file từ điển các từ thông dụng được cung cấp sẵn với nhiều thứ tiếng và sở thích, giới tính, ... Ta cũng có thể chạy thử một số chương trình crack có thể tìm thấy trong một số đĩa CDROM "Hacker".

***Một giải pháp khác*:** tại sao chúng ta không thiết lập quyền, không cho phép ai (trừ người dùng root), được phép đọc-ghi file CSDL này. Chúng ta hãy trở lại với quá trình đăng nhập hệ thống. File này ngoài trường mật khẩu - chỉ được sử dụng bởi chương trình login, nó còn chứa các thông tin về tên người dùng, thư mục chủ, uid, shell... Mỗi khi đăng nhập trình login sẽ quét các đề mục trong file /etc/passwd để kiểm tra và xác thực người dùng. Nếu quá trình xác thực thành công (người dùng được phép đăng nhập hệ thống). Trình login dựa vào file /etc/passwd thiết lập số định danh cho người dùng hệ thống, các biến môi trường, thư mục chủ, và các quyền tương ứng cho người dùng này. Các chương trình sau đó sẽ truy cập hệ thống chỉ thông qua số định danh (uid) này. Khi các chương trình cần sử dụng tên người dùng trên hệ thống nó sẽ đọc từ file /etc/passwd này để ánh xạ lại từ số

uid sang tên người dùng (hoặc từ gid sang tên nhóm với file /etc/group). Chẳng hạn, nếu ta thiết lập quyền không thể đọc-ghi cho mọi người trên hệ thống. Mỗi khi đăng nhập hệ thống với user thường, dấu nhắc bash sẽ có dạng như sau:

[I have no name@root]#

và rất nhiều các chương trình khác sẽ không thể hiển thị được tên người dùng hiện đang đăng nhập trên hệ thống như chương trình ps, ls -l, ... Do đó không thể thiết lập quyền không thể đọc cho file /etc/passwd.

Shadow là một giải pháp giải quyết vấn đề này bằng cách "giấu" các mật khẩu ra một file khác (thông thường là /etc/shadow) - và thiết lập quyền không ai (trừ root) có thể truy cập file này. Mỗi khi đăng nhập trình login sẽ quét file /etc/passwd, xác định người dùng và kiểm tra mật khẩu của người dùng (qua file /etc/shadow), xác định thư mục chủ, shell sẽ sử dụng. Bằng cách này sẽ ngăn chặn có hiệu quả với tấn công từ mật khẩu đã mã hoá bằng phương pháp tấn công từ điển. Thêm vào đó, shadow còn thêm một số chức năng khác:

- Một tệp cấu hình cho các thiết lập đăng nhập ngầm định (/etc/login.defs)
- Các tiện ích cho phép thêm, sửa đổi, và xoá tài khoản người dùng và nhóm.
- 'Tuổi mật khẩu' và hạn sử dụng.
- Hạn tài khoản và khoá.
- Các mật khẩu shadow nhóm (tuỳ chọn).

+ File /etc/shadow

Gói shadow là một trong các gói để xây dựng một hệ thống bảo mật hơn trên hệ thống Linux. Khi sử dụng mật khẩu shadow, trường mật khẩu trong file /etc/passwd được thay bằng dấu x. Định dạng của mỗi đề mục trong file /etc/shadow như sau:

username:passwd:last:may:must:warn:expire:disable:researved

ý nghĩa các trường:

username	: tên người dùng.
passwd	: mật khẩu đã mã hoá.
last	: ngày thay đổi mật khẩu cuối cùng tính từ 01/01/1970.
may	: ngày trước khi mật khẩu có thể bị thay đổi.
must	: ngày sau khi mật khẩu phải thay đổi.
warn	: ngày trước khi mật khẩn hết hạn để cảnh báo người dùng.
expire	: ngày sau khi mật khẩu hết hạn và tài khoản bị cấm.
disable	: Số ngày tính từ 01/01/1970 cho đến khi tài khoản bị cấm .
reserved	: Trường dự phòng.

Ví dụ:

username:Npge08pfz4wuk:9479:0:10000: : : :

2.4- Cracklib và cracklib_dict

Đây là một công cụ, được sử dụng kết hợp với module pam_cracklib (trong PAM) để kiểm tra độ mạnh mật khẩu và nhắc nhở bạn. Cracklib và Cracklib_dict

là 2 gói luôn đi kèm với nhau. Gói cracklib chứa các file thư viện sau:

/usr/include/cracklib.h

/usr/lib/libcrack.so (link tới /usr/lib/libcrack.so.2.7)

/usr/lib/libcrack.so.2.7

File thư viện này phục vụ cho các chương trình trong gói cracklib_dict. Gói này bao gồm các file sau:

/usr/lib/cracklib_dict.hwm (1024 bytes)

/usr/lib/{cracklib_dict.pwd, cracklib_dict.pwi}

/usr/sbin/{create-cracklib-dict, mkdict, packer}.

Hai file thư viện cracklib_dict.pwd, cracklib_dict.pwi chứa dữ liệu là các từ thông dụng (từ điển), và có kích thước phụ thuộc vào file "từ điển" để tạo ra nó. Shell script chính create-cracklib-dict sử dụng script mkdict và chương trình packer để tạo lại các file thư viện cracklib_dict.*.

Cracklib là một thư viện chứa các hàm, sẽ được sử dụng bởi chương trình passwd. Nó được tạo ra với ý tưởng: giúp người dùng tránh chọn các mật khẩu dễ đoán bởi các chương trình crack, sử dụng phương pháp tấn công từ điển. Cracklib tiến hành nhiều bước kiểm tra để có thể xác định xem bạn có chọn một mật khẩu tồi hay không?

+ Nó xét xem bạn có tạo các mật khẩu từ tên người dùng không?
+ Kiểm tra các mẫu đơn giản.
+ Nó kiểm tra xem mật khẩu bạn chọn có nằm trong từ điển không.

Nếu các bước kiểm tra này được thực hiện tốt đẹp, thì có thể mật khẩu bạn chọn là một mật khẩu tốt. Tuy nhiên cũng cần lưu ý rằng, nó được sử dụng thông qua module pam_cracklib được thiết lập trong các file cấu hình ở thư mục /etc/pam.d/.

3- PAM

3.1- PAM là gì?

Chúng ta có thể thấy rằng, có một số cách khác nhau trong việc xác thực thông tin người dùng trên hệ thống (mật khẩu shadow không có MD5, mã hoá MD5 không shadow...). Thế làm thế nào để các chương trình, chẳng hạn chương trình su, login biết để kiểm tra mật khẩu, xác thực người dùng. PAM (Pluggable Authentication Modules) sẽ giải quyết vấn đề này.

Với hệ thống Linux không sử dụng shadow & MD5, nếu một chương trình chẳng hạn su, passwd, login hay xlock cần xác thực một người dùng, nó đơn giản đọc thông tin người dùng trong tệp tin /etc/passwd. Khi sử dụng mã hoá MD5 và shadow, mỗi chương trình yêu cầu xác thực sẽ được PAM chỉ ra chương trình nào đang được sử dụng, thông tin người dùng được lưu trữ ở đâu một cách trong suốt.

Với PAM, khi một chương trình yêu cầu xác thực một người dùng, nó cung cấp một thư viện chứa các hàm cho lược đồ xác thực. Vì các thư viện này là các thư viện động, nên ta có thể thay đổi lược đồ xác thực chỉ đơn giản bằng việc soạn thảo

các file cấu hình. Tính phức tạp chính là một trong những sức mạnh lớn nhất của PAM.

Linux-PAM (Pluggable Modules for Linux) là các thư viện chia sẻ (shared libraries), cho phép quản trị hệ thống lựa chọn cách xác thực người dùng. Nói cách khác, ta không phải biên dịch lại các ứng dụng sử dụng PAM (PAM-aware), và vẫn có thể chuyển đổi cách xác thực khác nhau. Thay vào đó, chúng ta chỉ việc nâng cấp toàn bộ hệ thống xác thực cục bộ mà không phải đụng vào bất kỳ ứng dụng nào.

Mục đích của dự án Linux-PAM là phát triển nhánh phần mềm bảo mật và các lược đồ xác thực (authentication scheme) một cách độc lập. Nghĩa là nó sẽ cung cấp thư viện các hàm để các ứng dụng sử dụng cho yêu cầu xác thực người dùng. Thư viện PAM được sử dụng thông qua các tệp cấu hình trên máy cục bộ /etc/pam.conf (hoặc một các file trong thư mục /etc/pam.d/). Các tệp đối tượng khả nạp động (dynamically loadable objects files), được gọi là các modules, thường được đặt trong /lib/security (với các phiên bản Red Hat).

***Tóm lại*:**

PAM là một phương pháp dùng module để xác thực người dùng và điều khiển các truy cập tới các dịch vụ trên hệ thống. Để xác định xem hệ thống có sử dụng PAM hay không, ta có thể sử dụng lệnh sau:

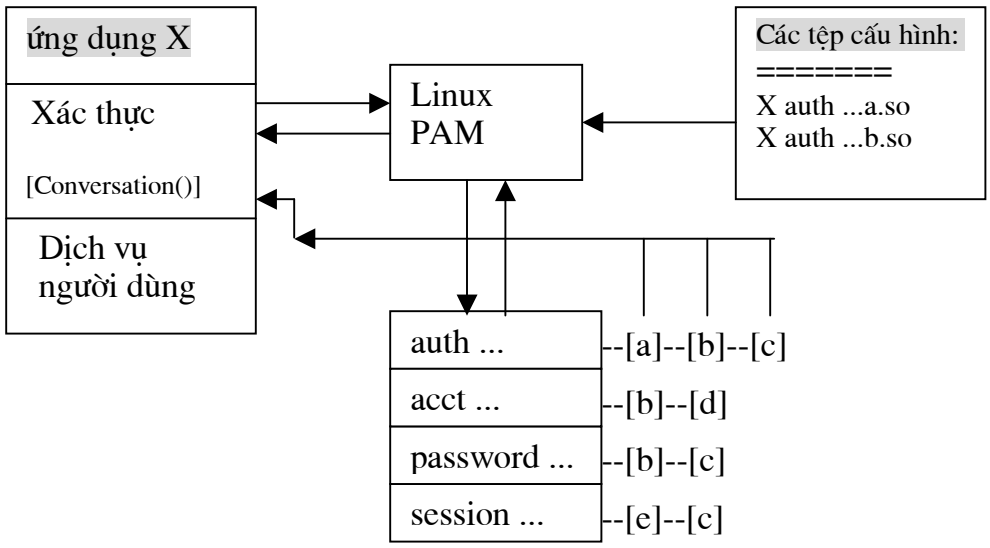
ldd /sbin/login

Nếu chương trình đăng nhập này có sử dụng thư viện libpam.so, thì hệ thống có sử dụng PAM. Các phiên bản Red Hat từ 5.0 trở lên, đều sử dụng PAM.

3.2- Tổng quan

Linux PAM có 4 kiểu tác vụ (quản lý) độc lập là: quản lý xác thực (authentication), quản lý tài khoản (account), quản lý phiên (session), và quản lý mật khẩu (password). Tổ hợp các lược đồ quản lý và cách đối xử với một ứng dụng được thiết lập bởi các đề mục trong file cấu hình của Linux PAM. Cú pháp của các file cấu hình này sẽ được mô tả ở phần dưới.

Ta có thể mô tả cách tổ chức tổng quát của Linux PAM.



Bên trái của hình biểu diễn ứng dụng X, giao diện ứng dụng này sử dụng thư viện Linux PAM, và nó không cần biết phương pháp xác thực được thiết lập cho nó. Thư viện Linux-PAM sẽ lấy nội dung file cấu hình PAM và nạp các modules phù hợp cho ứng dụng X. Các module này được đặt theo 4 nhóm quản lý và quản lý theo kiểu ngăn xếp theo thứ tự được đặt trong file cấu hình. Các module này khi được gọi, cho phép các cách xác thực khác nhau cho ứng dụng. Thông tin nguyên thuỷ cần thiết của người dùng sẽ được chuyển bằng hàm conversation().

Ta có thể sử dụng PAM để xác thực cho bất cứ chương trình nào nếu ta có quyền truy cập vào mã nguồn, và có thể thêm các hàm PAM phù hợp. Nói khác đi, nếu một chương trình sử dụng PAM, thì nó đã có các hàm ở trong chương trình. Nếu không có thì không thể sử dụng PAM. Để xác định được một chương trình có bao hàm PAM hay không, ta có thể xem các file trong thư mục /etc/pam.d/. Tên file cấu hình này đã được đặt trong mã chương trình, và thường trùng với tên của chương trình. Nếu có file này, thì ta có thể xác định được chương trình *prog* có hiểu PAM. Hoặc ta có thể sử dụng trình ldd; nếu chương trình không sử dụng các thư viện libpam và libpam_misc, thì chương trình không làm việc với PAM. Tuy nhiên phương pháp trên cũng không hẳn chính xác.

3.3- Cấu hình cho Linux PAM.

Các file cấu hình của PAM có thể được đặt trong các file thư mục /etc/pam.d/ trong một file /etc/pam.conf. Chúng ta sẽ trình bày cú pháp cấu hình theo kiểu các tệp trong thư mục /etc/pam.d/ trước. Cần chú ý rằng Linux PAM xác định các thẻ, các tham số trong file này phải là các chữ thường.

Cú pháp của file cấu hình:

PAM được cấu hình theo 2 cách khác nhau thông dụng: thiết lập cấu hình trong file /etc/pam.conf hoặc thông qua các tệp tin cấu hình trong thư mục /etc/pam.d. Tuy nhiên với 2 cách này cũng không có sự khác nhau lắm. Bản phân

phối của Red Hat thường sử dụng các file cấu hình trong /etc/pam.d/, như mô tả ở dưới:

Các file cấu hình PAM có cú pháp như sau:
type control module-path module-arguments

type báo cho PAM kiểu xác thực được sử dụng cho module này. Các module của cùng kiểu có thể được xếp chồng lên nhau. PAM nhận ra 4 kiểu xác thực sau:

account

Module này thực hiện quản lý tài khoản, không xác thực. Nó được sử dụng để hạn chế/cho phép truy cập tới một dịch vụ dựa trên thời gian hiện tại. Quyết định xem người dùng được phép truy cập dịch vụ, hoặc mật khẩu của họ đã hết hạn chưa, v.v.

auth

Quyết định xem người dùng là ai, thông thường xác thực bằng mật khẩu. Kiểu module này cung cấp cách xác thực người dùng. Đầu tiên nó xác định xem người dùng là ai bằng cách nhắc người dùng nhập mật khẩu hoặc cách xác định khác. Sau đó, module có thể xác nhận thành viên nhóm (dựa vào file /etc/group) hoặc các đặc quyền thông qua các thuộc tính uỷ quyền của nó.

password

Kiểu module này được yêu cầu để cập nhật xác thực được gán cho người dùng. Thông thường, có tương ứng một module cho mỗi kiểu module xác thực (auth). Cung cấp kỹ thuật cho người dùng thay đổi xác thực của họ, thông thường là thay đổi cách xác thực bằng mật khẩu của họ.

session

Các việc được thực hiện trước và/ hoặc sau khi người dùng được xác thực. Công việc bao gồm kết gắn/huỷ kết gắn thư mục home, đăng nhập login/logout, hạn chế/không hạn chế các dịch vụ khả dụng cho người dùng.

Trong file cấu hình login, sẽ có ít nhất một đề mục cho mỗi kiểu. Có thể hiểu rằng nó cần truy cập tất cả các kiểu xác thực khác nhau.

control

Trường này báo cho PAM cách thực hiện nếu như xác thực bằng module bị lỗi hoặc thành công. Khi các module được nạp, các cờ control này sẽ quyết định quan hệ quan trọng giữa mỗi module. PAM nhận ra 4 kiểu điều khiển sau:

requisite

Từ khoá này chỉ ra sự thành công của module này là cần thiết cho sự thành công của kiểu (type) module. Nếu xác thực bằng module này bị lỗi, thì ngay lập tức từ chối xác thực.

required

Từ khoá này chỉ ra rằng sự thành công của module này là cần thiết cho kiểu (type) module để thành công. Nếu gặp lỗi xác thực, PAM sẽ

vẫn gọi các modules (cùng kiểu) có trong danh sách ở dưới cho dịch vụ này trước khi từ chối xác thực.

sufficient

Nếu xác thực bằng module này thành công, PAM sẽ vẫn công nhận xác thực, mặc dù module 'required' trước đó bị lỗi.

optional

Như tên gọi của nó, cờ điều khiển này đánh dấu module này không có ý nghĩa quan trọng với sự thành công hoặc lỗi. Kết quả chỉ có thể là thành công hoặc lỗi mới có ý nghĩa cho dịch vụ này.

Trong file cấu hình cho login, chúng ta thấy module 'required' là pam_unix.so (module xác thực chính), module requisite là pam_securetty.so (kiểm tra để đảm bảo rằng người dùng đăng nhập trên một console bảo mật), và một modules 'optional' là pam_lastlogin.so (module thu lượm thông tin đăng nhập cuối cùng của người dùng gần nhất).

module-path

Chỉ ra module được sử dụng và đường dẫn để PAM tìm module đó. Hầu hết các file cấu hình chỉ chứa tên của các module, PAM sẽ tìm các module trong thư mục module ngầm định (thông thường là /lib/security).

module-arguments

Các module-arguments là danh sách các tham số được truyền tới cho module khi được gọi. Nói chung các tham số đúng là các tuỳ chọn và được xác định cho module cụ thể nào đó. Ví dụ: trong file cấu hình login, tham số "nullok" được truyền cho module pam_unix.so, chỉ ra rằng mật khẩu rỗng ("null") cũng chấp nhận được ("ok").

Bất kỳ một dòng nào đó trong file cấu hình được định dạng không đúng, sẽ tạo ra các lỗi tiến trình xác thực. Lỗi tương ứng sẽ được ghi ra file nhật ký hệ thống bằng cách gọi hàm syslog().

Ví dụ: file /etc/pam.d/login

```
#%PAM-1.0
auth       required /lib/security/pam_securetty.so
auth       required /lib/security/pam_pwdb.so shadow nullok
auth       required /lib/security/pam_nologin.so

account    required /lib/security/pam_pwdb.so

password   required /lib/security/pam_cracklib.so
password   required /lib/security/pam_pwdb.so nullok
use_authtok md5 shadow

session    required /lib/security/pam_pwdb.so
session    optional /lib/security/pam_console.so
```

File cấu hình pam.conf:

Là sự thay thế cho các file cấu hình trong thư mục /etc/pam.d/ (nếu bạn không thích để các file cấu hình trong thư mục này). File này có thể được lưu trữ trong thư mục /etc/pam.conf tương tự như các file cấu hình mô tả ở trên. Ví dụ: dòng sau trong /etc/pam.d/login:

```
auth required  pam_unix.so    nulok
```

sẽ trở thành:

```
login     auth required  pam_unix.so    nulok
```

Cú pháp cấu hình chung của file /etc/pam.conf có dạng như sau:

service-name type control module-path arguments

Trong đó type, control, module-path và argument hoàn toàn giống như đã mô tả ở trên. Service-name là tên của dịch vụ cần cấu hình với đề mục này. Thông thường Service-name là tên quy ước của ứng dụng. Ví dụ 'ftpd', 'login', 'other'... Cũng tương tự như các tên các file cấu hình theo kiểu cấu hình mô tả ở trên.

Ta có thể thấy rằng phương pháp cấu hình bằng các file trong thư mục /etc/pam.d/ có nhiều thuận tiện hơn so với cách cấu hình trên một tệp /etc/pam.conf:

- ít bị lỗi khi cấu hình một ứng dụng, ít hơn một trường.
- Dễ bảo trì; ứng dụng có thể được cấu hình lại độc lập với các ứng dụng khác.
- Quản lý gói đơn giản hơn.

Một số file cấu hình đặc biệt:

Mỗi một file trong thư mục /etc/pam.d/ chứa thông tin cấu hình cho một dịch vụ riêng nào đó. Thông thường tên của file cấu hình chính là tên của chương trình sử dụng nó. Ví dụ: file login là file cấu hình cho chương trình đăng nhập (/bin/login) hệ thống. File cấu hình /etc/pam.d/other là file cấu hình cho các dịch vụ, chương trình không có file cấu hình riêng. Ví dụ: khi một dịch vụ xyz nào đó cố gắng xác thực, PAM sẽ tìm file cấu hình /etc/pam.d/xyz, nếu không tìm thấy file này thì việc xác thực cho chương trình xyz được quyết định bằng file /etc/pam.d/other. Cấu hình ngầm định của file này như sau:

```
#%PAM-1.0
auth     required       /lib/security/pam_deny.so
account  required       /lib/security/pam_deny.so
password required       /lib/security/pam_deny.so
session  required       /lib/security/pam_deny.so
```

Với file cấu hình này, PAM từ chối xác thực (bằng module pam_deny.so) với các dịch vụ không được cấu hình. Ta có thể sử dụng file này để cấu hình hệ

thống một cách bảo mật hơn; xoá bỏ (hoặc đổi tên) các file cấu hình cho các dịch vụ, chương trình mà ta không sử dụng đến, ghi nhật ký các cảnh báo syslog bằng module module pam_warn.so. Cần lưu ý rằng: khi sử dụng file cấu hình này ta không nên xoá bỏ file cấu hình /etc/pam.d/login, nếu vô tình xoá nó thì không ai đăng nhập được hệ thống.

Ta có thể thấy rằng PAM, shadow là hai gói có vai trò khá quan trọng trong một hệ thống bảo mật. Tuy nhiên, từ các file cấu hình cơ bản cho PAM, ta sẽ chỉ giữ lại các module (các thư viện động) thông dụng, quan trọng cho hệ thống và bỏ các module dư thừa, không cần thiết cho mục đích tối thiểu hoá.

3.4-Các module khả dụng

Các module được PAM cung cấp trên bản Red Hat 6.2 đặt trong thư mục /lib/security/. Bao gồm 38 file: pam_access.so, pam_console.so, pam_cracklib.so, ..., pam_wheel.so, pam_warn.so, pam_xauth.so. Một số module được sử dụng với các file cấu hình /etc/security/*.conf, ngầm định các file này không thiết lập gì.

Trước hết, chúng tôi xin điểm qua tất cả các module:

STT	Tên module	Nhóm quản lý	Chức năng
1	pam_access.so	account	Cung cấp kiểu điều khiển truy cập đăng nhập dựa theo tên đăng nhập, tên máy (hoặc miền) địa chỉ internet, hoặc thiết bị đầu cuối (nếu không đăng nhập trên mạng).
2	pam_console.so	auth; session	
3	pam_cracklib.so	password	Cho phép kiểm tra độ mạnh của mật khẩu. Yêu cầu thư viện libcrack trong thư mục /usr/lib/cracklib_dict.
4	pam_deny.so	account, auth, password, session	Sử dụng để từ chối truy cập.
5	pam_env.so	auth	Cho phép thiết lập (hoặc không thiết lập) các biến môi trường.
6	pam_filter.so	account, auth, password, session	Module này cung cấp bộ lọc đơn giản bằng chuyển đổi các ký tự hoa và thường trong dòng dữ liệu vào và ra.
7	pam_ftp.so	auth	Mục đích của module này là cung cấp chế độ truy cập ftp nặc danh. Bằng cách chấp nhận tên người dùng ‘ftp’ hoặc
8	pam_group.so	auth	Module này cung cấp các thiết lập dựa theo tên người dùng và thiết bị đầu cuối (nó không xác thực người dùng). Nó yêu cầu file cấu hình /etc/security/groups.conf.
9	pam_krb4.so	auth, password, session	Module này cung cấp giao diện cho kiểm tra mật khẩu Kerberos của người dùng.

			Module này cần các thư viện libkrb, libdes, libcom_err, libkadm và các file tiêu đề Kerberos.
10	pam_issue.so	auth	Module này cho phép quyết định trước file issue (ngầm định /etc/issue) khi nhắc tên người dùng.
11	pam_lastlog.so	auth	Module này cung cấp thông điệp "Last login on ..." khi người dùng đăng nhập vào hệ thống. Nó bảo trì file /var/log/lastlogin. Có thể dùng để chỉ ra người dùng có thư mới khi đăng nhập.
12	pam_ldap.so		
13	pam_limits.so	session	Giới hạn tài nguyên hệ thống trong phiên người dùng. Cần file cấu hình /etc/security/limits.conf.
14	pam_listfile.so	auth	Module này cung cấp cách từ chối hoặc cho phép các dịch vụ.
15	pam_mail.so	auth, session	Module này tìm thư mục mail của người dùng và gửi thông điệp "You have mail" khi người dùng có thư trong đó.
16	pam_mkhomedir.so	session	Module này hữu ích cho hệ thống phân tán - các tài khoản người dùng được quản lý trong CSDL trung tâm (NIS, LDAP) và được truy cập qua nhiều hệ thống. Dùng để tạo các thư mục chủ ngầm định trên mỗi hệ thống.
17	pam_motd.so	session	Module này đưa nội dung của file motd (ngầm định là /etc/motd) ra màn hình mỗi khi đăng nhập thành công.
18	pam_nologin.so	auth	Nếu file /etc/nologin tồn tại, thì chỉ người dùng root được phép đăng nhập, các người dùng khác không được phép đăng nhập với một thông báo lỗi (nội dung của file nologin).
19	pam_permit.so	account; auth; password; session	Module này không phụ thuộc nhóm quản lý. Tác vụ của nó là cho phép truy cập, mà không thực hiện gì khác.
20	pam_pwdb.so	account; auth; password; session	Là module CSDL mật khẩu, có thể được thay thế cho các module pam_unix_..., nó sử dụng giao diện chung của thư viện CSDL mật khẩu. Yêu cầu cấu hình libpwdb đúng.
21	pam_radius.so	session	Module này dùng để cung cấp dịch vụ phiên cho xác thực người dùng với RADIUS server. Hiện tại, chỉ hỗ trợ sử dụng RADIUS server như server tài khoản (accounting server).
22	pam_rhosts_aut	auth	Module này cho phép xác thực các dịch

	h.so		vụ mạng chuẩn, bổ sung cho rlogin và rsh,... Dựa theo nội dung của các file /etc/hosts.equiv
23	pam_rootok.so	auth	Module này được dùng tình huống mà siêu người dùng không muốn nhập mật khẩu khi truy cập dịch vụ.
24	pam_securetty.so	auth	Cung cấp việc kiểm tra bảo mật của Unix chuẩn, chỉ xác thực cho người dùng root trên console được thiết lập trong /etc/securetty.
25	pam_shells.so		
26	pam_stress.so		
27	pam_tally.so		
28	pam_time.so	account	Module này cung cấp một số điều khiển thời gian truy cập tới các dịch vụ hệ thống. Có thể cấu hình để từ chối người dùng truy cập dựa theo tên, thời gian trong ngày, ngày trong tuần. Yêu cầu file cấu hình /etc/security/time.conf
29	pam_unix.so	account; auth; password; session	Đây là module xác thực Unix chuẩn. Nó sử dụng các lệnh gọi từ thư viện hệ thống để thiết lập thông tin tài khoản cũng như xác thực. Thông thường nó chỉ sử dụng các file cấu hình /etc/passwd và file /etc/shadow.
30	pam_userdb.so	auth	Sử dụng để kiểm tra cặp giá trị tên_người_dùng/mật_khẩu được lưu trữ trong CSDL Berkeley DB. CSDL này được sắp xếp theo các trường tên người dùng, mật khẩu (dạng không mã hoá).
31	pam_warn.so	auth; password	Ghi nhật ký thông tin về các dịch vụ, thiết bị đầu cuối, người dùng, người dùng từ xa và máy từ xa vào syslog.
32	pam_wheel.so	auth	Chỉ cho phép root truy cập tới các thành viên của nhóm wheel (gid=0).
33	pam_xauth.so		

Dưới đây chúng tôi chỉ mô tả một số module thư viện cơ bản - cần thiết cho hệ thống thông thường. Để biết thêm thông tin chi tiết về các module, tham khảo mục 6 “A reference guide for available modules” - trong [4].

Kiểm tra đô manh của mât khẩu pam_cracklib

a. Tóm tắt

Tên module: pam_cracklib

Nhóm quản lý: password.

Phụ thuộc hệ thống: Yêu cầu thư viện libcrack và từ điển hệ thống /usr/lib/cracklib_dict. Ta sẽ còn nói tới modules này trong phần cracklib, và

cracklib_dict ở phần dưới.

b. Mô tả module

Module này có thể được cấy thêm vào ngăn xếp password của ứng dụng cần xác thực để cung cấp một số plug-in kiểm tra độ mạnh cho mật khẩu. Module này làm việc như sau: đầu tiên nó gọi cracklib để kiểm tra độ mạnh của mật khẩu, sau đó module thực hiện thêm một số kiểm tra sau:

- Palindrome (xuôi ngược đều giống nhau): mật khẩu mới có là dạng đảo ngược của mật khẩu cũ?.
- Là mật khẩu cũ, nhưng thay đổi chữ hoa thành thường và ngược lại?.
- Tương tự: Mật khẩu mới giống với mật khẩu cũ?
- Đơn giản: Mật khẩu mới có quá ngắn không? Tuỳ chọn này được điều khiển bởi 5 tham số minlen, dcredit, urcedit, lcedit và ocredit.
- Mật khẩu mới có là dạng hoán vị của mật khẩu cũ?
- Đã được sử dụng trong quá khứ rồi? Thông tin các mật khẩu đã được dùng trước đó đặt /etc/security/opasswd.

c. Thành phần mật khẩu

Hoạt động của module này là nhắc người dùng vào mật khẩu và kiểm tra độ mạnh của mật khẩu bằng từ điển và tập các luật (rules) mô tả trên. Các tham số: debug; type=XXX; retry=N; difok=N; minlen=N; dcredit=N; lcredit=N; ocredit=N.

- debug: module viết thông tin ra syslog().
- type=XXX: ngầm định module sử dụng dấu nhắc dạng "New UNIX Password:". Sử dụng tuỳ chọn này bạn có thể thay từ UNIX bằng XXX.
- retry=N: số lần module yêu cầu mật khẩu mới (kiểm tra độ mạnh) từ người dùng đơn, ngầm định là 1.
- difok=N: số ký tự trong mật khẩu mới phải khác với mật khẩu cũ, ngầm định là 10. Thêm vào đó, nếu 1/2 các ký tự trong mật khẩu mới khác với khẩu cũ là có thể chấp nhận được.
- minlen=N: độ dài tối thiểu chấp nhận được cộng với 1.

Ví dụ: Thiết lập cấu hình file /etc/pam.d/passwd như sau

```
#%PAM-1.0
password    required    pam_cracklib.so      difok=3 minlen=15
password    required    pam_pwdb.so   use_authtok    nullok    md5
```

Module pam_deny

a. Tóm tắt

Tên module: pam_deny

Các nhóm quản lý: account; authentication; password; session.

b. Mô tả về module

Module này được dùng để từ chối truy cập. Module này phù hợp cho các đề mục ngầm định trong /etc/pam.d/other.

c. Thành phần tài khoản (Account component):

Thành phần này không thực hiện gì ngoài việc trả về lỗi. Ví dụ: Nếu ta thêm dòng sau vào /etc/pam.d/login, để không cho phép tài khoản nào được đăng nhập hệ thống.

```
account       required pam_deny.so
```

d. Thành phần xác thực (authentication component)

Thành phần này không thực hiện gì ngoài việc trả về một lỗi. Ví dụ: Để từ chối truy cập tới ứng dụng ngầm định bằng thành phần này của module pam_deny, bạn có thể thêm dòng sau vào file cấu hình PAM /etc/pam.d/other:

```
auth    required        pam_deny.so
```

e. Thành phần mật khẩu (Password component)

Thành phần của module này từ chối người dùng cơ hội thay đổi mật khẩu của họ. Nó luôn trả lại PAM_AUTHOK_ERR khi được gọi.

Ví dụ: để tránh login tự động nhắc mật khẩu mới khi mật khẩu cũ hết hạn bạn thêm dòng sau vào file cấu hình /etc/pam.d/login

```
login   password        required        pam_deny.so
```

f. Thành phần phiên (session component)

Không cho phép khởi động một session trên máy. Từ chối phiên người dùng trên hệ thống.

Module giới han tài nguyên hê thống

a. Tóm tắt

Tên module: pam_limits

Nhóm quản lý: session

Phụ thuộc hệ thống: Yêu cầu file cấu hình /etc/security/limits.conf và nhân hỗ trợ giới hạn tài nguyên. Nó cũng sử dụng thư viện libpwdb.

b. Mô tả về module

Thông qua một phiên được mở, module này thiết lập giới hạn tài nguyên hệ thống có thể từ phiên người dùng. Các tham số: debug; conf=/path/to/file.conf.

Thông qua nội dung của file cấu hình /etc/security/limits.conf, các giới hạn tài nguyên được đặt cho phiên người dùng. Sự hạn chế này không hiệu lực với siêu người dùng.

Ví dụ:

Trước tiên phải tạo một file chỉ đọc bởi người dùng root (ngầm định là /etc/security/limits.conf). File này mô tả tài nguyên giới hạn mà siêu người dùng muốn áp dụng cho người dùng và nhóm. Mỗi dòng có dạng sau:

<domain> <type> <item> <value>

Các trường có thể được điền như sau:

<domain> có thể là:

- tên người dùng
- tên nhóm, với cú pháp @group
- dấu *, cho đề mục ngầm định.

<type> có thể là:

- hard: để bắt buộc giới hạn tài nguyên cứng. Những giới hạn này được thiết lập bởi siêu người dùng và nhân Linux. Người dùng không thể nâng các giá trị này.
- soft: để bắt buộc giới hạn tài nguyên mềm. Với các giới hạn này, người dùng có thể chuyển lên hoặc xuống trong vùng cho phép.

<item> có thể là:

- core - giới hạn kích thước file gốc (KB).
- data - kích thước dữ liệu lớn nhất (KB).
- fsize - kích thước file lớn nhất (KB).
- memlock - không gian địa chỉ bộ nhớ-khoá lớn nhất (KB).
- cpu - thời gian CPU lớn nhất (MIN)
- nproc - số tiến trình lớn nhất.
- maxlogins - số đăng nhập lớn nhất cho người dùng này.

Ví dụ: ta soạn file /etc/security/limits.conf như sau:

#<domain>	<type>	<item>	<value>
*	soft	core	0
*	hard	rss	10000
@student	hard	nproc	20
@student	hard	maxlogin	4

Thêm vào file /etc/pam.d/login dòng sau:

```
session        required        pam_limits.so
```

Module no-login

a) Tóm tắt

Tên module: pam_nologin

Nhóm quản lý: authentication

b) Mô tả

Cung cấp xác thực nologin chuẩn của UNIX. Nếu file /etc/nologin tồn tại, chỉ người dùng root mới được phép đăng nhập; các người dùng khác không thể đăng nhập với một thông báo lỗi. Nếu không tồn tại file /etc/nologin, module này hoàn toàn im lặng.

Nó có thể được sử dụng với các phương thức required.

Module pam_permit

a. Tóm tắt

Tên module: pam_permit

Nhóm quản lý: account; authentication; password; session.

Module này rất nguy hiểm, nó thường được dùng với cảnh báo nguy hiểm, nó cho phép truy cập, mà chẳng làm gì cả.

b. Thành phần Account+Authentication+Password+Session

Module này đơn giản chỉ trả về mã thành công PAM_SUCCESS. Khi người quản trị hệ thống muốn tắt quản lý tài khoản trên một trạm làm việc, và tại thời điểm hiện hành vẫn cho phép đăng nhập, chỉ cần thêm vào file /etc/pam.d/login:

```
account        required        pam_permit.so
```

Module cơ sở dữ liệu mật khẩu.

a. Tóm tắt

Tên module: pam_pwdb

Nhóm quản lý: account; authentication; password; session

Module này là sự thay thế cho các module pam_unix_...

b. Thành phần account

Dựa vào các phần tử sau: expire; last_change; max_change; defer_change; warn_change, module này cho biết trạng thái của tài khoản, mật khẩu người dùng. Trong trường hợp sau đó, nó có thể khuyên người dùng thay đổi mật khẩu.

Ví dụ:

Trong chế độ tài khoản, module này được chèn vào file /etc/pam.d/login/ để đảm bảo rằng tài khoản người dùng và mật khẩu vẫn còn hoạt động.

```
account        required        pam_pwdb.so
```

c. Thành phần xác thực

Có thể sử dụng các tham số debug để ghi các thông tin về hoạt động của nó bằng syslog(). Hoạt động ngầm định là không cho phép người dùng truy cập vào dịch vụ nếu nếu khẩu của họ trống. Một file nhị phân, pwdb_chkpwd được cung cấp để kiểm tra mật khẩu của người dùng gọi nó.

d. Thành phần mật khẩu

Có thể sử dụng các tham số sau cho thành phần này: debug, nullok, not_set_pass, use_authtok, try_first_pass, use_first_pass, md5, bigcrypt, shadow, radius, unix.

Phần này của module pam_pwdb cung cấp tác vụ cập nhật mật khẩu của người dùng. Module này có thể chuyển mật khẩu của người dùng từ CSDL này sang CSDL khác, có thể bảo mật các đề mục CSDL của người dùng theo những dạng khác nhau, sử dụng các tham số shadow, radius và unix.

d. Thành phần session

Thành phần này không nhận dạng tham số nào. Các tác vụ của nó chỉ là ghi nhật ký tên người dùng và kiểu dịch vụ cho syslog(). Các thông điệp được ghi từ khi bắt đầu phiên cho tới cuối phiên.

Module securetty

a. Tóm tắt

Tên module: pam_securetty
Nhóm quản lý: authentication
Phụ thuộc: file cấu hình /etc/securetty

b. Thành phần xác thực

Cung cấp kiểm tra tty được phép chuẩn của Unix, chỉ nhằm kiểm tra cho phép người dùng root đăng nhập trên các tty trong file /etc/securetty. Mọi người dùng khác trên hệ thống đều được phép đăng nhập trên tất các console.

Kết luận

Có thể thấy rằng PAM cung cấp khá nhiều các module khác nhau, giúp người quản trị có thể dễ dàng thay đổi các phương pháp xác thực khác nhau. Tuy nhiên các module này yêu cầu (require) khá nhiều các gói chương trình, thư viện khác. Cụ thể, PAM require các gói và thư viện sau:

```
cracklib
cracklib-dicts
pwdb >= 0.54-2
initscripts >= 3.94
/sbin/ldconfig
ld-linux.so.2
libc.so.6
libcrypt.so.1
libnsl.so.1
libpwdb.so.0
libcrack.so.2
libdb.so.3
libdl.so.2
libglib-1.2.so.0
libpam.so.0
libc.so.6(GLIBC_2.0)
libc.so.6(GLIBC_2.1)
libcrypt.so.1(GLIBC_2.0)
libdb.so.3(GLIBC_2.1)
libdl.so.2(GLIBC_2.0)
libdl.so.2(GLIBC_2.1)
libnsl.so.1(GLIBC_2.0)
```

Tuy nhiên không phải hệ thống của chúng ta sử dụng mọi module khả dụng này. Ta có thể tối thiểu được khá nhiều các modules không cần thiết trong gói này phù hợp với mục đích, yêu cầu của mình.

Tài liệu tham khảo

[1] - Authentication HOWTO - Peter Hernberg
[2] - Shadow Password Howto - Michael H. Jackson mhjack@scnet.com
[3]- Security HOWTO
[4]- The Linux-PAM System Administrator's Guide, Adrew G. Morgan
[5]- Crypt()
[6]- Practical Unix Security - Simson Garfinkel and Gene Spafford
[7] - Các trang man getty(); mingetty(); login(); sulogin;
[8] - Text - Terminal HOWTO - David S. Lawyer **dave@lafn.org**

PHẦN II
AN TOÀN CỦA HỆ ĐIỀU HÀNH SUN SOLARIS

CHƯƠNG I- GIỚI THIỆU VÀ ĐÁNH GIÁ KHẢ NĂNG AN TOÀN CỦA SOLARIS

Chúng tôi sẽ cung cấp một tập hợp đầy đủ các chức năng an toàn và mô tả bốn mức bảo vệ trong Solaris™:

- Mức 1 bao gồm các tính năng và công cụ giúp những người quản trị kiểm soát chặt chẽ những người đăng nhập hệ thống.
- Mức 2 mô tả các công cụ cho phép những người quản trị thiết lập giải pháp an toàn tổng thể cho hệ thống.
- Mức 3 gồm các dịch vụ phân tán an toàn (Secure Distributed Services) và những nền tảng phát triển (Developers Platforms), mô tả cách Solaris hỗ trợ các cơ chế xác thực và mã hoá khác nhau.
- Mức 4 mô tả các công cụ điều khiển truy nhập tới mạng vật lý.

1.1-An toàn: Vấn đề cơ bản đối với công ty toàn cầu

Việc các trung tâm dữ liệu đứng độc lập cùng với các yêu cầu an toàn hoàn toàn tập trung đang giảm đi nhanh chóng trong các môi trường tính toán tập thể hiện đại đã được nói đến nhiều. Các môi trường phân tán hiệu năng cao và ưu thế hơn về giá cả, trong đó các hệ thống khách được tách khỏi các server trên mạng đang không ngừng tăng lên. Thêm vào đó, các mối liên kết giữa các tổ chức thương mại, cá nhân và chính phủ trên toàn thế giới đang mở rộng cộng đồng người dùng, họ có khả năng truy nhập tới những tài nguyên nội bộ của công ty.

Đồng thời, những người dùng ngày càng thông thạo và phức tạp hơn. Đáng tiếc, một số người đã dùng hiểu biết của họ với những mục đích không chính đáng. Mặc dù những hacker nổi tiếng luôn được đăng tải trên thông tin đại chúng, nhưng các nghiên cứu cho thấy phần lớn những hành động xâm phạm máy tính không bị phát hiện. Những xu hướng này đã làm nảy sinh những thay đổi về căn bản trong các yêu cầu an toàn đối với liên kết toàn cầu.

Không có gì ngạc nhiên khi mà an toàn nổi lên như là một vấn đề cốt lõi đối với các công ty mong muốn tận dụng những lợi ích trong việc thực thi các hệ thống phân tán toàn cầu, mà không làm nguy hiểm tới tính bí mật và toàn vẹn của thông tin nhạy cảm. Vì thế, những người quản trị hệ thống và mạng phải có khả năng lựa chọn những sản phẩm đáp ứng đầy đủ các tính năng nhằm vào những nhu cầu an toàn hay thay đổi của họ.

Giải pháp Solaris của Sun đưa ra một tập các tính năng an toàn hoàn chỉnh đầy đủ phù hợp với các yêu cầu khác nhau của các môi trường tính toán tập thể hiện thời:

- Nó đưa ra các công cụ tự động đơn giản hoá cấu hình trạng thái an toàn của hệ thống, và thông báo những điểm có thể bị mất an toàn.
- Nó cung cấp các dịch vụ file và thư mục được phân phối an toàn và cơ sở cho việc phát triển các ứng dụng an toàn.
- Nó đáp ứng các chuẩn quan trọng của Mỹ và quốc tế cũng như các tính năng an toàn Internet gần đây nhất.

- Cuối cùng, các sản phẩm nhóm thứ ba bổ sung vào cái mà trước đó Sun đã đưa ra cho các tổ chức một loạt bảo vệ chống lại những vi phạm an toàn có thể xảy ra.

1.2-Solaris: Giải pháp an toàn

Solaris của Sun bảo vệ chống lại những xâm nhập trái phép bằng một giải pháp an toàn hệ thống và mạng theo nhiều hướng. Sơ đồ bảo vệ Solaris bốn mức cho phép những người giám quản:

- Điều khiển người có thể đăng nhập vào hệ thống
- Điều khiển khả năng truy nhập tài nguyên hệ thống để tự động ghi chép của những người dùng và các chương trình.
- Thực thi các dịch vụ tệp và thư mục được phân phối an toàn và cung cấp những cơ sở cho việc phát triển các ứng dụng và dịch vụ an toàn.
- Điều khiển truy nhập tới mạng vật lý.

Những phần sau đây mô tả chi tiết hơn về bốn hướng an toàn Solaris.

1.3-Mức 1: Điều khiển đăng nhập trên Solaris

Mức thứ nhất của điều khiển an toàn Solaris bao gồm các tính năng và công cụ giúp những người giám quản kiểm soát chặt chẽ những người có thể đăng nhập vào hệ thống. Trọng tâm của mục tiêu này là việc sử dụng một mật khẩu có thể dùng để kiểm tra định danh của người đang cố gắng vào mạng. Nói cách khác, nếu ai đó có mật khẩu được xem là của riêng anh ta, thì nó có thể chứng minh rằng cá nhân đang cố gắng đăng nhập quả thực là "đáng tin cậy" và ngầm hiểu là đã được cấp "quyền" đăng nhập hệ thống.

Đương nhiên, hệ thống không thể xác định nếu mật khẩu đang bị một người khác không phải chủ nhân của nó sử dụng. Điều này chỉ ra rằng việc bảo vệ các mật khẩu là cực kỳ quan trọng. Solaris có nhiều tính năng điều khiển việc tạo, sử dụng và lưu trữ an toàn các mật khẩu. Những tính năng này được hiểu một cách chung chung là các tính năng quản lý mật khẩu, và chúng bao gồm:

- *Xác nhận mật khẩu*: Solaris so sánh mật khẩu người dùng cung cấp với mật khẩu đã đặt và lưu trữ của người dùng đó trong một file đặc biệt (xem "File mật khẩu bóng"). Nếu các mật khẩu tương hợp, thì người dùng được phép vào mạng.
- *Định thời gian có hiệu lực của mật khẩu*: Solaris cho phép người giám quản đặt ngày hết hạn của các mật khẩu. Solaris sẽ cảnh báo người dùng mật khẩu sắp hết hạn và yêu cầu đặt một mật khẩu mới. Khi mật khẩu hết hạn, nếu không đặt một mật khẩu mới, thì mọi cố gắng vào mạng sẽ bị từ chối. (Lưu ý rằng NIS+ trên Solaris 2.5 hỗ trợ định thời hạn mật khẩu phạm vi vùng).
- *Không cho phép mật khẩu cũ*: Lâu lâu, một mật khẩu bị lặp lại, có nhiều khả năng ai đó sẽ tìm ra nó. Tính năng này ngăn chặn người dùng sử dụng lại mật khẩu đã dùng trước đây.
- *Định tính mật khẩu*: Solaris giúp đảm bảo rằng bạn sẽ tạo được một mật khẩu khó phỏng đoán hơn. Nó làm điều này bằng cách kiểm tra xem mật khẩu có đủ số lượng ký tự và/hoặc ký hiệu hay không.

- *Che file mật khẩu*: File "ẩn" (được gọi là /etc./shadow) lưu tất cả những mật khẩu người dùng và chỉ có thể đọc từ gốc (root). Trước khi thực hiện che file mật khẩu, những người hợp pháp khác có thể truy nhập file mật khẩu.
- *Định thời hạn tài khoản*: Điều này cho phép người giám quản hệ thống đặt ngày hết hạn một tài khoản. Sau ngày hết hạn, tài khoản không còn hiệu lực.

Dòng sản phẩm quản lý an toàn của Sun tạo điều kiện cho các sản phẩm của Solaris có nhiều hạn chế đăng nhập hơn. Một số ví dụ như sau:

- *Hạn chế số giờ truy nhập*: Điều này nghĩa là những người dùng nào đó không được phép vào mạng trong thời gian mà người giám quản hệ thống đã định trước, ví dụ nửa đêm khi không có ai xung quanh.
- *Không cho phép vào mạng sau nhiều lần vô hiệu*: Điều này ngăn chặn những người dùng (và chương trình) phỏng đoán mật khẩu bằng việc thử đăng nhập nhiều lần.
- *Tự động khoá màn hình và ra khỏi mạng*: Sau một khoảng thời gian quy định không làm gì, trạm làm việc tự động khoá màn hình hoặc ra khỏi mạng.
- *Những điều khiển tăng cường đối với đặc quyền root/su*: Yêu cầu ai đó cung cấp những mật khẩu đặc biệt để truy nhập những đặc quyền root và superuser.

Bảo vệ truy nhập từ xa

Vì những truy nhập hệ thống trái phép có khả năng xảy ra cũng có thể thực hiện được trên đường điện thoại từ xa, nên Solaris cho phép các cổng modem có mật khẩu bảo vệ. Khi đã đặt một mật khẩu cổng modem, thì người dùng điện thoại từ xa trước hết sẽ được hỏi về nó. Nếu trả lời đúng, người dùng sẽ được phép tiến hành quá trình đăng nhập hệ thống bình thường.

Các sản phẩm bổ sung cung cấp bảo vệ truy nhập từ xa khả dụng với Solaris từ các nhóm thứ ba. Ví dụ, công nghệ thẻ số có từ các công ty như Enigma Logic và Security Dynamics. Thẻ số là mật khẩu được dùng "chỉ một lần" để điều khiển truy nhập từ xa. Một tập các thẻ số cầm tay đăng ký trước trợ giúp các thiết bị có thể phải mang đi xa. Theo lược đồ này, các mật khẩu không bao giờ bị dùng lại, cho nên an toàn hơn.

Một phiên bản phần mềm vùng công khai về thẻ số là S/Key đã có trên Internet và có thể truy nhập bằng cách truyền file tại địa chỉ sau: thumper.bellcore.com /pub/nmh.

1.4-Mức 2: Điều khiển truy nhập tài nguyên hệ thống

Một khi người dùng vào mạng thành công, anh ta có thể bắt đầu thử truy nhập các tài nguyên. Vì thế, Solaris cho phép những người giám quản kiểm soát khả năng truy nhập các tài nguyên chung trên hệ thống bằng việc cung cấp các công cụ có thể thiết lập trạng thái an toàn tổng thể cho hệ thống. Solaris cũng hỗ trợ các tính năng cho phép đặt các quyền truy nhập file thích hợp. Thêm nữa, khả năng kiểm soát cũng được cung cấp nhằm theo dõi những lần truy nhập. Các tính năng này được mô tả chi tiết dưới đây.

Thiết lập và kiểm tra thực trạng an toàn của Solaris
Hầu hết những người giám quản hệ thống đều đồng ý rằng có nhiều lần muốn đánh giá thực trạng an toàn tổng thể của hệ thống và/hoặc thiết lập nó cho phù hợp. Để thực hiện điều này, Solaris tính đến việc tự động đánh giá thực trạng của hệ thống cũng như đặt nó ở một trong ba mức an toàn xác định trước: thấp, trung bình, hoặc cao.

Bằng việc chạy định kỳ, ASET sẽ cảnh báo người giám quản về bất kỳ sự vi phạm an toàn nào có thể xảy ra. Dưới đây là một ví dụ về cái mà ASET kiểm tra:

- Sự tồn tại của mật khẩu EEPROM của hệ thống ngăn chặn cá nhân không được phép khởi động hệ thống trong chế độ chỉ có 1 người dùng.
- Sử dụng không an toàn biến UMASK điều khiển việc cài đặt những tuỳ chọn file ban đầu khi file được tạo ra.
- Sử dụng không an toàn biến PATH đưa ra thứ tự những thư mục sẽ được tìm kiếm đối với một lệnh hoặc chương trình thực hiện cụ thể.
- Những cài đặt quyền file hệ thống.
- Sự tồn tại của các chương trình cài đặt mới
- Các quyền thư mục ban đầu.
- Nội dung của các tệp:
  .rhosts, /etc/passwd, /etc/group
- Kích thước của các file trong /usr/bin và /bin

Lưu ý rằng những người giám quản có tuỳ chọn cảnh báo về các vấn đề có thể xảy ra bằng thư điện tử.

Khi sử dụng cài đặt hệ thống theo mô hình an toàn mức thấp, ASET đảm bảo các thuộc tính file (các quyền) được đặt ở các giá trị theo chuẩn ban hành. Một vài kiểm tra được thực hiện và những chỗ yếu về an toàn có thể có đều được thông báo.

Cài đặt an toàn mức trung bình đảm bảo an toàn thích đáng với hầu hết các môi trường. ASET sẽ thay đổi những cài đặt quyền của một số file hệ thống (ví dụ, ttytab, host.equiv) và các tham số hạn chế truy nhập hệ thống. Những kiểm tra an toàn bổ sung được thực hiện, những chỗ yếu và những thay đổi truy nhập được thông báo.

Cài đặt an toàn mức cao sản sinh một hệ thống cực kỳ an toàn. Nhiều file và tham số hệ thống được cài đặt chỉ cho phép truy nhập tối thiểu. Cài đặt an toàn mức cao cũng có thể huỷ bỏ việc chuyển tiếp IP. Tuy nhiên, chức năng này đã nhường hẳn cho một sản phẩm là Solstice™FileWall-1™ hoặc Solstice™SunScreen™ của Sun được trình bày sau trong tài liệu này.

Với ASET, người giám quản không cần lãng phí thời gian quý báu và cố gắng tìm kiếm bằng tay những lỗ hổng an toàn trên hệ thống. ASET cũng là một tính năng duy nhất của Solaris và không có ở các cơ sở hệ thống khác.

Bảo vệ file của Solaris
File là tài nguyên chính phải được bảo vệ trên bất kỳ cơ sở hệ thống nào từ PCs đến các máy tính lớn. Solaris thực thi hai biện pháp để bảo vệ file: cài đặt quyền "kiểu

Unix" truyền thống và các danh sách điều khiển truy nhập (ACLs). Với việc cài đặt quyền "kiểu Unix", có thể thiết lập những chỉ dẫn quyền đọc, ghi và thực hiện đối với người sở hữu file, các nhóm được lựa chọn, hoặc "mọi người" (cũng hiểu là "người khác"). Tuy nhiên, một nhược điểm của cài đặt quyền là việc truy nhập có thể chỉ bị hạn chế trên cơ sở nhóm và không thể tách riêng đặc quyền (hoặc hạn chế) cụ thể đối với các cá nhân.

Solaris hỗ trợ các danh sách điều khiển truy nhập (ACLs). Các danh sách điều khiển truy nhập chính là một danh sách điều khiển truy nhập tới các file. Với ACLs, các danh sách mở rộng về thông tin cấp phép có thể được duy trì đối với mỗi file cho phép phân nhỏ hơn điều khiển truy nhập file. Ví dụ, với ACLs truy nhập có thể được điều khiển trên cơ sở một người dùng thay vì trên cơ sở nhóm.

ACLs trên Solaris tuân theo đặc tả POSIX 1003.6. Chúng đã được thực thi với cả Hệ thống File Người dùng (UFS) cũng như với NFS phiên bản 2 và 3.

Kiểm toán

Kiểm toán nằm trong phần này vì nó giúp cho những người giám quản theo dõi các sự kiện liên quan tới an toàn bao gồm nhiều kiểu thử truy nhập khác nhau. Nếu một vi phạm xảy ra, nhật ký kiểm tra có thể giúp xác định cái gì đã xảy ra, và có thể còn giúp tìm ra ai là thủ phạm! Solaris có hai phương pháp kiểm tra: Nhật ký hệ thống Unix và kiểm tra C2. Cả hai được đề cập đến trong phần này.

Các nhật ký hệ thống Unix (syslogs) lưu giữ dấu vết của các sự kiện đăng nhập, sử dụng tài nguyên, các chỉ tiêu, và nhiều hơn nữa. Nhiều thiết bị hệ thống sử dụng các syslog để ghi chép hoặc thông báo cho người giám quản hệ thống về những sự kiện quan trọng. Các kịch bản chương trình khung, hoặc các gói có thể cũng được ghi vào các CSDL syslog để kiểm soát các tình huống đặc biệt.

Kiểm tra C2, còn gọi là Bảo vệ Truy nhập có điều khiển, có thể đưa ra một báo cáo kiểm tra chi tiết hơn. Vào những năm 1980, Cục Bảo vệ định nghĩa kiểm tra C2 như là một phần trong nguyên tắc chỉ đạo về các mức an toàn máy tính khác nhau. Các yêu cầu này được đưa ra trong Orange Book hay tiêu chuẩn đánh giá các hệ thống máy tính tin cậy (TC-SEC). Các mức an toàn được liệt kê bắt đầu với D cho mức thấp nhất, cho đến A1 với mức cao nhất. Trung tâm an toàn máy tính quốc gia (NCSC) đánh giá các hệ thống dựa vào tiêu chuẩn này.

C2 có thể tạo ra các nhật ký kiểm tra từ người dùng, sự kiện và lớp. Hơn nữa, với C2 nó có thể ghi lại bất kỳ sự kiện nào mà người giám quản hệ thống cho rằng "liên quan" tới an toàn. Kiểm tra C2 Solaris bao gồm chức năng mô hình an toàn cơ bản (BSM) cho phép ghi lại những sự kiện hoàn toàn ở mức gọi hệ thống.

Trong trường hợp không chắc xảy ra vi phạm an toàn, thì khả năng kiểm tra của Solaris cung cấp cho những người giám quản hệ thống một bản kê khai chi tiết về hành vi liên quan. Thông tin này có thể rất quan trọng giúp tìm ra nguồn gốc của vấn đề.

1.5-Mức 3: Các dịch vụ phân tán an toàn và những nền tảng phát triển

Môi trường điều hành nhân Solaris kết hợp chặt chẽ với họ dịch vụ phân tán ONC+™ được cấu hình một cách tuỳ ý để chạy với các tính năng an toàn bổ sung cho phép. Khi có điều kiện, ONC+ bao gồm dịch vụ đặt tên phân tán NIS+ an toàn, dịch vụ file phân tán NFS ™ an toàn và cơ sở gọi thủ tục từ xa an toàn độc lập với cơ cấu vận chuyển (TI-RPC) (còn được hiểu đơn giản là RPC an toàn) để xây dựng các ứng dụng và dịch vụ phân tán.

Sun cũng cung cấp họ dịch vụ DCE trong một sản phẩm rời gọi là DCE cho Solaris. Sản phẩm này bao gồm dịch vụ file phân tán DFS, dịch vụ đặt tên phân tán CDS và cơ sở phát triển dựa vào DCE RPC (và còn thêm các chi tiết khác không thích hợp với tài liệu này).

Cả hai dịch vụ ONC+ và DCE đều dựa vào cơ sở công nghệ được mô tả trong phần này.

Cơ sở công nghệ cho các dịch vụ an toàn

Trước khi người dùng trên hệ thống khách truy nhập tới các tài nguyên của server, server phải chắc chắn rằng người dùng được công nhận có quyền truy nhập "hợp pháp" tới server và các tài nguyên của nó. Vì thế, trong ngữ cảnh này, server phải có khả năng để:

1. Kiểm tra định danh của người dùng trên mạng. Chức năng này do dịch vụ xác thực cung cấp và cũng bao hàm các dịch vụ kê khai ở điểm 3 bên dưới.
2. Đảm bảo người dùng được phép truy nhập các tài nguyên mà anh ta đang cố gắng truy nhập một khi anh ta đã được xác thực một cách hợp lệ. Dịch vụ cấp phép cung cấp điều này.
3. Duy trì tính bí mật và toàn vẹn thông tin được trao đổi trên mạng. Những điều này nhờ cậy vào các dịch vụ bí mật và toàn vẹn.

Các dịch vụ xác thực, bí mật và toàn vẹn

Việc xác thực một người dùng trên mạng đòi hỏi thông tin "nhãn quyền" nhạy cảm phải được trao đổi giữa hệ thống khách và chủ. Vì không có cách gì đảm bảo thông tin này sẽ không bị chặn lại khi nó đi tới đích, nên nó phải được bảo vệ không bị dịch hoặc thay đổi trên suốt tuyến. Vì thế, dịch vụ phân tán an toàn phải có cách bảo vệ tính bí mật và toàn vẹn thông tin.

Các dịch vụ bí mật và toàn vẹn về cơ bản được gói cùng với dịch vụ xác thực. Dịch vụ bí mật cung cấp cách biến đổi thông tin thành một dạng mà chỉ người nhận định trước mới có thể dịch được. Điều này cũng được xem như là mã hoá dữ liệu. Người nhận có trách nhiệm giải mã dữ liệu hay noí cách khác biến đổi nó trở lại dạng có thể đọc được. Dịch vụ toàn vẹn cung cấp cách tính toán tổng kiểm tra thông tin, cái mà khi kiểm tra sẽ chỉ ra được thông tin đã bị biến đổi nội dung ban đầu hay chưa.

Hiện có nhiều dịch vụ xác thực đang tồn tại và nổi bật. Vì vậy, Solaris đáp ứng kiến trúc mềm dẻo cho phép tiếp cận tới các cơ chế xác thực cả bây giờ và trong tương lai. Hiện thời trên Solaris, các cơ chế xác thực có Kerberos, Diffie-Hellman và Unix-style được đáp ứng và tiếp cận thông qua giao diện TI-RPC an toàn. Thực tế, cả NIS+ an toàn và NFS an toàn đều đã được phát triển trên nền TI-RPC an toàn.

Các dịch vụ DCE an toàn trên Solaris tiếp cận xác thực Kerberos thông qua giao diện DCE RPC. Cả DFS và CDS đã được lập trình cho giao diện DCE RPC và vì thế có thể tận dụng cho dịch vụ xác thực Kerberos.

Sun hiện đang thực thi kiến trúc xác thực đã sửa lại dựa vào chuẩn Internet RFC 1508, còn hiểu là các dịch vụ an toàn API chung (GSSAPI- General Security Services Application Programming Interface). GSSAPI kết hợp nhiều giải pháp xác thực dưới một API. Nó cũng đảm bảo xác thực "tải thêm được" tin cậy. Điều này có nghĩa là một cơ chế xác thực mới có thể được đưa vào ghép nối mà không ảnh hưởng tới các ứng dụng và dịch vụ đang tồn tại.

Thuật ngữ GSSRPC biểu diễn sự tích hợp TI-RPC với GSSAPI của Sun trên Solaris. GSSRPC sẽ đưa ra các ứng dụng dựa vào RPC tiếp cận với các tuỳ chọn xác thực phức tạp.

Module xác thực có thể tải thêm (PAM- Pluggable Authentication Module)
Được phát triển ban đầu bởi Sun và OSF chấp nhận đưa vào CDE/Motif. PAM cung cấp mô hình có thể tải thêm cho các cơ chế xác thực hệ thống cũng như các dịch vụ liên quan khác chẳng hạn quản lý mật khẩu, tài khoản và phiên. Các dịch vụ này đặc biệt có lợi cho các ứng dụng cung cấp hoặc yêu cầu "đầu vào hệ thống" (hoặc đăng nhập) phải xác minh định danh người dùng cũng như thông tin tài khoản. Một số ví dụ phổ biến về các ứng dụng này là login, dtlogin, rlogin, rsh, telnet, ftp, ...

Các cơ chế an toàn có thể tiếp cận qua PAM được thi hành khi những người giám quản có thể cài đặt các module phần mềm chia xẻ, có khả năng tải động trong suốt đối với các ứng dụng. PAM cho phép người giám quản cấu hình cơ chế xác thực người dùng trên cơ sở từng ứng dụng. Ví dụ, một cổng yêu cầu xác thực mật khẩu S/Key để truy nhập Telnet trong khi đang cho phép các phiên đăng nhập giao tiếp với chính xác thực mật khẩu UNIX. Với PAM cũng có thể cấu hình nhiều cơ chế xác thực cho mỗi ứng dụng. Ví dụ, người giám quản muốn xác thực những người dùng bằng cả Kerberos và RSA. Cuối cùng, PAM cho phép những người dùng của các ứng dụng này cung cấp một mật khẩu đơn lẻ mà nhiều dịch vụ xác thực có thể sử dụng.

GSS-API
GSS-API là một chuẩn được đề xuất, như định nghĩa trong RFC-1508 và RFC-1509. Nó đang trở thành chuẩn thực tế, nói chung đó là kiểu tổng quát, để giao dịch với các dịch vụ an toàn (chẳng hạn xác thực, toàn vẹn và mã hoá). Các ứng dụng có thể chạy độc lập với cơ chế và các công nghệ an toàn bên dưới. Nó cũng cho phép có thể di chuyển mức nguồn.

PAM API và GSS-API là phần bù của nhau, trong khi PAM API hỗ trợ xác thực người dùng bằng các server lối vào hệ thống, thì GSSAPI hỗ trợ xác thực Client/Server dựa vào mạng. Vì thế, một khi những người dùng trên các hệ thống khách được xác thực thông qua PAM, thì họ có thể kết nối một cách an toàn với các hệ thống server sử dụng GSSAPI dựa vào các dịch vụ xác thực.

Các dịch vụ cấp phép
Dịch vụ hay cơ chế cấp phép cung cấp cách đảm bảo một người dùng đã được cấp

phép truy nhập thông tin mà anh ta đang cố gắng truy nhập từ xa. NFS cung cấp hai cơ chế cấp phép: các dấu hiệu về quyền file và POSIX 1003.6 tuân theo ACLs. Cả các quyền và các danh sách điều khiển truy nhập được đề cập trong phần có tiêu đề "Bảo vệ file trên Solaris". NIS+ an toàn dùng phương pháp gọi là các quyền truy nhập bảng biểu thị sự cho phép truy nhập thông tin lưu trong các bảng NIS+. Để có thêm thông tin về các quyền truy nhập bảng, hãy xem giải trình về NIS+.

DCE DFS sử dụng chuẩn ACL riêng của nó như DCE ACLs. Dịch vụ đặt tên phân tán DCE CDS khiến cho việc sử dụng DCE ACLs tốt hơn.

Các tiện ích an toàn từ xa
Bên cạnh các dịch vụ an toàn, Solaris có các tiện ích an toàn từ xa như là telnet, ftp, rcp, rsh và rlogin. Có một số liên quan tới các tiện ích "Kerberized" vì chúng thường dùng xác thực Kerberos.

Dấu hiệu khởi đầu riêng
Một vấn đề cơ bản đối với nhiều nhà sản xuất, các môi trường phân tán phải phù hợp với thực tế là trong đa số các trường hợp, mỗi máy chủ hoặc server đòi hỏi người dùng cung cấp một mật khẩu riêng để được truy nhập tới các dịch vụ. Một phương pháp gọi là dấu khởi đầu riêng nảy sinh để giải quyết vấn đề này. Với dấu khởi đầu riêng, người dùng đưa vào chỉ một mật khẩu để được truy nhập tới tất cả các hệ thống trong một môi trường phân tán.

Mặc dầu nó chỉ là một phần của giải pháp, giao diện PAM được nói đến trong phần có tiêu đề "Các dịch vụ xác thực, bí mật và toàn vẹn" giúp cho phép có thể ra dấu khởi đầu riêng nhờ khả năng tích hợp nhiều cơ chế xác thực của nó. Hiện tại, Sun đang phát triển dấu khởi đầu riêng để đáp ứng nhiều môi trường, chẳng hạn ONC+ và DCE. Dòng sản phẩm quản lý an toàn của Sun (kể cả Solaris) đồng nhất Solaris với các môi trường của nhiều nhà sản xuất khác nhau như Windows, IBM/MVS, VMS, ...

1.6-Mức 4: Điều khiển truy nhập tới mạng vật lý

Các mạng máy tính trước đây không được thiết kế để điều khiển an toàn chặt chẽ, vì nó được giả thiết rằng các cổng (và những người dùng) kết nối với mạng có khả năng tin cậy lớn. Thời gian và kinh nghiệm cho thấy đó không phải là một giả thiết đúng đắn. Thêm vào đó các mối đe doạ từ bên ngoài có thể xảy ra, những người dùng nội bộ được theo dõi cẩn thận có thể vô tình để lộ dữ liệu hay các dịch vụ của tổ chức từ trong mạng ra bên ngoài. Vì thế, người ta mong muốn ngăn chặn cả hai loại vấn đề xảy ra mà không yêu cầu mọi người trở thành các chuyên gia an toàn. Sun đáp ứng mức dịch vụ này - Solaris với Solstice Filewall-1 và các sản phẩm Solstice Sunscreen rời của nó.

Solstice Firewall-1
Mục đích của một "firewall", cũng được hiểu là hệ thống an toàn mạng, là đảm bảo rằng mọi kết nối giữa mạng cục bộ của tổ chức và một mạng mở tương thích với các chính sách an toàn mạng đã xác định của tổ chức. Một số ví dụ về các chính sách an toàn mạng có thể là "cho phép truy nhập tới tất cả các dịch vụ trừ khi bị từ chối thẳng thừng" hoặc "từ chối truy nhập tới tất cả các dịch vụ trừ khi được cho phép rõ ràng". Một khi các chính sách an toàn đã được thiết lập, Firewall-1 có thể trợ giúp

trong việc thực thi một môi trường mạng dựa vào các chính sách đã thiết lập.

Solstice Firewall-1 là một giải pháp phần cứng và phần mềm kết hợp được thiết kế cho phép hoặc không cho phép các goí đi vào mạng nội bộ dựa vào các chính sách an toàn đã thiết lập. Mạng mở rộng thường là một mạng công cộng, chẳng hạn Internet. Tuy nhiên, Firewall-1 cũng có thể được dùng để điều khiển lưu thông giữa các trụ sở khác nhau trong một mạng cục bộ.

Solstice Firewall-1 tổ hợp các tính năng như che gói đặc biệt "nhận ra" giao thức với mức ứng dụng và các cổng mạng để đảm bảo có một máy lọc gói an toàn, tổng quát và hiệu quả. Bổ sung vào công nghệ lọc, nó bao gồm một hệ thống ghi chép và biến đổi đủ mạnh để giúp phòng ngừa những vi phạm cố ý. Một giao diện người dùng trực quan, hướng đối tượng, cài đặt và cấu hình thuận tiện cũng được nói đến.

Solstice SunScreen
Solstice SunScreen kết hợp chức năng firewall với xác thực mức mạng (hay IP), còn gọi là SKIP (quản lý khoá đơn giản đối với IP). Nó là mạng, giao thức và ứng dụng độc lập. Kiến trúc "lấy lén" duy nhất của SunScreen cho phép các tổ chức có khả năng cài đặt mạng bí mật thực sự an toàn qua các kết nối mạng công cộng, chẳng hạn Internet. Vì nó không có router, nên các gói đi qua mà không bị ghi lại bất kỳ dấu hiệu nào về sự tồn tại của nó. Vì thế, Solstice SunScreen không cho những kẻ định xâm nhập thiếu hiểu biết lợi dụng có khả năng phát hiện .

Cấu hình SunScreen bao gồm một thiết bị phần cứng trung tâm (gọi là SunScreen SPF-100) và một trạm quản trị an toàn theo các quy tắc và tham số SunScreen được mô tả. Nó cho phép một vài mạng được quản trị như là mạng đơn lẻ với cùng dãy các địa chỉ IP. Điều này làm giảm nhu cầu về các địa chỉ IP và giao diện bổ sung trong khi vẫn đảm bảo vị trí trung tâm về ghi nhật ký và quản trị.

1.7-Các chuẩn an toàn

Solaris đáp ứng tập các chuẩn an toàn quan trọng mà Cục bảo vệ đưa ra, POSIX và kết nối Internet. Ví dụ:

- Solaris thoả mãn tập điều kiện mà Sách da cam của Cục bảo vệ đưa ra cho các hệ thống an toàn tính toán mức C2. Solaris 2.4SE có chứng nhận E2/F-C2 của ITSEC. Các chứng nhận Solaris đang được xúc tiến.
- Solaris UFS và NFS phiên bản 2 và 3 đáp ứng mô tả POSIX 1003.6 về các danh sách điều khiển truy nhập (ACLs).
- Internet RFC 1508, GSSAPI hiện tại đang triển khai.
- Trạm theo mô hình chia ngăn (CMW-B1) cung cấp cái gọi là "Solaris tin cậy" hiện đang triển khai đối với Solaris 2.X

Trong tương lai, Solaris sẽ tiếp tục hoàn thiện cái mới và đưa ra các chuẩn an toàn đáp lại các yêu cầu của khách hàng.

1.8-Solaris- giải pháp lựa chọn đối với môi trường phân tán an toàn

Không nghi ngờ gì về sự cần thiết của các giải pháp an toàn phức tạp trong môi trường mạng phân tán ngày nay. Solaris trang bị cho các tổ chức những công cụ để bảo vệ dữ liệu nhạy cảm của tổ chức tránh những kẻ xâm nhập với bốn mức an toàn

của nó: Điều khiển Truy nhập Hệ thống, Điều khiển Truy nhập Tài nguyên, các Dịch vụ phân tán an toàn và Bảo vệ Mạng Vật lý. Nó tôn trọng triệt để tập các chuẩn an toàn cao nhất do các tổ chức như Cục bảo vệ và POSIX đưa ra cũng như cung cấp công nghệ mới nhất để truy nhập mạng công cộng an toàn với các sản phẩm rời như Solstic Firewall-1 và Solstice SunScreen. Một số lượng lớn các sản phẩm nhóm ba bổ sung đầy đủ vào những gì đã có ở Sun. Cuối cùng, Solaris đảm bảo tính mềm dẻo với một loạt lựa chọn thoả mãn những nhu cầu thách thức đa dạng phong phú của tổ chức.

CHƯƠNG II -QUẢN LÝ HỆ THỐNG AN TOÀN

Việc giữ an toàn thông tin của hệ thống là một trách nhiệm giám quản hệ thống quan trọng. Chương này cung cấp thông tin tổng quan về quản lý an toàn hệ thống ở mức file, hệ thống và mạng.

Sau đây là danh sách thông tin tổng quan trong chương này.

- "Cấp quyền truy nhập tới hệ thống máy tính" ở mục 2.1
- "An toàn file" ở mục 2.2
- "An toàn hệ thống" ở mục 2.3
- "An toàn mạng" ở mục 2.4

Ở mức file, hệ điều hành SunOS 5.7 cung cấp một số tính năng an toàn chuẩn có thể sử dụng để bảo vệ file, thư mục và thiết bị. Ở các mức hệ thống và mạng, các vấn đề an toàn hầu như giống nhau. Theo vị trí làm việc, một số hệ thống kết nối với server có thể được xem như là một hệ thống đa diện lớn. Người giám quản hệ thống có trách nhiệm bảo vệ hệ thống hay mạng lớn này. Nó không chỉ quan trọng đối với việc bảo vệ mạng tránh những kẻ xâm nhập cố tình truy nhập mạng, mà còn đóng vai trò quan trọng trong việc đảm bảo tính toàn vẹn của dữ liệu trên các hệ thống trong mạng.

2.1-Cho phép truy nhập tới hệ thống máy tính

Bước đầu tiên của việc bảo vệ an toàn là điều khiển truy nhập tới hệ thống của bạn. Bạn có thể điều khiển và theo dõi truy nhập hệ thống bằng:

- Duy trì an toàn cổng vật lý
- Duy trì điều khiển đăng nhập
- Hạn chế truy nhập tới dữ liệu trong các file
- Duy trì điều khiển mạng
- Kiểm soát việc sử dụng hệ thống
- Đặt biến đường dẫn một cách đúng đắn
- An toàn các file
- Theo dõi việc đăng nhập của siêu người dùng (gốc)
- Cài đặt firewall
- Sử dụng công cụ tăng cường an toàn tự động (ASET-Automated Security Enhancement Tool)

Duy trì an toàn cổng vật lý

Để điều khiển truy nhập tới hệ thống của bạn, bạn phải duy trì an toàn vật lý môi trường tính toán của bạn. Chẳng hạn, nếu một hệ thống được nối máy và không bị giám sát, thì bất kỳ ai có thể dùng hệ thống đó đều có thể truy nhập được tới hệ điều hành và mạng. Bạn cần có nhận thức về môi trường xung quanh máy tính của bạn và bảo vệ nó về mặt vật lý tránh những truy nhập trái phép.

Duy trì điều khiển đăng nhập và truy nhập
Bạn cũng cần phải hạn chế những đăng nhập bất hợp pháp vào hệ thống hay mạng, bạn có thể làm điều đó thông qua mật khẩu và điều khiển đăng nhập. Tất cả các tài khoản trên hệ thống nên có một mật khẩu. Một tài khoản không có mật khẩu tạo nên khả năng truy nhập toàn bộ mạng của bạn cho bất kỳ ai có thể đoán được tên người dùng.

Phần mềm hệ thống Solaris 5.7 hạn chế điều khiển của các thiết bị hệ thống nào đó đối với người dùng đăng ký tài khoản. Tiến trình chỉ chạy khi người dùng cao cấp (superuser) hay người dùng bàn điều khiển có thể truy nhập chuột, bàn phím, vùng đệm chính, hay thiết bị âm thanh trừ khi /etc/logindevperm bị cắt xén. Xem logindevperm(4) để có thêm chi tiết.

Hạn chế truy nhập tới dữ liệu trong các file
Sau khi bạn đã thiết lập những hạn chế đăng nhập, bạn có thể điều khiển truy nhập tới dữ liệu trên hệ thống của bạn. Bạn có thể tuỳ ý cho phép một số người đọc một số file, và cho những người khác quyền thay đổi hay xoá một số file. Bạn có thể có một số dữ liệu mà bạn không muốn bất kỳ ai xem nó. Chương 3 đề cập tới việc đặt các quyền đối với file như thế nào.

Duy trì điều khiển mạng
Các máy tính thường là một phần cấu hình của của các hệ thống gọi là mạng. Một mạng cho phép các hệ thống kết nối trao đổi thông tin và truy nhập dữ liệu và các tài nguyên khác có trong các hệ thống được kết nối với mạng. Nối mạng đã tạo nên cách tính toán mạnh và phức tạp. Tuy nhiên, nối mạng cũng gây nguy hiểm cho an toàn máy tính.

Ví dụ, trong mạng máy tính, các hệ thống riêng lẻ được mở cho phép chia xẻ thông tin. Vì nhiều người có thể truy nhập tới mạng, nên cũng có nhiều cơ hội cho phép truy nhập ngoài ý muốn, đặc biệt thông qua lỗi của người dùng (ví dụ, thông qua việc dùng mật khẩu quá đơn giản).

Giám sát sử dụng hệ thống
Khi giám quản hệ thống, bạn cần giám sát hành vi của hệ thống, nhận biết tất cả các khía cạnh trong hệ thống của bạn, gồm:

- Cái gì là tải thông thường?
- Ai đã truy nhập hệ thống?
- Khi nào các cá nhân truy nhập hệ thống?

Với loại thông tin này, bạn có thể sử dung các công cụ sẵn có để kiểm tra việc sử dụng hệ thống và giám sát các hành động của những người dùng đơn lẻ. Việc giám sát rất có lợi khi nghi ngờ có sự vi phạm an toàn.

Đặt đường dẫn đúng
Đặt biến đường dẫn của bạn một cách đúng đắn là quan trọng; nếu không, bạn có thể tình cờ chạy một chương trình theo chỉ dẫn của ai đó có hại cho dữ liệu hay hệ thống của bạn. Loại chương trình tạo nên mối nguy hiểm cho an toàn này được gọi là "Trojan horse". Ví dụ, chương trình su thay thế có thể được đặt ở thư mục công cộng nơi mà bạn có thể chạy nó với tư cách người giám quản hệ thống. Chẳng hạn,

một bản trông giống như một lệnh su thông thường; vì nó tự xoá bỏ sau khi thực hiện, nên bạn khó biết rằng thực tế bạn đã chạy Trojan horse.

Biến đường dẫn được đặt tự động tại thời điểm đăng nhập thông qua các file starup: .login, .profile, và .cshrc. Việc thiết lập đường dẫn tìm kiếm người dùng sao cho thư mục hiện thời (.) trở thành thư mục cuối cùng ngăn bạn hay những người dùng của bạn không chạy kiểu Trojan horse này. Biến đường dẫn đối với superuser không nên có thư mục hiện thời. Tiện ích ASET kiểm tra các file starup để đảm bảo rằng biến đường dẫn được thiết lập đúng và không chứa đầu vào chấm (.).

An toàn các file

Vì hệ điều hành SunOS 5.7 là một hệ đa người dùng, an toàn hệ thống file là những rủi ro an toàn cơ bản và quan trọng nhất trên một hệ thống. Bạn có thể dùng cả bảo vệ file UNIX truyền thống hay các danh sách điều khiển truy nhập (ACLs) an toàn hơn để bảo vệ các file của bạn.

Ngoài ra, nhiều chương trình thực hiện phải được chạy với quyền root (đó là, với superuser) để làm việc đúng đắn. Những chương trình thực hiện này chạy với tập ID người dùng bằng 0 (setuid=0). Bất kỳ ai quản lý những chương trình này chạy chúng với ID gốc đều có thể gây nên một vấn đề an toàn nếu không suy nghĩ về an toàn khi viết các chương trình.

Ngoại trừ các tệp thực hiện được đóng gói với setuid được đặt cho root, bạn không nên cho phép sử dụng chương trình setuid, hoặc ít nhất là giới hạn và giữ việc sử dụng ở mức tối thiểu.

Cài đặt Firewall

Một cách khác bảo vệ mạng của bạn là sử dụng firewall hoặc hệ thống cổng an toàn. Firewall là một hệ thống chuyên dùng tách hai mạng, mỗi một trong chúng xem một cái khác như là mạng không tin cậy. Khi bắt buộc, bạn sẽ quan tâm đến cài đặt này ở giữa mạng nội bộ của bạn và các mạng bên ngoài bất kỳ, chẳng hạn Internet, mà bạn muốn những người dùng mạng nội bộ kết nối.

Firewall cũng có thể hữu ích giữa một số mạng nội bộ. Ví dụ, firewall hay máy tính cổng an toàn sẽ không gửi gói dữ liệu giữa hai mạng trừ phi máy tính cổng là địa chỉ khởi đầu hay đích của gói. Firewall cũng sẽ thiết lập gửi các gói chỉ theo các giao thức cụ thể. Ví dụ, bạn có thể cho phép các gói truyền thư, nhưng không cho các gói này telnet hoặc rlogin. Khi chạy ở mức an toàn cao, tiện ích ASET làm mất khả năng gửi các gói giao thức Internet (IP).

Thông báo các vấn đề an toàn

Nếu bạn có một sự vi phạm an toàn đáng ngờ, bạn có thể liên lạc với Computer Emergency Response Team/Coordination Center (CERT/CC), nơi có một dự án Nhà nước của Defense Advanced Research Projects Agency (DARPA) đặt ở Viện Kỹ thuật Phần mềm tại trường tổng hợp Carnegie Mellon. Nó có thể giúp bạn về bất kỳ vấn đề an toàn nào mà bạn có. Nó cũng có thể giới thiệu bạn với các nhóm Computer Emergency Response có khả năng thích hợp hơn với các nhu cầu cụ thể của bạn. Bạn có thể gọi CERT/CC theo đường dây nóng 24 giờ của nó: (412) 268-7090, hoặc liên lạc với nhóm qua email cert@cert.sei.cmu.edu.

2.2-An toàn file

Hệ điều hành SunOS 5.7 là hệ đa người dùng, nghĩa là tất cả người dùng đăng nhập vào hệ thống có thể đọc và sử dụng các file thuộc về một người khác, cụ thể hơn họ có quyền làm như vậy. Bảng 2-1 mô tả ccác lệnh giám quản hệ thống file. Xem chương 3 để có những hướng dẫn từng bước về an toàn file.

Các lệnh quản lý file

Bảng 2-1 liệt kê các lệnh quản lý file mà bạn có thể sử dụng trên các file hoặc thư mục.

Bảng 2-1 Các lệnh quản lý file

Lệnh	Mô tả
ls(1)	Liệt kê các file trong một thư mục và thông tin về chúng
chown(1)	Thay đổi quyền sở hữu một file
chgrp(1)	Thay đổi quyền sở hữu nhóm của một file
chmod(1)	Thay đổi các quyền trên một file. Bạn có thể hoặc sử dụng kiểu trừu tượng (các ký tự hoặc ký hiệu), hoặc kiểu chính xác (các số hệ 8) để thay đổi các quyền trên một file.

Mã hoá file

Việc đặt file nhạy cảm vào thư mục không thể truy nhập được (kiểu 700) và làm cho những người khác không thể đọc được file (kiểu 600) sẽ giữ cho nó an toàn trong hầu hết các trường hợp. Tuy nhiên, một số người doán được mật khẩu của bạn hay mật khẩu gốc có thể đọc và ghi file đó. Ngoài ra, file nhạy cảm được bảo quản trên các băng sao lưu mỗi lần bạn sao dự phòng các file hệ thống vào băng.

May thay, một tầng an toàn bổ sung sẵn có cho tất cả những người dùng phần mềm hệ thống SunOS 5.7 ở Mỹ - bộ công cụ mã hoá file tuỳ chọn. Bộ công cụ mã hoá gồm lệnh crypt(1) xáo trộn dữ liệu để không thể đoán được văn bản.

Các danh sách điều khiển truy nhập (ACLs)

ACLs có thể đảm bảo điều khiển tốt hơn trên các quyền file khi sự bảo vệ file UNIX truyền thống theo hệ điều hành SunOS không đủ. Bảo vệ file UNIX truyền thống cung cấp các quyền đọc, ghi, và thực hiện cho ba lớp người dùng: người sở hữu, nhóm và còn lại. Một ACL đảm bảo an toàn file tốt hơn bằng việc cho phép bạn định nghĩa các quyền file đối với người sở hữu, nhóm sở hữu, những người khác, những người dùng và những nhóm đặc biệt, và các quyền ngầm định cho mỗi loại này. Xem "Sử dụng các danh sách điều khiển truy nhập (ACLs) ở chương III để có các chỉ dẫn từng bước về việc dùng ACLs.

Bảng 2-2 liệt kê các lệnh ACL mà bạn có thể dùng trên các file hoặc thư mục.

Bảng 2-2 Các lệnh ACL

Lệnh	Mô tả
setfacl(1)	Chọn, thêm, thay đổi và xoá những khoản mục ACL
getfacl(1)	Hiển thị các khoản mục ACL

2.3- An toàn hệ thống

Phần này mô tả cách làm thế nào để bảo vệ hệ thống của bạn chống lại truy nhập trái phép, chẳng hạn làm thế nào để ngăn chặn một kẻ đột nhập đăng nhập vào hệ thống của bạn, làm thế nào để lưu giữ các file mật khẩu, và làm thế nào để ngăn chặn superuser trái phép truy nhập tới các file và các chương trình hệ thống nhạy cảm.

Bạn có thể cài đặt hai hàng rào an toàn trên hệ thống. Hàng rào an toàn thứ nhất là chương trình đăng nhập. Để vượt qua hàng rào này và truy nhập được vào một hệ thống, người dùng phải cung cấp tên người dùng và mật khẩu tương ứng mà hệ thống cục bộ hoặc dịch vụ tên (NIS hoặc NIS+) nhận biết được.

Hàng rào an toàn thứ hai đảm bảo rằng chỉ superuser có thể thay đổi hoặc di chuyển các file và các chương trình hệ thống. Một superuser muốn làm được phải cung cấp tên người dùng gốc và mật khẩu đúng của nó.

Những hạn chế đăng ký truy nhập

Khi người dùng đăng nhập vào hệ thống, chương trình đăng nhập tra cứu cơ sở dữ liệu thích hợp theo thông tin liệt kê trong file /etc/nsswitch.conf. Những đầu vào trong file này có thể gồm files (chỉ những file /etc), nis (chỉ cơ sở dữ liệu NIS) và nisplus (chỉ cơ sở dữ liệu NIS+). Xem Solaris Naming Administration Guide hoặc nsswitch.conf(4) về đặc tả của file này.

Chương trình đăng nhập phê chuẩn tên và mật khẩu người dùng đưa vào. Nếu tên người dùng không nằm trong file mật khẩu hoặc mật khẩu không đúng với tên người dùng, chương trình đăng nhập từ chối truy nhập hệ thống. Khi người dùng cung cấp tên có trong file mật khẩu và mật khẩu đúng với tên, hệ thống cho phép người dùng truy nhập hệ thống.

Các cách đăng nhập đặc biệt

Có hai cách phổ biến để truy nhập một hệ thống - dùng cách đăng nhập người dùng quy ước hay dùng cách đăng nhập gốc. Bên cạnh đó, một số cách đăng nhập hệ thống đặc biệt cho phép người dùng thi hành các lệnh quản trị mà không dùng tài khoản gốc. Người giám quản gán mật khẩu cho các tài khoản đăng nhập này.

Bảng 2-3 liệt kê các tài khoản đăng nhập hệ thống và cách sử dụng chúng. Các cách đăng nhập hệ thống thực hiện các chức năng cụ thể, và mỗi cách có số định danh nhóm riêng của nó (GID). Mỗi một cách đăng nhập này sẽ có mật khẩu riêng của nó mà sẽ được phân phối trên cơ sở need-to-know.

Bảng 2-3 Các cách đăng nhập hệ thống

Tài khoản đăng nhập	GID	Cách dùng
root	0	Hầu như không có hạn chế và không tính đến tất cả các cách đăng nhập, bảo vệ và các quyền khác. Tài khoản gốc có quyền truy nhập tới hệ thống đầu vào. Mật khẩu để đăng nhập gốc cần được bảo vệ rất cẩn thận .
daemon	1	Điều khiển xử lý nền
bin	2	Sở hữu đa số lệnh

sys	3	Sở hữu nhiều file hệ thống
adm	4	Sở hữu các file quản trị nào đó.
lp	71	Sở hữu các file đối tượng và dữ liệu tác vụ song song với máy in.
uucp	5	Sở hữu các file đối tượng và dữ liệu tác vụ song song với UUCP, chương trình sao chép UNIX-to-UNIX.
nuucp	9	Được dùng cho các hệ thống từ xa đăng nhập hệ thống và bắt đầu các cuộc truyền file.

Bạn cũng nên đặt an toàn của lệnh eeprom đòi hỏi mật khẩu. Xem eeprom(1M) để biết thêm chi tiết.

Quản lý thông tin mật khẩu

Khi đăng nhập vào hệ thống, người dùng phải đưa vào cả tên người dùng và mật khẩu. Mặc dù các cách đăng nhập được biết công khai, các mật khẩu phải được giữ bí mật và chỉ người dùng biết. Bạn nên đề nghị những người dùng của bạn chọn mật khẩu của họ một cách cẩn thận, và thay đổi chúng thường xuyên.

Các mật khẩu được tạo ra từ đầu khi bạn cài đặt tài khoản người dùng. Để duy trì an toàn cho các tài khoản người dùng, bạn có thể cài đặt thời hạn mật khẩu để buộc người dùng định kỳ thay đổi mật khẩu của họ, và bạn cũng có thể làm mất hiệu lực tài khoản người dùng băng việc chốt mật khẩu. Xem "Quản lý các tài khoản và nhóm người dùng (Tổng quan)" trong System Adminitration Guide, Volumn I để có thông tin chi tiết về cách cài đặt và lưu giữ các mật khẩu.

File mật khẩu NIS+

Nếu mạng của bạn sử dụng NIS+, thông tin về mật khẩu được lưu giữ trong cơ sở dữ liệu NIS+. Thông tin trong cơ sở dữ liệu NIS+ có thể được bảo vệ bằng việc hạn chế truy nhập của những người dùng hợp pháp. Bạn có thể sử dụng Solstice User Manager hoặc lệnh passwd(1) để thay đổi mật khẩu NIS+ của người dùng.

File mật khẩu NIS

Nếu mạng của bạn sử dụng NIS, thông tin về mật khẩu được lưu giữ trong bản đồ mật khẩu NIS. NIS không cung cấp thời hạn mật khẩu. Bạn có thể sử dụng Solstice™ User Manager hoặc lệnh passwd(1) để thay đổi mật khẩu NIS của người dùng.

Các file /etc

Nếu mạng của bạn sử dụng các file /etc, thông tin về mật khẩu được lưu giữ trong các file /etc/passwd và /etc/shadow. Tên người dùng và thông tin khác được lưu giữ trong file mật khẩu /etc/passwd, trong khi bản thân mật khẩu được mã hoá lưu giữ trong file shadow riêng, /etc/shadow. Đó là biện pháp an toàn ngăn chặn người dùng truy nhập tới các mật khẩu đã mã hoá. Trong khi tất cả mọi người có thể đăng nhập vào máy đều có thể tiếp cận với file /etc/passwd, thì chỉ superuser mới có thể đọc được file /etc/shadow. Bạn có thể sử dụng Solstice AdminSuite's User Manager, Admintool, hoặc lệnh passwd(1) để thay đổi mật khẩu của người dùng trên hệ thống cục bộ.

Sử dụng Shell hạn chế
Shell chuẩn cho phép người dùng mở các file, thực hiện các lệnh và .....Shell hạn chế có thể được dùng để giới hạn khả năng thay đổi thư mục và thực hiện các lệnh của người dùng. Shell hạn chế (rsh) được đặt trong thư mục /usr/lib. (Lưu ý rằng đó không phải là Shell điều khiển từ xa, đó là /usr/sbin/rsh). Shell hạn chế khác với Shell thông thường ở những điểm sau:

- Người dùng bị hạn chế ở thư mục đầu (không thể dùng cd để thay đổi thư mục).
- Người dùng chỉ có thể sử dụng các lệnh trong tập PATH của người giám quản hệ thống (không thể thay đổi biến PATH).
- Người dùng chỉ có thể truy nhập những file trong thư mục đầu và các thư mục con của nó (không thể đặt tên các lệnh hoặc file sử dụng tên đường dẫn đầy đủ).
- Người dùng không thể gửi đi một lần nữa với > hoặc >>.

Shell hạn chế cho phép người giám quản hệ thống hạn chế khả năng người dùng lạc vào các file hệ thống, và chủ yếu nhằm dựng lên người dùng cần thực hiện những công việc cụ thể. Tuy nhiên, rsh không hoàn toàn an toàn, nó chỉ nhằm giữ cho những người dùng không thạo khỏi lâm vào rắc rối.

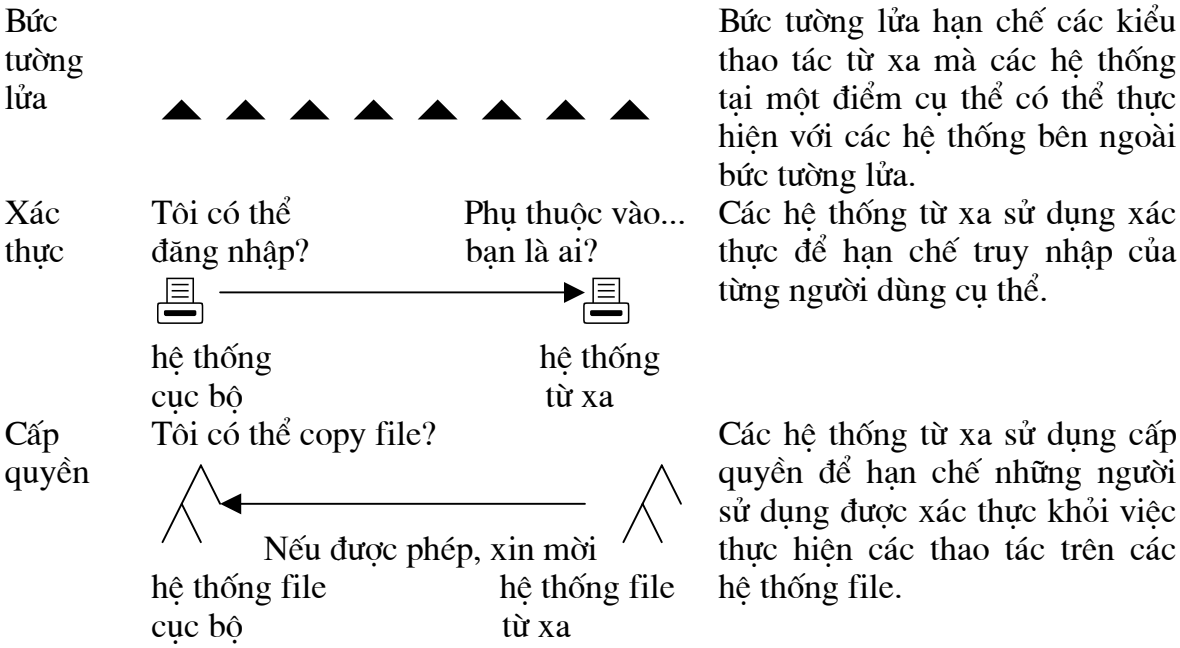
Xem sh(1) để có thông tin về shell hạn chế.

Theo dõi đăng nhập superuser (gốc)
Hệ thống của bạn yêu cầu mật khẩu gốc cho chế độ superuser. Trong cấu hình ngầm định, người dùng không thể từ xa đăng nhập vào hệ thống như là gốc. Khi đăng nhập từ xa, người dùng phải đăng nhập theo bản thân anh ta và sau đó dùng lệnh su để trở thành gốc. Điều này cho phép bạn theo dõi ai đang dùng các quyền superuser trên hệ thống của bạn.

Giám sát superuser hay những người dùng khác
Bạn phải dùng lệnh su để biến đổi thành một người dùng khác, ví dụ khi bạn muốn trở thành superuser. Để có những lý lẽ an toàn, bạn có thể muốn giám sát ai đã dùng lệnh su, đặc biệt ai đang cố gắng có được quyền truy nhập superuser của người dùng này.

Xem "Làm thế nào để giám sát người dùng lệnh su" ở chương III để có những thông tin chi tiết.

2.4-An toàn mạng
Truy nhập có giá trị hơn khi qua mạng, nó thuận lợi hơn đối với các hệ thống nối mạng. Tuy nhiên, truy nhập tự do và chia xẻ dữ liệu và tài nguyên tạo nên các vấn đề về an toàn. An toàn mạng thường dựa vào việc hạn chế hay chặn các thao tác từ các hệ thống từ xa. Hình 2-1 mô tả những hạn chế an toàn mà bạn có thể đặt trên các thao tác điều khiển từ xa.



Hình 2-1 Các hạn chế an toàn đối với các thao tác từ xa

Các hệ thống Firewall

Bạn có thể thiết lập một hệ thống firewall để bảo vệ tài nguyên trong mạng của bạn khỏi truy nhập bên ngoài. Một hệ thống firewall là một máy chủ an toàn hoạt động như một hàng rào giữa mạng nội bộ của bạn và các mạng bên ngoài.

Firewall có hai chức năng. Nó hoạt động như là một cổng cho dữ liệu đi qua giữa các mạng, và nó hoạt động như là một hàng rào chặn gói dữ liệu tự do đến và đi khỏi mạng. Firewall yêu cầu người dùng trên mạng nội bộ đăng nhập hệ thống firewall để truy nhập tới các maý chủ trên các mạng từ xa. Một cách tương tự, người dùng trên một mạng bên ngoài phải đăng nhập hệ thống firewall trước khi được phép truy nhập một máy chủ trên mạng nội bộ.

Ngoài ra, tất cả thư điện tử gửi từ mạng nội bộ là gửi tới hệ thống firewall để truyền tới máy chủ trên mạng mở rộng. Hệ thống firewall nhận tất cả thư điện tử đến và phân phát tới các máy chủ trên mạng nội bộ.

Chú ý - Firewall ngăn chặn những người dùng trái phép truy nhập các máy chủ trên mạng của bạn. Bạn sẽ duy trì chặt chẽ và đòi hỏi khắt khe về an toàn trên firewall, nhưng an toàn trên các máy chủ khác trên mạng có thể bị xao nhãng hơn. Tuy nhiên, một kẻ đột nhập có thể can thiệp vào hệ thống firewall của bạn, sau đó có thể truy nhập tới tất cả các máy chủ khác trên mạng nội bộ.

Hệ thống firewall sẽ không có bất kỳ máy chủ tin cậy nào. (Một máy chủ tin cậy là máy mà người dùng có thể đăng nhập không đòi hỏi kiểu mật khẩu.) Nó sẽ không chia xẻ bất kỳ file hệ thông nào của nó, hay phần lớn các file hệ thống với các server khác.

ASET có thể dùng để làm cho một hệ thống thành firewall và tuân theo mức an toàn cao trên hệ thống firewall, như mô tả trong chương 6.

Bắt gói

Hầu hết các mạng cục bộ truyền dữ liệu giữa các máy tính theo các khối gọi là gói. Thông qua một thủ tục gọi là packet smashing, những người dùng trái phép có thể làm hỏng hoặc phá hoại dữ liệu. Packet smashing bao hàm việc bắt các gói trước khi chúng tới đích, cấy dữ liệu tuỳ ý vào nội dung, sau đó gửi những gói này trở lại hướng ban đầu của chúng. Trên một mạng cục bộ, không thể packet smashing, vì tại cùng thời điểm các gói đi đến tất cả các hệ thống kể cả server. Tuy nhiên, trên một cổng có thể packet smashing, những phải chắc chắn là tất cả các cổng trên mạng được bảo vệ.

Những tấn công nguy hiểm nhất là những tấn công ảnh hưởng tới tính toàn vẹn của dữ liệu. Những tấn công như thế liên quan tới việc thay đổi nội dung các gói hoặc mạo nhận người dùng. Những tấn công liên quan đến việc nghe trộm - ghi lại các cuộc nói chuyện và nghe lại chúng sau đó không cần thủ vai người dùng - không làm tổn thương tính toàn vẹn. Tuy nhiên, những tấn công này ảnh hưởng tới tính bí mật. Bạn có thể bảo vệ bí mật của thông tin nhạy cảm bằng cách mã hoá dữ liệu lưu thông trên mạng.

Xác thực và cấp phép

Xác thực là một cách hạn chế truy nhập đối với những người dùng cụ thể khi truy nhập một hệ thống từ xa, có thể cài đặt ở cả hai mức hệ thống hay mạng. Một khi người dùng truy nhập được tới một hệ thống từ xa, thì cấp phép là một cách để hạn chế các thao tác mà người dùng có thể thực hiện trên hệ thống từ xa. Bảng 2-4 liệt kê các kiểu xác thực và cấp phép có thể trợ giúp bảo vệ các hệ thống của bạn trên mạng chôngs lại việc sử dụng trái phép.

Bảng 2-4 Các kiểu Xác thực và Cấp phép.

Kiểu	Mô tả	Nơi tìm thông tin
NIS+	Dịch vụ đặt tên NIS+ có thể cung cấp cả xác thực và cấp phép ở mức mạng.	Solaris Naming Adminitration Guide.
Các chương trình đăng nhập từ xa.	Các chương trình đăng nhập từ xa (rlogin, rcp, ftp) cho phép người dùng đăng nhập vào một hệ thống từ xa trên mạng và sử dụng tài nguyên của nó. Nếu bạn có một "máy chủ tin cậy", xác thực là tự động, ngược lại, bạn được yêu cầu xác thực bản thân bạn.	Solaris System Administration Guide, Chương 8.
RPC an toàn	RPC an toàn hoàn thiện an ninh của các môi trường mạng bằng việc xác thực người dùng đưa ra các yêu cầu trên các hệ thống từ xa. Bạn có thể dùng hoặc UNIX, DES, hoặc hệ thống xác thực Kerberos cho RPC an toàn. RPC cũng có thể được dùng để tăng	NFS Adminitration Guide. "NFS Sevices and

	cường an toàn cho môi trường NFS, gọi là NFS an toàn.	Secure RPC" ở chương V.
Mã hoá DES	Các hàm mã hoá theo chuẩn mã hoá dữ liệu (DES) dùng khoá 56-bit để mã hoá khoá bí mật.	"DES Encryption " ở chương V.
Xác thực Diffie-Hellman	Phương pháp xác thực này dựa vào khả năng của hệ thống gửi dùng khoá công khai để mã hoá thời hiện tại, mà hệ thống nhận có thể giải mã và kiểm tra dựa vào thời hiện tại của nó.	"Diffie-Hellman Authentication" ở chương V.
Kerberos Version 4.	Kerberos dùng mã DES để xác thực người dùng khi đăng nhập vào hệ thống.	"Kerberos Version 4" ở chương V.
Solstice AdminSuite	Sản phẩm Solstice AdminSuite cung cấp các cơ chế xác thực và cấp phép để quản lý các hệ thống từ xa với các công cụ AdminSuite	Solstice AdminSuite 2.3 Adminitration Guide.

Chia xẻ các file

Một file server mạng có thể điều khiển những file nào có thể dùng chung. Nó cũng có thể điều khiển những client nào đã truy nhập tới file và kiểu truy nhập nào là hợp pháp đối với các client này. Nói chung, file server có thể cấp quyền truy nhập đọc/ghi hoặc chỉ đọc hoặc cho tất cả client hoặc cho một số client cụ thể. Điều khiển truy nhập được mô tả khi các tài nguyên khả dụng với lệnh share.

Một server có thể dùng file /etc/dfs/dfstab để liệt kê các hệ thống file khả dụng đối với client trên mạng. Xem "NFS Adminitration Guide" để có thêm thông tin về chia xẻ file.

Hạn chế truy nhập superuser (gốc)

Nói chung, superuser không được phép truy nhập gốc đối với các hệ thống file dùng chung qua mạng. Trừ khi server cấp các quyền cho superuser một cách cụ thể, còn lại người dùng đăng nhập với vai trò superuser trên client không thể truy nhập gốc tới các file được đặt từ xa trên client. Hệ thống NFS thực hiện điều này bằng cách thay đổi ID người dùng của người yêu cầu thành ID người sử dụng ứng với tên người dùng nobody; trong trường hợp chung là 60001. Các quyền truy nhập của người dùng nobody cũng tương tự như các quyền đưa ra cho public (hoặc người dùng không được uỷ quyền) đối với file cụ thể. Ví dụ, nếu public chỉ có quyền thực hiện đối với một file, thì người dùng nobody chỉ có thể thực hiện file đó.

Một server NFS có thể cấp các quyền superuser trên hệ thống file dùng chung trên cơ sở từng máy chủ dùng lựa chọn root=hostname với lệnh share.

Sử dụng các cổng bí mật

Nếu bạn không muốn chạy RPC an toàn, thì cái có thể thay thế là cơ chế "cổng bí mật" Solaris. Một cổng bí mật do superuser xây dựng với số cổng ít hơn 1024. Sau khi hệ thống client được xác thực uỷ quyền của client, nó thiết lập kết nối với server thông qua cổng bí mật. Sau đó server phê chuẩn uỷ quyền client bằng việc kiểm tra số cổng của kết nối.

Tuy nhiên các client Non-Solaris không có khả năng kết nối thông qua cổng bí mật. Vì chúng không thể, nên bạn sẽ thấy các thông báo lỗi chẳng hạn như:

```
" Weak Authentication
NFS request from unprivileged port"
```

Sử dụng công cụ tăng cường an toàn tự động (ASET)

Gói an toàn ASET cung cấp các công cụ quản trị tự động cho phép bạn điều khiển và giám sát an toanf hệ thống của bạn. Bạn mô tả mức độ an toàn - thấp, trung bình, hoặc cao - khi ASET chạy. Ở mỗi mức cao hơn, các hàm điều khiển file của ASET tăng cường giảm quyền truy nhập file và siết chặt an toàn hệ thống của bạn.

Xem chương VI để biết thêm thông tin.

CHƯƠNG III- CÁC TÁC VỤ AN TOÀN FILE

Chương này mô tả các thủ tục về an toàn file. Dưới đây là một danh sách những chỉ dẫn từng bước trong chương này.

- "Cách hiển thị thông tin file" ở mục 3.2.1
- "Thay đổi quyền sở hữu file như thế nào" ở mục 3.3.1
- "Thay đổi quyền sở hữu file của nhóm như thế nào" ở mục 3.3.2
- "Cách thay đổi các quyền theo kiểu trực tiếp" ở mục 3.4.1
- "Cách thay đổi các quyền đặc biệt theo kiểu trực tiếp" ở mục 3.4.2
- "Cách thay đổi các quyền theo kiểu ký hiệu" ở mục 3.4.3
- "Cách tìm các file có các quyền setuid" ở mục 3.5.1
- "Làm thế nào để các chương trình không sử dụng các stack khả thi" ở mục 3.6.1
- "Cách cài đặt một ACL trên một file" ở mục 3.7.3
- "Cách kiểm tra một file có ACL " ở mục 3.7.5
- "Cách thay đổi các đầu vào ACL (ACL entry) trên một file" ở mục 3.7.6
- "Cách hiển thị các ACL entry của một file" ở mục 3.7.8

3.1-Các tính năng an toàn file

Phần này mô tả các tính năng tạo nên an toàn của một file.

3.1.1-Các lớp người dùng

Đối với mỗi file, có ba lớp người dùng định rõ ba mức an toàn:

- Người sở hữu file hoặc thư mục (owner) - thường là người dùng tạo ra file. File owner có thể quyết định ai có quyền đọc, ghi (thay đổi) nó, hoặc khi nó là một lệnh thì thực hiện nó.
- Các thành viên của một nhóm (group).
- Tất cả những người dùng khác không phải owner hay group.

Chỉ file owner hoặc root có thể gán hoặc thay đổi các quyền file.

3.1.2-Các quyền file

Bảng 3-1 liệt kê và mô tả các quyền mà bạn có thể đưa ra cho mỗi lớp người dùng đối với file.

Bảng 3-1 Các quyền file

Ký hiệu	Quyền	Cho phép người được gán quyền
r	Read	Có thể mở và đọc nội dung file
w	Write	Có thể ghi vào file (thay đổi nội dung của nó), bổ sung, hay xoá file.
x	Execute	Có thể thực hiện file (nếu nó là một chương trình hay shell script), hoặc chạy file với một trong các lời gọi exec(1).
-	Denied	Không thể đọc, ghi hay thi hành file.

Các quyền file này áp dụng cho các file đặc biệt chẳng hạn các thiết bị, sockets và named pipes (FIFOs) như là thực hiện với các file thông thường.

Đối với liên kết ký hiệu, các quyền áp dụng là các quyền của file mà liên kết hướng tới.

3.1.3-Các quyền thư mục

Bảng 3-2 liệt kê và mô tả các quyền mà bạn có thể đưa ra cho mỗi lớp người dùng đối với một thư mục.

Bảng 3-2 Các quyền thư mục

Ký hiệu	Quyền	Cho phép người được gán quyền
r	Read	Hiển thị các file trong thư mục
w	Write	Thêm hoặc xoá các file hoặc các liên kết trong thư mục.
x	Execute	Mở hoặc thi hành các file trong thư mục. Cũng có thể tạo thư mục và các thư mục hiện thời bên trong nó.

3.1.4-Các quyền file đặc biệt (setuid, setgid và Sticky Bit)

Ba loại quyền đặc biệt này có thể có đối với các file khả thi và các thư mục public. Khi các quyền này được đặt, thì bất kỳ người dùng nào chạy file khả thi đều được gán user ID của người sở hữu (hoặc nhóm) file khả thi.

Bạn phải rất cẩn thận khi đặt các quyền đặc biệt này, vì các quyền đặc biệt tạo nên rủi ro về an toàn. Ví dụ, một người có thể có được quyền superuser bằng việc thực thi chương trình cài đặt user ID cho root. Ngoài ra, tất cả người dùng có thể đặt các quyền đặc biệt đối với file mà họ sở hữu tạo nên một mối lo lắng an toàn khác.

Bạn nên giám sát hệ thống của bạn đối với việc tuỳ tiện sử dụng các quyền setuid và setgid để có được các quyền superuser. Xem "Cách tìm thấy các file có quyền setuid" ở trong chương này để tìm kiếm các hệ thống file và in ra danh sách tất cả các chương trình sử dụng các quyền này. Danh sách đáng ngờ sẽ là danh sách cấp quyền sở hữu chương trình như thế cho người dùng mà không phải là bin hay sys.

Quyền setuid

Khi quyền đặt định danh người dùng (setuid) được thiết lập trên một file khả thi, thì một tiến trình chạy file này được cấp quyền truy nhập dựa vào file owner (thường là root), mà không phải là người dùng đang chạy file khả thi. Điều này cho phép người dùng truy nhập các file và thư mục thường là người sở hữu sẵn có. Ví dụ, quyền setuid ở lệnh passwd khiến nó có thể làm cho người dùng thay đổi các mật khẩu công nhận các quyền của root ID:

```
-r-sr-sr-x  1  root  sys  10322 May  3 08:23  /usr/bin/passwd
```

Điều này chỉ ra một rủi ro về an toàn, vì một số người dùng xác định có thể tìm cách duy trì các quyền đã cấp cho họ bằng tiến trình setuid ngay cả sau khi tiến trình đã kết thúc thực hiện.

Ghi chú - Việc dùng các quyền setuid với UIDs (0-99) dành riêng có thể không cài đặt UID có hiệu quả một cách đúng đắn. Nên dùng shell script thay thế hoặc tránh dùng UIDs dành riêng với các quyền setuid.

Quyền setgid
Quyền đặt định danh nhóm (setgid) tương tự với setuid, ngoại trừ là ID nhóm có hiệu lực của tiến trình do nhóm sở hữu file thay đổi, và người dùng được cấp quyền truy nhập dựa vào các quyền đã cấp cho nhóm đó. Chương trình /usr/bin/mail có các quyền setgid:

```
-r-x—s—x 1 bin mail 62504 May  3 07:58  /usr/bin/mail
```

Khi quyền setgid được áp dụng cho một thư mục, các file được tạo ra trong thư mục này thuộc về nhóm sở hữu thư mục, không phải nhóm tạo ra nó. Bất kỳ người dùng nào có các quyền ghi và thực hiện trong thư mục đều có thể tạo file ở đó - tuy nhiên, file không thuộc về nhóm của người dùng, mà thuộc về nhóm của thư mục.

Bạn nên giám sát hệ thống của bạn tránh việc tuỳ tiện sử dụng các quyền setuid và setgid để có được các quyền superuser. Xem "Cách tìm thấy các file có quyền setuid" ở trong chương này để tìm kiếm các hệ thống file và in ra danh sách tất cả các chương trình sử dụng các quyền này. Danh sách đáng ngờ sẽ là danh sách cấp quyền sở hữu chương trình như thế cho người dùng mà không phải là bin hay sys.

Sticky Bit
Sticky bit là bit quyền bảo vệ file trong một thư mục. Nếu thư mục có sticky bit được đặt, thì chỉ có file owner, người sở hữu thư mục hay root mới có thể xoá file. Điều này ngăn chặn việc một người dùng xoá file của những người dùng khác khỏi thư mục public chẳng hạn /tmp:

```
drwxrwxrwt 7  sys  sys   517 Mar   6  02:01  tmp
```

Để chắc chắn nên đặt sticky bit một cách thủ công khi bạn tạo một thư mục public trên hệ thống file TMPFS.

3.1.5-Umask mặc định
Khi bạn tạo một file hay thư mục, nó có một tập quyền mặc định. Các quyền mặc định này được xác định bởi giá trị của umask(1) trong file hệ thống /etc/profile, hoặc trong file .cshrc hay .login của bạn. Theo mặc định, hệ thống đặt các quyền trên một file text là 666, cấp quyền đọc và ghi cho người dùng, nhóm và những lớp khác, và là 777 trên một thư mục hay file khả thi.

Giá trị do umask gán không được mặc định. Điều này có tác dụng từ chối các quyền cùng kiểu mà chmod cấp cho chúng. Ví dụ, trong khi lệnh chmod 022 cấp quyền ghi cho nhóm và những lớp khác, thì umask 022 từ chối quyền ghi đối với nhóm và những lớp khác.

Bảng 3-3 đưa ra một số cài đặt umask điển hình và có hiệu lực trên file khả thi.

Bảng 3-3 Các cài đặt umask cho các mức an toàn khác nhau

Mức an toàn	umask	**Từ chối**
Permissive (744)	022	w đối với nhóm và những lớp khác

Moderate (740)	027	w đối với nhóm, rwx đối với các lớp khác
Moderate (741)	026	w đối với nhóm, rw đối với các lớp khác
Severe (700)	077	rw đối với nhóm và những lớp khác

3.2-Hiển thị thông tin file

3.2.1- Cách hiển thị thông tin file

Hiển thị thông tin về tất cả các file trong một thư mục bằng cách dùng lệnh ls.

```
$ ls -la
```

-l Hiển thị dạng chi tiết
-a Hiển thị tất cả các file kể cả file ẩn bắt đầu bằng dấu chấm (.).

Mỗi dòng có thông tin sau về file:

- Kiểu file

Một file có thể có một trong sáu kiểu. Bảng 3-4 liệt kê các kiểu file có thể có.

Bảng 3-4 Các kiểu file

Ký hiệu	Kiểu
-	Text hoặc chương trình
d	Thư mục
b	File block đặc biệt
c	File ký tự đặc biệt
p	Named pipe (FIFO)
l	Symblic link

- Các quyền; xem mô tả ở bảng 3-1 và bảng 3-2
- Số liên kết cứng
- File owner
- Nhóm của file
- Kích thước file tính bằng byte
- Ngày tạo file hay ngày cập nhật file cuối cùng
- Tên file

Ví dụ - Hiển thị thông tin về file

Ví dụ sau đây hiển thị danh sách các file cụ thể trong thư mục /sbin

```
$ cd /sbin
$ ls -la
total 11652
drwxrwxr-x   2  root  sys      512  Jun  2  11:47  . /
drwxr-xr-x  30  root  root     512  Jun  3  14:13  .. /
-r-xr-xr-x   1  bin   bin    199224  May  6   21:23  autopush*
lrwxrwxrwx   1  root  root       21  Jun  2  11:47  bpgetfile -> ...
-r-xr-xr-x   1  bin   bin    467856  May  6   21:23  dhcpagent*
-r-xr-xr-x   1  bin   bin    430172  May  6   21:23  dhcpinfo*
-r-xr-xr-x   1  bin   bin    251500  May  6   21:23  fdisk*
```

```
-r-xr-xr-x    1  bin    bin      762136  May   6    21:29  hostconfig*
-r-xr-xr-x    1  bin    bin      533272  May   6    21:30  ifconfig*
-r-xr-xr-x    1  root   sys      515296  May   6    21:25  init*
-r-xr-xr-x    2  bin    root     256272  May   6    21:27  jsh*
-r-xr-xr-x    1  bin    bin      223448  May   7    20:06  mount*
-r-xr-xr-x    1  root   sys        6935  Jan   1     1970  mountall*
      .
      .
      .
```

3.3-Thay đổi quyền sở hữu file

3.3.1-Cách thay đổi file owner

1. Trở thành superuser
Theo mặc định, người sở hữu không thể sử dụng lệnh chown để thay đổi file owner hay thư mục. Tuy nhiên, bạn có thể cho phép người sở hữu dùng chown bằng cách bổ sung dòng sau đây vào file /etc/system của hệ thống và khởi động lại hệ thống.

```
set  rstchown = 0
```

Xem chown(1) để biết thêm chi tiết. Ngoài ra, nhận thấy rằng có thể có những hạn chế khác trong việc thay đổi quyền sở hữu trên các hệ thống file NFS.

2. Thay đổi file owner bằng cách dùng lệnh chown.

```
#  chown  newowner   filename
```

newowner đặc tả tên người dùng hoặc UID của file owner hay thư mục mới.
filename đặc tả file hay thư mục.

3. Xác nhận file owner bị thay đổi

```
#  ls  -l  filename
```

Ví dụ - Thay đổi file owner
Ví dụ sau đây đặt quyền sở hữu trên myfile cho người dùng rimmer.

```
#  chown  rimmer  myfile
#  ls  -1 myfile
-rw-r--r--      1  rimmer     scifi    112640  May  24  10:49  myfile
```

3.3.2-Cách thay đổi quyền sở hữu nhóm của một file

1. Trở thành superuser
Theo mặc định, người sở hữu chỉ có thể dùng lệnh chgrp để thay đổi nhóm của file thành nhóm trong đó có người sở hữu. Ví dụ, nếu file owner chỉ thuộc vào các nhóm

staff và sysadm, thì người sở hữu chỉ có thể thay đổi nhóm của file thành nhóm staff hay sysadm mà thôi.

Tuy nhiên, bạn có thể cho phép người sở hữu thay đổi nhóm của file thành nhóm trong đó không có người sở hữu bằng cách bổ sung dòng sau đây vào file /etc/system của hệ thống và khởi động lại hệ thống.

```
set  rstchown = 0
```

Xem chgrp(1) để có thêm chi tiết. Ngoài ra, nhận thấy rằng có thể có những hạn chế khác trong việc thay đổi các nhóm trên các hệ thống file NFS.

2. Thay đổi nhóm sở hữu file bằng cách dùng lệnh chgrp.

```
$  chgrp  group  filename
```

group Đặc tả tên nhóm hay GID của nhóm mới của file hay thư mục.
filename Đặc tả file hay thư mục

Xem chương 13 để có thông tin về cách soạn thảo các tài khoản nhóm.

3. Xác nhận nhóm sở hữu file bị thay đổi

```
$  ls  -l  filename
```

Ví dụ - Thay đổi quyền sở hữu nhóm của file
Ví dụ sau đây đặt quyền sở hữu nhóm trên myfile thành nhóm scifi.

```
$  chgrp  scifi  myfile
$  ls  -l  myfile
-rwxrw--  1  rimmer  scifi  12985  Nov  12  16:28  myfile
```

3.4-Thay đổi các quyền file

Lệnh chmod cho phép bạn thay đổi các quyền trên file. Bạn phải là superuser hoặc file owner hay thư mục khi thay đổi các quyền của nó.

Bạn có thể sử dụng lệnh chmod để đặt các quyền theo một trong hai kiểu:

- **Kiểu trực tiếp** - Dùng các số biểu diễn các quyền file (phương pháp này được dùng để đặt các quyền phổ biến nhất). Khi bạn thay đổi các quyền bằng cách dùng kiểu trực tiếp, bạn biểu diễn các quyền đối với mỗi bộ ba bằng một số kiểu octal.
- **Kiểu ký hiệu** - Dùng các dãy các ký tự và ký hiệu để bổ sung hay loại bỏ các quyền.

Bảng 3-5 liệt kê các giá trị octal để đặt các quyền file theo mô hình tuyệt đối. Bạn sử dụng những số này trong các tập ba số để đặt các quyền cho người sở hữu, nhóm và những người khác (theo thứ tự). Ví dụ, giá trị 644 đặt các quyền đọc/ghi cho người sở hữu, và các quyền chỉ đọc cho nhóm và những người khác.

Bảng 3-5 Đặt các quyền file theo kiểu trực tiếp

Giá trị octal	Tập các quyền file	Mô tả các quyền
0	---	Không có quyền gì
1	--x	Chỉ có quyền thực thi
2	-w-	Chỉ có quyền ghi
3	-wx	Quyền ghi và thực thi
4	r--	Chỉ có quyền đọc
5	r-x	Quyền đọc và thực thi
6	rw-	Quyền đọc và ghi
7	rwx	Quyền đọc, ghi và thực thi

Bạn có thể đặt các quyền đặc biệt trên một file theo kiểu trực tiếp và ký hiệu. Theo kiểu trực tiếp, bạn đặt các quyền đặc biệt bằng cách bổ sung giá trị octal mới vào bên trái bộ ba quyền. Bảng 3-6 liệt kê các giá trị octal để đặt các quyền đặc biệt trên file.

Bảng 3-6 Đặt các quyền đặc biệt theo kiểu trực tiếp.

Giá trị octal	Tập các quyền đặc biệt
1	Sticky bit
2	setguid
4	setuid

Bảng 3-7 liệt kê các ký hiệu để đặt các quyền file theo kiểu ký hiệu. Các ký hiệu có thể đặc tả các quyền được đặt hay thay đổi, thao tác được thực hiện, hoặc các quyền được gán hay thay đổi.

Bảng 3-7 Đặt các quyền file theo kiểu ký hiệu

Ký hiệu	Chức năng	Mô tả
u	Who	Người dùng (người sở hữu)
g	Who	Nhóm
o	Who	Others
a	Who	Tất cả
=	Operation	Gán
+	Operation	Thêm
-	Operation	Xoá
r	Permission	Đọc
w	Permission	Ghi
x	Permission	Thực thi
l	Permission	Khoá bắt buộc, setuid bit là on, group execution bit là off.
s	Permission	setuid hoặc setgid bit là on
S	Permission	suid bit là on, user execution bit là off
t	Permission	Sticky bit là on, execution bit của others là on
T	Permission	Sticky bit là on, execution bit của others là off

Các chỉ định Who Operator Permission ở cột chức năng mô tả các ký hiệu thay đổi quyền trên file hoặc thư mục.

Who Đặc tả các quyền của ai được thay đổi

Operation	Đặc tả thao tác thực hiện
Permissions	Đặc tả các quyền nào được thay đổi.

3.4.1-Thay đổi quyền theo kiểu trực tiếp như thế nào

1. Nếu bạn không phải là file owner hay thư mục, thì trở thành superuser
Chỉ người sở hữu hiện thời hay superuser mới có thể dùng lệnh chmod thay đổi các quyền file trên file hay thư mục.

2. Dùng lệnh chmod thay đổi quyền theo kiểu trực tiếp

```
$ chmod nnn filename
```

nnn	Đặc tả các giá trị octal biểu diễn các quyền của file owner, file group, và others theo thứ tự. Xem bảng 3-5 để có danh sách các giá trị octal có hiệu lực.
filename	Đặc tả file hay thư mục

Ghi chú - Nếu bạn dùng lệnh chmod thay đổi các quyền của nhóm sở hữu file trên một file bởi các ACL entry, thì cả các quyền của nhóm sở hữu file và ẩn trong ACL đều được biến đổi thành các quyền mới. Suy ra rằng các quyền ẩn trong ACL mới có thể thay đổi các quyền có hiệu lực đối với những người dùng và nhóm bổ sung có các ACL entry trên file. Dùng lệnh getfacl để chắc chắn các quyền thích hợp được đặt cho tất cả ACL entry.

3. Xác nhận các quyền của file đã thay đổi

```
$ ls -l filename
```

Ví dụ - Thay đổi quyền theo kiểu trực tiếp

Ví dụ sau đây chỉ ra cách thay đổi các quyền của một thư mục public từ 744 (đọc /ghi /thực thi, chỉ đọc và chỉ đọc) thành 755 (đọc /ghi /thực thi, đọc /thực thi, và đọc/thực thi).

```
$ ls -ld public_dir
drwxr--r-- 1 ignatz staff 6023 Aug 5 12:06 public_dir
$ chmod 755 public_dir
$ ls -ld public_dir
drwxr-xr-x 1 ignatz staff 6023 Aug 5 12:06 public_dir
```

Ví dụ sau đây chỉ ra cách thay đổi các quyền của shell script thực thi từ đọc/ghi thành đọc/ghi/thực thi.

```
$ ls -l my_script
rw------- 1 ignatz staff 6023 Aug 5 12:06 my_script
$ chmod 700 my_script
$ ls -l my_script
-rwx------ 1 ignatz staff 6023 Aug 5 12:06 my_script
```

3.4.2-Thay đổi các quyền đặc biệt theo kiểu tuyệt đối như thế nào

1. Nếu bạn không phải là người sở hữu file hay thư mục, thì trở thành superuser
Chỉ người sở hữu hiện thời hay superuser mới có thể dùng lệnh chmod thay đổi các quyềnđặc biệt trên file hay thư mục.

2. Dùng lệnh chmod thay đổi các quyền đặc biệt theo kiểu trực tiếp

```
$ chmod nnnn filename
```

nnnn Đặc tả các giá trị octal thay đổi các quyền trên file hay thư mục. Giá trị octal thứ nhất về bên trái đặt các quyền đặc biệt trên file Xem bảng 3-6 để có danh sách các giá trị octal có hiệu lực đối với các quyền đặc biệt.

filename Là file hay thư mục

Ghi chú - Nếu bạn dùng lệnh chmod thay đổi các quyền của nhóm sở hữu file trên một file bởi các ACL entry, thì cả các quyền của nhóm sở hữu file và ẩn trong ACL đều được biến đổi thành các quyền mới. Suy ra rằng các quyền ẩn trong ACL mới có thể thay đổi các quyền có hiệu lực đối với những người dùng và nhóm bổ sung có các ACL entry trên file. Dùng lệnh getfacl(1) để chắc chắn các quyền thích hợp được đặt cho tất cả ACL entry.

3. Xác nhận các quyền của file đã thay đổi

```
$ ls -l filename
```

Ví dụ - Đặt các quyền đặc biệt theo kiểu trực tiếp

Ví dụ sau đây đặt quyền setuid trên file dbprog

```
$ chmod 4555 dbprog
$ ls -l dbprog
-r-sr-xr-x  1 db      staff   12095 May  6 09:29 dbprog
```

Ví dụ sau đây đặt quyền setgid trên file dbprog2

```
$ chmod 2551 dbprog2
$ ls -l dbprog2
-r-xr-s--x  1 db      staff   24576 May  6 09:30 dbprog2
```

Ví dụ sau đây đặt quyền sticky bit trên thư mục pubdir

```
$ chmod 1777 pubdir
```

3.4.3-Thay đổi quyền theo kiểu ký hiệu như thế nào

1. Nếu bạn không phải là người sở hữu file hay thư mục, thì trở thành superuser
Chỉ người sở hữu hiện thời hay superuser mới có thể dùng lệnh chmod thay đổi các quyền file trên file hay thư mục.

2. Dùng lệnh chmod thay đổi quyền theo kiểu ký hiệu

```
$ chmod who operation permission filename
```

who operation permission who đặc tả các quyền của ai bị thay đổi, operator đặc tả thao tác thi hành, và permission đặc tả những quyền nào bị thay đổi.
Xem bảng 3-7 về danh sách các ký hiệu có hiệu lực.

filename Là file hay thư mục

3. Xác nhận các quyền của file đã thay đổi

```
$ ls -l filename
```

Ví dụ - Thay đổi các quyền theo kiểu ký hiệu
Ví dụ sau đây loại bỏ quyền đọc của others

```
$ chmod o-r filea
```

Ví dụ sau đây thêm các quyền đọc và thực thi cho user, group và others

```
$ chmod a+rx fileb
```

Ví dụ sau đây gán các quyền đọc, ghi và thực thi cho group.

```
$ chmod g=rwx filec
```

3.5-Kiểm soát các quyền đặc biệt

Bạn nên giám sát hệ thống của bạn đối với bất kỳ việc sử dụng trái phép các quyền setuid và setgid nào để có được các quyền superuser. Một danh sách đáng ngờ là danh sách cấp quyền sở hữu chương trình nào đó cho user mà không phải là bin hay sys.

3.5.1-Tìm những file có quyền setuid như thế nào

1. Trở thành superuser
2. Dùng lệnh find tìm những file có đặt quyền setuid

```
# find directory -user root -perm -4000 -exec ls -ldb { }\; >/tmp/filename
```

find *directory*	Kiểm tra tất cả các đường dẫn đã thiết lập bắt đầu từ thư mục đặc biệt, có thể là root (/), sys, bin hay mail.
- user root	Chỉ hiển thi những file của root
-perm -4000	Chỉ hiển thị những file có quyền đặt là 4000
-exec ls -ldb	Hiển thị đầu ra của lệnh find dưới dạng ls -ldb
>/tmp/*filename*	Ghi kết quả vào file này

3. Hiển thị các kết quả trong /tmp/filename
Nếu bạn cần thông tin cơ bản về quyền setuid, hãy xem mục "Quyền setuid".

Ví dụ - Tìm những file có quyền setuid

```
# find  / -user root -perm -4000 -exec ls -ldb { }\ ; >/tmp/ckperm
# cat /tmp/ckperm
-r-sr-xr-x  1  root  bin  38836  Aug  10  16:16  /usr/bin/at
-r-sr-xr-x  1  root  bin  19812  Aug  10  16:16  /usr/bin/crontab
---s--x--x  1  root  sys  46040  Aug  10  15:18  /usr/bin/ct
-r-sr-xr-x  1  root  sys  12092  Aug  11  01:29  /usr/lib/mv_dir
-r-sr-xr-x  1  root  bin  33208  Aug  10  15:55  /usr/lib/lpadmin
-r-sr-xr-x  1  root  bin  38696  Aug  10  15:55  /usr/lib/lpsched
---s--x---  1  root  rar   45376  Aug  18  15:11  /usr/rar/bin/sh
-r-sr-xr-x  1  root  bin  12524  Aug  11  01:27  /usr/bin/df
-rwsr-xr-x  1 root  sys  21780  Aug  11  01:27  /usr/bin/newgrp
-r-sr-sr-x  1  root  sys  23000  Aug  11  01:27  /usr/bin/passwd
-r-sr-xr-x  1  root  sys  23824  Aug  11  01:27  /usr/bin/su
#
```

Một người dùng trái phép (rar) đã tạo một bản sao /usr/bin/sh riêng, và đã đặt các quyền như setuid cho root. Điều này có nghĩa là rar có thể thực hiện /usr/rar/bin/sh và trở thành người dùng hợp pháp. Nếu bạn muốn ghi đầu ra này để tham khảo về sau, hãy chuyển file ra khỏi thư mục /tmp.

3.6-Các stack khả thi và an toàn

Một số lỗi kỹ thuật về an toàn liên quan tới các stack khả thi mặc định khi các quyền của chúng được đặt là đọc, ghi và thực thi. Trong khi các stack có tập các quyền thực thi chịu ràng buộc của SPARC ABI và Intel ABI, thì hầu hết các chương trình có thể vận hành một cách đúng đắn mà không dùng các stack khả thi.

Biến noexec_user_stack đang bắt đầu khả dụng trong Solaris 2.6 loại bỏ cái cho phép bạn đặc tả sự sắp xếp stack có khả thi hay không. Theo ngầm đinh, biến là 0 đảm bảo tuân theo ABI. Nếu biến này được đặt khác 0, thì hệ thống sẽ đánh dấu stack của mọi tiến trình trong hệ thống là có thể đọc và ghi, nhưng không khả thi.

Một khi biến này được đặt, thì các chương trình cố gắng thực hiện mã trên stack của chúng sẽ được gửi tín hiệu SIGSEGV thường xuất hiện khi kết thúc chương trình với việc xổ lõi nhớ (core dump). Những chương trình như thế cũng đưa ra lời cảnh báo gồm tên của chương trình vi phạm, process ID, và UID thực của user chạy chương trình. Ví dụ:

```
a.out[347] attempt  to execute code on stack by uid 555
```

Thông báo này được ghi lại nhờ trình syslogd(1M) khi tiện ích syslog kern được đặt ở mức notice. Theo ngầm định, bản ghi này được đặt trong file syslog.conf(4), nghĩa là thông báo được gửi tới cả console và file /var/adm/messages.

Thông báo này thuận lợi cho việc theo dõi các vấn đề an toàn có thể xảy ra, cũng như xác định các chương trình hợp lệ dựa vào những stack khả thi đã bị hoạt động sai lạc do đặt biến này. Nếu người giám quản không muốn có bất kỳ thông báo nào

được ghi lại, thì có thể đặt biến noexec_user_stack_log bằng 0 để làm mất hiệu lực của nó trong file /etc/system, mặc dù tín hiệu SIGSEGV có thể tiếp tục dẫn chương trình đang thực thi tới core dump.

Bạn có thể dùng mprotect(2) nếu bạn muốn các chương trình đánh dấu chúng một cách rõ ràng khi stack khả thi.

Vì những hạn chế về phần cứng, nên khả năng bắt và thông báo các vấn đề về stack khả thi chỉ khả dụng trên nền sun4m, sun4d, và sun4u.

3.6.1-Làm thế nào để các chương trình không dùng stack khả thi

1. Trở thành superuser

2. Soạn thảo file /etc/system và bổ sung dòng sau.

```
set  noexec_user_stack=1
```

3. Khởi động lại hệ thống

```
# init 6
```

3.6.2-Làm thế nào để không ghi lại thông báo về stack khả thi

1. Trở thành superuser
2. Soạn thảo file /etc/system và bổ sung dòng sau.

```
set  noexec_user_stack_log=0
```

3. Khởi động lại hệ thống

```
# init 6
```

3.7-Sử dụng các danh sách điều khiển truy nhập (ACLs)

Bảo vệ file UNIX truyền thống đảm bảo các quyền đọc, ghi và thực thi cho ba lớp người dùng: người sở hữu file (file owner), nhóm file (file group) và những người dùng khác (others). ACL bảo đảm an toàn file tốt hơn bằng việc cho phép bạn định nghĩa các quyền file cho file owner, file group, other, những người dùng và nhóm người dùng đặc biệt, và các quyền mặc định cho mỗi lớp này.

Ví dụ, nếu bạn muốn mọi người trong một nhóm có thể đọc file, đơn giản là bạn sẽ đặt cho nhóm quyền đọc file đó. Bây giờ, giả sử bạn chỉ muốn một người trong nhóm có thể ghi file đó. UNIX chuẩn không cung cấp mức an toàn file đó. Tuy nhiên, tình thế này được hoàn thiện đối với ACLs.

Các ACL entry là cách định nghĩa ACL trên một file, và chúng được đặt thông qua lệnh setfacl(1). Đầu vào ACL gồm các trường sau được viết cách nhau dấu hai chấm:

```
entry_type:[uid\gid]:perms
```

entry_type	Kiểu ACL entry trên cái đặt các quyền file. Ví dụ, entry_type có thể là user (file owner) hay mặt nạ (ACL mask).
uid	Tên hay số định danh người dùng
gid	Tên hay số định danh nhóm
perms	Biểu diễn các quyền được đặt trên entry_type. perms có thể được định rõ bằng các ký tự ký hiệu rwx hay một số (các số tương ứng với các quyền sử dụng với lệnh chmod).

Ví dụ sau đây cho thấy ACL entry đặt các quyền đọc/ghi cho user nathan

```
user : nathan : rw-
```

Chú ý- Các thuộc tính của hệ thống file UFS, chẳng hạn ACLs, được đáp ứng chỉ trong các hệ thống UFS. Điều đó có nghĩa là nếu bạn phục hồi hay sao chép các file với các ACL entry vào thư mục /tmp thường gắn với hệ thống file TMPFS, thì các ACL entry sẽ bị mất. Dùng thư mục /var/tmp để lưu trữ tạm thời các file UFS.

3.7.1-Các ACL entry của đối với các file

Bảng 3-8 liệt kê các ACL entry có hiệu lực. Ba ACL entry đầu tiên đảm bảo sự bảo vệ file UNIX cơ bản.

Bảng 3-8 Các ACL entry của các file

Đầu vào ACL	**Mô tả**
u[ser]::perms	Các quyền sở hữu file
g[roup]::perms	Các quyền của file group
o[ther]::perms	Các quyền đối với những người dùng không phải file owner hay thành viên của file group
m[ask]::perms	ACL mask. Đầu vào mặt nạ định rõ các quyền tối đa cho phép những người dùng (other hơn là owner) và các nhóm. Mặt nạ là cách thay đổi nhanh các quyền của tất cả những người dùng và nhóm.
u[ser]:uid:perms	Các quyền của người dùng cụ thể. Đối với uid, bạn có thể đặc tả hoặc tên người dùng hoặc UID kiểu số.
g[roup]:gid:perms	Các quyền của một nhóm cụ thể. Đối với gid, bạn có thể đặc tả tên nhóm hay GID kiểu số.

3.7.2-Các ACL entry của các thư mục

Để có các ACL entry mô tả trong bảng 3-8, bạn có thể đặt các ACL entry mặc định trên thư mục. Các file hay thư mục tạo ra trong một thư mục có các ACL entry mặc định sẽ có cùng các đầu vào ACL như các ACL entry mặc định. Bảng 3-9 liệt kê các đầu vào mặc định cho các thư mục.

Khi lần đầu tiên bạn đặt các ACL entry mặc định cho những người dùng và nhóm cụ thể, bạn cũng phải đặt các ACL entry mặc định cho file user, file group, others, và

ACL mask (những điều này được quy định và là bốn ACL entry mặc định đầu tiên trong bảng 3-9).

Đầu vào ACL mặc định	Mô tả
d[efault]:u[ser]::perms	Các quyền mặc định của file owner.
d[efault]:g[roup]::perms	Các quyền mặc định của file group.
d[efault]:o[ther]::perms	Các quyền mặc định đối với những người dùng không phải là file owner hay các thành viên của file group.
d[efault]:m[ask]:perms	ACL mask mặc định.
d[efault]:u[ser]:uid:perms	Các quyền mặc định của người dùng cụ thể. Đối với uid, bạn có thể đặc tả hoặc tên người dùng, hoặc UID bằng số.
d[efault]:g[roup]:gid:perms	Các quyền mặc định của nhóm cụ thể. Đối với gid, bạn có thể đặc tả hoặc tên nhóm hoặc GID bằng số.

3.7.3-Cài đặt ACL trên một file như thế nào

1. Dùng lệnh setfacl để cài đặt ACL trên một file

```
$ setfacl -s user::perms, group::perms, mask:perms,acl_entry_list
filename ...
```

-s	Đặt ACL trên file. Nếu file sẵn có một ACL, thì nó bị thay thế. Lựa chọn này đòi hỏi ít nhất các đầu vào file owner, file group và other.
user::perms	Đặc tả các quyền file owner.
group::perms	Đặc tả các quyền file group.
other:perms	Đặc tả các quyền đối với những người dùng không phải file owner, hay thành viên của file group.
mask:perms	Đặc tả các quyền đối với ACL mask. Mặt nạ chỉ rõ các quyền tối đa cho phép đối với những người dùng (other hơn là owner) và nhóm.
acl_entry_list	Là danh sách một hay nhiều ACL entry cài đặt đối với những người dùng và nhóm cụ thể trên file hay thư mục. Bạn cũng có thể đặt các ACL entry mặc định trên thư mục. Bảng 3-8 và bảng 3-9 trình bày các ACL entry hợp lệ.
filename	File hay thư mục mà trên đó cài đặt ACL.

2. Xác nhận ACL đã cài đặt trên file, xem "Cách kiểm tra một file có ACL" ở phần sau. Dùng lệnh getfacl để thẩm tra các ACL entry đã cài đặt trên file.

```
$ getfacl filename
```

Ghi chú - Nếu ACL đã tồn tại trên file, tuỳ chọn -s sẽ thay thế toàn bộ ACL bằng ACL mới.

Các ví dụ - Cài đặt ACL trên file

Ví dụ sau đây đặt các quyền file owner là đọc/ghi, các quyền file group là chỉ đọc, và các quyền other là none trên file ch1.doc. Ngoài ra, người dùng george được gán cho các quyền đọc/ghi trên file, và các quyền ACL mask được đặt là đọc/ghi, nghĩa là không có người dùng hay nhóm nào có các quyền thực thi.

```
$ setfacl -s user::rw-, group::r--, other:---, mask:rw-, user:george:rw- ch1.doc
$ ls -l
-rw-r-----+ 1 nathan sysadmin 34816 Nov 11 14:16 ch1.doc
-rw-r--r--  1 nathan sysadmin 20167 Nov 11 14:16 ch2.doc
-rw-r--r--  1 nathan sysadmin  8192 Nov 11 14:16 notes
$ getfacl ch1.doc
# file: ch1.doc
# owner: nathan
# group:sysadmin
user:: rw-
user:george:rw-      #effective: rw-
group:: r--          #effective: r--
mask: rw-
other: ---
```

Ví dụ sau đây cài đặt các quyền file owner là đọc/ghi/thực thi, các quyền file group là chỉ đọc, các quyền other là none, và các quyền ACL mask là chỉ đọc trên file ch2.doc. Ngoài ra, người dùng george được gán cho các quyền đọc/ghi; tuy nhiên, theo ACL mask, các quyền có hiệu lực đối với george là chỉ đọc.

```
$ setfacl -s u::7, group::4, o:0, m:4, user:george:7 ch2.doc
$ getfacl ch2.doc
# file: ch2.doc
# owner: nathan
# group:sysadmin
user:: rwx
user:george:rwx      #effective: r--
group:: r--          #effective: r--
mask: r--
other: ---
```

3.7.4-Cách sao chép ACL

Sao chép ACL của một file cho một file khác bằng cách đổi hướng đầu ra của getfacl

```
$ getfacl filename1 | setfacl --f - filename2
```

file1 Đặc tả file, nơi sao chép ACL
file2 Đặc tả file, nơi đặt bản sao ACL.

Ví dụ - Sao chép ACL

Ví dụ sau đây sao chép ACL trên ch1.doc sang ch3.doc.

```
$ getfacl ch2.doc | setfacl --f - ch3.doc
```

3.7.5-Cách kiểm tra một file có ACL

Dùng lệnh ls kiểm tra một file có ACL

```
$ ls -l filename
```

file name Đặc tả file hay thư mục

Dấu '+' bên phải trường kiểu chỉ file có ACL.

Ghi chú - File được xem là có ACL "tầm thường" và dấu '+' sẽ không hiển thị, trừ khi bạn đã thêm các ACL entry cho những người dùng hay nhóm bổ sung trên một file.

Ví dụ - Kiểm tra một file có ACL

Ví dụ sau đây chỉ ra rằng ch1.doc có một ACL, vì danh sách có dấu '+' bên phải trường kiểu.

```
$ ls -l ch1.doc
-rwxr-----+ 1 nathan sysadmin 167 Nov 11 11:13 ch1.doc
```

3.7.6-Cách thay đổi các ACL entry trên một file

1. Dùng lệnh setfacl để thay đổi các ACL entry

```
$ setfacl -m acl_entry_list filename1 [filename2 ...]
```

-m	Thay đổi ACL entry đang tồn tại
acl_entry_list	Đặc tả danh sách một hay nhiều ACL entry để thay đổi trên file hay thư mục. Bạn cung có thể thay đổi các ACL entry mặc định trên một thư mục. Bảng 3-8 và bảng 3-9 chỉ ra các ACL entry hợp lệ.
filename ...	Đặc tả file hoặc thư mục.

2. Dùng lệnh getfacl để xác nhận các ACL entry đã bị thay đổi trên file

```
$ getfacl filename
```

Các ví dụ - Thay đổi các ACL entry trên file

Ví dụ sau đây thay đổi các quyền đối với người dùng george thành đọc/ghi.

```
$ setfacl -m user:george:6 ch3.doc
$ getfacl ch3.doc
# file: ch3.doc
# owner: nathan
# group: staff
user:: rw-
```

```
user::george:rw-      #effective: r--
group:: r-            #effective: r--
mask: r--
other: r-
```

Ví dụ sau đây thay đổi các quyền mặc định đối với nhóm staff thành đọc và các quyền ACL mask mặc định thành đọc/ghi trên thư mục book.

```
$ setfacl -m default:group:staff:4, default:mask:6 book
```

3.7.7-Cách xoá các ACL entry khỏi file

1. Dùng lệnh setfacl để xoá các ACL entry khỏi file

```
$ setfacl -d acl_entry_list filename ...
```

-d	Xoá các ACL entry cụ thể.
acl_entry_list	Đặc tả danh sách các ACL entry (không mô tả các quyền) cần xoá khỏi file hay thư mục. Bạncó thể chỉ xoá các ACL entry và ACL entry mặc định đối với những người dùng và nhóm cụ thể. Bảng 3-8 và bảng 3-9 cho thấy các ACL entry hợp lệ.
filename ...	Đặc tả file hoặc thư mục.

Bên cạnh đó, bạn có thể dùng lệnh setfacl -s để xoá tất cả các ACL entry trên một file và thay chúng bởi các ACL entry mới được mô tả.

2. Dùng lệnh getfacl để xác nhận các ACL entry đã bị xoá khỏi file

```
$ getfacl filename
```

Ví dụ - Xoá các ACL entry trên một file

Ví dụ sau đây xoá người dùng george khỏi file ch4.doc

```
$ setfacl -d user:george ch4.doc
```

3.7.8-Làm thế nào để hiển thị các ACL entry của một file

Dùng lệnh getfacl để hiển thị các ACL entry của một file

```
$ getfacl [-a] [-d] filename ...
```

-a	Hiển thị tên file, file owner, file group, và các ACL entry của file hoặc thư mục cụ thể.
-d	Hiển thị tên file, file owner, file group, và các ACL entry mặc định của thư mục cụ thể.
filename ...	Đặc tả file hoặc thư mục

Nếu bạn đặc tả nhiều tên file trên dòng lệnh, thì các ACL entry được tách nhau bởi một dòng trống.

Các ví dụ - Hiển thị các ACL entry của file

Ví dụ sau đây cho thấy các ACL entry của file ch1.doc. Ghi chú #effective: bên cạnh đầu vào user và group chỉ rõ các quyền sau khi bị thay đổi bởi ACL mask là gì.

```
$ getfacl ch1.doc
# file: ch1.doc
# owner: nathan
# group: sysadmin
user:: rw-
user::george:r--        #effective: r--
group:: rw-             #effective: rw-
mask: rw-
other: ---
```

Ví dụ sau đây cho thấy các ACL entry của thư mục book.

```
$ getfacl -d book

# file: book
# owner: nathan
# group: sysadmin
user:: rwx
user:george:r-x         #effective: r-x
group:: rwx             #effective: rwx
mask: rwx
other: ---
default:user:: rw-
default:user:george:r--
default:group:: rw-
default:mask: rw-
default:other: ---
```

CHƯƠNG IV-CÁC TÁC VỤ AN TOÀN CÁC HỆ THỐNG

Chương này mô tả các thủ tục an toàn hệ thống. Sau đây là danh sách những chỉ dẫn từng bước trong chương này.

- "Cách hiển thị trạng thái đăng nhập của người dùng" ở mục 4.1
- " Cách hiển thị những người dùng thiếu mật khẩu" ở mục 4.2
- " Cách vô hiệu hoá tạm thời đăng nhập của người dùng" ở mục 4.3
- "Cách lưu lại những cuộc đăng nhập thất bại" ở mục 4.4
- "Cách tạo một mật khẩu quay số" ở mục 4.5
- "Cách vô hiệu hoá tạm thời các cuộc đăng nhập bằng quay số" ở mục 4.6
- "Cách hạn chế Superuser (root) đăng nhập tới thiết bị điều khiển" ở mục 4.7
- "Cách giám sát những người sử dụng lệnh su" ở mục 4.8
- "Cách hiển thị những lần thử truy nhập tới thiệt bị điều khiển của Superuser (root)" ở mục 4.9

4.1-Cách hiển thị trạng thái đăng nhập của người dùng

1. Trở thành superuser
2. Dùng lệnh logins để hiển thị trạng thái đăng nhập của người dùng.

```
# logins -x -l username
```

-x	Hiển thị tập thông tin trạng thái đăng nhập mở rộng.
-l *username*	Hiển thị trạng thái đăng nhập của người dùng cụ thể. *uername* là tên đăng nhập của người dùng. Nhiều tên đăng nhập phải được mô tả bằng danh sách viết cách nhau dấu phẩy.

Lệnh logins(1M) dùng file /etc/passwd cục bộ và các cơ sở dữ liệu mật khẩu NIS hoặc NIS+ để có được trạng thái đăng nhập của người dùng.

Ví dụ - Hiển thị trạng thái đăng nhập của người dùng
Ví dụ sau đây hiển trạng thái đăng nhập của người dùng rimmer

```
# logins -x -l rimmer
rimer          500    staff          10   Arnold J. Rimmer
                      /export/home/rimmer
                      /bin/sh
                      PS 010170 10 7 -1
```

Trong ví dụ này,

rimmer	Tên đăng nhập của người dùng.
500	UID (ID người dùng).
staff	Nhóm ban đầu của người dùng.
10	GID (ID nhóm).
Arnold J. Rimmer	Lời chú giải

/export/home/rimmer	Thư mục gốc của người dùng.
/bin/sh	Cấu trúc đăng nhập
PS 010170 10 7 -1	Thông tin thời hạn mật khẩu: • Ngày cuối cùng mật khẩu bị thay đổi • Số ngày yêu cầu giữa các lần thay đổi • Số ngày cho phép trước khi có đòi hỏi thay đổi • Khoảng thời gian cảnh báo

4.2-Cách hiển thị những người dùng không có mật khẩu

Bạn nên chắc chắn rằng tất cả người dùng có một mật khẩu hợp lệ.

1. Trở thành superuser.
2. Dùng lệnh logins để hiển thị những người dùng không có mật khẩu.

```
# logins -p
```

-p Hiển thị danh sách những người dùng không có mật khẩu

Lệnh logins dùng file /etc/passwd cục bộ và các cơ sở dữ liệu mật khẩu NIS hoặc NIS+ để có được trạng thái đăng nhập của người dùng.

Ví dụ - Hiển thị những người dùng không có mật khẩu

Ví dụ sau đây hiển thị pmorph người dùng không có mật khẩu.

```
# logins -p
pmorph      501    other          1       Polly Morph
```

4.3-Vô hiệu hoá tạm thời các cuộc đăng nhập của người dùng

Bạn có thể vô hiệu hoá tạm thời các cuộc đăng nhập của người dùng bằng cách:

- Tạo file /etc/nologin.
- Làm cho hệ thống chạy mức 0 (mức người dùng đơn lẻ). Xem "Đóng hệ thống (các tác vụ)" trong *System Adminitration Guide, Volume I* để có thông tin về cách làm cho hệ thống về mô hình người dùng đơn lẻ.

Tạo file /etc/nologin

Tạo file này để không cho phép người dùng đăng nhập và thông báo cho người dùng về việc hệ thống sẽ không khả dụng trong khoảng thời gian quá hạn để đóng hệ thống hoặc bảo dưỡng định kỳ.

Nếu người dùng đăng nhập tới hệ thống mà ở đó có file này tồn tại, thì nội dung của file nologins(4) được hiển thị, và cuộc đăng nhập của người dùng bị ngắt. Những cuộc đăng nhập của superuser không bị ảnh hưởng.

Cách vô hiệu hoá tạm thời các cuộc đăng nhập của người dùng

1. Trở thành superuser.
2. Dùng trình soạn thảo tạo file /etc/nologin.

```
# vi /etc/nologin
```

3. Kèm theo thông báo về tính khả dụng của hệ thống.
4. Đóng và ghi file.

Ví dụ - Vô hiệu hoá các cuộc đăng nhập của người dùng
Ví dụ sau đây trình bày cách làm thế nào để thông báo cho người dùng về việc hệ thống không khả dụng.

```
# vi /etc/nologin
(Thêm thông báo về hệ thống ở đây)

# cat /etc/nologin
***No logins permitted.***

***The system will be unavailable until 12 noon.***
```

4.4-Lưu lại các cuộc đăng nhập không thành công

Bạn có thể lưu lại các cuộc đăng nhập thất bại bằng cách tạo file /var/adm/loginlog mà chỉ root mới có quyền đọc và ghi. Sau khi bạn tạo file loginlog, tất cả các hành động đăng nhập lỗi đều được tự động ghi vào file này sau năm cuộc thất bại. Xem "Cách lưu lại các cuộc đăng nhập không thành công" ngay sau đây để có những chỉ dẫn chi tiết.

File loginlog chứa một đầu vào cho mỗi lần lỗi. Mỗi đầu vào chứa tên đăng nhập của người dùng, thiết bị hiển thị kiểu điện báo (tty), và thời gian của cuộc đăng nhập lỗi. Nếu một người tiến hành ít hơn năm cuộc không thành công, thì không có cuộc nào được ghi lại.

File loginlog có thể tăng một cách nhanh chóng. Để sử dụng thông tin trong file này và ngăn chặn file trở nên quá lớn, thỉnh thoảng bạn phải kiểm tra và xoá nội dung của nó. Khi file này lưu nhiều hành động, nó có thể cho thấy một cuộc đột phá vào hệ thống máy tính. Để biết thêm thông tin về file này, xem **loginlog**(4).

Cách lưu lại các cuộc đăng nhập không thành công

1. Trở thành superuser
2. Tạo file loginlog trong thư mục /var/adm

```
# touch /var/adm/loginlog
```

3. Đặt các quyền đọc và ghi cho root trên file loginlog

```
# chmod 600 /var/adm/loginlog
```

4. Thay đổi thành viên nhóm đối với sys trên file loginlog

```
# chgrp sys /var/adm/loginlog
```

5. Thử đăng nhập vào hệ thống năm lần với mật khẩu sai để chắc chắn rằng log làm việc sau khi file loginlog đã được tạo. Sau đó hiển thị file /var/adm/loginlog.

```
# more /var/adm/loginlog
pmorph: /dev/pts/4:Mon Jun 8 11: 08: 27 1998
pmorph: /dev/pts/4:Mon Jun 8 11: 08: 49 1998
pmorph: /dev/pts/4:Mon Jun 8 11: 09: 03 1998
pmorph: /dev/pts/4:Mon Jun 8 11: 09: 22 1998
pmorph: /dev/pts/4:Mon Jun 8 11: 09: 36 1998
#
```

4.5-Bảo vệ mật khẩu bằng cách dùng các mật khẩu quay số

Bạn có thể thêm một tầng an toàn vào cơ chế mật khẩu của bạn bằng việc yêu cầu mật khẩu quay số (dial-up password) của người dùng truy nhập hệ thống thông qua modem hay cổng quay số. Một mật khẩu dial-up là một mật khẩu bổ sung mà người dùng phải đưa vào trước khi được phép truy nhập tới hệ thống.

Chỉ superuser mới có thể tạo hay thay đổi mật khẩu dial-up. Để đảm bảo toàn vẹn hệ thống, mật khẩu nên được thay đổi mỗi tháng một lần. Việc sử dụng cơ chế này hiệu quả nhất là đòi hỏi mật khẩu dial-up để có được quyền truy nhập tới hệ thống gateway.

Liên quan tới việc tạo mật khẩu dial-up là hai file: /etc/dialups và /etc/d_passwd. File thứ nhất chứa danh sách các cổng yêu cầu mật khẩu dial-up, và file thứ hai chứa danh sách các chương trình khung yêu cầu mật khẩu mã hoá với tư cách mật khẩu dial-up bổ sung.

File **dialups(4)** là một danh sách các thiết bị cuối, ví dụ:

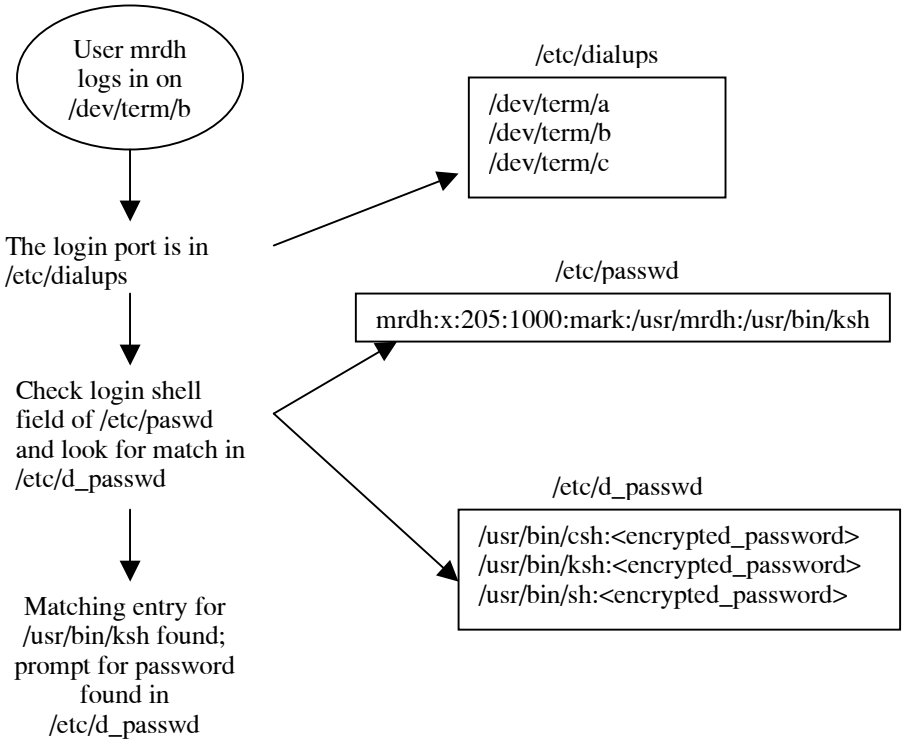
```
/dev/term/a
dev/term/b
```

File **d_passwd(4)** có hai trường. Trường thứ nhất là cấu trúc (shell) đăng nhập sẽ đòi hỏi mật khẩu, và trường thứ hai là mật khẩu mã hoá. Các file /etc/dialups và /etc/d_passwd làm việc như sau:

Khi người dùng thử đăng nhập bất kỳ cổng nào được liệt kê trong /etc/dialups, chương trình đăng nhập xem xét mục đăng nhập của người dùng lưu trong /etc/passwd, và so sánh shell đăng nhập với các mục trong /etc/d_passwd. Các mục này sẽ xác định xem người dùng có phải đưa vào mật khẩu dial-up hay không.

```
/usr/lib/uucp/uucico:encrypted_password:
/usr/bin/csh:encrypted_password:
/usr/bin/ksh:encrypted_password:
/usr/bin/sh:encrypted_password:
```

Dãy các mật khẩu dial-up cơ bản được trình bày ở hình 4-1



Hình 4-1 Dãy mật khẩu cơ bản của dial-up

File /etc/d_passwd

Vì hầu hết người dùng sẽ quản lý một shell khi họ đăng nhập, nên tất cả các chương trình khung nên đưa vào /etc/d_passwd. Những chương trình như thế bao gồm uucico, sh, ksh và csh. Nếu một số người dùng thực hiện một cái gì đó không phải shell đăng nhập của họ, thì cũng có nghĩa là không phải shell đăng nhập trong file.

Nếu chương trình đăng nhập của người dùng (cụ thể trong /etc/passwd) không tìm thấy trong /etc/d_passwd, hay nếu trường shell đăng nhập trong /etc/passwd là null, thì mật khẩu của /usr/bin/sh được sử dụng.

- Nếu shell đăng nhập của người dùng trong /etc/passwd tương ứng với một mục trong /etc/passwd, thì người dùng phải đưa vào một mật khẩu dial-up.
- Nếu shell đăng nhập của người dùng không tìm thấy trong /etc/d_passwd, thì người dùng phải đưa vào mật khẩu mặc định. Mật khẩu mặc định là một mục của /usr/bin/sh.
- Nếu trường shell đăng nhập trong /etc/passwd là rỗng, thì người dùng phải đưa vào mật khẩu mặc định (một mục của /usr/bin/sh).
- Nếu /etc/d_passwd không có mục nào của /usr/bin/sh, thì những người dùng mà trường shell đăng nhập của họ trong /etc/passwd là rỗng hay không tương ứng với mục nào trong /etc/d_paswd sẽ không được nhắc đưa vào mật khẩu dial-up.
- Những đăng nhập bằng dial-up bị vô hiệu hoá nếu /etc/d_paswd chỉ có mục sau: /usr/bin/sh:*:

Cách tạo mật khẩu dial-up

Chú ý - Lần đầu tiên khi bạn thiết lập mật khẩu dial-up, bạn chắc chắn vẫn duy trì việc ghi lại trên ít nhất một thiết bị cuối trong khi kiểm tra mật khẩu trên một thiết bị cuối khác. Nếu bạn gây lỗi khi cài đặt mật khẩu phụ và kết thúc kiểm tra mật khẩu mới, thì bạn có thể không có khả năng khởi hoạt trở lại. Nếu bạn vẫn ghi lại trên một thiết bị cuối khác, bạn có thể quay lại và sửa chữa lỗi của bạn.

1. Trở thành superuser.
2. Tạo một file /etc/dialups lưu danh sách các thiết bị cuối, bao gồm tất cả các cổng sẽ đòi hỏi bảo vệ bằng mật khẩu dial-up.
   File /etc/dialups nên được thấy như sau:

```
/dev/term/a
/dev/term/b
/dev/term/c
```

3. Tạo file /etc/d_passwd lưu các chương trình đăng nhập sẽ đòi hỏi mật khẩu dial-up và mật khẩu dial-up mã hoá.
   Bao gồm các chương trình đăng nhập mà người dùng có thể chạy khi đăng nhập, ví dụ uucico, sh, ksh và csh. File /etc/d_passwd nên được thấy như sau:

```
 /usr/lib/uucp/uucico:encrypted_password:
 /usr/bin/csh:encrypted_password:
/usr/bin/ksh:encrypted_password:
/usr/bin/sh:encrypted_password:
```

4. Đặt quyền sở hữu cho root trên hai file này.

```
# chown root /etc/dialups /etc/d_passwd
```

5. Đặt quyền sở hữu nhóm cho root trên hai file này.

```
# chogrp root /etc/dialups /etc/d_passwd
```

6. Đặt các quyền đọc và ghi cho root trên hai file này
   # chmod 600 /etc/dialups /etc/d_passwd

7. Tạo các mật khẩu mã hoá
   a. Tạo người dùng tạm thời.

```
# useradd user-name
```

   b. Tạo mật khẩu cho người dùng tạm thời.

```
# passwd user-name
```

   c. Gán mật khẩu mã hoá.

```
# grep user-name /etc/shadow > user-name.temp
```

   d. Soạn thảo file user-name.tmp.
   Xoá tất cả các trường, ngoại trừ mật khẩu mã hoá (trường thứ hai).
   Ví dụ, trong dòng sau mật khẩu mã hoá là U9gp9SyA/JlSk

```
temp:U9gp9SyA/JlSk:7967:::::7988
```

e. Xoá người dùng tạm thời

```
# userdel user-name
```

8. Sao chép mật khẩu mã hoá từ file user-name.tmp vào file /etc/d_passwd

Bạn có thể tạo một mật khẩu khác cho mỗi shell đăng nhập, hay sử dụng cùng một mật khẩu.

4.6-Cách vô hiệu hoá tạm thời các cuộc đăng nhập dial-up

1. Trở thành superuser
2. Đặt chính mục sau vào file /etc/d_passwd:

```
/usr/bin/sh:*:
```

4.7-Hạn chế truy nhập Superuser (root) trên thiệt bị điều khiển

Hệ điều hành dùng tài khoản của superuser để thự hiện các chức năng cơ bản và có điều khiển phạm vi rộng trên toàn bộ hệ điều hành. Nó truy nhập và có thể thực hiện các chương trình hệ thống thiết yếu. Vì vậy, hầu như không có những hạn chế đối với bất kỳ chương trình nào mà superuser thực hiện.

Bạn có thể bảo vệ tài khoản của superuser bằng cách hạn chế truy nhập tới thiết bị cụ thể thông qua file /etc/default/login. Ví dụ, nếu truy nhập của superuser bị hạn chế ở thiệt bị điều khiển, thì bạn có thể đăng nhập hệ thống với tư cách superuser (superuser login) chỉ từ thiệt bị điều khiển. Khi bất kỳ ai đăng nhập hệ thống từ xa nhằm thực hiện chức năng giám quản, thì trước hết họ phải đăng nhập với tư cách người dùng của họ và sau đó dùng lệnh **su**(1M) để trở thành superuser. Xem "Cách hạn chế Superuser (root) Login tới thiệt bị điều khiển" để có những chỉ dẫn chi tiết.

Ghi chú - Hạn chế superuser login tới thiệt bị điều khiển được thiết lập mặc định khi bạn cài đặt hệ thống.

Cách hạn chế Superuser (root) login tới thiệt bị điều khiển

1. Trở thành superuser.
2. Soạn thảo file /etc/default/login

3. Không đưa ra dòng sau

```
CONSOLE=/dev/console
```

Bất kỳ người dùng nào thử đăng nhập hệ thống này từ xa trước hết phải đăng nhập với tư cách người dùng, và sau đó dùng lệnh su để trở thành superuser.

4. Thử đăng nhập từ xa tới hệ thống này với tư cách superuser, và thẩm tra thao tác lỗi.

4.8-Giám sát người dùng lệnh su

Bạn có thể bắt đầu kiểm soát các cố gắng su thông qua file /etc/default/su. Qua file này, bạn có thể làm cho file /var/adm/sulog có khả năng kiểm soát mỗi lần dùng

lệnh su để thay đổi thành người dùng khác. Xem "Cách giám sát người dùng lệnh su" để có những chỉ dẫn từng bước.

File sulog liệt kê tất cả người dùng lệnh su, không chỉ những người sử dụng để chuyển từ người dùng thành superuser. Các mục đưa ra ngày và thời gian mà lệnh được thực hiện, nó thành công hay không (+ hoặc -), cổng mà từ đó lệnh phát ra, và cuối cùng là tên người dùng và định danh đã chuyển đổi.

Thông qua file /etc/default/su, bạn cũng có thể thiết lập hệ thống hiển thị trên thiệt bị điều khiển mỗi lần dùng lệnh su để có được quyền truy nhập của superuser từ hệ thống từ xa. Đây là cách tốt để phát hiện kịp thời ai đó cố gắng có được quyền truy nhập của superuser trên hệ thống mà hiện tại bạn đang làm việc. Xem "Cách hiển thị các cuộc truy nhập của superuser tới thiệt bị điều khiển" để có những chỉ dẫn chi tiết.

Cách kiểm soát người dùng lệnh su

1. Trở thành superuser.
2. Soạn thảo file /etc/default/su.
3. Bỏ chú giải ở dòng sau

```
SULOG=/var/adm/sulog
```

4. Sau khi biến đổi file /etc/default/su, dùng lệnh su vài lần và hiển thị file /var/adm/sulog. Bạn nên xem danh sách mỗi lần bạn dùng lệnh su.

```
#  more  /var/adm/sulog
SU  12/20  16:26  +  pts/0  nathan-root
SU  12/21  10:59  +  pts/0  nathan-root
SU  01/12  11:11  +  pts/0 root-joebob
SU  01/12  14:56  +  pts/0  pmorph-root
SU  01/12  14:57  +  pts/0 pmorph-root
```

4.9-Cách hiển thị những lần truy nhập của superuser (root) tới thiết bị điều khiển

1. Trở thành superuser.
2. Soạn thảo file /etc/default/su.
3. Không đưa ra dòng sau

```
CONSOLE=/dev/console
```

4. Dùng lệnh su để trở thành root, và kiểm lại thông báo được in ra trên thiết bị điều khiển hệ thống.

CHƯƠNG V-SỬ DỤNG CÁC DỊCH VỤ XÁC THỰC

Phần đầu tiên của chương này cung cấp thông tin về các cơ chế xác thực mà RPC an toàn có thể sử dụng. Cả hai kiểu xác thực Diffie-Hellman và Kerberos version 4 đều được hỗ trợ. Phần thứ hai gồm kiến thức về module xác thực tải thêm được (PAM). PAM cung cấp cách thức để "tải vào" các dịch vụ xác thực và đảm bảo trợ giúp nhiều dịch vụ xác thực.

Đây là danh sách những chỉ dẫn từng bước trong chương này.

- "Cách thiết lập các nhãn quyền NIS+ đối với xác thực Diffie-Hellman" ở mục 5.2.2.
- "Cách thiết lập các nhãn quyền NIS đối với xác thực Diffie-Hellman" ở mục 5.2.3.
- "Cách chia xẻ và gán file bởi xác thực Diffie-Hellman" ở mục 5.2.4
- "Cách chia xẻ và gán file bởi xác thực Kerberos" ở mục 5.3.1
- "Cách tiếp nhận nhãn Kerberos của Superuser trên Client" ở mục 5.3.2
- "Cách đăng nhập dịch vụ Kerberos" ở mục 5.3.3
- "Cách truy nhập một thư mục với xác thực Kerberos" ở mục 5.3.5
- "Cách bổ sung module PAM" ở mục 5.6.2
- "Cách ngăn chặn truy nhập trái phép từ các hệ thống từ xa bằng PAM" ở mục 5.6.3
- "Cách khởi hoạt trình báo lỗi PAM" ở mục 5.6.4

5.1-Tổng quan về RPC an toàn

RPC an toàn là cách thức xác thực xác nhận cả máy chủ và người dùng đưa ra yêu cầu. RPC an toàn dùng xác thực hoặc Diffie-Hellman hoặc Kerberos. Cả hai cơ chế xác thực này dùng mã DES. Các ứng dụng dùng RPC an toàn gồm NFS và dịch vụ tên NIS+.

5.1.1-Các dịch vụ NFS và RPC an toàn

Phần mềm NFS cho phép một vài máy chủ chia xẻ file trên mạng. Theo hệ thống NFS, một server chứa dữ liệu và các tài nguyên của một vài client. Các client truy nhập tới hệ thống file mà server đưa ra cho client. Những người dùng đăng nhập vào máy client có thể truy nhập các hệ thống file bằng cách kéo từ server ra. Đối với người dùng trên máy client, nó xuất hiện như là các file được đặt tại client. Một trong những cách dùng môi trường NFS phổ biến nhất là cho cài đặt các hệ thống trong các công sở, trong khi lưu giữ những file người dùng ở địa điểm trung tâm. Một số tính năng của hệ thống NFS, chẳng hạn mount -nosuidoption, có thể được dùng để ngăn chặn những người dùng trái phép mở các thiết bị cũng như các file hệ thống.

Môi trường NFS dùng RPC an toàn để xác thực người dùng đưa ra yêu cầu trên mạng. Điều này được hiểu như là NFS an toàn. Cơ chế xác thực, AUTH_DH, dùng mã DES với xác thực Diffie-Hellman để bảo đảm truy nhập hợp pháp. Cơ chế AUTH_DH còn được gọi là AUTH_DES. Cơ chế AUTH_KERB4 dùng mã DES với xác thực Kerberos. Cơ chế này còn được gọi là AUTH_KERB.

NFS Adminitration Guide mô tả cách thiết lập và quản trị NFS an toàn. Cách thiết lập các bảng NIS+ và đưa các tên vào bảng cred được đề cập trong *Solaris Naming Adminitration Guide*. Xem "Thực thi xác thực Diffie-Hellman" để có nét phác thảo về các bước liên quan tới xác thực RPC.

5.1.2-Mã DES

Các hàm mã theo chuẩn mã hoá dữ liệu (DES) dùng khoá 56-bit để mã hoá khoá bí mật. Nếu hai người dùng có tác quyền (hay các nguyên lý) biết cùng một khoá DES, họ có thể kết nối với tư cách cá nhân dùng khoá để mã hoá và giải mã văn bản. DES là một cơ chế mã hoá tương đối nhanh. Một DES chip mã hoá còn nhanh hơn, nhưng khi không có chip, thì thực thi phần mềm thay thế.

Rủi ro của việc chỉ dùng khoá DES là với đủ thời gian kẻ xâm nhập có thể thu thập đầy đủ các thông điệp văn bản mã được mã hoá bởi cùng một khoá, do đó có thể phát hiện ra khoá và gải mã các thông điệp. Vì lý do này, các hệ thống an toàn như NFS an toàn thường xuyên thay đổi khoá.

5.1.3-Xác thực Diffie-Hellman

Phương pháp xác thực người dùng Diffie-Hellman là rất quan trọng đối với một kẻ xâm nhập để bẻ khoá. Mỗi client và server có khoá sở hữu riêng của nó (đôi khi gọi là khoá bí mật) được dùng cùng với khoá công khai để tạo ra một khoá công cộng. Họ dùng khoá công cộng để kết nối với những người khác dùng hàm mã hoá/giải mã đã thoả thuận (ví dụ DES). Phương pháp này được gọi là xác thực DES trong các ấn phẩm Solaris trước đây.

Xác thực dựa vào khả năng hệ thống gửi sử dụng khoá công cộng mã hoá phiên hiện hành mà hệ thống nhận có thể giải mã và kiểm tra ngược lại phiên hiện hành của nó. Bạn phải đảm bảo đồng bộ thời gian trên client và server.

Các khoá công khai và bí mật được lưu trong cơ sở dữ liệu NIS hoặc NIS+. NIS lưu các khoá theo sơ đồ publickey, và NIS+ lưu các khoá trong bảng cred. Các file này chứa khoá công khai và khoá bí mật cho tất cả người dùng trong tương lai.

Người giám quản hệ thống có trách nhiệm thiết lập các bảng NIS hoặc NIS+ và sinh một khoá công khai và một khoá bí mật cho mỗi người dùng. Khoá bí mật được lưu đã mã hoá cả mật khẩu của người dùng. Điều này làm cho khoá bí mật chỉ có người dùng biết.

Sau đây chúng tôi mô tả một loạt giao dịch trong một phiên client-server dùng trình cấp phép DH (AUTH_DH).

Sinh các khoá công khai và bí mật

Đôi khi trước một giao dịch, người giám quản thực hiện hoặc lệnh newkey hoặc lệnh nisaddcred sinh khoá công khai và khoá bí mật. (Mỗi người dùng có một khoá công khai và một khoá bí mật duy nhất). Khoá công khai được lưu trong cơ sở dữ liệu công khai; khoá bí mật được lưu dưới dạng mã trong cùng cơ sở dữ liệu. Dùng lệnh chkey để thay đổi cặp khoá này.

Thực hiện lệnh keylogin
Thông thường, mật khẩu đăng nhập giống hệt mật khẩu RPC an toàn. Trong trường hợp này, không đòi hỏi keylogin. Nếu các mật khẩu khác nhau, thì những người dùng phải đăng nhập, và sau đó thực hiện keylogin một cách rõ ràng.

Trình keylogin nhắc người dùng về mật khẩu RPC an toàn và dùng mật khẩu này để giải mã khoá bí mật. Sau đó trình keylogin chuyển khoá bí mật đã giải mã tới trình gọi là keyserver. (Keyserver là một dịch vụ RPC an toàn với bộ dữ liệu cục bộ trên mỗi máy tính). Keyserver lưu lại khoá bí mật đã giải mã và chờ người dùng khởi hoạt một giao dịch RPC an toàn với server.

Nếu các mật khẩu giống nhau, trình đăng nhập chuyển khoá bí mật tới keyserver. Nếu các mật khẩu được yêu cầu khác nhau và người dùng luôn phải chạy keylogin, thì trình keylogin có thể được bao hàm trong file cấu hình môi trường của người dùng, chẳng hạn ~/.login, ~/.cshrc, hoặc ~/.profile, sao cho nó tự động chạy mỗi khi người dùng đăng nhập.

Sinh khoá giao tiếp
Khi người dùng khởi hoạt một giao dịch với server:
1. Keyserver ngẫu nhiên sinh một khoá giao tiếp.
2. Kernel dùng khoá giao tiếp mã hoá tem thời gian của client (giữa những thứ khác).
3. Keyserver tìm khoá công khai của server trong CSDL khoá công khai (xem trang chỉ dẫn **publickey(4)**).
4. Keyserver dùng khoá bí mật của client và khoá công khai của server tạo khoá công cộng.
5. Keyserver mã hoá khoá giao tiếp bằng khoá công cộng.

Liên lạc thứ nhất với server
Cuộc truyền bao gồm tem thời gian đã mã và khoá giao tiếp đã mã khi đó được gửi tới server. Cuộc truyền này chứa một nhãn quyền và một nhãn xác minh. Nhãn quyền gồm ba thành phần:

- Tên mạng của client
- Khoá giao tiếp mã hoá bởi khoá công cộng
- Một "cửa sổ" ("window") mã hoá bởi khoá giao tiếp

Window là độ vi sai mà client cho là hợp lệ giữa đồng hồ của server và tem thời gian của client. Nếu độ vi sai giữa đồng hồ của server và tem thời gian của client lớn hơn window, thì server sẽ loại bỏ yêu cầu của client. Theo những tình huống thông thường điều này sẽ không xảy ra vì client trước hết phải đồng bộ với server trước khi bắt đầu phiên RPC.

Nhãn xác minh của client gồm:

- Tem thời gian đã mã
- Một nhãn xác minh window đã mã giảm dần tới 1

Nhãn xác minh window cần thiết trong trường hợp ai đó muốn mạo nhận người dùng và ghi một chương trình chỉ bổ sung các bit ngẫu nhiên, thay cho việc điền các trường nhãn quyền và xác minh đã mã. Server sẽ giải mã khoá giao tiếp thành một khoá ngẫu nhiên nào đó và dùng nó để thử giải mã window và tem thời gian. Kết quả sẽ được các số ngẫu nhiên. Tuy nhiên, sau vài nghìn thử nghiệm, có khả năng cặp window/tem thời gian ngẫu nhiên sẽ vượt qua hệ thống xác thực. Nhãn xác minh window làm cho việc phỏng đoán nhãn quyền đúng khó khăn hơn nhiều.

Giải mã khoá giao tiếp

Khi server tiếp nhận cuộc truyền từ client:

1. Keyserver yêu cầu server tìm khoá công khai của client trong CSDL khoá công khai.
2. Keyserver dùng khoá công khai của client và khoá bí mật của server suy ra khoá công cộng - giống khoá công cộng do client tính. (Chỉ có client và server có thể tính khoá công cộng vì việc làm này đòi hỏi biết một khoá bí mật hoặc một khoá khác).
3. Kernel dùng khoá công cộng giải mã khoá giao tiếp.
4. Kernel gọi keyserver để giải mã tem thời gian của client bằng khoá giao tiếp đã giải mã.

Lưu thông tin trên server

Sau khi server giải mã tem thời gian của client, nó lưu bốn mục thông tin vào bảng nhãn quyền:

- Tên máy tính của client
- Khoá giao tiếp
- Độ vi sai (window)
- Tem thời gian của client

Server lưu ba mục đầu tiên cho lần dùng sau. Nó lưu tem thời gian để bảo vệ chống lặp lại. Server chỉ chấp nhận những tem thời gian theo thứ tự thời gian lớn hơn tem cuối cùng nhìn thấy, nên các giao dịch lặp lại chắc chắn bị loại bỏ.

Ghi chú - Ẩn trong các thủ tục này là tên của người gọi cần được xác thực theo cách nào đó. Keyserver không thể dùng xác thực DES để làm điều này vì nó sẽ tạo ra tắc nghẽn. Để giải quyết vấn đề này, keyserver lưu các khoá bí mật bằng UID và chỉ thừa nhận các yêu cầu đối với các tiến trình root cục bộ.

Gửi trả nhãn xác minh cho client

Server trả lại nhãn xác minh cho client, gồm:

- Index ID mà server ghi trong cache nhãn quyền của nó
- Tem thời gian của client trừ 1 mã hoá bằng khoá giao tiếp

Lý do trừ 1 ở tem thời gian là để chắc chắn rằng tem thời gian bị sai và không thể dùng lại làm nhãn xác minh client.

Client xác thực server

Client nhận nhãn xác minh và xác thực server. Client biết rằng chỉ có server mới có thể gửi nhãn xác minh vì chỉ có server mới biết client gửi tem thời gian nào.

Các giao dịch phụ
Với mỗi giao dịch sau giao dịch đầu tiên, client trả lại index ID cho server trong giao dịch thứ hai của nó và gửi một tem thời gian đã mã khác. Serser gửi trở lại tem thời gian của client trừ 1 mã hoá bằng khoá giao tiếp.

5.1.4-Kerberos version 4
Kerberos là hệ thống xác thực đã được phát triển tại Viện Công nghệ Massachusetts. Kerberos dùng mã DES để xác thực người dùng khi đăng nhập hệ thống. Xác thực dựa vào khả năng hệ thống gửi dùng khoá công cộng để mã phiên hiện hành mà hệ thống nhận có thể giải mã và kiểm tra ngược lại phiên hiện hành của nó. Kerberos version 4 được trợ giúp ban đầu phiên bản Solaris 2.6.

Kerberos tiến hành xác thực mật khẩu đăng nhập của người dùng. Người dùng đưa vào lệnh kinit để thu được thẻ đã phê chuẩn thời gian của phiên (hoặc 8 giờ, là thời gian phiên mặc định) từ server xác thực Kerberos. Khi người dùng logout, thẻ có thể bị huỷ (dùng lệnh kdestroy).

Phầnmềm Kerberos có từ dự án Athena của MIT, và không phải là một phần của phần mềm SunOS 5.7. Phần mềm SunOS 5.7 cung cấp:
- Các lệnh và APIs mà client dùng để tạo, yêu cầu và phê chuẩn các thẻ
- Tuỳ chọn xác thực cho RPC an toàn
- Trình hoạt động độc lập bên cạnh client, **kerbd**(1M)

Sau đây chúng tôi mô tả các bước của cách "Cài đặt xác thực Kerberos với NFS" nhằm đưa ra một cái nhìn tổng quan về cách làm việc của thủ tục xác thực Kerberos.

Ghi chú - Solaris cung cấp khả năng kết nối với tiện ích Kerberos. Nó không cung cấp trọn gói Kerberos. Tuy nhiên, bạn có thể ftp nguồn Kerberos 4 từ athena-dist.mit.edu dùng anonymous làm tên người dùng và địa chỉ email của bạn làm mật khẩu. Nguồn được đặt trong thư mục pub/kerberos.

Quá trình sau giả thiết rằng trung tâm phân phối khoá (KDC) đã được cài đặt sẵn trên mạng dùng nguồn sẵn có công khai từ dự án Athena của MIT.
1. Tiện ích /usr/sbin/kerbd phải đang chạy trên client và server NFS.
   Tiện ích này thông thường được bắt đầu khi cần thiết bằng inetd. Có thể dùng lệnh rpcinfo để chắc chắn rằng dịch vụ kerbd đã được đăng ký. kerbd là tiện ích kiểu người dùng. Nó ghép nối với kernel RPC và KDC. Nó sinh và phê chuẩn các thẻ xác thực. mount
2. Người giám quản hệ thống gán cho NFS server dùng xác thực Kerberos.
   Phần mềm Kerberos của MIT thường đăng ký các tên chính yếu ở trung tâm phân phối khoá (KDC) trên Kerberos server. Các mục sau được yêu cầu:
   - root.*hostname* (đòi hỏi đối với mỗi client)
   - nfs.*hostname* (đòi hỏi đối với mỗi NFS server)
3. Người dùng thiết lập hệ thống file dùng chung.
   Người dùng trên client phải có một thẻ của root trên client để thiết lập hệ thống file dùng chung.
4. Người dùng dùng lệnh kinit để đăng nhập dịch vụ Kerberos.

Server xác thực Kerberos xác nhận yêu cầu và cấp một thẻ cho dịch vụ cấp thẻ.

5. Người dùng truy nhập thư mục đã thiết lập.
   Tiện ích kerbd tự động bảo vệ thẻ đại diện cho client đối với NFS server xuất hệ thống file. Tại điểm này, có hai thẻ có hiệu lực, thẻ cấp thẻ (ticket-granting ticket) ban đầu và thẻ của server.
6. Người dùng huỷ các thẻ ở cuối phiên để ngăn làm tổn thương chúng.
   Lệnh kdestroy huỷ các thẻ xác thực Kerberos có hiệu lực của người dùng bằng cách ghi các số 0 vào file chứa thẻ. Bạn có thể đặt lệnh kdestroy trong file .logout của bạn, sao cho tất cả các thẻ Kerberos được huỷ tự động khi bạn thoát khỏi hệ thống.
7. Nếu các thẻ bị huỷ trước khi phiên làm việc kết thúc, người dùng phải yêu cầu một thẻ mới bằng lệnh kinit.

5.2-Phân phối xác thực Diffie-Hellman

Người giám quản hệ thống có thể thực thi các chính sách trợ giúp an toàn mạng. Mức an toàn yêu cầu đối với mỗi vị trí (site) khác nhau. Phần này cung cấp những chỉ dẫn về một số tác vụ kết hợp với an toàn mạng.

5.2.1-Cách khởi động Keyserver

1. Trở thành superuser
2. Xác nhận tiện ích keyserv (keyserver) hiện không chạy

```
# ps -ef | grep keyserv
root   100     1   16  Apr 11 ?          0:00 /usr/sbin/keyserv
root  2215  2211    5  09:57:28 pts/0    0:00 grep keyserv
```

3. Khởi động keyserver nếu nó hiện không chạy

```
# /usr/sbin/keyserv
```

5.2.2-Cách thiết lập nhãn quyền NIS+ đối với xác thực Diffie-Hellman

Để có mô tả chi tiết về an toàn NIS+, xem Solaris Naming Administration Guide.

Đặt một khoá mới cho root trên NIS+ client

1. Trở thành superuser.
2. Soạn thảo file /etc/nsswitch.conf và bổ sung dòng sau:

```
publickey:  nisplus
```

3. Khởi hoạt NIS+ client.

```
# nisinit -cH hostname
```

hostname là tên của NIS+ server tin cậy chứa đầu vào của máy client trong các bảng của nó.

4. Bổ sung client vào bảng cred bằng cách gõ các lệnh sau.

```
# nisaddcred local
# nisaddcred des
```

5. Xác nhận cài đặt bằng lệnh keylogin.

Nếu bạn được nhắc mật khẩu, thì thủ tục đã thành công.

Ví dụ về cách đặt một khoá mới cho root trên NIS+ client

Ví dụ sau đây dùng máy chủ pluto để đặt earth là một NIS+ client. Bạn có thể bỏ qua những lời cảnh báo. Việc chấp nhận lệnh keylogin xác nhận rằng đã đặt earth đúng như là một NIS+ client an toàn.

```
# nisinit -cH pluto
NIS Server/Client setup utility
This machine is in the North.Abc.COM. directory.
Setting up NIS+ client ...
All done
# nisaddcred local
# nisaddcred des
DES principal name : unix.earth@North.Abc.COM
Adding new key for unix.earth@North.Abc.Com (earth.North.Abc.COM.)

Network password: xxx <Press Return>
Warning, password differs from login password.
Retype password: xxx <Press Return>

# keylogin
Password:
#
```

Đặt một khoá mới cho NIS+ user:

1. Bổ sung người dùng vào bảng cred trên root master server bằng cách gõ lệnh sau:

```
# nisaddcred -p unix. UID@domainname -p username.domainname. des
```

Lưu ý rằng trong trường hợp này *username.domainname* phải kết thúc bằng dấu chấm (.)

2. Xác nhận cài đặt bằng cách đăng nhập như client và gõ lệnh keylogin

Ví dụ về đặt một khoá mới cho NIS+ user

Ví dụ sau đưa ra xác thực an toàn DES cho người dùng george.

```
# nisaddcred -p unix.1234@North.Abc.com -p george.North.Abc.COM. des
DES principal name : unix.1234@North.Abc.COM
Adding new key for unix.1234@North.Abc.COM (george.North.Abc.COM.)

Password:
Retype password:

# rlogin rootmaster -l george
# keylogin
Password:
#
```

5.2.3-Cách đặt nhãn quyền NIS cho xác thực Diffie-Hellman

Tạo một khoá mới cho superuser trên client

1. Trở thành superuser trên client.
2. Soạn thảo file /etc/nsswitch.conf và bổ sung dòng sau:

```
publickey:  nis
```

3. Tạo một cặp khoá bằng lệnh newkey

```
# newkey  -h  hostname
```

hostname là tên của client

Ví dụ về việc cài đặt NIS+ client để dùng an toàn Diffie-Hellman

Ví dụ sau đây cài đặt earth là một NIS+ client

```
# newkey  -h  earth
Adding new key for unix.earth@North.Abc.COM

New password:
Retype password:
Please wait for the database to get updated...
Your new key has been successfully stored away.
#
```

Tạo một khoá mới cho user

1. Đăng nhập tới server với tư cách superuser.

Chỉ người giám quản hệ thống đăng nhập tới NIS+ server mới có thể sinh một khoá mới cho user.

2. Tạo khoá mới cho user

```
# newkey  -u  username
```

username là tên của người dùng. Hệ thông nhắc đưa vào mật khẩu. Người giám quản hệ thống có thể goc mật khẩu chung. Khoá bí mật được lưu đã mã hoá với mật khẩu chung.

```
# newkey  -u  george
Adding new key for unix.earth@North.Abc.COM

New password:
Retype password:
Please wait for the database to get updated...
Your new key has been successfully stored away.
#
```

3. Lệnh cho người dùng đăng nhập và gõ lệnh chkey -p.

Điều này cho phép người dùng mã lại khoá bí mật của họ với mật khẩu chỉ có người dùng biết.

```
earth%  chkey  -p
Updating nis publickey database.
```

```
Reencrypting key for unix.1234@Abc.North.Acme.COM
Please enter the Secure-RPC password for george:
Please enter the login password for george:
Sending key change request to pluto...
#
```

Ghi chú - Lệnh chkey có thể được dùng để tạo một cặp khoá mới cho user

5.2.4-Cách chia xẻ và gắn các file với xác thực Diffie-Hellman

Điều kiện đầu

Cho phép xác thực publickey Diffie-Hellman trên mạng. Xem mục "Cách đặt nhãn quyền NIS+ cho xác thực Diffie-Hellman" và "Cách đặt nhãn quyền NIS cho xác thực Diffie-Hellman".

Dùng chung hệ thống file với xác thực Diffie-Hellman

1. Trở thành superuser.
2. Dùng chung hệ thống file với xác thực Diffie-Hellman.

```
# share -F nfs -o sec=dh /filesystem
```

Gắn hệ thống file với xác thực Diffie-Hellman

1. Trở thành superuser.

2. Gắn hệ thống file với xác thực Difie-Hellman.

```
# mount -F nfs -o sec=dh server.resource mountpoint
```

Tùy chọn -o sec=dh gắn hệ thống với xác thực AUTH_DH

5.3-Quản trị xác thực Kerberos version 4

Người giám quản hệ thống có thể thực thi các chính sách trợ giúp an toàn mạng. Mức an toàn yêu cầu đối với mỗi vị trí (site) khác nhau. Phần này cung cấp những chỉ dẫn về một số tác vụ kết hợp với an toàn mạng.

5.3.1-Cách chia xẻ và gắn các file với xác thực Kerberos

Điều kiện đầu

Phải cho phép xác thực Kerberos trên mạng.

Dùng chung hệ thống file với xác thực Kerberos

1. Trở thành superuser.
2. Dùng chung hệ thống file với xác thực Kerberos.

```
# share -F nfs -o sec=krb4 /filesystem
```

Gắn hệ thống file với xác thực Diffie-Hellman

1. Trở thành superuser.
2. Gắn hệ thống file với xác thực Kerberos.

```
# mount -F nfs -o sec=krb4 server.resource mountpoint
```

Tùy chọn -o sec=krb4 gắn hệ thống file với xác thực AUTH_KRB

5.3.2-Cách lấy thẻ Kerberos cho superuser trên client

Nếu không gán được hệ thống file NFS mà bạn cần truy nhập, bạn cần có một thẻ của superuser trên client trước khi vào nó.

Lấy thẻ cho hệ thống file chưa sẵn sàng (not-yet-mounted)

1. Trở thành superuser
2. Lấy thẻ Kerberos trên client

```
# kinit root.hostname
```

hostname là tên của hệ thống client

```
# kinit root.earth
Pasword:
#
```

Lấy thẻ của hệ thống file đã sẵn sàng

Nếu mục root.*hostname* của client đã được đưa vào file cấu hình /etc/srvtab, bạn có thể dùng lệnh ksrvtgt để lấy thẻ cho superuser. Trong trường hợp này, bạn không bị yêu cầu đưa ra mật khẩu superuser. Tham khảo tài liệu của MIT để có thông tin về cách khởi hoạt file /etc/srvtab.

1. Trở thành superuser
2. Lấy thẻ của hệ thống file gán được.

```
# ksrvtgt root.hostname
```

Ví dụ - Lấy thẻ Kerberos cho superuser trên client

```
# ksrvtgt root.earth
#
```

5.3.3-Cách đăng nhập tới dịch vụ Kerberos

Dùng lệnh kinit -l username để đăng nhập tới dịch vụ Kerberos

```
earth% kinit -l username
```

Lệnh kinit nhắc bạn về thời gian có hiệu lực của thẻ (tuỳ chọn -l), và mật khẩu của bạn. Nó in ra thông tin về thẻ theo kiểu sao y (tuỳ chọn -v).

Ví dụ về đăng nhập dịch vụ Kerberos

```
earth% kinit -l jjones
SunOS (earth)
Kerberos Initialization for "jjones"
Kerberos ticket lifetime (minutes): 480
Password:
earth%
```

5.3.4-Cách liệt kê các thẻ Kerberos

```
earth% klist
```

Ví dụ về liệt kê các thẻ Kerberos

```
earth% klist
Ticket file:  /tmp/tkt8516
Principal:  jjones@North.Abc.COM
  Issued              Expires             Principle
  Jan 14 20:40:54  Jan
15:04:40:54  krbtgt.North.Abc.COM@North.Abc.COM
```

5.3.5-Cách truy nhập thư mục với xác thực Kerberos

Gõ cd /*mountpoint*

Truy nhập thư mục đã gán được, chỉ như là bạn sẽ có một thư mục gán được khác nào đó. Bạn có thể liệt kê các file trong thư mục bằng lệnh ls, hoặc liệt kê các thẻ Kerberos với lệnh klist. mount

Ví dụ về cách truy nhập thư mục với xác thực Kerberos

Trong ví dụ sau, người dùng jjones có thể biến đổi thư mục mntkrb gán được và liệt kê các file trong thư mục này.

Tiện ích kerbd tự động bảo vệ thẻ đại diện cho người dùng đối với NFS server xuất hệ thống file. Tại điểm này có hai thẻ có hiệu lực - thẻ cho việc cấp thẻ và thẻ của server. Hai thẻ này được liệt kê bằng klist.

```
earth% cd /mntkrb
earth% ls -l /mntkrb
-rw-r--r--     1 marks staff   29 Jul 14 12:22 sports
drwxr-xr-x   3 jjones staff 512 Sep 13 13:44 market

earth% klist
Ticket file:  /tmp/tkt8516
Principal:  jjones@North.Abc.COM
  Issued              Expires             Principle
  Jan 14 20:40:54  Jan
15:04:40:54  krbtgt.North.Abc.COM@North.Abc.COM
  Jan 14 20:43:21  Jan 15:04:43:21     nfs.pluto@North.Abc.COM
```

5.3.6-Cách huỷ thẻ Kerberos

Gõ vào kdestroy

Huỷ các thẻ Kerberos khi phiên làm việc kết thúc, sao cho người dùng trái phép không thể có được quyền truy nhập nó. Nếu bạn muốn khởi hoạt lại xác thực Kerberos, thì dùng lệnh kinit.

Ví dụ về việc huỷ thẻ Kerberos
Ví dụ sau trình bày cách huỷ thẻ Kerberos. Sau đó nếu người dùng thử thay đổi hay hiển thị thư mục có bảo vệ Kerberos, ticket server từ chối truy nhập.

```
 earth%  kdestroy
Tickets destroyed
earth%  ls  /mntkrb
Can't get Kerberos key:  No ticket file (tf_util)
NFS getattr failed for server pluto: RPC: Authentication error
can not access directory /mntkrb.
```

5.4-Giới thiệu về PAM
Khung module xác thực khả cắm (PAM) cho phép bạn "cắm thêm" công nghệ xác thực mới mà không cần thay đổi các dịch vụ hệ thống tiếp nhận như login, ftp, telnet và ..... Bạn cũng có thể dùng PAM để tích hợp UNIX login với các cơ chế an toàn khác như DCE hoặc Kerberos. Cũng có thể "cắm" khung này vào các cơ chế để quản lý tài khoản, phiên và mật khẩu.

5.4.1-Những lợi ích của việc dùng PAM
PAM cho phép người giám quản hệ thống lựa chọn bất kỳ tổ hợp các dịch vụ nào hệ thống tiếp nhận (ftp, login, telnet, hoặc rsh làm ví dụ) để xác thực người dùng. Một số lợi ích mà PAM cung cấp là:

- Chính sách cấu hình linh hoạt
    - Chính sách xác thực từng ứng dụng.
    - Khả năng lựa chọn cơ chế xác thực mặc định
    - Nhiều mật khẩu trên các hệ thống an toàn cao.
- Dễ dùng đối với người dùng cuối
    - Không phải gõ lại các mật khẩu nếu chúng giống nhau đối với các cơ chế khác nhau.
    - Khả năng dùng mật khẩu riêng cho nhiều phương thức xác thực bằng tính năng ánh xạ mật khẩu, ngay cả khi các mật khẩu gắn với mỗi phương thức xác thực khác nhau.
    - Khả năng nhắc người dùng các mật khẩu đối với nhiều phương thức xác thực mà không bắt người dùng đưa vào nhiều lệnh.
- Khả năng bỏ qua các tham số tùy chọn đối với các dịch vụ xác thực người dùng.

PAM dùng các module khả cắm thời gian thực để bảo đảm xác thực cho các dịch vụ hệ thống tiếp nhận. Các module này được chia thành bốn kiểu khác nhau dựa vào chức năng của chúng: xác thực, quản lý tài khoản, quản lý phiên và quản lý mật khẩu. Tính năng xếp chồng (stacking) được cung cấp cho phép bạn xác thực những người dùng nhiều dịch vụ, cũng như tính năng ánh xạ mật khẩu không đòi hỏi người dùng nhớ nhiều mật khẩu.

5.4.2-Các kiểu PAM module
Có hiểu biết về các kiểu PAM module là rất quan trọngvì kiểu module xác định cách ghép nối với module. Có bốn kiểu PAM module thời gian thực:

- Các *module xác thực* đảm bảo xác thực những người dùng và cho phép đặt, làm tươi và huỷ các nhãn quyền. Chúng cung cấp công cụ giám quản có giá trị đối với việc định danh người dùng.
- Các *module tài khoản* kiểm tra thời hạn mật khẩu, làm mất hiệu lực tài khoản và các hạn chế về thời gian truy nhập. Sau khi người dùng được định danh thông qua các module xác thực, các module tài khoản xác định khi nào người dùng sẽ được truy nhập.
- Các *module phiên* quản lý việc mở và đóng một phiên xác thực. Chúng có thể ghi lại hành động hay đảm bảo dọn dẹp sau khi phiên kết thúc.
- Các *module mật khẩu* cho phép thay đổi mật khẩu hiện tại.

5.4.3-Tính năng stacking

PAM cung cấp một phương pháp xác thực người dùng nhiều dịch vụ bằng *stacking*. Phụ thuộc vào cấu hình, người dùng có thể được nhắc về các mật khẩu với mỗi phương thức xác thực. Trật tự mà theo đó các dịch vụ xác thực được dùng được xác định thông qua file cấu hình PAM.

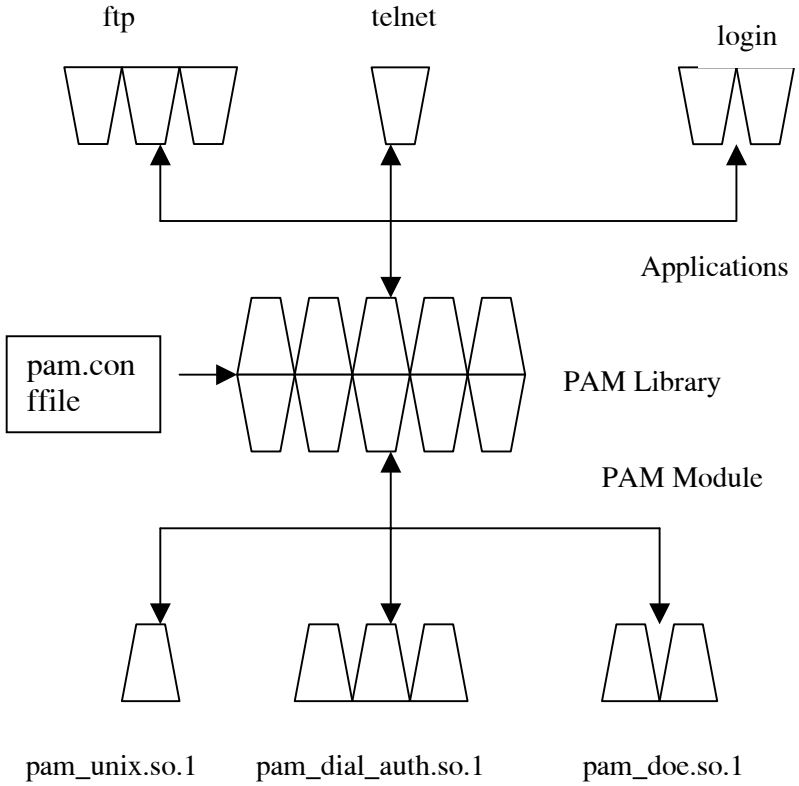
5.4.4-Tính năng ánh xạ mật khẩu

Phương pháp stacking có thể đòi hỏi một người dùng nhớ một vài mật khẩu. Với tính năng *ánh xạ mật khẩu*, mật khẩu chính được dùng để giải mã các mật khẩu khác, nên người dùng không cần nhớ hay đưa vào nhiều mật khẩu. Sự lựa chọn khác này là đồng bộ các mật khẩu chéo qua mỗi cơ chế xác thực. Lưu ý rằng, điều này có thể làm tăng rủi ro về an toàn, vì an toàn của mỗi cơ chế bị hạn chế do dùng mật khẩu ít an toàn nhất trong stack.

5.5-Chức năng tiện ích PAM

Phần mềm PAM gồm một thư viện, một vài module và một file cấu hình. Các phiên bản mới của một vài lệnh hay tiện ích hệ thống tiếp nhận mang đến ưu điểm của các ghép nối PAM cũng được bao hàm trong đó.

Hình 5-1 minh hoạ mối quan hệ giữa các ứng dung, thư viện PAM, file pam.conf và các PAM module.



Hình 5-1 PAM làm việc như thế nào

Các ứng dụng (ftp, telnet, và login) dùng thư viện PAM để truy nhập module thích hợp. File pam.conf xác định dùng các module nào, và sử dụng theo trật tự nào với mỗi ứng dụng. Những câu trả lời từ các module được truyền lại tới ứng dụng thông qua thư viện.

Phần sau mô tả mối quan hệ này.

5.5.1-Thư viện PAM

Thư viện PAM, /usr/lib/libpam cung cấp khung tải các module thích hợp và quản lý tiến trình stacking. Nó đảm bảo cấu trúc chung cho tất cả các module cắm vào.

5.5.2-Các PAM module

Mỗi PAM module thực thi một cơ chế cụ thể. Khi cài đặt xác thực PAM, bạn cần đặc tả cả module và kiểu module xác định module sẽ làm gì. Có thể có nhiều kiểu module (xác thực, tài khoản, phiên, hay mật khẩu) được kết hợp với mỗi module.

Danh sách sau mô tả từng PAM module

- Module pam_unix, /usr/lib/security/pam_unix.so.1, đảm bảo trợ giúp xác thực, quản lý tài khoản, quản lý phiên và quản lý mật khẩu. Có thể dùng bất kỳ một trong bốn định nghĩa kiểu module với module này. Nó dùng các mật khẩu UNIX để xác thực. Trong môi trường Solaris, điều khiển sự lựa chọn các dịch vụ tên thích hợp để lấy các bản ghi mật khẩu thông qua file /etc/nsswitch.conf. Xem **pam_unix**(5) để có thêm thông tin.

- Module dial_auth, /usr/lib/security/pam_auth.so.1, có thể chỉ dùng để xác thực. Nó dùng dữ liệu lưu trong các file /etc/dialups và /etc/d_passwd để xác thực. Modle này chủ yếu là login dùng. Xem **pam_dial_auth**(5) để có thêm thông tin.
- Module rhosts_auth, /usr/lib/security/pam_rhosts_auth.so.1, có thể cũng chỉ dùng để xác thực. Nó dùng dữ liệu lưu trong các file ~/.rhosts và /etc/host.equip thông qua ruserok(). Module này chủ yếu là các lệnh rlogin và rsh dùng. Xem **pam_rhosts_auth**(5) để có thêm thông tin.

Vì các lý do an toàn, các file module này phải do root sở hữu và không thể ghi thông qua các quyền group và other. Nếu file không do root sở hữu, thì PAM sẽ không tải module.

5.5.3-File cấu hình PAM

File cấu hình PAM, /etc/pam.conf, xác định các dịch vụ xác thực được sử dụng, và thứ tự sử dụng chúng. Có thể soạn thảo file này để chọn các cơ chế xác thực cho mỗi ứng dụng hệ thống tiếp nhận.

Cú pháp file cấu hình

File cấu hình PAM gồm các mục với cú pháp như sau:

service_name module_type control_flag module_path module_options

service_name	Tên của dịch vụ (ví dụ, ftp, login, telnet).
module_type	Kiểu module cho dịch vụ.
control_flag	Xác định tiếp tục hay giải nghĩa lỗi của module.
module_path	Đường dẫn tới đối tương thư viện thực thi chức năng dịch vụ.
module_options	Các tùy chọn đặc biệt được truyền tới các module dịch vụ.

Bạn có thể bổ sung lời dẫn giải vào file pam.conf bằng cách gõ dấu # (dấu pao) ở đầu dòng. Dùng dấu khoảng cách để phân biệt các trường.

Ghi chú - Một mục trong file cấu hình bị bỏ qua nếu tồn tại một trong những điều kiện sau đây: dòng có ít hơn bốn trường, đưa ra một giá trị sai cho *module_type* hay *control_flag*, hoặc module tên không tìm thấy.

Các tên dịch vụ hợp lệ

Bảng 5-1 liệt kê một số tên dịch vụ hợp lệ , các kiểu module có thể dùng được với dịch vụ đó, và tiện ích hay lệnh kết hợp với tên dịch vụ.

Có một vài kiểu module không tương thích với từng dịch vụ. Ví dụ, kiểu module password chỉ được đặc tả để đi với lệnh passwd. Không có kiểu module auth kết hợp với lệnh này vì nó không liên quan tới xác thực.

Bảng 5-1 Các tên dịch vụ hợp lệ đối với /etc/pam.conf

Tên dịch vụ	Tiện ích hay Lệnh	Kiểu module
dtlogin	/usr/dt/bin/dtlogin	auth, account, session
ftp	/usr/sbin/in.ftpd	auth, account, session
init	/usr/sbin/init	session

login	/usr/bin/login	auth, account, session
passwd	/usr/bin/passwd	passwd
rexd	/usr/sbin/rpc.rexd	auth
rlogin	/usr/sbin/in.rlogind	auth, account, session
rsh	/usr/sbin/in.rshd	auth, account, session
sac	/usr/lib/saf/sac	session
su	/usr/bin/su	auth, account, session
telnet	/usr/sbin/in.telnetd	auth, account, session
ttymon	/usr/lib/saf/ttymon	session
uucp	/usr/sbin/in.uucpd	auth, account, session

Các cờ điều khiển

Để xác định tiếp tục hay hay xử lý lỗi từ một module trong tiến trình xác thực, bạn phải chọn một trong bốn cờ điều khiển cho mỗi mục. Các cờ điều khiển chỉ rõ cách cuộc thử thành công hay thất bại thông qua mỗi module được thực hiện. Mặc dù các cờ này áp dụng cho tất cả các kiểu module, lời giải nghĩa sau đây giả thiết rằng các cờ này để dùng cho các module xác thực. Các cờ điều khiển như sau:

- required - Module này phải trả lại kết quả để có được kết quả tổng thể.
  Nếu tất cả các module có nhãn required, thì tất cả các module phải có kết quả cho người dùng được xác thực.

  Nếu một số module lỗi , thì thông báo giá trị lỗi từ module bị lỗi đầu tiên.

  Nếu lỗi xảy ra ở một module được gán cờ required, tất cả các module trong stack vẫn tiếp tục nhưng trả lại lỗi.

  Nếu không có module nào được gán cờ required, thì ít nhất một trong các đầu vào của dịch vụ đó phải có kết quả cho người dùng được xác thực.
- requisite - Module này phải trả lại kết quả để tiến hành xác thực bổ sung.
  Nếu hỏng hóc xảy ra với một module được gán cờ requisite, thì một lỗi được trả lại tức thì cho ứng dụng và không có xác thực bổ sung nào được thực hiện. Nếu stack không có các module ưu tiên được gán cờ required bị hỏng, thì lỗi từ module này bị trả lại. Nếu một module ban đầu được gán nhãn required bị hỏng, thì trả lại thông báo lỗi từ module required.
- optional - Nếu module này hỏng, có thể có kết quả tổng thể nếu một module khác trong stack này trả lại kết quả.
  Cờ optional sẽ được dùng khi một kết quả trong stack cũng đủ xác thực người dùng. Cờ này sẽ chỉ được sử dụng nếu nó không quan trọng đối với sự thành công của cơ chế cụ thể này.
  Nếu những người dùng của bạn cần có quyền kết hợp với một cơ chế đặc biệt để bắt đầu tiến hành công việc của họ, thì bạn không nên dán nhãn optional.
- sufficient - Nếu module này thành công, bỏ qua các module còn lại trong stack, kể cả nếu chúng được gán nhãn required.
  Cờ sufficient chỉ rõ rằng một xác thực thành công cũng đủ để người dùng được cấp quyền truy nhập.
  Mục "Cấu hình PAM" sẽ cung cấp hông tin thêm về các cờ này, mô tả file /etc/pam.conf mặc định

File pam.conf tổng quát
Sau đây là một ví dụ về file pam.conf tổng quát

```
#  PAM  configuration
#  Authentication management
#
login auth required /usr/lib/security/pam_unix.so.1
login auth required /usr/lib/security/pam_dial.so.1
rlogin auth sufficient  /usr/lib/security/pam_rhost_auth.so.1
rlogin auth required /usr/lib/security/pam_unix.so.1
dtlogin auth required /usr/lib/security/pam_unix.so.1
telnet auth required /usr/lib/security/pam_unix.so.1
su auth required /usr/lib/security/pam_unix.so.1
ftp auth required /usr/lib/security/pam_unix.so.1
uucp auth required /usr/lib/security/pam_unix.so.1
rsh auth required /usr/lib/security/pam_rhost_auth.so.1
OTHER auth required /usr/lib/security/pam_unix.so.1
#
#  Account management
#
login account required /usr/lib/security/pam_dial.so.1
rlogin account required /usr/lib/security/pam_unix.so.1
dtlogin account required /usr/lib/security/pam_unix.so.1
telnet account required /usr/lib/security/pam_unix.so.1
ftp account required /usr/lib/security/pam_unix.so.1
OTHER account required /usr/lib/security/pam_unix.so.1
#
#  Session management
#
login session required /usr/lib/security/pam_dial.so.1
rlogin session required /usr/lib/security/pam_unix.so.1
dtlogin session required /usr/lib/security/pam_unix.so.1
telnet session required /usr/lib/security/pam_unix.so.1
uucp session required /usr/lib/security/pam_unix.so.1
OTHER session required /usr/lib/security/pam_unix.so.1
#
#  Password management
#
passwd password required /usr/lib/security/pam_unix.so.1
OTHER password required /usr/lib/security/pam_unix.so.1
```

File pam.conf tổng quát này định rõ:
1. Khi chạy login, xác thực phải có kết quả cho cả hai module pam_unix và pam_dial_auth.
2. Đối với rlogin, xác thực qua module pam_unix phải có kết quả, nếu xác thực qua pam_rhost_auth hỏng.

3. Cờ điều khiển sufficient chỉ rõ rằng đối với rlogin xác thực do module pam_rhost_auth cung cấp thành công là đủ và đầu vào tiếp theo sẽ được bỏ qua.
4. Hầu hết các lệnh khác đòi hỏi xác thực yêu cầu xác thực qua module pam_unix thành công.
5. Xác thực đối với rsh phải có kết quả qua module pam_rhost_auth.

Tên dịch vụ OTHER cho phép một mặc định được đặt cho bất kỳ lệnh nào khác đòi hỏi xác thực mà không có trong file này. Tuỳ chọn OTHER làm cho nó dễ dàng hơn đối với người quản trị file, vì nhiều lệnh đang dùng cùng một module có thể dùng chỉ một đầu vào là đủ. Ngoài ra, tên dịch vụ OTHER, khi dùng như là một "catch-all", có thể đảm bảo mỗi truy nhập được kiểm soát bằng một module. Theo quy ước, đầu vào OTHER được đề cập ở cuối phần này cho mỗi kiểu module.

Các mục còn lại trong file điều khiển việc quản lý tài khoản, phiên và mật khẩu.

Bằng việc dùng tên dịch vụ mặc định, OTHER, file cấu hình PAM tổng quát được đơn giản hoá thành:

```
#  PAM  configuration
#
#  Authentication management
#
login auth required /usr/lib/security/pam_unix.so.1
login auth required /usr/lib/security/pam_dial.so.1
rlogin auth sufficient  /usr/lib/security/pam_rhost_auth.so.1
rlogin auth required /usr/lib/security/pam_unix.so.1
rsh auth required /usr/lib/security/pam_rhost_auth.so.1
OTHER auth required /usr/lib/security/pam_unix.so.1
#
#  Account management
#
OTHER account required /usr/lib/security/pam_unix.so.1
#
#  Session management
#
OTHER session required /usr/lib/security/pam_unix.so.1
#
#  Password management
#
OTHER password required /usr/lib/security/pam_unix.so.1
```

Thông thường, đầu vào cho module_path là "root-relative". Nếu tên file bạn đưa vào cho *module_path* không được bắt đầu bằng dấu /, thì đường dẫn /usr/lib/security/ được đặt trước tên file. Các module ở các thư mục khác phải dùng một tên đường dẫn đầy đủ.

Các giá trị cho *module_options* có thể được tìm thấy ở các trang trợ giúp của module. (Ví dụ, **pam_unix**(5)).

Các tuỳ chọn use_first_pass và try_first_pass do module pam_unix cung cấp cho phép người dùng dùng lại mật khẩu xác thực mà không phải gõ lại nó.

Nếu login đặc tả xác thực qua cả pam_local và pam_unix, thì người dùng được nhắc đưa vào mật khẩu cho mỗi module. Trong trường hợp các mật khẩu giống nhau, tuỳ chọn module use_first_pass chỉ yêu cầu một mật khẩu và dùng mật khẩu đó để xác thực người dùng cho cả hai module. Nếu các mật khẩu khác nhau, xác thực lỗi. Nói chung, tuỳ chọn này nên dùng với cờ điều khiển optional, như trình bày dưới đây, để chắc chắn rằng người dùng vẫn có thể đăng nhập.

```
#  Authentication management
#
login auth required /usr/lib/security/pam_unix.so.1
login auth optional /usr/lib/security/pam_local.so.1  use_first_pass
```

Nếu tuỳ chọn module try_first_pass được dùng thay thế, thì local module yêu cầu mật khẩu thứ hai khi các mật khẩu không tương thích hoặc gây lỗi. Nếu cả hai phương pháp xác thực đều cần cho người dùng để có được quyền truy nhập tới tất cả các công cụ họ cần, thì việc dùng tuỳ chọn này có thể dẫn đến một số xáo trộn đối với người dùng vì người dùng có thể có được quyền truy nhập chỉ với một kiểu xác thực.

5.6-Cấu hình PAM

Phần bên dưới đề cập tới một số tác vụ có thể được yêu cầu để làm cho PAM đầy đủ chức năng. Cụ thể, bạn nên biết một số vấn đề an toàn kết hợp với file cấu hình PAM.

5.6.1-Lập sơ đồ cho PAM

Khi quyết định cách dùng PAM tốt nhất trong môi trường của bạn, thì bạn nên bắt đầu bằng việc xem xét các vấn đề sau:

- Xác định các nhu cầu của bạn là gì, đặc biệt bạn nên chọn những module nào.
- Xác định các dịch vụ cần chú ý đặc biệt; dùng OTHER nếu thích hợp.
- Quyết định thứ tự chạy các module
- Chọn cờ điều khiển cho module đó.
- Chọn các tuỳ chọn cần cho module

Ở đây có một số gợi ý đáng quan tâm trước khi thay đổi file cấu hình PAM:

- Dùng đầu vào OTHER cho mỗi kiểu module sao cho mỗi ứng dụng không phải nói đến.
- Chắc chắn đã xem xét các hiệu ứng an toàn kéo theo của các cờ điều khiển sufficient và optional.
- Xem lại các trang trợ giúp đi cùng với các moduleđể hiểu cách hoạt động của các module, các tuỳ chọn nào khả dụng và sự tương tác giữa các module xếp chồng.

Chú ý - Nếu file cấu hình PAM bị cấu hình thiếu hay có sai lạc, thì có thể ngay cả superuser cũng không thể đăng nhập. Vì sulogin không dùng PAM, nên khi đó superuser sẽ được yêu cầu khởi động máy theo kiểu người dùng đơn lẻ và sửa chữa lỗi.

Sau khi thay đổi file /etc/pam.conf, xem lại nó kỹ càng trong khi vẫn đăng nhập với tư cách superuser. Kiểm tra tất cả các lệnh có thể bị ảnh hưởng do các thay đổi của bạn. Ví dụ, nếu bạn bổ sung module mới vào dịch vụ telnet, dùng lệnh telnet và xác nhận rằng những thay đổi mà bạn tạo ra hoạt động như mong đợi.

5.6.2-Cách bổ sung PAM module

1. Trở thành superuser
2. Xác định các cờ điều khiển và các tuỳ chọn khác sẽ sử dụng
Tham khảo mục "PAM modules" thông tin về module.
3. Sao chép module mới vào /usr/lib/security.
4. Đặt các quyền sao cho root sở hữu file module và các quyền là 555.
5. Soạn thảo file cấu hình PAM, /etc/pam_conf, và bổ sung module này vào các dịch vụ thích hợp.

Kiểm tra

Một số việc kiểm tra là rất quan trọng trước khi hệ thống được khởi động lại trong trường hợp file cấu hình bị sai. Chạy rlogin, su, và telnet trước khi khởi động lại hệ thống. Nếu dịch vụ là một tiện ích đổi chỉ khi hệ thống khởi động, thì có thể cần khởi động lại hệ thống trước khi bạn có thể kiểm tra module đã được bổ sung.

5.6.3-Cách ngăn chặn truy nhập trái phép từ các hệ thống từ xa bằng PAM

Loại bỏ mục rlogin auth rhosts_auth.so.1 khỏi file cấu hình PAM. Điều này ngăn việc đọc các file ~/.rhosts trong suốt phiên rlogin và vì thế ngăn chặn truy nhập trái phép tới hệ thống cục bộ từ các hệ thống từ xa. Tất cả truy nhập rlogin yêu cầu một mật khẩu, bỏ qua sự hiện diện hay nội dung của bất kỳ file ~/.rhosts hay /etc/hosts.equip nào.

Ghi chú - Để ngăn chặn truy nhập trái phép khác tới các file ~/.rhosts, hãy nhớ làm mất hiệu lực file rsh. Cách tốt nhất là loại bỏ đầu vào dịch vụ khỏi /etc/inetd.conf. Thay đổi file cấu hình PAM không ngăn được dịch vụ bị khởi động.

5.6.4-Cách kích hoạt thông báo lỗi của PAM

1. Soạn thảo /etc/syslog.conf để bổ sung bất kỳ mục nào trong các mục thông báo lỗi PAM sau đây:

- auth.alert - các thông báo về các điều kiện sẽ được sửa tức thì
- auth.crit - các thông báo khẩn
- auth.err - các thông báo lỗi
- auth.info - các thông báo về thông tin
- auth.debug - các thông báo gỡ rối

2. Khởi động tiện ích syslog hay gửi tín hiệu SIGHUP tới nó để kích hoạt thông báo lỗi PAM.

Ví dụ - Kích hoạt thông báo lỗi PAM

Ví dụ sau hiển thị tất cả các thông báo báo động trên thiết bị điều khiển. Các thông báo khẩn được gửi cho root. Các thông báo về thông tin và gỡ rối được bổ sung vào

file /var/log/pamlog.

```
auth.alert  /dev/console
auth.crit  'root'
auth.info;auth.debug  /var/log/pamlog
```

Mỗi dòng trên bản ghi gồm một nhãn thời gian, tên hệ thống sinh ra thông báo, và bản thân thông báo. File pamlog có khả năng ghi lại một số lượng lớn thông tin.

CHƯƠNG VI-SỬ DỤNG CÔNG CỤ TĂNG CƯỜNG AN TOÀN TỰ ĐỘNG

Chương này mô tả cách dùng công cụ tăng cường an toàn tự động (ASET-Automated Security Enhancement Tool) để giám sát hoặc hạn chế truy nhập tới các file hệ thống và các thư mục.

Sau đây là danh sách các chỉ dẫn từng bước trong chương này.

- "Cách chạy ASET trực tuyến" ở mục 6.2.1
- "Cách chạy ASET định kỳ" ở mục 6.2.2
- "Cách ngừng chạy ASET định kỳ" ở mục 6.2.3
- "Cách tập hợp các báo cáo trên server" ở mục 6.2.4

6.1-Công cụ tăng cường an toàn tự động (ASET)

Phần mềm hệ thống SunOS 5.7 bao hàm công cụ tăng cường an toàn tự động (ASET). ASET giúp bạn giám sát và điều khiển an toàn hệ thống bằng cách tự động thực hiện các tác vụ mà lẽ ra bạn sẽ làm theo cách thủ công.

Gói an toàn ASET cung cấp các công cụ quản trị tự động cho phép bạn điều khiển và giám sát an toàn hệ thống của bạn. Bạn đặc tả một mức an toàn - thấp, trung bình, hoặc cao - ở nơi mà ASET sẽ chạy. Ở mỗi mức cao hơn, các chức năng điều khiển file của ASET tăng lên nhằm giảm truy nhập file và siết chặt an toàn hệ thống của bạn.

Có bảy tác vụ liên quan tới ASET, mỗi tác vụ thực hiện kiểm tra và hiệu chỉnh các file hệ thống. Các tác vụ ASET siết chặt các quyền file, kiểm tra nội dung các file hệ thống quan trọng đối với những chỗ thiếu an toàn, và giám sát các vùng quyết định. ASET có thể bảo vệ một mạng bằng cách chú tâm vào các yêu cầu cơ bản của hệ thống firewall đối với hệ thống mà server giống như hệ thống gateway.

ASET dùng các file cơ bản để cấu hình. Các file cơ bản, báo cáo và những file ASET khác nằm trong thư mục /usr/aset. Những file này có thể thay đổi cho phù hợp với các yêu cầu cụ thể đối với site của bạn.

Mỗi tác vụ sinh ra một báo cáo ghi lại những điểm thiếu an toàn được phát hiện và thay đổi tác vụ đã tạo ra các file hệ thống. Khi chạy ở mức an toàn cao nhất, ASET sẽ cố gắng biến đổi tất cả những điểm thiếu an toàn của hệ thống. Nếu nó không thể hiệu chỉnh sự cố an toàn có thể xảy ra, thì ASET thông báo sự tồn tại của sự cố này.

Bạn có thể khởi hoạt phiên ASET bằng cách dùng lệnh /usr/aset trực tuyến, hoặc bạn cũng có thể cài đặt ASET để chạy định kỳ bằng cách đặt một đầu vào trong file crontab.
Các tác vụ ASET cần nhiều đĩa và có thể can thiệp vào các hoạt động thường xuyên. Để giảm tối đa ảnh hưởng tới sự thực thi hệ thống, lịch trình ASET nên chạy khi mức hoạt động của hệ thống là thấp nhất, ví dụ, cứ 24 hoặc 48 giờ một lần vào nửa đêm.

6.1.1-Các mức an toàn ASET

ASET có thể được cài đặt để hoạt động ở một trong ba mức an toàn: thấp, trung bình và cao. Ở mỗi mức cao hơn, các chức năng điều khiển file của ASET tăng lên nhằm giảm truy nhập file và nâng cao an toàn hệ thống. Các chức năng này đi từ giám sát an toàn hệ thống mà không hạn chế truy nhập file của người dùng tới việc siết chặt hơn các quyền truy nhập cho đến khi hệ thống thực sự an toàn.

Ba mức được phác thảo dưới đây:

- *An toàn thấp* - Mức này đảm bảo các thuộc tính của các file hệ thống được đặt bằng các giá trị không chuẩn. ASET thực hiện một vài kiểm tra và báo cáo về những điểm thiếu an toàn có thể xảy ra. Ở mức này, ASET không hành động và không làm ảnh hưởng tới các dịch vụ của hệ thống.
- *An toàn trung bình* - Mức này cung cấp điều khiển an toàn thích đáng cho hầu hết các môi trường. ASET thay đổi một số cài đặt các file và các tham số hệ thống, hạn chế truy nhập hệ thống làm giảm những rủi ro từ những cuộc xâm phạm an toàn. ASET thông báo những điểm thiếu an toàn và bất kể những thay đổi nào mà nó tạo ra để hạn chế truy nhập. Ở mức này, ASET không ảnh hưởng tới các dịch vụ hệ thống.
- *An toàn cao* - Mức này mang lại hệ thống an toàn cao. ASET chỉnh lý nhiều file hệ thống và các cài đặt tham số nhằm giảm tối đa các quyền truy nhập. Hầu hết các ứng dụng và các lệnh của hệ thống tiếp tục hoạt động bình thường, nhưng ở mức này, những mối quan tâm về an toàn trước hết đặt ở cách xử lý hệ thống khác.

Ghi chú - ASET không thay đổi các quyền của một file làm cho nó thiếu an toàn, trừ phi hạ thấp mức an toàn hay cố tình trở lại hệ thống với những cài đặt tồn tại chu kỳ chạy ASET.

6.1.2-Các tác vụ ASET

Phần này đề cập tới vấn đề ASET làm cái gì. Bạn nên hiểu mỗi tác vụ ASET - các mục tiêu của nó là gì, nó thực hiện những thao tác gì, nó ảnh hưởng tới những thành phần nào của hệ thống - để hiểu và sử dụng các báo cáo một cách hiệu quả.

Các file báo cáo chứa những thông báo mô tả tỉ mỉ đến mức có thể bất kỳ sự cố nào do từng tác vụ ASET phát hiện. Những thông báo này có thể giúp bạn chẩn đoán và sửa chữa những sự cố này. Tuy nhiên, sử dụng ASET thành công chỉ có trên cơ sở bạn có hiểu biết chung về quản trị hệ thống và các thành phần của hệ thống. Nếu bạn là một người giám quản mới, bạn có thể tham khảo tài liệu quản trị hệ thống SunOS 5.7 khác và các trang tập sử dụng có liên quan để trang bị cho mình về quản trị ASET.

Tiện ích taskstat nhận diện các tác vụ đã hoàn thành và những tác vụ vẫn đang thực hiện. Mỗi tác vụ đã hoàn thành sản sinh một file báo cáo. Để có mô tả đầy đủ về tiện ích taskstat, tham khảo **taskstat**(1M).

Kiểm chứng các quyền với các file hệ thống

Tác vụ này thiết lập các quyền trên những file hệ thống ở mức an toàn mà bạn thiết kế. Nó chạy khi hệ thống được cài đặt. Nếu sau đó bạn quyết định thay đổi các mức

đã thiết lập trước đây, thì thực hiện lại tác vụ này. Ở mức an toàn thấp, các quyền được đặt bằng các giá trị thích hợp với môi trường chia xẻ thông tin mở. Ở mức an toàn trung bình, các quyền được siết chặt để đưa ra mức an toàn thích đáng cho đa số các môi trường. Ở mức an toàn cao, chúng được siết chặt nhằm hạn chế khắt khe truy nhập.

Mọi biến đổi mà tác vụ này tạo ra trên các quyền với các file hệ thống hay những cài đặt tham số được ghi lại trong file tune.rpt. Mục "Tune Files" ở trong 6.1.11 trình bày một ví dụ về các file mà ASET tra cứu khi đặt các quyền.

Kiểm soát các file hệ thống
Tác vụ này kiểm tra các file hệ thống và so sánh từng file với mô tả về file đó được kê khai trong file cơ bản. File cơ bản được tạo ra lần đầu tiên khi ASET thực hiện tác vụ này. File cơ bản chứa những cài đặt file hệ thống mà checklist bắt buộc đối với mức an toàn cụ thể.

Một danh sách các thư mục mà các file của chúng cần kiểm tra được xác định đối với mỗi mức an toàn. Bạn có thể dùng danh sách mặc định, hoặc bạn có thể biến đổi nó, đặc tả các thư mục khác nhau đối với mỗi mức.

Đối với mỗi file, tiêu chuẩn sau được kiểm tra:
- Người sở hữu và nhóm
- Các bit quyền
- Kích thước và tổng kiểm tra (checksum)
- Số các liên kết
- Thời gian thay đổi lần cuối

Mọi sự không nhất quán tìm thấy đều ghi trong file cklist.rpt. File này chứa các kết quả so sánh kích thước file hệ thống, quyền và các giá trị tổng kiểm tra với file cơ bản.

Kiểm soát người dùng/nhóm
Tác vụ này kiểm tra tính nhất quán và tính toàn vẹn của các tài khoản người dùng và các nhóm đã định nghĩa trong các file passwd và group. Nó kiểm tra các file mật khẩu cục bộ và NIS hay NIS+. Những sự cố file mật khẩu NIS+ được thông báo nhưng không được hiệu chỉnh. Tác vụ này kiểm tra những vi phạm sau đây:
- Các tên hoặc ID đúp
- Những đầu vào có định dạng sai
- Các tài khoản không có mật khẩu
- Các thư mục đăng nhập vô hiệu
- Tài khoản nobody
- Mật khẩu nhóm null
- Tín hiệu cộng (+) trong file /etc/passwd trên NIS (hoặc NIS+) server

Những điều không nhất quán được ghi trong file usrgrp.rpt

Kiểm soát các file cấu hình hệ thống
Trong suốt tác vụ này, ASET kiểm tra các bảng hệ thông khác nhau, hầu hết chúng ở thư mục /etc. Các file này là:

- /etc/default/login
- /etc/hosts.equip
- /etc/inetd.conf
- /etc/aliases
- /var/adm/utmp
- /var/adm/utmpx
- /.rhosts
- /etc/vfstab
- /etc/dfs/dfstab
- /etc/ftpusers

ASET thực hiện những kiểm tra và biến đổi khác nhau trên các file này, và ghi lại tất cả các vấn đề trong file sysconf.rpt.

Kiểm tra môi trường
Tác vụ này kiểm tra các biến môi trường PATH và UMASK được đặt như thế nào đối với root và những người dùng khác trong các file /.profile, /.login và /.cshrc.

Những kết quả kiểm tra môi trường về an toàn được ghi trong file env.rpt.

Kiểm tra eeprom
Tác vụ này kiểm tra giá trị tham số an toàn eeprom nhằm bảo đảm rằng nó được đặt ở mức an toàn thích hợp. Bạn có thể đặt tham số an toàn eeprom bằng none, command, hoặc full.

ASET không thay đổi cài đặt này, nhưng đưa ra những khuyến cáo của nó trong file eeprom.rpt.

Thiết lập firewall
Tác vụ này đảm bảo hệ thống có thể sử dụng an toàn như rơle mạng. Nó bảo vệ mạng nội bộ tách khỏi các mạng công cộng bên ngoài bằng cách thiết lập hệ thống dành riêng làm firewall, xem mô tả trong "Firewall Systems" ở chương II. Hệ thống firewall tách hai mạng, mỗi mạng xem mạng kia là không tin cậy. Firewall cài đặt tác vụ làm mất hiệu lực chuyển tiếp các gói giao thức Internet (IP) và che thông tin dẫn đường từ mạng bên ngoài.

Tác vụ firewall chạy ở tất cả các mức an toàn, nhưng đưa ra hành động chỉ ở mức an toàn cao nhất. Nếu bạn muốn chạy ASET ở mức an toàn cao, nhưng nhận thấy rằng hệ thống của bạn không yêu cầu bảo vệ bằng firewall, bạn có thể loại trừ tác vụ firewall bằng cách soạn thảo file asetnev.

Mọi thay đổi tạo ra được ghi trong file firewall.rpt.

6.1.3-Ghi nhật ký thực hiện ASET

ASET sinh nhật ký thực hiện khi nó chạy trực tuyến hoặc theo nền. Theo mặc định, ASET sinh file nhật ký ở đầu ra chuẩn. Nhật ký thực hiện khẳng định ASET đã chạy trong thời gian thiết kế, và ngoài ra chứa các thông báo lỗi xử lý. Lệnh aset -n gửi nhật ký được phân phối bằng thư điện tử thẳng tới người thiết kế. Để có một danh sách các tuỳ chọn ASET đầy đủ, tham khảo **aset**(1M).

Ví dụ về file nhật ký thực hiện

```
ASET running at security level low
Machine=example;  Current time = 0325_08:00
aset: Using  /usr/aset  as  working directory

Excuting task list...
        firewall
        env
        sysconfig
        usrgrp
        tune
        cklist
        eeprom
All tasks excuted. Some background tasks may still be running.

Run  /usr/aset/util/taskstat to check their status:
$/usr/aset/util/taskstat        aset_dir
Where aset_dir is ASET's operating directory, currently=/usr/aset

When the tasks complete, the reports can be found in:
/usr/aset/reports/latest/*.rpt
You can view them by:
more  /usr/aset/reports/latest/*.rpt
```

Đầu tiên nhật ký trình bày hệ thống và thời gian ASET đã chạy. Sau đó nó liệt kê từng tác vụ khi nó khởi hoạt.

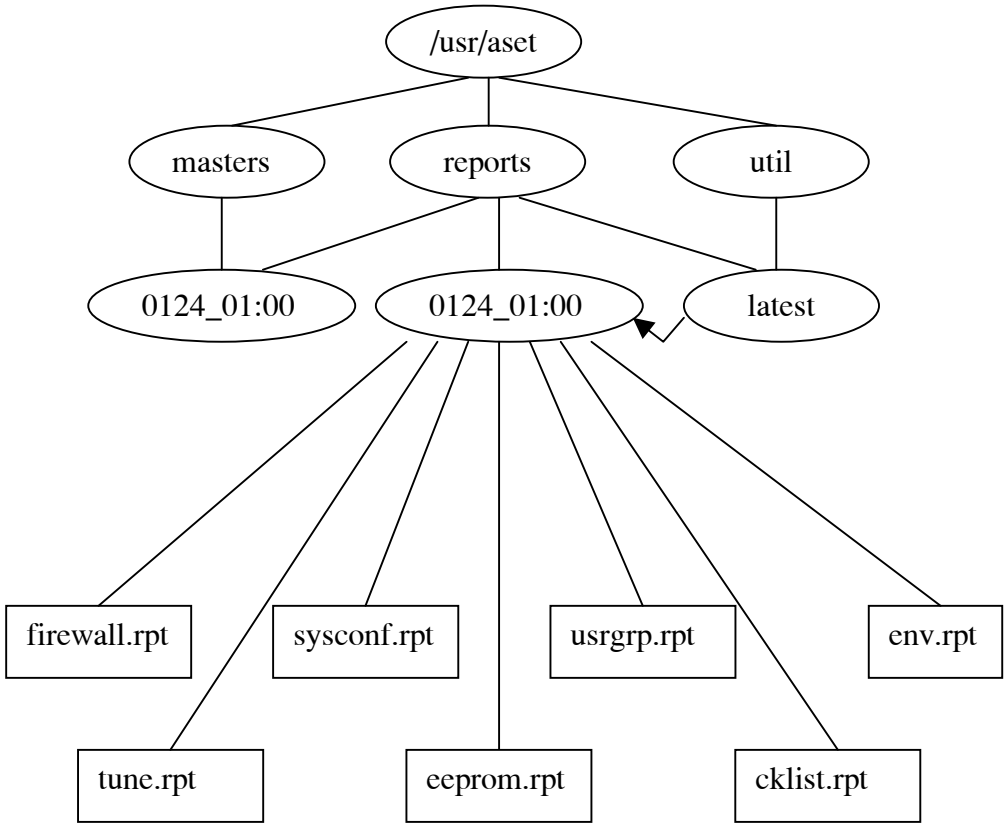
ASET gọi trình nền cho mỗi tác vụ này, những tác vụ được mô tả trong "ASET tasks" ở mục 6.1.2. Tác vụ được hiển thị trong nhật ký thực hiện khi nó khởi hoạt; điều này không có nghĩa nó đã hoàn thành. Để kiểm tra các tác vụ nền, dùng tiện ích taskstat.

6.1.4-Các báo cáo ASET

Tất cả file báo cáo sinh ra từ các tác vụ ASET nằm trong các thư mục con của thư mục /usr/aset/reports. Phần này mô tả cấu trúc của thư mục /usr/aset/reports, và cung cấp những chỉ dẫn về việc quản lý các file báo cáo.

ASET đặt các file báo cáo trong các thư mục con được đặt tên phản ánh thời gian và ngày sinh ra báo cáo. Điều này cho phép bạn lưu giữ dấu vết của các bản ghi trạng thái hệ thống biến đổi giữa các xử lý ASET một cách có thứ tự. Bạn có thể theo dõi và so sánh các báo cáo này để xác định sự an toàn đầy đủ của hệ thống của bạn.

Hình 6-1 đưa ra một ví dụ về cấu trúc thư mục reports.



Hình 6-1 Cấu trúc thư mục reports

Hai thư mục báo cáo được trình bày trong ví dụ này:

- 0124_01:00
- 0125_01:00

Tên các thư mục con chỉ rõ ngày và giờ các báo cáo được sinh ra. Mỗi tên thư mục báo cáo có định dạng như sau:

monthdate_hour:minute

trong đó month, date, hour, và minute đều là các số có hai chữ số. Ví dụ, 0125_01:00 biểu diễn January 25, lúc 1 giờ sáng.

Mỗi cặp thư mục báo cáo gồm một tập các báo cáo sinh ra từ một lần thực hiện ASET.

Thư mục latest là một liên kết ký hiệu luôn chỉ thư mục con chứa các báo cáo gần nhất. Vì thế, nhìn vào các báo cáo gần nhất mà ASET đã sinh ra, bạn có thể đi tới thư mục /usr/aset/reports/latest. Có một file báo cáo trong thư mục này cho mỗi tác vụ mà ASET thực hiện trong lần thực hiện gần đây nhất của nó.

Định dạng các file báo cáo

Mỗi file báo cáo được đặt tên sau khi tác vụ sinh ra nó. Xem bảng 6-1 để có danh sách các tác vụ và các báo cáo của chúng.

Bảng 6-1 Các tác vụ ASET và các báo cáo kết quả

Các tác vụ	**Báo cáo**
Điều chỉnh các quyền của file hệ thống (tune)	tune.rpt
Danh sách kiểm tra các file hệ thống (cklist)	cklist.rpt
Kiểm tra người dùng/nhóm (usrgrp)	usrgrp.rpt
Kiểm tra các file cấu hình hệ thống (sysconf)	sysconf.rpt
Kiểm tra môi trường (env)	env.rpt
Kiểm tra eeprom (eeprom)	eeprom.rpt
Cài đặt firewall (firewall)	firewall.rpt

Trong mỗi file báo cáo, các thông báo được đặt giữa dòng tiêu đề bắt đầu và kết thúc. Đôi khi một tác vụ kết thúc sớm; ví dụ, khi một thành phần của ASET ngẫu nhiên bị loại bỏ hoặc tổn thương. Trong hầu hết các trường hợp, file báo cáo sẽ chứa một thông báo gần thông báo cuối chỉ rõ nguyên nhân kết thúc sớm.

Sau đây là một file báo cáo mẫu, usrgrp.rpt

```
*** Begin User and Group Checking ***

Checking /etc/passwd ...
Warning! Password file, line 10, no passwd
:sync::1:1::/:/bin/sync
..end user check; starting group check ...
Checking  /etc/group ...
*** End User and Group Checking ***
```

Kiểm tra các file báo cáo

Sau lần chạy hay cấu hình lại ASET đầu tiên, bạn nên kiểm tra sít sao các file báo cáo. (Cấu hình lại bao gồm việc biến đổi file asetenv hay các file cơ bản trong thư mục con masters, hoặc thay đổi mức an toàn mà ASET hoạt động). Các báo cáo ghi lại mọi lỗi được đưa vào khi bạn cấu hình lại. Bằng cách theo dõi chặt chẽ các báo cáo, bạn có thể đối phó và giải quyết các sự cố khi chúng nảy sinh.

Đối chiếu các file báo cáo

Sau một chu kỳ theo dõi các file báo cáo mà không có những thay đổi cấu hình hay cập nhật hệ thống, bạn có thể thấy nội dung của các báo cáo mở đầu ổn định và nó chứa ít thông tin không momg đợi nếu có. Bạn có thể dùng tiện ích diff để đối chiếu các báo cáo.

6.1.5-Các file cơ bản ASET

Các file cơ bản của ASET, tune.high, tune.low, tune.med, và uid_aliases được đặt trong thư mục /usr/aset/masters. ASET dùng các file cơ bản để định nghĩa các mức an toàn.

Tune Files

Các tệp cơ bản tune.low, tune.med và tune.high định ra các mức an toàn có thể của ASET. Chúng chỉ ra các thuộc tính của các tệp hệ thống tại mỗi mức và định ra các mức an toàn.

File uid_aliases
File uid_aliases chứa một danh sách nhiều tài khoản người dùng dùng chung ID. Thông thường, ASET cảnh báo về nhiều tài khoản người dùng như thế vì thực tế này làm giảm khả năng giải trình. Bạn có thể cho phép loại trừ quy tắc này bằng cách liệt kê những loại trừ trong file uid_aliases. ASET không đưa ra những mục vào với các user ID đúp nếu các mục này được mô tả trong file uid_aliases.

Cần tránh có nhiều tài khoản người dùng (các mục vào mật khẩu) dùng chung user ID. Bạn nên xem xét các phương pháp khác thực hiện mục tiêu của bạn. Ví dụ, nếu bạn định cho một vài người dùng dùng chung một tập quyền, thì bạn có thể tạo một tài khoản nhóm. Dùng chung các user ID là phương sách cuối cùng của bạn, chỉ dùng khi thật cần thiết và khi các phương pháp khác không đạt tới các mục tiêu của bạn.

Bạn có thể dùng biến môi trường UID_ALIASES để đặc tả file các bí danh lần lượt kế tiếp nhau. Mặc định là /usr/aset/masters/uid_aliases.

Các file Checklist
Các file cơ bản dùng phù hợp với các hệ thống sắp xếp danh sách kiểm tra được sinh ra khi bạn thucj hiện ASET lần đầu, hay khi bạn chạy ASET sau khi bạn thay đổi mức an toàn.

Những file kiểm tra bằng tác vụ này được định nghĩa phù hợp với các biến môi trường sau:
- CKLISTPATH_LOW
- CKLISTPATH_MED
- CKLISTPATH_HIGH

6.1.6- File môi trường ASET (asetenv)
File môi trường, asetenv, chứa một danh sách các biến tác động tới các tác vụ ASET. Các biến có thể bị thay đổi làm giảm nhẹ hiệu quả ASET.

6.1.7-Cấu hình ASET
Phần này đề cập tới cách cấu hình ASET và môi trường mà nó hoạt động.

ASET đòi hỏi quản trị và cấu hình tối thiểu, và trong đa số trường hợp, bạn có thể chạy nó với các giá trị mặc định. Tuy nhiên, bạn có thể điều chỉnh chính xác một số tham số ảnh hưởng tới hoạt động và cách xử lý của ASET làm tăng tối đa lợi ích của nó. Trước khi thay đổi các giá trị mặc định, bạn nên hiểu ASET làm việc như thế nào, và nó tác động ra sao tới các thành phần trong hệ thống của bạn.

ASET dựa vào bốn file cấu hình để điều khiển cách xử lý của các tác vụ của nó:
- /usr/aset/asetenv
- /usr/aset/masters/tune.low
- /usr/aset/masters/tune.med
- /usr/aset/masters/tune.high

Thay đổi file môi trường (asetenv)
File /usr/aset/asetenv có hai phần chính:

- Phần các tham số có thể cấu hình người dùng
- Phần các biến môi trường bên trong

Bạn có thể biến đổi phần các tham số có thể cấu hình người dùng. Tuy nhiên, các cài đặt trong phần các biến môi trường bên trong chỉ để dùng nội bộ và không nên thay đổi.

Bạn có thể soạn thảo các mục vào trong phần các tham số cấu hình người dùng để:

- Chọn các tác vụ để chạy
- Đặc tả các thư mục cho tác vụ checklist
- Lập kế hoạch thực hiện ASET
- Đặc tả file các bí danh
- Mở rộng kiểm tra tới các bảng NIS+

Chọn những tác vụ nào để chạy: TASKS
Mỗi tác vụ ASET thực hiện giám sát một vùng an toàn hệ thống cụ thể. Trong hầu hết các môi trường hệ thống, tất cả các tác vụ đều cần thiết để bảo đảm sự giám sát an toàn tương xứng. Tuy nhiên, bạn có thể quyết định loại trừ một hoặc nhiều tác vụ.

Ví dụ, tác vụ firewall chạy ở tất cả các mức an toàn, nhưng chỉ đưa ra hành động ở mức an toàn cao. Bạn có thể muốn chạy ASET ở mức an toàn cao, nhưng không yêu cầu bảo vệ bằng firewall.

Có thể cài đặt ASET chạy ở mức cao mà không có tính năng firewall bằng cách soạn thảo danh sách các biến môi trường TASKS trong file asetenv. Theo mặc định, danh sách TASKS chứa tất cả các tác vụ ASET. (Ví dụ trình bày bên dưới). Để xoá một tác vụ, loại bỏ cài đặt tác vụ khỏi file. Trong trường hợp này, bạn sẽ xoá biến môi trường firewall khỏi danh sách. Lần tiếp theo chạy ASET, tác vụ bị loại trừ sẽ không được thực hiện.

```
TASKS=" env  sysconfig  usrgrp  tune  cklist  eeprom  firewall"
```

Đặc tả các thư mục cho tác vụ Checklist: CKLISTPATH
Các file hệ thống kiểm tra thuộc tính các file ở các thư mục hệ thống được chọn trong những lần kiểm tra. Bạn xác định những thư mục để kiểm tra bằng cách dùng các biến môi trường checklist path sau:

- CKLISTPATH_LOW
- CKLISTPATH_MED
- CKLISTPATH_HIGH

Biến CKLISTPATH_LOW xác định các thư mục được kiểm tra ở mức an toàn thấp. Các biến môi trường CKLISTPATH_MED và CKLISTPATH_HIGH có chức năng tương tự đối với các mức an toàn trung bình và cao.

Danh sách thư mục xác định qua một biến ở mức an toàn thấp sẽ là tập con của danh sách thư mục xác định ở mức cao hơn tiếp theo. Ví dụ, tất cả các thư mục trong

CKLISTPATH_LOW sẽ bao hàm trong CKLISTPATH_MED, và tất cả các thư mục trong CKLISTPATH_MED sẽ bao hàm trong CKLISTPATH_HIGH.

Những lần kiểm tra thực hiện trên các thư mục này không đệ quy; ASET chỉ kiểm tra những thư được liệt kê rõ ràng trong biến. Nó không kiểm kiểm tra các thư mục con của chúng.

Bạn có thể soạn thảo các định nghĩa biến này để bổ sung hoặc xoá các thư mục mà bạn muốn ASET kiểm tra. Lưu ý rằng những checklist này chỉ có lợi đối với các file hệ thống mà thông thương không thay đổi hàng ngày. Ví dụ, thư mục gốc của người dùng nói chung là quá động để làm một ứng cử viên cho checklist.

Lập kế hoạch thực hiện ASET: PERIODIC_SCHEDULE
Khi bạn khởi động ASET, bạn có thể trực tiếp khởi động nó , hoặc dùng tuy chọn -p để yêu cầu chạy các tác vụ ASET tại một thời điểm và chu kỳ đã đặt. Bạn có thể chạy ASET định kỳ, tại thời điểm đó hệ thống yêu cầu có đèn báo. Ví dụ, ASET tra cứu PERIODIC_SCHEDULE để xác định các tác vụ ASET thường xuyên thực hiện như thế nào, và vào thời điểm nào thì chạy chúng. Để có những chỉ dẫn chi tiết về cách cài đặt ASET chạy định kỳ, xem mục "How to Run ASET Periodically" 6.2.2.

Định dạng của PERIODIC_SCHEDULE theo sau định dạng những mục vào crontab. Xem crontab(1) để có thông tin đầy đủ.

Đặc tả file bí danh: UID_ALIASES
Biến UID_ALIASES mô tả file các bí danh liệt kê các userID dùng chung. Mặc định là /usr/aset/masters/uid_aliases.

Kiểm tra mở rộng đối với các bảng NIS+: YPCHECK
Biến môi trường YPCHECK đặc tả ASET có kiểm tra các bảng của file cấu hình hệ thống hay không. YPCHECK là một biến logic; bạn có thể gán true hoặc false cho nó. Giá trị mặc định là false, không kiểm tra bảng NIS+.

Để hiểu biến này làm việc như thế nào, hãy xét ảnh hưởng của nó trên file passwd. Khi biến này được đặt là false, ASET kiểm tra file passwd cục bộ. Khi nó được đặt là true, tác vụ còn kiểm tra file passwd NIS+ của domain hệ thống.

Chú ý - Mặc dầu ASET tự động sửa chữa các bảng cục bộ, nhưng nó chỉ thông báo những sự cố có thể xảy ra trong các bảng NIS+; nó không thay đổi chúng.

Biến đổi các file điều chỉnh (tune)
ASET dùng ba file điều chỉnh cơ bản, tune.low, tune.med, và tune.high. ASET dùng chúng để làm giảm hoặc siết chặt quyền truy nhập tới các file hệ thống quan trọng. Các file cơ bản này được đặt trong thư mục /usr/aset/masters, và có thể biến đổi chúng cho phù hợp với môi trường của bạn. Để có thêm thông tin, xem "Tune Files" thuộc mục 6.1.11.

File tune.low đặt các quyền bằng các giá trị thích hợp với những cài đặt hệ thống mặc định. File tune.med hạn chế hơn nữa các quyền này và gồm những mục vào không có trong tune.low. File tune.high hạn chế các quyền còn nhiều hơn nữa.

Chú ý - Biến đổi những cài đặt trong file điều chỉnh bằng cách bổ sung hoặc xoá đi những đầu vào file. Đặt một quyền bằng giá trị ít hạn chế hơn cài đặt hiện tại không có tác dụng; các tác vụ ASET không buông lỏng các quyền trừ phi bạn hạ mức an toàn hệ thống của bạn xuống mức thấp hơn.

6.1.8-Khôi phục các file hệ thống do ASET biến đổi

Khi ASET được thực hiện lần đầu tiên, nó ghi và lưu trữ các file hệ thống ban đầu. Tiện ích aset.restore sắp đặt lại các file này. Nó cũng đưa vào lịch trình ASET, nếu hiện tại nó được lên lịch để thực hiện định kỳ. Tiện ích aset.restore được đặt trong thư mục /usr/aset, thư mục điều hành ASET.

Những thay đổi tạo ra với các file hệ thống bị mất khi bạn chạy aset.restore.

Bạn nên dùng aset.restore:

- Khi bạn muốn loại bó những thay đổi ASET và khôi phục hệ thống ban đầu. Khi bạn muốn đình hoạt ASET lâu dài, bạn có thể loại bỏ nó khỏi lịch trình cron nếu lệnh aset đã được bổ sung vào crontab của root trước đó. Đối với các thư mục dùng cron để loại trừ thực hiện tự động, xem mục "How to Stop Running ASET Perodically" 6.2.3.
- Sau một thời gian ngắn thử nghiệm ASET, khôi phục lại trạng thái hệ thống ban đầu.
- Khi một số chức năng tiện ích của hệ thống làm việc không hoàn hảo và bạn ngờ rằng ASET là nguyên nhân của vấn đề.

6.1.9-Điều hành mạng dùng hệ thống NFS

Nói chung, ASET được dùng theo kiểu trạm độc lập, ngay cả trên hệ thống là một phần của mạng. Với tư cách người giám quản hệ thống đối với hệ thống độc lập của bạn, bạn có trách nhiệm về an toàn hệ thống của bạn, chạy và quản lý ASET để bảo vệ hệ thống của bạn.

Bạn cũng có thể dùng ASET trong môi trường phân tán NFS. Với tư cách người giám quản mạng, bạn có trách nhiệm cài đặt, chạy và quản lý các tác vụ quản trị đối với tất cả client của bạn. Để tiện quản lý ASET đi qua một vài hệ thống client, bạn có thể đưa ra những thay đổi cấu hình được áp dụng tổng thể cho tất cả client, loại trừ nhu cầu của bạn đăng nhập tới từng hệ thống để lặp lại tiến trình.

Khi quyết định cách cài đặt ASET trên các hệ thống mạng của bạn, bạn nên xem xét vấn đề bạn muốn những người dùng điều khiển an toàn trên các hệ thống riêng của họ đến đâu, và bạn muốn tập trung trách nhiệm điều khiển an toàn đến đâu.

Cung cấp cấu hình tổng thể cho mỗi mức an toàn

Một trường hợp nảy sinh khi bạn muốn cài đặt nhiều cấu hình mạng. Ví dụ, bạn có thể muốn thiết lập một cấu hình cho các client được thiết kế với mức an toàn thấp, một cấu hình khác với mức trung bình, và một cấu hình khác nữa với mức cao.

Nếu bạn cần tạo một cấu hình mạng ASET riêng rẽ cho mỗi mức an toàn, bạn có thể tạo ba cấu hình ASET trên server - mỗi cấu hình cho một mức. Bạn sẽ đưa ra từng cấu hình cho các client với mức an toàn thích hợp. Một số thành phần ASET là chung cho tất cả ba cấu hình có thể được dùng chung bằng các liên kết.

Thu thập các báo cáo ASET
Bạn không chỉ có thể tập trung các thành phần ASET trên server mà các client có hoặc không có đặc quyền superuser được truy nhập, mà bạn còn có thể thiết lập một thư mục trung tâm trên server để thu thập tất cả các báo cáo do các tác vụ chạy trên các client khác sinh ra. Để có những chỉ dẫn về cài đặt cơ chế thu thập, xem mục "How to Collect Reports on a Server" 6.2.4.

Thiết lập tập hợp các báo cáo trên server cho phép bạn xem lại các báo cáo của tất cả client từ một vị trí. Bạn có thể dùng cách này để biết một client có đặc quyền superuser hay không. Như một sự lựa chọn, bạn có thể cho phép có thư mục các báo cáo trên hệ thống cục bộ khi bạn muốn những người dùng theo dõi các báo cáo ASET riêng của họ.

6.1.10-Các biến môi trường
Bảng 6-2 liệt kê các biến môi trường ASET và các giá trị mà chúng mô tả.

Bảng 6-2 Các biến môi trường và ý nghĩa của chúng

Biến môi trường	**Mô tả**
ASETDIR (Xem bên dưới)	Thư mục làm việc của ASET
ASETSECLEVEL (Xem bên dưới)	Mức an toàn
PERIOD_SCHEDULE	Lịch trình chu kỳ
TASKS	Các tác vụ chạy
UID_ALIASES	File các bí danh
YPCHECK	Mở rộng kiểm tra NIS và NIS+
CKLISTPATH_LOW	Danh sách thư mục với mức an toàn thấp
CKLISTPATH_MED	Danh sách thư mục với mức an toàn trung bình
CKLISTPATH_HIGH	Danh sách thư mục với mức an toàn cao

Các biến môi trường liệt kê bên dưới được tìm thấy trong file /usr/aset/asetenv. Các biến ASETDIR và ASETSECLEVEL là tuỳ chọn và có thể đặt chỉ qua một shell bằng cách dùng lệnh aset. Có thể đặt các biến môi trường khác bằng cách soạn thảo file. Các biến được mô tả dưới đây.

Biến ASETDIR
ASETDIR đặc tả thư mục làm việc của ASET.

Từ C Shell, gõ:

```
% setenv ASETDIR pathname
```

Từ Bourne Shell hoặc Korn Shell, gõ:

```
$ ASETDIR=pathname
$ export ASETDIR
```

Đặt pathname là tên đường dẫn đầy đủ của thư mục làm việc của ASET

Biến ASETSECLEVEL
Biến ASETSECLEVEL mô tả mức an toàn mà các tác vụ ASET được thực hiện.

Từ C Shell, gõ:

```
% setenv ASETSECLEVEL level
```

Từ Bourne Shell hoặc Korn Shell, gõ:

```
$ ASETDIR=level
$ export ASETDIR
```

Trong các câu lệnh ở trên, level có thể được đặt một trong các giá trị sau:

low	Mức an toàn thấp
med	Mức an toàn trung bình
high	Mức an toàn cao

Biến PERIODIC_SCHEDULE
Giá trị của PERIODIC_SCHEDULE theo cùng định dạng như file crontab. Mô tả giá trị biến bằng một xâu ký tự gồm năm trường đặt trong cặp dấu nháy, mỗi trường phân cách nhau bởi một dấu cách.

```
"minutes hours day-of-month month day-of-week"
```

Bảng 6-3 Các giá trị biến PERIODIC_SCHEDULE

Biến	Giá trị
minutes hours	Đặc tả thời gian bắt đầu bằng số phút theo giờ (0-59) và giờ (0-23)
day-of-month	Đặc tả ngày trong tháng khi ASET sẽ chạy, dùng các giá trị từ 1 đến 31
month	Đặc tả tháng trong năm khi ASET sẽ chạy, dùng các giá trị từ 1 đến 12
day-of-week	Đặc tả ngày trong tuần khi ASET sẽ chạy, dùng các giá trị từ 0 đến 6; Chủ nhật là ngày 0 theo lược đồ này

Các quy tắc áp dụng như sau:

- Bạn có thể đặc tả danh sách các giá trị cho bất kỳ trường nào, mỗi giá trị phân cách bằng dấu phẩy.
- Bạn có thể đặc tả giá trị bằng một số, hoặc bạn có thể đặc tả bằng một khoảng; nghĩa là, một cặp số nối với nhau bằng dấu nối. Một khoảng chỉ ra rằng các tác vụ ASET sẽ được thực hiện trong tại mỗi thời điểm trong khoảng.
- Bạn có thể đặc tả giá trị của một trường bất kỳ bằng dấu *. Dấu * mô tả tất cả các giá trị có thể của trường, bao gồm tất cả.

Đầu vào mặc định của biến PERIODIC_SCHEDULE dẫn đến ASET thực hiện vào 12 giờ đêm hàng ngày:

```
PERIODIC_SCHEDULE="0  0  *  *  *"
```

Biến TASKS
Biến TASKS kê khai các tác vụ mà ASET thực hiện. Mặc định là danh sách tất cả bảy tác vụ:

```
TASKS="env  sysconfig  usrgrp  tune  cklist  eeprom  firewall"
```

Biến UID_ALIASES
Biến UID_ALIASES đặc tả file các bí danh. Nếu có, ASET tra cứu file này để có danh sách các bí danh phức tạp cho phép. Định dạng là UID_ALIASES=*pathname*. *pathname* là tên đường dẫn đầy đủ của file các bí danh.

Mặc định là:

```
UID_ALIASES=${ASETDIR}/masters/uid_aliases
```

Biến YPCHECK
Biến YPCHECK mở rộng tác vụ kiểm tra các bảng hệ thống bao gồm các bảng NIS hoặc NIS+. Nó là biến logic, có thể đặt bằng hoặc true hoặc false.

Mặc định là false, hạn chế việc kiểm tra ở các bảng hệ thống cục bộ:

```
YPCHECK=false
```

Biến CKLISTPATH_level
Ba biến đường dẫn checklist kê khai các thư mục được kiểm tra bằng tác vụ checklist. Các định nghĩa sau của các biến được đặt mặc định; chúng minh hoạ mối quan hệ giữa các biến ở các mức khác nhau:

```
CKLISTPATH_LOW=${ASETDIR}/tasks:${ASETDIR}/util:${ASETDIR}/masters
:/etc
CKLISTPATH_MED=${CKLISTPATH_LOW:/usr/bin:/usr/ucb
CKLISTPATH_HIGH=${CKLISTPATH_MED}:/usr/lib:/sbin:/usr/sbin:/usr/ucblib
```

Các giá trị của các biến môi trường đường dẫn checklist tương tự với các giá trị này của các biến đường dẫn khung, trong đó chúng là các danh sách tên thư mục ngăn cách bằng dấu hai chấm (:). Bạn dùng dấu bằng (=) để kết nối tên biến với giá trị của nó.

6.1.11-Các ví dụ file ASET
Phần này có các ví dụ về một số file ASET bao gồm các file điều chỉnh và file các bí danh.

Các file điều chỉnh
ASET duy trì ba file điều chỉnh. Định dạng đầu vào trong tất cả ba file điều chỉnh được mô tả trong bảng 6-4

Bảng 6-4 Định dang đầu vào file điều chỉnh

Đầu vào	**Mô tả**
pathname	Tên đường dẫn đây đủ của file
mode	Số năm chữ số biểu diễn cài đặt quyền
owner	Người sở hữu file
group	Nhóm sở hữu file
type	Kiểu file

Các quy tắc áp dụng như sau:

- Bạn có thể dùng các ký tự thay thế khung hợp lệ trong tên đường dẫn của các tham chiếu phức tạp, chẳng hạn dấu * và dấu ?. Xem **sh**(1) để có thêm thông tin.
- *mode* biểu diễn giá trị hạn chế tối thiểu. Nếu cài đặt hiện tại có sẵn nhiều hạn chế hơn giá trị được mô tả, thì ASET không nới lỏng những cài đặt quyền. Ví dụ, nếu giá trị mô tả là 00777, thì quyền sẽ giữ nguyên không thay đổi, vì 00777 luôn ít hạn chế hơn bất kỳ cái gì hiện được cài đặt.

  Đây là cách ASET thực hiện cài đặt mode, trừ phi mức an toàn sẽ bị hạ xuống hoặc bạn sẽ loại bỏ ASET. Khi bạn giảm mức an toàn từ mức mà nó có trong lần thực hiện trước đó, hoặc khi bạn muốn khôi phục các file hệ thống về trạng thái mà chúng có trước khi ASET được thực hiện lần đầu tiên, ASET nhận biết bạn đang làm gì và giảm mức bảo vệ.
- Bạn phải dùng các tên của *owner* và *group* thay cho các ID bằng số.
- Bạn có thể dùng dấu chấm hỏi (?) ở vị trí của *owner*, *group* và *type* để ngăn chặn ASET khỏi việc thay đổi các giá trị đang tồn tại của các tham số này.
- *type* có thể là symlink (liên kết ký hiệu), thư mục, hay file (mọi thứ còn lại).
- Các file điều chỉnh mức an toàn cao đặt lại các quyền file có ít hạn chế nhất so với chúng có ở các mức an toàn thấp hơn. Ngoài ra, ở các mức cao hơn, các file bổ sung được thêm vào danh sách.
- Một file có thể tương thích với nhiều đầu vào file điều chỉnh. Ví dụ, etc/passwd tương thích với các đầu vào etc/pass* và /etc/*.
- Ở đâu hai đầu vào có các quyền khác nhau, thì quyền file được đặt bằng giá trị hạn chế nhất. Trong ví dụ sau, quyền của /etc/passwd sẽ được đặt bằng 00755 có nhiều hạn chế hơn 00755 và 00770.

```
/etc/pass*      00755   ?   ?   file
          /etc/*     00770    ?   ?   file
```

- Nếu hai đầu vào có các định danh owner hoặc group khác nhau, thì đầu vào cuối đưa ra quyền ưu tiên. Ví dụ sau trình bày một vài dòng đầu tiên của file tune.low.

```
/  02755  root  root  directory
/bin  00777  root  bin  symlink
/sbin  02775  root  sys  directory
/usr/sbin  02775  root  bin  directory
/etc  02755  root  sys  directory
/etc/chroot  00777  bin  bin  symlink
```

File các bí danh
File các bí danh chứa một danh sách các bí danh dùng cùng userID.

Mỗi đầu vào ở dạng này:
uid=alias1=alias2=alias3=...
uid userID dùng chung
aliasn Tài khoản người dùng dùng chung userID.

Ví dụ, đầu vào sau kê khai userID 0 được sysadm và root dùng chung:
0=root=sysadm

6.2-Chạy ASET
Phần này mô tả cách chạy ASET trực tuyến hoặc định kỳ.

6.2.1-Cách chạy ASET trực tuyến
1. Trở thành superuser
2. Chạy ASET trực tuyến bằng cách dùng lệnh aset

```
# /usr/aset/aset -l level -d pathname
```

level Đặc tả mức an toàn. Các giá trị hợp lệ là low, medium, hoặc high. Đặt mặc định là low. Xem mục "ASET Security Levels" để có thông tin chi tiết về các mức an toàn.
pathname Đặc tả thư mục làm việc của ASET. Mặc định là /usr/aset.

3. Kiểm chứng ASET đang chạy bằng cách xem nhật ký thực hiện ASET được hiển thị trên màn hình.

Thông báo nhật ký thực hiện cho thấy những tác vụ nào được chạy.

Ví dụ - Cách chạy ASET trực tuyến
Ví dụ sau chạy ASET ở mức an toàn thấp với thư mục làm việc mặc định.

```
# /usr/aset/aset -l low
====== ASET Excution Log ======

ASET running at security level low

Machine = jupiter;  Current time = 0111_09:26

aset: Using  /usr/aset as working directory

Excuting task list ...
firewall
env
sysconf
usrgrp
tune
cklist
eeprom
```

```
All tasks executed. Some background tasks may still be running.

Run  /usr/aset/util/taskstat to check their status:
   /usr/aset/util/taskstat  [aset_dir]

where aset_dir is ASET's operating
directory, currently=/usr/aset.

When the tasks complete, the reports can be found in:
  /usr/aset/reports/latest/*.rpt

You can view them by:
 more  /usr/aset/reports/latest/*.rpt
```

6.2.2-Cách chạy ASET định kỳ

1. Trở thành superuser
2. Nếu cần, thiết lập thời gian mà bạn muốn ASET chạy định kỳ.

Bạn nên cho ASET chạy khi hệ thống yêu cầu có đèn báo. Biến môi trường PERIODIC_SCHEDULE trong file /usr/aset/asetenv được dùng để cài đặt thời gian cho ASET chạy định kỳ. Theo mặc định, thời gian được đặt vào nửa đêm 24 giờ hàng ngày.

Nếu bạn muốn đặt thời gian khác, soạn thảo biến PERIODIC_SCHEDULE trong file /usr/aset/asetenv. Xem mục "PERIODIC_SCHEDULE Variable" để có thông tin chi tiết về cách đặt biến PERIODIC_SCHEDULE .

3. Bổ sung một mục vào file crontab, dùng lệnh aset

```
# /usr/aset/aset  -p
```

-p Chèn một dòng vào file crontab mà ASET bắt đầu chạy tại thời điểm được xác định bằng biến môi trường PERIODIC_SCHEDULE trong file /usr/aset/asetenv.

4. Hiển thị đầu vào crontab để kiểm chứng khi ASET sẽ chạy.

```
# crontab  -l  root
```

6.2.3-Cách ngừng chạy ASET định kỳ

1. Trở thành superuser
2. Soạn thảo file crontab

```
# crontab  -e  root
```

3. Xoá đầu vào ASET
4. Ghi lại những thay đổi và thoát.
5. Hiển thị đầu vào crontab để kiểm chứng đầu vào ASET bị xoá.

```
# crontab  -l  root
```

6.2.4-Cách tập hợp các báo cáo trên server

1. Trở thành superuser

2. Thiết lập một thư mục trên server:
a. Chuyển tới thư mục /usr/aset

```
mars#  cd  /usr/aset
```

b. Tạo thư mục rptdir

```
mars#  mkdir  rptdir
```

c. Chuyển tới thư mục rptdir và tạo thư mục client_rpt.

```
mars#  cd  rptdir
mars#  mkdir  client_rpt
```

d. Bước này tạo thư mục con (*client_rpt*) cho client. Lặp lại bước này đối với mỗi client mà bạn cần thu thập báo cáo.

Ví dụ sau tạo thư mục all_reports, và các thư mục con pluto_rpt và neptune_rpt.

```
mars#  cd  /usr/aset
mars#  mkdir  all_reports
mars#  cd  all_reports
mars#  mkdir  pluto_rpt
mars#  mkdir  neptune_rpt
```

3. Bổ sung các thư mục client_rpt vào file /etc/dfs/dfstab
Các thư mục sẽ có tuỳ chọn đọc/ghi.

Ví dụ, các mục sau trong dfstab được chia xẻ với các quyền đọc/ghi

```
share  -F  nfs  -o  rw=pluto  /usr/aset/all_reports/pluto_rpt
share  -F  nfs  -o  rw=neptune  /usr/aset/all_reports/neptune_rpt
```

4. Làm cho tài nguyên trong file dfstab khả dụng với các client

```
#  shareall
```

5. Trên mỗi client, gán thư mục con client từ server ở tại điểm gán, /usr/aset/masters/reports

```
#  mount  server:/usr/aset/client_rpt  /usr/aset/masters/reports
```

6. Soạn thảo file /etc/vfstab để tự động thiết lập thư mục khi khởi động.
Mục mẫu sau trong /etc/vfstab trên neptune cho thấy thư mục được gán từ mars, /usr/aset/all_reports/neptune_rpt, và điểm gán trên neptune, /usr/aset/reports. Tại thời điểm khởi động, các thư mục kê khai trong vfstab được tự động gán.

```
mars:/usr/aset/all_reports/neptune_rpt  /usr/aset/reports  nfs  -  yes
hard
```

6.3-Sửa chữa các sự cố ASET

Phần này đưa ra các thông báo lỗi do ASET sinh ra.

ASET failed: no mail program found.

Nguyên nhân lỗi xảy ra	**Cách khắc phục**
ASET được lệnh gửi nhật ký thực hiện tới người dùng, nhưng không tìm thấy chương trình mail.	Cài đặt chương trình mail

Usage: aset [-n user[@host]] in /bin/mail or /usr/ucb/mail. Cannot decide current and previous security levels.

Nguyên nhân lỗi xảy ra	**Cách khắc phục**
ASET không thể xác định các mức an toàn là gì những lần gọi hiện tại và trước đây	Đảm bảo mức an toàn hiện tại được đặt hoặc thông qua tuỳ chọn dòng lệnh hoặc biến môi trường ASETDIR/archives/asetseclevel.arch phản ánh đúng mức an toàn trước đó. Nếu các giá trị này không được đặt hoặc không đúng, thì mô tả chúng

ASET working directory undefined. To specify, set ASETDIR environment variable or use command line option -d ASET startup unsuccessful.

Nguyên nhân lỗi xảy ra	**Cách khắc phục**
Thư mục làm việc (điều hành) ASET không được định nghĩa, hoặc định nghĩa sai.	Dùng biến môi trường ASETDIR hoặc tuỳ chọn dòng lệnh -d để mô tả nó một cách đúng đắn, và khởi động lại ASET.

ASET working directory $ASETDIR missing. ASET startup unsuccessful.

Nguyên nhân lỗi xảy ra	**Cách khắc phục**
Thư mục làm việc (điều hành) ASET không được định nghĩa, hoặc định nghĩa sai. Điều này có thể là do biến ASETDIR hay tuỳ chọn dòng lệnh -d tham chiếu tới một thư mục không tồn tại.	Đảm bảo thư mục đúng - nghĩa là, thư mục chứa hệ thống phân cấp thư mục ASET - được tham chiếu tới một cách đúng đắn.

Cannot expand $ASETDIR to full pathname

Nguyên nhân lỗi xảy ra	**Cách khắc phục**
ASET không thể mở rộng tên thư mục do biến ASETDIR hoặc tuỳ chọn dòng lệnh -d đưa ra thành tên đường dẫn đầy đủ.	Đảm bảo rằng tên thư mục được đưa ra đúng đắn, và nó tham chiếu tới một thư mục đang tồn tại mà người dùng đã truy nhập.

aset: invalid/undefined security level To specify, set ASETSECLEVEL environment variable or use command line option -l, with argument= low/med/high.

Nguyên nhân lỗi xảy ra	**Cách khắc phục**
Mức an toàn không được định nghĩa hoặc nó được định nghĩa không đúng. Chỉ các giá trị low, med, hoặc high được chấp nhận.	Dùng biến ASETSECLEVEL hoặc tuy chọn dòng lệnh -l đặc tả một trong ba giá trị.

ASET environment file asetenv not found in $ASETDIR. ASET startup unsuccessful.

Nguyên nhân lỗi xảy ra	**Cách khắc phục**
ASET không thể định vị file asetenv trong thư mục làm việc của nó.	Đảm bảo có một file asetenv trong thư mục làm việc của ASET. Xem **asetenv**(4) để có chi tiết về file này.

filename doesn't exist or is not readable

Nguyên nhân lỗi xảy ra	**Cách khắc phục**
File tham chiếu tới bằng filename không tồn tại hoặc không thể đọc. Điều này có thể xảy ra rõ ràng khi dùng tuỳ chọn -u mà bạn có thể đặc tả file chứa danh sách những người dùng bạn muốn kiểm tra	Đảm bảo đối số đối với tuỳ chọn -u tồn tại và có thể đọc được.

ASET task list TASKLIST undefined.

Nguyên nhân lỗi xảy ra	**Cách khắc phục**
Danh sách tác vụ ASET lẽ ra được định nghĩa trong file asetenv, lại không được định nghĩa. Điều này có nghĩa là file asetenv bị lỗi.	Kiểm tra file asetenv của bạn. Đảm bảo danh sách tác vụ được định nghĩa trong phần User Configurable. Ngoài ra kiểm tra các bộ phận khác của file để đảm bảo file còn nguyên vẹn. Xem **asetenv**(4) để có nội dung file asetenv tốt.

ASET task list $TASKLIST missing. ASET startup unsuccessful.

Nguyên nhân lỗi xảy ra	**Cách khắc phục**
Danh sách tác vụ ASET lẽ ra được định nghĩa trong file asetenv, lại không được định nghĩa. Điều này có thể có nghĩa là file asetenv bị lỗi.	Kiểm tra file asetenv của bạn. Đảm bảo danh sách tác vụ được định nghĩa trong phần User Configurable. Ngoài ra kiểm tra các bộ phận khác của file để đảm bảo file còn nguyên vẹn. Xem **asetenv**(4) để có nội dung file asetenv tốt.

Schedule undefined for periodic invocation.

No tasks excuted or scheduled. Check asetenv file	
Nguyên nhân lỗi xảy ra	**Cách khắc phục**
Lập lịch ASET đòi hỏi dùng tuỳ chọn -p, nhưng biến PERIODIC_ SCHEDULE không được định nghĩa trong file asetenv.	Kiểm tra phần User Configurable của file asetenv để chắc chắn biến được định nghĩa và có định dạng hoàn chỉnh.

Warning! Duplicate ASET excution scheduled. Check crontab file.	
Nguyên nhân lỗi xảy ra	**Cách khắc phục**
ASET được lập lịch nhiều hơn một lần. Nói cách khác, lập lịch được yêu cầu trong khi một lịch trình đang có tác động. Điều này không cần một lỗi nếu quả thực nhiều lịch trình được yêu cầu, chỉ cần một lời cảnh báo mà thông thường điều này là không cần thiết vì bạn nên dùng định dạng lập lịch **crontab**(1) nếu bạn muốn nhiều hơn một lịch trình.	Kiểm chứng thông qua giao diện lệnh **crontab**(1) mà lịch trình đúng đang tác động. Đảm bảo không cần đặt các mục crontab cho ASET

Tài liệu sử dụng

1. Solaris System Administration Guide, Chapter 12 -> Chapter 16
2. Software White Paper: Solaris Security, Tài liệu từ Internet

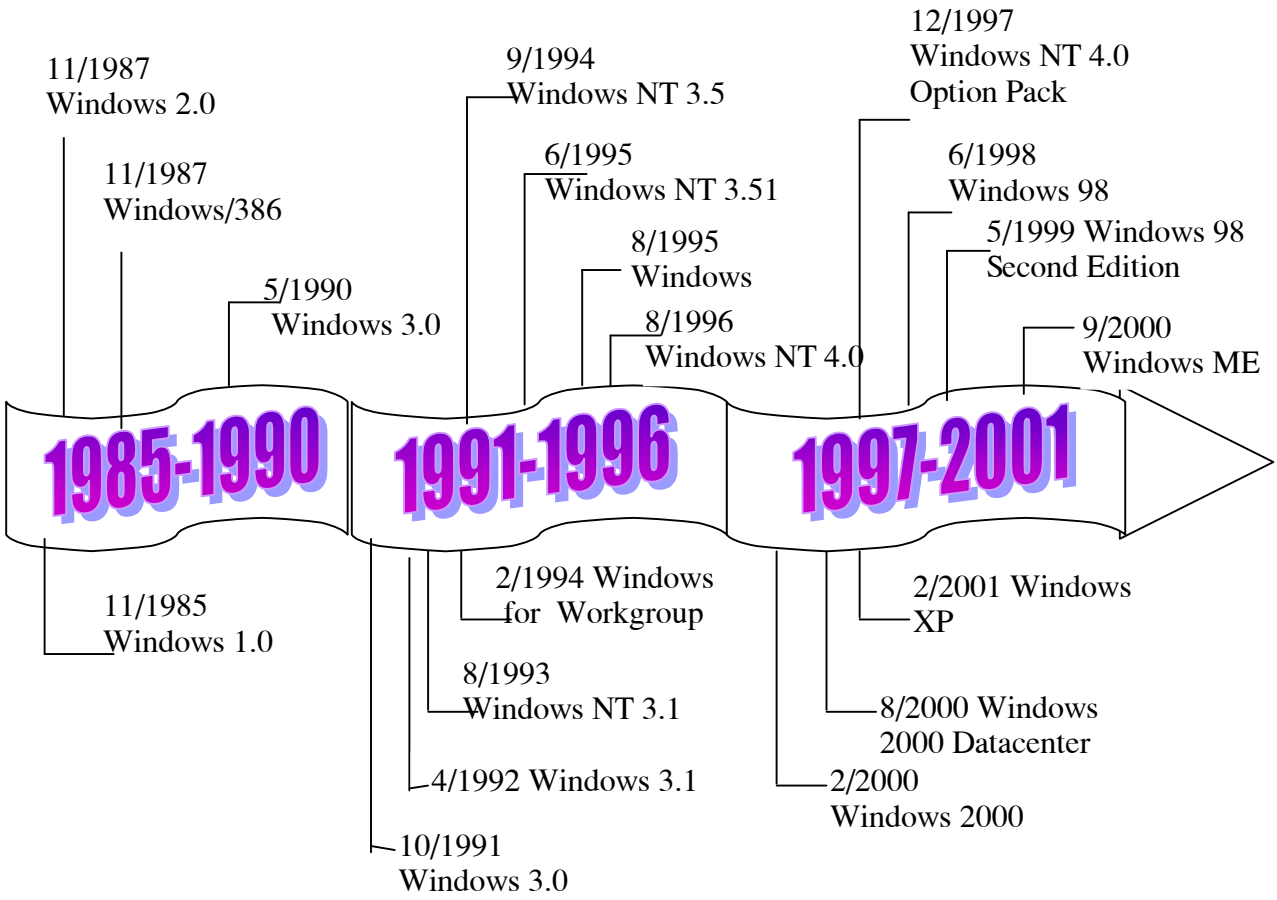
PHẦN III
AN NINH CỦA CÁC HỆ ĐIỀU HÀNH HỌ MICROSOFT WINDOWS

Các từ tiếng Anh được sử dụng trong tài liệu

Access Control List (ACL)	Danh sách điều khiển truy cập
Active Dierectory (AD)	
CD-ROM File System (CDFS)	Được sử dụng để đọc dữ liệu từ các ổ CD-ROM
Discretionary Access Controls (DAC)	Kiểm soát truy cập phân tách cụ thể (nhiệm ý)
Data Encryption Standard (DES)	Chuẩn mật mã dữ liệu
Exteded DES (DESX)	DES mở rộng
Domain model	Mô hình miền
Encrypting File System (EFS)	Mã hoá hệ thống tệp
File Allocation Table (FAT)	Bảng xác định vị trí tệp
File Encryption Key (FEK)	Khoá mã tệp
Local Security Authority (LSA).	Thẩm quyền bảo mật cục bộ
New Technology File System (NTFS)	Hệ thống tệp công nghệ mới
One-way function (OWF)	hàm một chiều
OS (Operating System)	Hệ điều hành (HĐH)
Password	Mật khẩu
Permission	Được dùng để giao hoặc khước từ quyền truy cập vào một đối tượng (ví dụ tệp hay thư mục).
Remote Access Service (RAS)	Dịch vụ truy cập từ xa
Security Account Manager (SAM)	Trình quản lý tài khoản bảo mật
Security Reference Monitor (SRM)	Trình giám sát tham chiếu bảo mật
Unique security identifier (SID)	mã định danh bảo mật duy nhất
Windows NT LAN Manager (NTLM)	Quản trị mạng LAN của Windows NT
Workgroup model	Mô hình nhóm làm việc

CHƯƠNG 1. TỔNG QUAN

Từ khi ra đời với hệ điều hành (HĐH) MS - DOS, cho đến những phiên bản đầu tiên của Windows được tung ra như là một chương trình bổ sung cho HĐH này, và đến nay đã có những phiên bản mới nhất mà sự hiện diện của MS - DOS bị che dấu đi hoặc bị loại bỏ hẳn. Chúng ta có thể thấy được sự phát triển của dòng HĐH của Microsoft trên Hình 1. Hãng Microsoft vẫn giữ được vị trí độc quyền như ngày nay cũng vì một phần Hãng đã chú trọng phát triển các phiên bản HĐH ngày càng tiện dụng, đảm bảo an toàn thông tin không những trên máy tính cá nhân mà còn khi làm việc trên mạng. Để xem xét vấn đề an ninh, an toàn mạng của họ các HĐH Microsoft, chúng ta cần nhắc lại ở đây các mô hình mạng Windows.



Hình 1. Quá trình phát triển của họ các hệ điều hành của Microsoft

1. Mô hình lập mạng trong môi trường Windows:

Mạng được hình thành gồm có hai phần chính: chủ (server) điều hành và cung cấp các dịch vụ, khách (client) nhận dịch vụ và chịu sự điều hành. Về cơ bản có hai mô hình lập mạng trong môi trường Windows: mô hình nhóm làm việc (workgroup model) và mô hình miền (domain model).

1.1 Mô hình nhóm làm việc:

Là một mô hình mạng đơn giản ở đó người sử dụng tại trạm làm việc của mình tham gia vào một nhóm ngươì dùng khác để chia sẻ tài nguyên. Người sử dụng cục bộ có thể chịu trách nhiệm về việc giao quyền truy cập các tài nguyên trên máy tính của mình cho những người sử dụng khác trong nhóm làm việc. Mọi phiên bản của các hệ điều hành của Microsoft (Windows 9x/NT/2000) hiện hành đều hỗ trợ tính năng lập mạng nhóm làm việc. Trong mô hình này, tên các máy tính đóng một vai trò quan trọng. Có hai phương pháp điều khiển truy cập trên các trạm làm việc Windows: điều khiển truy cập cấp dùng chung và điều khiển truy cập cấp người sử dụng. Phương pháp đầu sẵn dùng trên mọi mạng Windows, trong khi phương pháp sau đòi hỏi phải gắn với mạng một máy tính Windows NT để điều quản các tài nguyên người sử dụng. Trong mô hình cấp dùng chung, một người sử dụng quyết định chia sẻ các tài nguyên trên máy tính của mình với những người sử dụng khác trên mạng. Các thư mục được chia sẻ trên cơ sở Read-Only (chỉ đọc), Full (toàn quyền) và có Depend on Password (tuỳ thuộc vào mật khẩu) hay không. Đây là một lược đồ tuy cơ động, song không bảo mật vì các mật khẩu thường được trao đổi tuỳ ý. Trong mô hình cấp người sử dụng, việc truy cập các tài nguyên được giao cho từng người sử dụng riêng lẻ thay vì cho mọi người. Tên người sử dụng có trong các cơ sở dữ liệu tài khoản người sử dụng được lưu trữ trên một máy tính Windows NT. Ta chọn trong danh sách người sử dụng đó, họ sẽ được thẩm định quyền bởi máy tính Windows NT đang quản lý tài khoản người sử dụng rồi giao quyền truy cập. Người sử dụng không cần gõ mật khẩu vì họ đã làm việc này khi đăng nhập vào các hệ phục vụ. Theo cách này mật khẩu được giữ bí mật. Các điều khiển truy cập trong mô hình nhóm sẽ hạn chế người sử dụng trên mạng tự do truy cập các tệp trên một máy tính mạng khác. Tính năng bảo mật trên các máy chạy các HĐH Windows 9x, Windows 3.1, và MS-DOS không tồn tại bởi mọi người đều có thể bật máy tính và chép các tệp tin ra đĩa mềm. Không có tiến trình đăng nhập hoặc hệ cấp phép để ngăn cản người sử dụng truy cập các tệp. Các máy tính Windows NT đều mang tính bảo mật hiểu theo nghĩa này.

***1.2 Mô hình miền**:*

Miền là tập hợp các máy tính và người sử dụng máy tính được quản lý theo một thẩm quyền tập trung. Trong mô hình này việc truy cập các tài nguyên được điều khiển chặt chẽ bởi một điều hành viên trung tâm quản lý một máy tính Windows NT Server đang chạy một dịch vụ quản lý miền. Mô hình này thực thi các tài khoản người sử dụng hợp lệ bắt buộc phải có để cấp quyền sử dụng các tài nguyên dùng chung. Mô hình miền thực tế là một dạng cao cấp của mô hình nhóm làm việc. Tập hợp các máy tính nhóm làm việc đơn giản trở thành một miền ở đó tính năng bảo mật tài khoản người sử dụng

được điều quản bởi một hệ điều khiển miền. Tuy nhiên, ngay cả khi dùng mô hình miền, mọi hệ khách đều vẫn có thể chọn chia sẻ một tài nguyên trên máy tính của mình với một máy tính khác trên mạng khi dùng điều khiển truy cập cấp người sử dụng.

2. Khái quát về an toàn, an ninh mạng làm việc trong môi trường Windows

2.1 *Trong môi trường Windows 9x*

Như phần trên đã xem xét khi thiết lập mạng theo mô hình nhóm làm việc và mô hình miền, ta thấy rằng mặc dù các HĐH Windows 9x/NT/2000 đều hỗ trợ tính năng làm việc trên mạng, nhưng vấn đề quan trọng nhất mà chúng ta cần nhận thức rõ là Windows 9x không được thiết kế như một HĐH an ninh như Windows NT. Thực tế, có vẻ như trong nhiều trường hợp Microsoft đã muốn hi sinh tính năng bảo mật để đổi lấy khả năng dễ dùng khi hoạch định kiến trúc Windows 9x. Ví dụ như hộp thoại kiểm tra đăng nhập vào Windows 9x có thể dễ dàng bỏ qua hay tên người dùng/mật hiệu được lưu trữ cục bộ theo ngầm định của Windows 9x.

Tuy nhiên, mức đơn giản của Windows 9x cũng có điều hay về an ninh. Do nó không được thiết kế để trở thành một HĐH đa người dùng thực sự, nên nó có tính năng điều hành từ xa rất hạn chế. Ta không thể thi hành các lệnh từ xa trên các hệ thông Windows 9x thông qua các công cụ cài sẵn, và việc truy cập từ xa vào Windows 9x Registry chỉ có thể nếu các yêu cầu truy cập được chuyển đầu tiên thông qua một hệ thống cung cấp bảo mật như hệ phục vụ Windows NT. Windows 9x không thể tác động như một hệ phục vụ thẩm định quyền cấp người sử dùng.

Để xây dựng một mạng thực sự an toàn thì chỉ có thể dựa trên HĐH Windows NT. Điều quan trọng hơn cả trong mô hình an toàn này là kiểm soát truy nhập. Nó bao gồm việc kiểm soát ai có thể truy cập vào các tệp, các dịch vụ và các thư mục. Nó quan tâm đến cả thời gian mà truy cập này xảy ra. Phần tiếp theo sẽ xem xét HĐH mạng Windows NT

2.2 *Giới thiệu về hệ bảo mật Windows NT*

Hệ bảo mật NT là một cách để điều khiển việc truy cập của người sử dụng vào hệ thống mà ta có thể so sánh với các giấy chứng nhận tháo khoán an ninh cho một doanh nghiệp hoặc thậm chí một đặc khu quân sự. Windows NT dùng các điều khiển nhiệm ý (Discretionary Access Controls - DAC) cho phép điều khiển chính xác người sử dụng nào được quyền truy cập cái gì. Cấp truy cập cũng có thể thay đổi tuỳ theo các khu vực khác nhau.

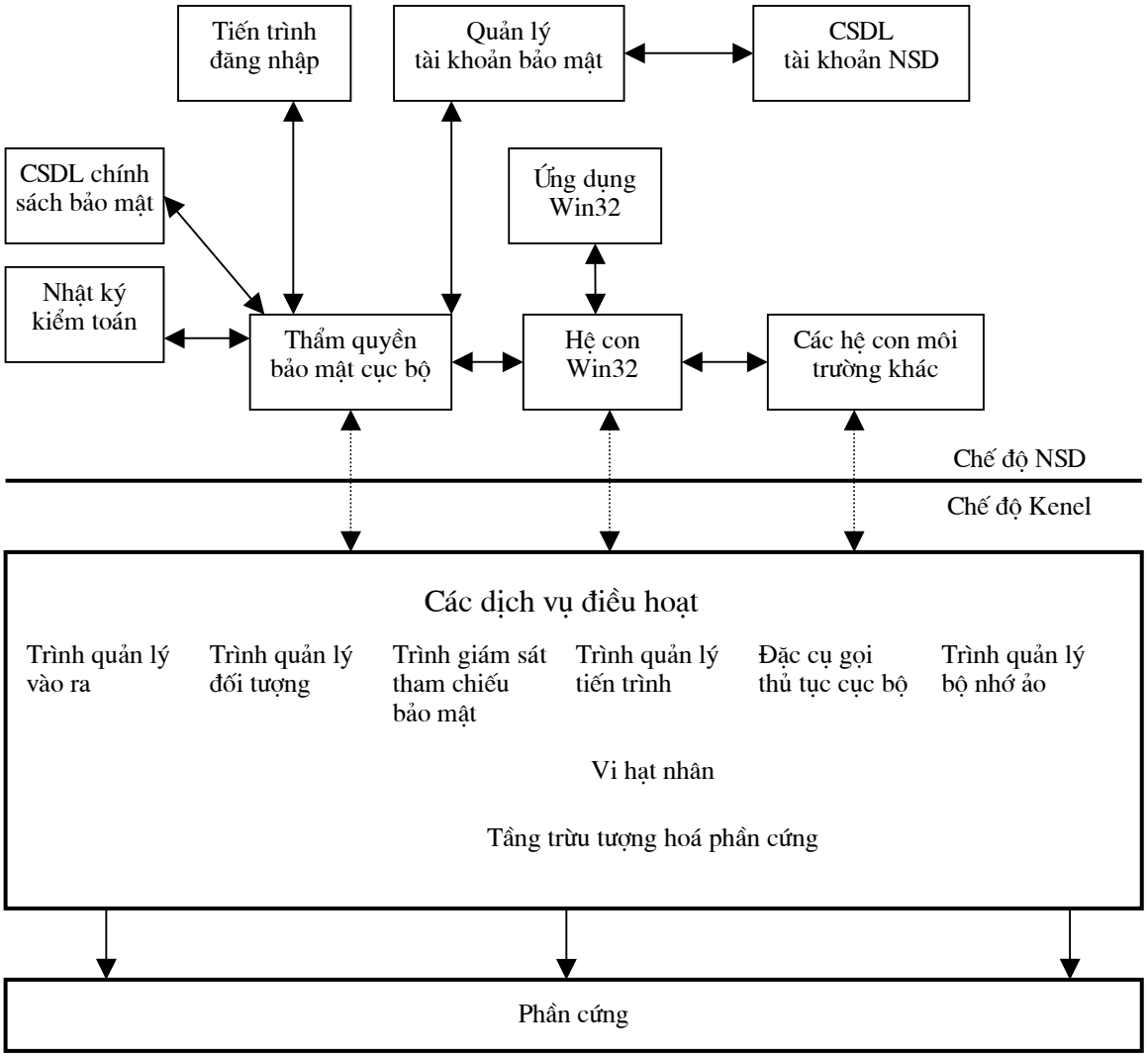
Hệ bảo mật Windows NT tương tự như hệ thống bảo vệ-và khoá thẻ: một chốt bảo vệ tại cửa Windows NT sẽ hợp lệ hoá người sử dụng và cho phép họ vào, sau đó giao cho họ một khoá thẻ để họ có thể truy nhậpcác tài nguyên bảo mật trên hệ thống.

Windows NT sử dụng tính năng thẩm định quyền "hai chiều". Trước tiên, người sử dụng phải có quyền truy cập một tài nguyên nào đó trên hệ thống, sau đó tài nguyên phải cho phép nó.
Phía "quyền" của bối cảnh trên được điều khiển bởi các tài khoản người sử dụng. Trong môi trường Windows NT, một tài khoản người sử dụng có thể được ví như hồ sơ cá nhân của một nhân viên. Nó chứa thông tin về người sử dụng và các quyền mà họ có trên hệ thống. Tài khoản được gán cho một mã định danh bảo mật duy nhất (unique security identifier - SID).

Nếu người sử dụng có một tài khoản, mật khẩu hợp lệ, và họ có giấy phép truy cập hệ thống, một thẻ bài truy cập bảo mật (security access token) sẽ được phát sinh khi người sử dụng đăng nhập một máy tính Windows NT. Thẻ bài truy cập chứa SID của người sử dụng ngoài các ID dành cho nhóm người mà người sử dụng đó thuộc về. Khi người sử dụng truy cập các tài nguyên trên hệ thống, thông tin trong thẻ bài truy cập được đối chiếu với thông tin mà tài nguyên lưu giữ được chính xác về người có thể truy cập. Đây là lúc tính năng thẩm định hai chiều ra tay. Mọi tài nguyên đều lưu giữ một danh sách người sử dụng có thể truy cập nó; các mục trong danh sách này được đối chiếu với các mục trong thẻ bài truy cập của người sử dụng. Sở dĩ Windows NT có được tính năng an toàn cao một phần là nhờ các đặc điểm về cấu trúc hệ thống và thiết kế

2.2.1 Cấu trúc hệ thống
Các thành phần cung cấp tính năng bảo mật trong Windows NT đã hình thành hệ con bảo mật. Trên H.2 giới thiệu cách kết hợp hệ thống này trong kiến trúc Windows NT.

Cũng như hầu hết các mô hình khác, nó được xếp tầng với phần cứng máy tính nằm ở dưới cùng và các ứng dụng cấp cao nằm ở trên cùng. Người sử dụng tương tác với các thành phần cấp cao nhất. Tất cả các tầng ở giữa sẽ cung cấp các dịch vụ cho các tầng phía trên và tương tác với các tầng thấp hơn.



Hình 2. Cấu trúc bảo mật Windows NT

Như trên hình vẽ, kết cấu được chia làm hai chế độ (mode): mode kernel, nơi mã đặc quyền cao yêu cầu truy cập trực tiếp đến bộ nhớ và các thao tác phần cứng; và mode người sử dụng, nơi đặt các ứng dụng và các hệ con Windows.

Trong mode người sử dụng, hệ điều hành đặt các không gian nhớ, và các ứng dụng và các hệ con được bảo vệ lẫn nhau và từ mã của hệ điều hành đặc quyền. Trong khi nhiều

hệ con an toàn được đặt ở mode người sử dụng thì hai thành phần kiểm soát không hạn chế truy cập tới toàn bộ tài nguyên hệ thống là trình Kiểm soát tham chiếu bảo mật (Security Reference Monitor) và trình Quản lý đối tượng (Object Manager) được đặt ở mode kernel.

2.2.2 Thiết kế hướng đối tượng

Windows NT được thiết kế dưới dạng một tập hợp đối tượng có quan hệ qua lại với nhau, cung cấp dịch vụ của HĐH. Nhờ vào thiết kế hướng đối tượng mà nó có thể cung cấp độ tin cậy trong bảo mật. Các đối tượng là chìa khoá cung cấp một cấp bảo mật cao trong hệ điều hành Windows NT. Đối tượng như một chiếc hộp chứa thông tin và các chức năng để điều tác thông tin đó. Việc bảo mật được thực hiện như sau. Trước tiên, các đối tượng che giấu dữ liệu của chúng với bên ngoài và chỉ cung cấp thông tin theo một số cách nhất định. Điều này ngăn cấm các tiến trình bên ngoài trực tiếp truy cập các dữ liệu bên trong. Sở dĩ Windows NT đạt được các cấp bảo mật cao đó là nhờ không bao giờ nó cho phép các chương trình trực tiếp truy cập các đối tượng. Mọi hành động trên một đối tượng đều được cấp quyền và được thực hiện bởi hệ điều hành.

2.2.3 Hệ con bảo mật Windows NT

Hệ con bảo mật cung cấp một hệ thống đơn lẻ thông qua đó mọi truy cập đến các đối tượng, kể cả các tập tin trên đĩa, các tiến trình trong bộ nhớ, hoặc các cổng ra các thiết bị bên ngoài đều được kiểm tra để không có ứng dụng hay người sử dụng nào có thể truy cập mà không có quyền hạn đúng đắn. Các thành phần hệ con bảo mật bao gồm:

Thẩm quyền bảo mật cục bộ (Local Security Authority - LSA). Đây là thành phần trung tâm của hệ con bảo mật. LSA có trách nhiệm xác nhận tính hợp lệ của toàn bộ các đăng nhập của người sử dụng tại chỗ cũng như từ xa, phát ra các thẻ bài truy cập, và quản lý chính sách an toàn cục bộ, bao gồm cả việc kiểm soát chính sách kiểm toán. LSA cũng có trách nhiệm ghi lại các bản ghi sự kiện bất kỳ một thông báo kiểm toán nào do trình tham chiếu bảo mật tạo ra.

Tiến trình đăng nhập (Logon Process). Tiến trình đăng nhập này đăng nhập cho cả người sử dụng cục bộ lẫn từ xa.

Trình quản lý tài khoản bảo mật (Security Account Manager - SAM). Hệ thống này có trách nhiệm kiểm soát và duy trì cơ sở dữ liệu về các tài khoản người sử dụng đã được cấp quyền truy cập và xác minh người sử dụng trong tiến trình đăng nhập (phê chuẩn người sử dụng cho LSA). Cơ sở dữ liệu SAM chứa các thông tin tài khoản của tất cả những (nhóm) người sử dụng và cung cấp các thông tin này hỗ trợ người sử dụng hợp lệ trong quá trình đăng nhập. Nó so sánh hàm hash mật mã của mật khẩu khi đăng nhập với mật khẩu theo giá trị hàm hash được lưu giữ trong cơ sở dữ liệu SAM. Sau đó,

nó cấp SID của (nhóm) người sử dụng về lại LSA. Sau đó các SID được sử dụng để tạo ra một thẻ bài truy cập bảo mật cho phiên hiện thời của người sử dụng đó.
Mỗi hệ Windows NT có cơ sở dữ liệu SAM tại chỗ. Mỗi máy trạm hay máy chủ có một cơ sở dữ liệu SAM cho người sử dụng cục bộ và các nhóm cụ thể với máy tính đó. Mỗi trình kiểm soát vùng có một cơ sở dữ liệu SAM để nhận dạng (nhóm) người sử dụng có thể sử dụng toàn bộ máy tính trong vùng đó. Cơ sở dữ liệu SAM được cập nhật và nhân bản cho phép bất kỳ một hệ điều khiển vùng nào cũng có thể đáp ứng yêu cầu xác thực. Chỉ có máy phục vụ và trạm làm việc Windows NT mới có SAM, và đây là lý do vì sao máy tính Windows 95 và nhiều máy tính khác thực sự không thể gia nhập vùng.

Trình giám sát tham chiếu bảo mật (Security Reference Monitor - SRM). Đây là một thành phần của chế độ Kernel; nó ngăn cấm mọi tiến trình hay người sử dụng trực tiếp truy cập các đối tượng. Nó hợp lệ hoá mọi tiến trình truy cập các đối tượng. Nó cũng phát sinh các thông báo kiểm toán thích hợp.

SRM có trách nhiệm buộc tất cả các truy cập hợp lệ và chính sách kiểm toán làm việc đúng đắn bên trong nội quy bảo mật cục bộ. Như vậy, nó hạn chế việc truy cập trực tiếp vào đối tượng bởi bất kỳ người sử dụng hay tiến trình nào, vì vậy bảo đảm rằng việc bảo vệ là như nhau được cung cấp cho các đối tượng bên trong hệ thống. SRM làm việc kết hợp với trình quản lý đối tượng nhằm hợp lệ hoá các truy cập đối tượng và phát ra các thông điệp kiểm toán nào đó theo yêu cầu. Khi truy lục một đối tượng được yêu cầu, SRM so sánh nội dung của ACL của đối tượng với nội dung của phiếu truy cập của người sử dụng. Nếu truy cập được cấp cho đối tượng, SRM bố trí một điều khiển cho quá trình này và điều khiển này được sử dụng đối với tất cả các yêu cầu truy cậpkhác tương tự mà không cần phải kiểm tra truy cập thêm nữa.

Cơ sở dữ liệu thư mục (Directory database). Trong môi trường mạng, cơ sở dữ liệu này có thể tồn tại trên một số máy. Khi một người sử dụng đăng nhập một máy cục bộ, SAM trên máy đó sẽ truy lục các ID người sử dụng từ cơ sở dữ liệu này. Trong môi trường mạng vùng Windows NT, thông tin tài khoản người sử dụng có thể được lưu trữ trong một cơ sở dữ liệu thư mục trên một hay nhiều hệ phục vụ có tên hệ điều khiển vùng (domain controllers), chúng chia sẻ và cập nhật thông tin tài khoản. Cơ sở dữ liệu dùng chung này cho phép người sử dụng đăng nhập một lần để truy cập các tài nguyên trên toàn mạng

Kiểm soát truy cập nhiệm ý (Discretionary Access Controls - DAC). Nó cho phép kiểm soát đầy đủ các tệp và tài nguyên nào có thể được truy cập bởi người sử dụng tại thời điểm đã cho.

Mức độ mà hệ thống DAC có thể kiểm soát tệp và thư mục được truy cập gọi là Gnanularity. Đó là chỉ số đo xem kiểm soát truy cập có thể cụ thể đến mức nào. Ví dụ,

ta có thể hạn chế truy cập đến năm tệp trong một thư mục cho một nhóm người sử dụng nào đó trong khi ta lại cho phép tất cả các người sử dụng truy cập đến các tệp còn lại trong thư mục này.

Trong cơ chế DAC nếu kẻ bẻ khoá không có quyền truy cập đến các tệp thì họ không thể bẻ khoá được máy tính. Do đó đặt quyền truy cập tệp đúng đắn là bước đầu tiên làm an toàn máy tính Windows NT. Muốn vậy máy tính phải sử dụng hệ thống tệp công nghệ mới (New Technology File System - NTFS).

NTFS tuy vậy vẫn chưa được hoàn thiện. Đối với phiên bản 3.51 người sử dụng không có quyền ưu tiên gì cũng có thể xoá được tệp. Một ví dụ khác là khi chạy File Manager cũng của phiên bản này thì các quyền truy cập tệp có thể bị bỏ qua. Tuy nhiên Windows NT DAC là hoàn toàn tốt. Đó là nguyên lý ai tạo ra tệp thì người đó là chủ. Với một tệp có thể đọc bởi người này, viết bởi người kia và chạy lại bởi những người khác nữa. Có thể nói rằng DAC là phức tạp.

Tóm lại, trên các HĐH như MS-DOS, Windows thực tế không thể đạt được một hệ bảo mật cao cấp. Các HĐH này được thiết kế để người sử dụng có thể truy cập các tài nguyên hệ thống với rất ít hạn chế, nếu có. HĐH quá yếu để cho phép bổ sung hệ bảo mật mạnh. Windows NT là một HĐH hướng đối tượng, và hệ bảo mật của nó được xây dựng ngay trong các cấp thấp nhất của cấu trúc đối tượng. Điều này khiến Windows NT dễ bảo vệ an toàn hơn so với hầu hết các HĐH khác.

3. Những nội dung chính cần nghiên cứu

Việc đảm bảo an toàn mạng ngoài việc thiết kế theo các mô hình mạng đã đề cập ở trên nó còn phụ thuộc vào các tính năng an toàn của các HĐH được sử dụng, mà ở đây là các HĐH Microsoft. Có thể thấy rằng chỉ từ khi có sự ra đời của HĐH Windows 95 các tính năng mạng mới được chú ý. Tuy nhiên lúc này Microsoft lại chú ý nhiều hơn đến các tính năng tiện dùng hơn là các tính năng bảo mật. Chỉ với các HĐH Windows NT và sau này, các tính năng bảo mật mới được quan tâm thích đáng hơn và rõ ràng, tính năng an toàn mạng tăng lên đáng kể. Theo một bản báo cáo của ITSEC (phát hành 20/6/2000) nếu WIDOWS/DOS và Windows 9x có tính năng bảo mật tối thiểu thì tính năng bảo mật của Windows NT có thể nói là rất tốt với các phiên bản mới nhất.

An ninh an toàn mạng là một vấn đề lớn và liên quan đến nhiều yếu tố. Trong phần này ta sẽ xem xét an ninh an toàn mạng đối với các HĐH của Microsoft, mà chủ yếu là HĐH Windows NT thông qua các nội dung sau:

- Đăng nhập, sử dụng dịch vụ
- Phân quyền đối với thư mục, tệp
- NTFS

CHƯƠNG II. ĐĂNG NHẬP, SỬ DỤNG DỊCH VỤ

Mặc dù các tài nguyên mạng đã được bảo vệ ở các mức khác nhau, nhưng ngay việc chúng ta muốn truy cập vào vùng hay một máy cụ thể nào đó cũng vẫn phải qua một cửa là cơ chế bảo mật đăng nhập. Giống như khi chìa hộ chiếu vào sân bay hay xuất trình thẻ vào cơ quan, chúng ta phải khai báo tên, mật khẩu, tên vùng hay máy mà mình muốn vào để tiến trình đăng nhập (logon) xác nhận chúng ta đúng là người của vùng với các quyền hạn nhất định về tài nguyên, hệ thống và cho phép nhập vùng.

Hiện nay, truy cập tới một hệ thống được thực hiện nhờ đăng nhập bắt buộc. Đây là một yêu cầu an toàn cơ bản tuân thủ nguyên tắc bảo mật cấp C2. Sự nhận diện và xác thực này là nền tảng của hệ an toàn. Không có nhận diện và xác thực người sử dụng đăng nhập vào hệ thống, việc truy cập đến đối tượng không được kiểm soát, các quyền và thẩm quyền của người sử dụng không có hiệu lực, và trách nhiệm giải trình không thể được duy trì nhờ kiểm toán. Yêu cầu an toàn thống nhất trong toàn HĐH và là thể hiện đầu tiên cho người sử dụng trong quá trình đăng nhập. Quá trình đăng nhập sai khác nhỏ phụ thuộc vào việc người sử dụng đăng nhập tới hệ cục bộ hay tới một vùng. Có hai cách có thể đăng nhập: cục bộ hoặc từ xa. Đăng nhập cục bộ ngụ ý việc đăng nhập trực tiếp vào máy tính đang được đề cập. Đăng nhập từ xa ám chỉ một cuộc đăng nhập "ngang qua mạng" để truy cập tài nguyên dùng chung. Việc đăng nhập đòi hỏi phải có tên đăng nhập và mật khẩu hợp lệ.

Vì sao Windows 9x không có thể được coi là hệ điều hành an toàn? Mặc dù các HĐH Windows 9x được thiết kế không phải cho đa người sử dụng thực sự và hạn chế được các tấn công từ xa song nếu một ai đó có thể tiếp cận một máy Windows 9x, anh ta có thể dễ dàng lấy cắp được thông tin đăng nhập (tên người sử dụng, mật khẩu, tên vùng và sử dụng nó để đăng nhập. Ngoài ra còn một số điểm yếu khác nữa về mật khẩu, mật mã...của Windows 9x, mà ta sẽ bàn đến sau, làm cho nó bị coi là HĐH không an toàn.

Nói chung, các bước trong tiến trình đăng nhập là xin giấy uỷ nhiệm (credentials) và thẩm định quyền. Đầu tiên, người sử dụng xin giấy uỷ nhiệm bằng cách nhập thông tin người sử dụng (tên người sử dụng, mật khẩu, tên vùng). Thông tin này được so sánh với thông tin về người sử dụng mà hệ thống đã lưu giữ. Nếu chính xác thì người sử dụng được thẩm định quyền và có quyền truy cập hệ thống. Nếu sai người sử dụng đó bị từ chối truy cập. Như vậy mức độ an toàn của thông tin người sử dụng và cách thức thẩm định quyền sẽ quyết định mức độ an toàn khi đăng nhập.

1. An toàn mật khẩu

Mật khẩu thường được lưu giữ trong cơ sở dữ liệu. Trong Windows 9x mật khẩu được mật mã yếu, mật khẩu LAN Manager, và đặt trong tệp .pwl. Các dịch vụ tệp LAN

Manager đã được sử dụng cho các dịch vụ tệp PC trong những năm 1980 và đầu những năm 1990 và vẫn có trong các HĐH khác. Windows 9x chỉ thực hiện xác thực quản lý. Nó dựa trên bộ kí tự OEM chuẩn, không phân biệt chữ hoa chữ thường và có thể dài tới 14 kí tự. Mật khẩu LAN Manager được mật mã khi sử dụng thuật toán Chuẩn mật mã dữ liệu (DES). Mật khẩu này rất dễ bị tấn công do không phân biệt chữ hoa chữ thường. Mật khẩu này cũng dễ bị giải mã khi sử dụng kí tự thông thường và số lượng kí tự khác 7 hay 14.

Windows NT có sử dụng thêm mật khẩu Windows NT. Mật khẩu này dựa trên bộ kí tự Unicode và nó là mật khẩu có phân biệt chữ hoa chữ thường và có thể dài tới 128 kí tự. Mật khẩu NT được mật mã bằng việc sử dụng thuật toán Message Digest 4 (MD4). Trong Windows NT mỗi mật khẩu được mã hoá kép trong cơ sở dữ liệu SAM. Lần mã hoá đầu tiên là một phiên bản hàm một chiều (one-way function - OWF) của mật khẩu rõ. Sau đó, mật khẩu này được mã hoá lại để khiến nó trở nên khó hiểu hơn. Các mật khẩu mã hoá một chiều thường được xem là không thể giải mã. Thực tế, cơ sở dữ liệu không hề được giải mã. Thậm chí, chính SAM cũng không thể giải mã các mật khẩu trong cơ sở dữ liệu. Điều này ngăn cấm ai đó viết một chương trình dùng các SAM API để đọc các mật khẩu đã mã hoá ra khỏi cơ sở dữ liệu. Một chương trình như vậy có thể được dùng để tiến hành một cuộc tiến công từ điển trên cơ sở dữ liệu SAM.

Trong Windows 2000 thông tin người sử dụng được mã hoá và cất giấu trong SAM của hệ thống cục bộ hay AD (Active Dierectory) của hệ thống. AD thay thế cho vùng cơ sở dữ liệu SAM của registry trên điều khiển vùng và là một thành phần tin cậy của LSA. Khác với cơ sở dữ liệu SAM có cấu trúc phẳng thì AD lại có cấu trúc phân cấp cho phép hạn chế các truy cập trái phép.

Ngoài ra các HĐH Windows NT và Windows 2000 cho phép nhấn CTRL-ALT-DEL để khởi phát một tiến trình đăng nhập mới. Như vậy sẽ tránh được mối nguy hiểm của các chương trình Trojan Horse.

2. Thẩm định quyền

HĐH Windows NT quản lý truy cập mạng thông qua cơ chế xác thực người sử dụng gồm tên người sử dụng và mật khẩu tương ứng. Mật mã được đưa từ trạm đăng nhập mạng về trung tâm theo cách mã hoá đặc biệt theo một trong hai cách sau đây:

Một là, Windows NT dùng DES làm hàm một chiều để mã hoá mật khẩu của người sử dụng. Mật khẩu này dùng DES làm hàm một chiều để mã hoá một hằng số qui ước rồi chuyển giá trị mã khoá này đến cơ sở dữ liệu người sử dụng. Ở đây giá trị này được đem so sánh với giá trị đã lưu trong cơ sở dữ liệu. Nếu trùng khớp thì được đăng nhập mạng nếu không sẽ bị từ chối. Mật khẩu không bị lộ vì nó không thể tính ngược do tính chất một chiều của DES.

Hai là, sử dụng giao thức thẩm định quyền mã hoá Windows NT LAN Manager (NTLM). Khi đăng nhập thì SERVER sẽ gửi một giá trị nonce dài 16 byte cho trạm CLIENT. Mật khẩu của người sử dụng được dùng để lập mã nonce và gửi về SERVER. Mật khẩu đầu tiên được dùng làm khoá để mã một hằng số qui ước trước, sau đó giá trị một chiều này được dùng làm khoá để mã hoá nonce và kết quả trả về SERVER. Một mặt SERVER nhận giá trị này, mặt khác nó lấy giá trị một chiều ở cơ sở dữ liệu của người sử dụng 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 là trùng nhau SERVER cho phép đăng nhập mạng. Ngược lại, nó từ chối đăng nhập mạng. Nó mã hoá bằng một khoá số nhị phân 56 bit với 72 nghìn triệu triệu tổ hợp khả dĩ. Khoá được phát sinh ngẫu nhiên cho mỗi phiên làm việc để tạo một khuôn mẫu mã hoá thường được xem là không thể bẻ khoá nếu không có khoá giải mã.

Để thẩm định quyền người sử dụng trong Windows 2000 ngầm định sử dụng giao thức thẩm định quyền Kerberos Version 5. Giao thức thẩm định quyền Kerberos Version 5 là một giao thức thẩm định quyền an toàn phân tán dựa trên an toàn chuẩn Internet. Nó thay thế NTLM, được dùng trong Windows NT Server 4.0, như một giao thức an toàn chính khi truy cập các tài nguyên trong hoặc ngang qua mạng vùng Windows 2000 Server. Hỗ trợ Kerberos bảo đảm đăng nhập an toàn, một lần và nhanh đến các tài nguyên dựa trên Windows 2000 Server cũng như các môi trường khác có hỗ trợ giao thức này.

Với Windows NT và các HĐH sau này, khi tài khoản người sử dụng cần được hợp thức hoá, nhưng máy tính cục bộ không thể tự hợp thức được thì mật khẩu luôn được mật mã hoá và truyền trên một kênh mật được thiết lập trước.

Thẩm định quyền người sử dụng hai yếu tố.
Có thể dùng các thiết bị bảo mật bên thứ ba để cải thiện hệ bảo mật cho người sử dụng quay số vượt trên mức bảo mật sẵn có của các dịch vụ Windows NT RAS (Remote Access Service - Dịch vụ truy cập từ xa). Các thiết bị bảo mật thường là các thẻ khoá [keycards]: đó là các thiết bị bảo mật có kích cỡ bằng thẻ tín dụng hiển thị một mã số khác nhau theo từng phút. Thẻ khoá được đồng bộ hoá với một thiết bị tương tự tại hệ phát sinh cùng mã số. Khi người sử dụng đăng nhập, mã số trên thẻ khoá của người sử dụng được gửi đến hệ phục vụ quay số dưới dạng một biện pháp bổ trợ cho thủ tục đăng nhập bình thường. Kỹ thuật này bảo đảm chỉ người sử dụng hợp pháp có các mật khẩu và các mã số thẻ khoá hợp lệ mới có thể đăng nhập hệ thống. Hai yếu tố trong lược đồ này là mật khẩu mà người sử dụng biết và giá trị thẻ khoá mà họ có vào lúc đăng nhập. Các thiết bị bảo mật tồn tại theo cả dạng phần cứng và phần mềm. Các thiết bị phần cứng thường có kích cỡ như các thẻ tín dụng và có một màn hình LCD nhỏ để nêu mã số truy cập. Các thiết bị phần mềm là các chương trình chạy trên máy tính người sử dụng và thực hiện cùng chức năng như các thiết bị phần cứng. Nói chung, các thiết bị phần mềm tiện dụng hơn bởi vì chúng tự động hoá tiến trình và không yêu cầu khoá

của người sử dụng trong mã số truy cập. Tuy nhiên, các thiết bị phần mềm thường ít an ninh hơn, bởi các hắc cơ có cơ hội để bẻ khoá thông tin có thể nằm trong bộ nhớ hoặc trên đĩa. Việc bổ trợ đăng nhập từ xa theo cách này sẽ cho ta một cấp bảo mật cao.
Cùng với sự ra đời Windows 2000 là ứng dụng thẻ thông minh trên nó.Thẻ thông minh bảo đảm lưu giữ chống lục lọi nhằm bảo vệ các khoá riêng tư, các số tài khoản, mật khẩu và các thông tin cá nhân khác. Các thẻ thông minh nâng cao các giải pháp phần mềm bao gồm cả thẩm định quyền hệ khách. Thẻ thông minh là một thành phần chủ chốt của cơ sở hạ tầng khoá an toàn mà Microsoft tích hợp trong HĐH Windows 2000.
Trong tương lai, mật khẩu có thể được bảo mật thêm nhờ các phương pháp nhận dạng sinh học sử dụng các đặc tính cá thể như vân tay, mẫu võng mạc, mồ hôi, DNA, sự thay đổi giọng nói và nhịp điệu đánh máy trên bàn phím.

CHƯƠNG III. PHÂN QUYỀN ĐỐI VỚI THƯ MỤC, TỆP

Phần trên đã đề cập đến giai đoạn đầu, giai đoạn "quyền" của tính năng thẩm định quyền "hai chiều" trong hệ thống mạng an toàn. Phần này sẽ trình bày giai đoạn hai , giai đoạn cho phép "permission" của một đối tượng cụ thể.

Như đã trình bày ở trên, các đối tượng trong các HĐH của Microsoft bao gồm mọi thứ từ các tệp, các cổng truyền thông, đến các xâu thi hành. Mỗi đối tượng đều có thể được phân quyền riêng lẻ hoặc dưới dạng một nhóm tuỳ thuộc vào HĐH. Các đối tượng có các kiểu permission khác nhau được dùng để giao hoặc khước từ quyền truy cập chính chúng. Trong chương này đề cập đến đối tượng cần phân quyền là thư mục và tệp. Đối với đối tượng này có thể có permission Read, Write, và Execute. Các thư mục là các đối tượng "thùng chứa" lưu giữ các tệp, do đó permission gán cho "thùng chứa" đều được thừa kế bởi các đối tượng tệp chứa trong nó. Để xem xét tính năng phân quyền thư mục và tệp, chúng ta cần tìm hiểu các hệ thống tệp được các hệ điều hành Microsoft hỗ trợ, và sau đó là các permission của chúng.

Nên lưu ý rằng, các điều khiển truy cập và các quyền tài khoản người sử dụng là hai khía cạnh khác nhau của hệ bảo mật Windows NT. Hệ bảo mật tài khoản người sử dụng định danh và hợp lệ hoá người sử dụng, trong khi các điều khiển truy cập lại hạn chế những người sử dụng nào mới có thể làm việc với các đối tượng.

Cũng như mọi đối tượng khác, đối tượng thư mục và tệp có một dấu mô tả bảo mật (security descriptor) để mô tả các thuộc tính bảo mật. Dấu mô tả bảo mật bao gồm:

- ID bảo mật của người sử dụng sở hữu đối tượng, thường là những người tạo ra đối tượng và được gọi là chủ nhân (owner)
- ACL (Access Control List - danh sách điều khiển truy cập), lưu giữ thông tin về những người sử dụng và nhóm nào có thể truy cập đối tượng
- ACL hệ thống, có liên quan đến hệ kiểm toán
- ID bảo mật nhóm, dược dùng bởi hệ con POSIX

Các ACL là điểm then chốt của phần thảo luận này. Về cơ bản, ACL là một danh sách người sử dụng và nhóm có permission truy cập vào đối tượng. Đối tượng thư mục và tệp có ACL riêng của nó. Các chủ nhân có thể tạo các mục trong ACL thông qua các công cụ như File Manager hoặc bằng cách ấn định các tính chất cho các tệp và thư mục (trong Windows NT 4.0). Network và Services trong Control panel cũng là những trình tiện ích khác dùng để ấn định permission.

Người sử dụng có thể có nhiều mục trong ACL của một đối tượng, cung cấp các mức truy cập khác nhau cho chúng. Ví dụ, một người sử dụng có thể có giấy phép Read đối với một tệp dựa trên tài khoản người sử dụng của họ và giấy phép Read/Write dựa trên

tư cách là thành viên của một nhóm. Mỗi giấy phép này được nêu trong một mục riêng biệt trong ACL.

Khi người sử dụng truy cập một đối tượng, họ thường có một quyền truy cập thoả đáng nhất định, như Read hay Read/Write. Để giao (hay khước từ) truy cập, SRM sẽ đối chiếu thông tin trong thẻ bài truy cập của người sử dụng với các mục trong ACL. Thẻ bài truy cập chứa các ID bảo mật và danh sách các nhóm mà người sử dụng đó thuộc về. SRM sẽ đối chiếu thông tin này với một hay nhiều mục trong ACL cho đến khi tìm thấy đủ giấy phép để trao quyền truy cập thoả đáng. Nếu không thấy đủ giấy phép, việc truy cập bị khước từ.

Nếu SRM tìm thấy vài mục dành cho người sử dụng, nó sẽ xem xét từng mục để xem (tổ hợp các) mục đó có thể giao cho người sử dụng giấy phép thoả đáng để dùng đối tượng đó hay không.

1. Các hệ thống tệp được các hệ điều hành Microsoft hỗ trợ:

Một trong những yêu cầu của HĐH là công tác quản lý dữ liệu: có thể dùng loại đĩa nào với HĐH đó, cách thức HĐH chia đĩa thành nhiều phần nhỏ, dữ liệu và tệp được lưu giữ theo cách thức nào, và nhiều vấn đề khác. Mục này sẽ cung cấp tổng quan về khả năng hỗ trợ các hệ thống tệp của họ các HĐH của Microsoft (được liệt kê trong Bảng 1). Các hệ thống tệp này có những tính năng khác nhau như độ dài tên tệp, tính năng bảo mật, dung lượng tối đa của tệp và phân hoạch

Bảng 1

Các hệ điều hành	Hỗ trợ các hệ thống tệp
Windows NT, Microsoft Windows 95/98, MS-DOS, IBM OS/2	File Allocation Table (FAT)
Windows NT/2000	Windows NT File System (NTFS), New Technology File System
Windows NT, Microsoft Windows	CD-ROM File System (CDFS)
OS/2, Windows NT	High Perfomance File System (HPFS)

- CDFS được sử dụng để đọc dữ liệu từ các ổ CD-ROM. Vì CDFS là hệ thống tệp đặc biệt chỉ đọc (read-only) nên phạm vi ứng dụng của nó bị hạn chế.
- FAT mà chúng ta quen gọi là bảng xác định vị trí tệp đã được sử dụng nhiều năm trên các máy chạy MS - DOS, chạy các HĐH Microsoft Windows 9x, IBM OS/2, Windows NT. FAT hỗ trợ qui ước tên tệp 8.3 (số ký tự phần bên trái dấu chấm không quá 8 và số ký tự phần bên phải dấu chấm không quá 3) cho các phiên bản của các HĐH này. Ngoài ra FAT còn hỗ trợ thêm qui ước đặt tên dài cho tệp/thư mục, vốn được áp dụng ở Windows 95/98/NT.

Trong hệ thống tệp này, mỗi tệp và thư mục tồn tại ở cấp gốc (root) trong phân chia FAT chỉ đến một mục nhập FAT nhận diện con số bắt đầu cho tệp/thư mục đó. Nếu tệp lớn hơn một cụm (cluster) sector đơn lẻ (có kích hước phụ thuộc vào kích thước phân chia), cụm sector này chỉ đến cụm kế tiếp. FAT không hề cố gắng tối ưu hoá tệp : cụm sector kế tiếp của tệp sẽ là cụm kế tiếp khả dụng trên đĩa, bất chấp vị trí của cụm trước đó. Cụm sector cuối cùng mà tệp chiếm dụng có dấu hiệu End of File.

Thư mục gốc của FAT bị giới hạn ở 512 mục nhập (có thể là tệp hoặc thư mục con). Thư mục con (subdirectory) là tệp liệt kê các tệp và thư mục con khácchứa trong nó, với một dấu hiệu cho biết đây là thư mục con. Thư mục con có thể chứa thư mục con và tệp trực thuộc với số lượng bất kỳ.

Hệ thống tệp FAT bị giới hạn ở số lượng nhập nhất định: mặc dù ban đầu MS-DOS hỗ trợ tối đa 4096 mục nhập, nhưng Windows 95/98/NT lại hỗ trợ đến 65536 mục nhập trong FAT. Vì FAT bị giới hạn ở số lượng cluster cố định, nên một cluster sẽ không có kích thước như nhau trên hai volume không cùng kích thước. Chỉ duy nhất một tệp được chỉ định cho mỗi cluster, và bất kỳ không gian thừa nào ở cluster cuối cùng được gán cho tệp đều bị bỏ phí.

Không thể bảo vệ được các phân hoạch FAT bằng tính năng bảo mật thư mục hoặc tệp cục bộ (local file) trên các HĐH này. Duy nhất có một chế độ bảo mật cho các phân hoạch FAT trên mạng: chế độ này được cung cấp thông qua các nguyên tắc chia sẻ của các HĐH. Điều đó có nghĩa rằng trên một phân hoạch FAT, các HĐH không hỗ trợ các tính năng bảo mật đến mức tệp; nếu muốn thiết đặt để không thể truy cập được một tệp nào đó, ta phải khởi tạo thư mục, thiết đặt trạng thái không chia sẻ (không dùng chung) cho thư mục đó và đặt tệp nói trên trong thư mục đó. Một trong những nhược điểm của việc chia sẻ là rất khó quản lý vì nếu giả sử có hàng trăm người sử dụng trên một máy chủ và mỗi người lại có một thư mục riêng, chúng ta phải thiết lập hàng trăm chia sẻ, và đôi khi những chia sẻ này lại chồng chéo nhau nên gây thêm những phiền toái.

- Hệ thống tệp công nghệ mới (New Technology File System - NTFS) được hỗ trợ trong Windows NT/2000 là hệ thống tệp thích hợp nhất cho Windows NT/2000 do một số lí do, đặc biệt là lý do bảo mật. Khác với FAT, NTFS không bị giới hạn ở một số lượng sector nhất định trong mỗi cluster. ở hệ thống tệp này, cluster là đơn vị cơ sở. Thừa số cluster được định nghĩa là một số lượng byte, và việc định dạng một volume theo NTFS sẽ bảo đảm rằng thừa số cluster là bội số của kích thước sector trên ổ đĩa. Vì NTFS nhận diện mọi thứ theo số hiệu cluster, nên hệ thống tệp không tính đến kích thước sector. Do vậy, số lượng sector trong mỗi cluster là một giá trị có tính đề nghị thay vì giá trị cố định. NTFS cho phép điều chỉnh số lượng sector mặc định trong mỗi cluster sao cho thích hợp nhất với mức độ sử dụng thực tế của volume. NTFS còn tìm kiếm không gian đĩa liền nhau trước khi ghi hoặc sao chép tệp vào đĩa.

Chúng ta nên dùng phân hoạch NTFS khi có yêu cầu bảo mật cho các máy chủ hoặc các máy cá nhân. NTFS hỗ trợ điều khiển truy cậpvà các đặc quyền riêng rất quan trọng để đảm bảo tính thống nhất của dữ liệu. Mặc dù các thư mục trên các máy chạy Windows NT/2000 có thể được gán thêm permission chia sẻ không phụ thuộc vào hệ thống tệp đang dùng, với các tệp và các thư mục NTFS, ta vẫn có thể gán permission để chúng được dùng chung hay không. NTFS là hệ thống tệp duy nhất trên Windows NT/2000 cho phép ta khả năng thiết đặt permission tới các tệp và các thư mục riêng.

2. Phân quyền đối với thư mục và tệp thực chất là bảo mật các tài nguyên mạng thông qua permission chia sẻ

Các thư mục được chia sẻ (hay được dùng chung - shared folders) cho phép người sử dụng truy cập vào các tệp trên các ổ đĩa mạng.

2.1. Giới thiệu chung

Khái niệm chia sẻ tài nguyên là một khái niệm quan trọng đối với môi trường làm việc trên mạng. Nếu làm việc trên một máy tính cục bộ, chúng ta hoàn toàn có thể truy cập và khai thác các tài nguyên trên máy đó. Nhưng tình hình sẽ khác đi nếu chúng ta muốn truy cập vào một chương trình nào đó hoặc vào một cơ sở sở dữ liệu được cài trên một máy khác. Muốn sử dụng một tài nguyên của mạng (không có ở trên máy mình), tài nguyên đó phải được chia sẻ.

Việc chia sẻ các tài nguyên trên mạng mang lại những lợi ích căn bản. Trước hết, những người sử dụng truy cập các tài nguyên đó theo cách thức tập trung. Đối với những người quản trị, điều này có nghĩa rằng việc nắm quyền kiểm soát các tài nguyên trên mạng được thực hiện một cách dễ dàng hơn. Lợi ích thứ hai trong việc chia sẻ chính là việc sử dụng các tài nguyên một cách có hiệu quả và kinh tế hơn rất nhiều. Cuối cùng, với những chính sách thích hợp, việc bảo mật các tài nguyên sẽ được thực hiện một cách hữu hiệu hơn.

Các HĐH hỗ trợ mạng của Microsoft đều cho phép thiết đặt để có thể chia sẻ các tài nguyên cho người khác. Tuy nhiên mức độ cho phép người khác dùng đến đâu là do người chia sẻ quyết định.

a. Các thư mục được chia sẻ

Một thư mục được chia sẻ là một thư mục được thiết đặt sao cho những người có quyền hợp pháp có thể kết nối tới thư mục đó và khai thác các tài nguyên lưu giữ trong đó. Ngoài ra, với Windows NT chúng ta có thể thiết đặt permission chia sẻ (shared permission) cho các tài khoản người sử dụng (và các tài khoản nhóm) để điều khiển những người sử dụng có thể thực hiện được điều gì với nội dung của thư mục được chia sẻ. Những người sử dụng trên các máy khác có thể dùng tiện ích duyệt (Browser) như Explorer của Windows 9x, Windows NT Explorer để truy cập được tài nguyên trên một

thư mục được chia sẻ hay kết nối đến thư mục đó như một phân hoạch ảo (E, F, hay G) của máy mình; chúng ta gọi các ổ ảo này là các ổ đĩa mạng. Tất cả các thư mục trên mọi hệ thống tệp (FAT, NTFS, CDFS, HPFS) đều có thể chia sẻ được. Muốn chia sẻ thư mục cần có các điều kiện sau:

- Serser Service đã được khởi động
- Người thao tác có quyền chia sẻ là những người thuộc các nhóm: Administrators, Server Operators, Power Users

Nếu một phân hoạch được định dạng theo hệ thống tệp FAT, việc thiết đặt chia sẻ hay không chia sẻ các thư mục là cách thức duy nhất đảm bảo tính bảo mật cho các tài nguyên trên phân hoạch đó. Nếu phân hoạch được định dạng theo hệ thống tệp NTFS, ngoài chia sẻ thư mục chúng ta có thể thết đặt thêm permission NTFS để đảm bảo tính bảo mật cao hơn.

b. Permission trên các thư mục được chia sẻ

Để điều khiển người sử dụng truy cập vào một thư mục được chia sẻ, chúng ta có thể gán permission chia sẻ (share permision) cho những người sử dụng, cho các nhóm hoặc cho cả hai. Mọi giấy phép đối với một thư mục có tác dụng đối với mọi tệp và thư mục con được chứa trong đó. Bảng 2 liệt kê permission đối với các thư mục được chia sẻ.

Bảng 2

Giấy phép	Dùng để
No access (không được truy cập)	Đặt ở chế độ này, người sử dụng có thể nhìn thấy thư mục trong mạng nhưng không truy cập vào được cũng như không nhìn thấy và không khai thác được các tệp hay thư mục con của nó
Read (Đọc)	Có thể xem tên tệp và thư mục con, xem dữ liệu và thuộc tính của tệp, chạy các tệp chương trình và truy cập tới các thư mục con chứa trong thư mục đó
Change (Thay đổi)	Có thể tạo các thư mục con, thêm tệp, thay đổi dữ liệu cũng như thêm dữ liệu vào tệp, thay đổi thuộc tính tệp, xoá các tệp và thư mục con
Full control (Toàn quyền)	Có thể làm mọi việc của chế độ Change, thay đổi permission của tệp, lấy quyền sở hữu đối với các tệp, thư mục ở NTFS.

Như ta thấy, bảng trên liệt kê bốn loại giấy phép, mức độ giấy phép tăng dần theo thứ tự liệt kê, từ giấy phép gán quyền hạn chế nhất tới giấy phép rộng rãi nhất.

c. Phạm vi tác dụng của permission trên các thư mục được chia sẻ. Có thể đặt permission trên bất kỳ một tài nguyên được chia sẻ nào mà không phụ thuộc vào hệ thống tệp trên phân hoạch đó. Tuy nhiên những giấy phép này chỉ có tác dụng khi ta truy cập thông qua mạng. Permission chia sẻ đối với một thư mục không có tác dụng khi một người nào đó đăng nhập một cách cục bộ thành công vào máy có chứa thư mục

đó. Khi đó anh ta có toàn quyền trong việc truy cập các thư mục và các tệp. Hiển nhiên, việc klhai thác tài nguyên tệp của anh ta có thành công hay không cũng còn phụ thuộc vào các tính năng bảo mật của chương trình ứng dụng tạo ra tệp đó (như tệp văn bản Word được cài đặt mật khẩu chẳng hạn).

Nếu người đó không có quyền đăng nhập cục bộ (Log on locally) trên máy chủ Windows NT Server thì điều này không gây ra phiền toái gì. Mặt khác, đối với các máy chạy Windows NT Workstation, người sử dụng được gán quyền đó một cách tự động thì vấn đề trên phức tạp hơn; họ có thể bỏ qua permission chia sẻ để truy cập tới các tệp trên máy cục bộ.

d. Hiệu lực kết hợp của permission

Chúng ta có thể gán permission chia sẻ thư mục một cách trực tiếp cho người sử dụng cũng như gán permission đó cho một nhóm mà người sử dụng đó là thành viên. Nếu một người sử dụng là thành viên của nhiều nhóm thì cần định rõ giấy phép thực tế áp dụng cho người sử dụng này dựa theo hai nguyên tắc sau: thứ nhất, mức giấy phép thực tế của người đó là mức giấy phép ít bị hạn chế nhất trong số permission đó; thứ hai, nếu trong số tất cả các mức giấy phép có giấy phép No Access thì mức giấy phép thực tế của người đó là No Access.

2.2. Chia sẻ các thư mục

Để thiết đặt một thư mục có được chia sẻ hay không chúng ta chỉ cần phải thực hiện một vài thao tác đơn giản. Nhưng trước khi thực hiện "một vài thao tác đơn giản" đó là cả một quá trình suy ngẫm công phu để hoạch định nên chính sách chia sẻ tài nguyên. Hệ thống có hoạt động tốt, đáp ứng vừa đủ các nhu cầu đa dạng của người sử dụng hay không là hệ quả tất yếu của quá trình hoạch định này.

a. Hoạch định các thư mục được chia sẻ

Trước khi chia sẻ chúng ta phải trả lời được câu hỏi: chia sẻ tài nguyên gì và cho ai. Để một mạng máy tính hoạt động trơn tru, người sử dụng hợp thức phải dễ dàng truy cập được các chương trình mạng, các dữ liệu dùng chung cũng như các thư mục chứa các tệp tài nguyên khác. Sau đây là một vài gợi ý:

- Hãy xác định xem người sử dụng sẽ truy cập vào những thư mục nào của mình. Hãy tổ chức lại các thư mục có cùng một mức bảo mật vào trong một thư mục. Chẳng hạn, nên đưa những thư mục chứa các tệp chỉ cho phép đọc vào trong một thư mục
- Sử dụng các tên chia sẻ trực quan để người sử dụng có thể dễ dàng đoán nhận và truy cập tới đó.
- Sử dụng các tên chia sẻ và tên thư mục đọc được bằng các HĐH của tất cả các máy trạm. Đối với các máy chạy HĐH Windows NT và Windows 95 thì tên chia sẻ và tên tệp có thể tối đa là 255 ký tự, còn đối với các máy chạy HĐH MS – DOS,

Windows 3.x và Windows for Workgroup thì tên chia sẻ và tên tệp đều phải tuân theo qui tắc 8.3

b. Hoạch định để gán permission trên các thư mục được chia sẻ

Cũng như việc chia sẻ, việc gán permission trên các thư mục được chia sẻ đến từng đối tượng, từng nhóm sử dụng cũng là một công việc đòi hỏi sự hoạch định và tính toán kỹ lưỡng. Sau đây là các gợi ý:

- Xác định nhóm nào có nhu cầu truy cập tới các tài nguyên gì và mức độ truy cập cần thiết đối với họ.
- Tạo ra các nhóm cục bộ (local group) đối với các tài nguyên được chia sẻ. Nếu các thư mục được chia sẻ nằm trên các máy chủ thành viên hay máy chạy Windows NT Workstation thì nhóm cục bộ đối với các tài nguyên được chia sẻ sẽ được tạo trên chính các máy đó. Nếu tài nguyên nằm trên các máy Điều khiển vùng thì nhóm cục bộ có thể tạo ra trên bất kỳ máy naò có chạy User Manager for Domains.
- Chỉ gán permission cho những nhóm thực sự có nhu cầu truy cập tới tài nguyên.
- Gán giấy phép hạn chế nhất cho nhóm cục bộ trên các tài nguyên, song phải đảm bảo cho phép đến mức để họ có thể thực hiện được công việc của mình. Chẳng hạn, nếu người sử dụng chỉ có nhu cầu đọc các tệp trên một thư mục thì chỉ nên gán giấy phép Read cho họ.
- Để đảm bảo tính bảo mật cao, hãy xoá bỏ giấy phép Full control của nhóm Everyone. Nếu muốn mọi người sử dụng đều có thể truy cập được tài nguyên, tốt nhất nên sử dụng nhóm Users. Trong một vùng nhóm Users chỉ bao gồm các tài khoản người sử dụng vùng mà chúng ta tạo ra. Trong một nhóm công tác, Users chứa mọi bgười sử dụng cục bộ

Ngoài ra, tuỳ theo tính chất của từng loại tài nguyên được chia sẻ (chương trình ứng dụng hay dữ liệu), chúng ta cần có chiến lược gán permission một cách phù hợp hơn. Đối với các mạng lớn, có thể có một hay nhiều máy chủ giữ vai trò lưu giữ các chương trình. Khi đó chúng ta cần:

- Tạo một thư mục được chia sẻ dùng để lưu các chương trình
- Gán giấy phép Full control cho nhóm Administrators để họ có thể truy cập và quản trị các chương trình.
- Xoá bỏ giấy phép Full control của nhóm Everyone và gán giấy phép Read cho nhóm Users để đảm bảo tính bảo mật cao.
- Gán giấy phép Change cho nhóm những người chịu trách nhiệm nâng cấp hay giải quyết các vấn đề về phần mềm.
- Với các thư mục chứa dữ liệu công cộng cũng như các dữ liệu nhạy cảm chúng ta cũng cần phải có những hoạch định tương tự. Thiết lập chia sẻ tới mức tệp chỉ có trong NTFS mà chúng ta sẽ xem ở phần sau.

CHƯƠNG IV. NTFS

Trong chương này chỉ đề cập đến các tính năng của hệ thống tệp NTFS, các ưu điểm và nhược điểm của nó.

1. Giới thiệu chung

Trong phần II.2 đưa ra tổng quan về các hệ thống tệp mà các HĐH của Microsoft hỗ trợ. Phần này đi sâu vào NTFS. NTFS hỗ trợ các tính năng sau:

- Hỗ trợ tên tệp dài. Các tên tệp và thư mục có thể dài tới 255 ký tự, bao gồm cả phần mở rộng.
- Hỗ trợ tính bảo mật cục bộ. Chúng ta nên dùng phân hoạch NTFS khi có yêu cầu bảo mật cho các máy chủ hoặc các máy cá nhân. NTFS hỗ trợ điều khiển truy cập và các đặc quyền riêng rất quan trọng để đảm bảo tính thống nhất của dữ liệu. Mặc dù các thư mục trên các máy chạy Windows NT/2000 có thể được gán thêm shared permission không phụ thuộc vào hệ thống tệp đang dùng, với các tệp và các thư mục NTFS, ta vẫn có thể gán permission để chúng được dùng chung hay không. NTFS là hệ thống tệp duy nhất trên Windows NT/2000 cho phép ta khả năng thiết đặt permission tới các tệp và các thư mục riêng.
- Kích thước của phân hoạch và tệp NTFS: phụ thuộc vào phần cứng của máy, cỡ của tệp lớn nhất nằm trong khoảng 4 GB và 64 GB. Do việc sử dụng không gian đĩa liên quan liên quan đến việc dùng NTFS, cỡ tối thiểu của một phân hoạch NTFS nên lớn hơn 50 MB.
- Một trong những đặc tính cơ bản của NTFS là khả năng nén tệp. Tỉ lệ nén, trên thực tế, thay đổi tuỳ vào bản chất của tệp được nén. Việc nén các tệp làm giảm cỡ của các tệp trong các ứng dụng văn bản và cỡ các tệp dữ liệu khoảng 50% và giảm cỡ của các tệp thực hiện khoảng 40%. Chương trình ứng dụng nào truy cập tệp được nén trên NTFS sẽ không biết rằng tệp này được giải nén khi yêu cầu. Tệp được nén khi được đóng hoặc lưu lại. Chỉ có hệ thống tệp NTFS mới có thể đọc được nội dung đã nén của dữ liệu. Khi một trình ứng dụng hoặc lệnh, Copy chẳng hạn, yêu cầu truy cập tệp, NTFS sẽ giải nén tệp trước khi hoạt động sao chép diễn ra. Sao chép tệp tin được nén vào một thư mục cũng được nén thật ra phải trải qua nhiều công đoạn, bao gồm giải nén, sao chép, và nén lại tệp. Có thể cho phép nén tệp/thư mục cá thể, hoặc nén toàn volume, nhưng không nên nén ở môi trường máy phục vụ
- Các tính năng phụ: NTFS có các tính năng phụ để nó trở thành một hệ thống tệp mạnh và năng động
    - Khả năng khôi phục lại dựa trên các tác vụ (transaction). NTFS có độ tin cậy cao. Nó là một hệ thống tệp có khả năng khôi phục bằng cách sử dụng việc cập nhật nhật ký tác vụ của tất cả các thư mục và các tệp một cách tự động. Nhật ký này được Windows NT sử dụng để lặp lại hoặc khôi phục các thao tác bị hỏng xảy ra do sự cố hệ thống bị hỏng hoặc mất điện.

- Hỗ trợ việc tái ánh xạ các chùm (cluster remapping). Nếu một lỗi xảy ra bởi có một cung (sector) bị hỏng trên đĩa cứng, NTFS sẽ cấp phát một chùm mới để thay thế một chùm có cung bị hỏng. Sau đó NTFS lưu các địa chỉ của chùm chứa cung bị hỏng, do vậy cung bị hỏng không được sử dụng lại
- Hỗ trợ các tệp Macitosh
- Hỗ trợ các yêu cầu POSIX

2. Dùng chế độ bảo mật của NTFS

Trên các phân hoạch NTFS, chúng ta có thể đặt permission NTFS trên các thư mục và tệp. Permission NTFS bảo mật các tài nguyên trên máy cục bộ và khi người sử dụng nối tới các tài nguyên đó trên mạng.

2.1. Một số khái niệm:

- Permission NTFS. Permission NTFS là permission chỉ có trên một phân hoạch được định dạng qua hệ thống tệp của Windows NT/2000. Permission cung cấp bảo mật ở mức độ cao hơn vì chúng có thể gán tới các thư mục và tới các tệp cụ thể. Permission NTFS cho thư mục và tệp được áp dụng cả với người sử dụng làm việc tại máy nơi có thư mục hoặc tệp lưu trữ và cả những người truy cập thư mục hoặc tệp đó từ mạng thông qua việc nối tới một thư mục được chia sẻ
- Kiểm soát (Audit). Ghi vào nhật ký những vấn đề liên quan đến bảo mật có thể xảy ra và sau này dùng chức năng Event Viewer (xem sự kiện) để xem lại
- Lập nhật ký các sự kiện (Event log). Khi một tệp hay một thư mục được sửa đổi, Dịch vụ Tệp nhật ký (Log File Service) theo dõi mọi thông tin về các thao tác redo hay undo cho việc sửa đổi. Những thông tin redo cho phép NTFS tạo lại các sửa đổi trong trường hợp hệ thống có sự cố. Thông tin về undo cho phép NTFS bỏ những sửa đổi nếu như nó không thể thực hiện được hoàn toàn chính xác. NTFS luôn cố gắng redo một giao dịch và chỉ undo khi không thể redo
- Quyền sở hữu (Ownership). Người tạo ra tệp hay thư mục hay nhà quản trị có toàn quyền sử dụng hay cho phép người khác sử dụng tài nguyên này. Chỉ có họ mới có thể thay đổi cấp độ truy cập áp dụng cho một đối tượng. Họ không thể chuyển giao quyền sở hữu cho người sử dụng khác mà chỉ có thể cấp quyền "giành quyền sở hữu". Người sử dụng phải dành quyền sở hữu thư mục/tệp sau khi đã được cấp quyền làm thế.

2.2. Sử dụng permission NTFS

Chúng ta có thể sử dụng permission NTFS để bảo vệ các nguồn tài nguyên, tránh những người sử dụng có thể truy cập máy theo những con đường sau:

- Một cách cục bộ, ngồi làm việc tại máy nơi lưu trữ các tài nguyên
- Từ xa, bằng cách nối tới một thư mục được chia sẻ

Chúng ta có thể đặt permission tệp tới các mức chi tiết. Thí dụ, chúng ta có thể đặt permission khác nhau cho mỗi tệp trong một thư mục. Chúng ta có thể cho phép một người sử dụng đọc nội dung của một tệp và thay đổi nó, cho phép người khác chỉ được đọc tệp và ngăn cản mọi người khác truy cập vào tệp. Trên một phân hoạch NTFS, người tạo ra một thư mục hoặc tệp là chủ nhân của thư mục hoặc tệp đó. Nếu người đó là thành viên quản trị thì nhóm quản trị trở thành chủ nhân của thư mục hoặc tệp đó. Chủ nhân luôn có thể gán và thay đổi permission trên các thư mục hoặc tệp của họ.

Permission NTFS được gán cho các tài khoản người sử dụng và các tài khoản nhóm theo cùng một cách mà permission chia sẻ đã gán. Một người sử dụng có thể được gán permission NTFS một cách trực tiếp hoặc như là thành viên của một hay nhiều nhóm
Permission thư mục NTFS được áp dụng như sau:

- Giống như các phép chia sẻ, permission NTFS cung cấp permission hiệu quả cho người sử dụng bao gồm các tổ hợp permission nhóm và permission ngươì sử dụng, trừ trường hợp ngoại lệ No Access. Giấy phép No Access đứng trên tất cả permission khác.
- Không giống như permission chia sẻ, permission NTFS bảo vệ các tài nguyên cục bộ và có thể được gán cho các thư mục và các tệp khác trong cùng cây phân cấp.

Permission tệp NTFS có quyền cao hơn permission được gán cho thư mục mà tệp đó thuộc vào. Thí dụ, nếu một người sử dụng có giấy phép Read tới một thư mục và giấy phép Write tới một tệp trong thư mục đó, thì người đó sẽ có thể ghi vào tệp nhưng không thể tạo ra một tệp mới trong thư mục đó.

2.3. Các mức giấy phép truy cập tệp NTFS

Permission truy cậpthể hiện sự điều khiển đối với người sử dụng tài nguyên tệp cũng như mức độ sử dụng. Đối với tệp, các mức độ giấy phép truy cậptương ứng với các quyền sau:

Giấy phép	R	X	W	D	P	O
No Access (Không được truy nhập)						
Read (Đoc)						
Change (Thay đổi)	☯	☯	☯	☯		
Full control (Toàn quyền)	☯	☯	☯	☯	☯	☯
Special File Access (Truy cập tệp đặc biệt)	Có thể chọn tổ hợp các chế độ trên					

Trong đó:
R: Xem được dữ liệu, thuộc tính, người sở hữu và các mức giấy phép

X: Chạy được tệp (thí dụ đối với các tệp .exe)
W: Ghi vào tệp hay thay đổi các thuộc tính của nó
D: Xoá tệp
P: Thay đổi permission đối với tệp
O: Lấy quyền sở hữu

2.4. Các mức giấy phép truy cập thư mục NTFS

Permission truy cậpthư mục thể hiện sự điều khiển đối với người sử dụng tài nguyên thư mục cũng như mức độ sử dụng thư mục đó. Đối với thư mục, các mức độ giấy phép truy cậptương ứng với các quyền sau:

Giấy phép	R	X	W	D	P	O
No Access (Không được truy nhập)						
List (Liệt kê)	☯					
Read (Đọc)	☯	☯				
Add (Thêm)		☯	☯			
Add & Read (Thêm & Đọc)	☯	☯	☯			
Change (Thay đổi)	☯	☯	☯	☯		
Full control (Toàn quyền)	☯	☯	☯	☯	☯	☯
Special File Access (Truy cậptệp đặc biệt)	Có thể chọn tổ hợp các chế độ trên					

Trong đó:
R: Hiện dữ liệu của thư mục, thuộc tính, người sở hữu và các mức giấy phép
X: Chạy được tệp trong thư mục (thí dụ đối với các tệp .exe)
W: Tạo các tệp mới trong thư mục, sửa đổi các tệp hay thay đổi các thuộc tính của thư mục
D: Xoá các tệp trong thư mục
P: Thay đổi các mức giấy phép đối với thư mục
O: Lấy quyền sở hữu

2.4.1. Sự thay đổi các mức giấy phép trong trường hợp sao chép hay di chuyển.

Thao tác sao chép (copy) một tệp khác với di chuyển (move) tệp đó. Một tệp được di chuyển sau khi sao tệp đó vào vị trí mới và xoá ở vị trí cũ. Trên thực tế, khi di chuyển một tệp thì chỉ có con trỏ trong cấu truc tệp thay đổi, còn về mặt vật lý, tệp vẫn ở nguyên chỗ cũ. Ta chỉ di chuyển được các tệp trên cùng một ổ logic. Còn sao tệp giữ

hai bản tại hai vị trí khác nhau. Theo logic thông thường, ngưpì ta đặt chính sách bảo mật khác nhau cho hai thao tác này.

Trong trường hợp sao chép, tệp đích được coi là một tệp mới và như vậy permission hiện có sẽ được thay thế bằng permission như đối với các tệp mới trong thư mục đích. Khi một tệp hay thư mục được sao chép, nó thừa hưởng những giấy phép của thư mục đích cùng với giấy phép ngầm định đối với một tệp mới. Người sao chép trở thành chủ sở hữu của bản sao và có mọi quyền

Trong trường hợp di chuyển, mọi mức giấy phép cũng như chủ sở hữu giữ nguyên như cũ. Muốn di chuyển, người sử dụng phải có quyền đưa các tệp cũng như thư mục con vào thư mục đích.

2.4.2. Tương quan giữa giấy phép cá nhân và giấy phép của nhóm
Giấy phép truy cập đối với cá nhân kết hợp với giấy phép nhóm mà anh ta là thành viên sẽ cho phép anh ta nhiều nhất từ đó, ngoại trừ trường hợp No Access, khi đó tổng hợp permission cũng là No Access.

Theo ngầm định, người tạo ra tệp hay thư mục chính là người chủ sở hữu tệp hay thư mục đó. Ta không thể đưa trực tiếp quyền sở hữu tệp hay thư mục cho một người khác nhưng ta có thể cho người đó được phép đoạt quyền sở hữu. Người quản trị (Administrator) luôn luôn có thể đoạt quyền sở hữu tệp hay thư mục, thậm chí ngay cả khi họ bị từ chối quyền truy cậptệp hay thư mục đó. Khi một thành viên của nhóm Administrators đoạt quyền sở hữu thì tất cả mọi thành viên của nhóm này cũng có quyền sở hữu tại đó.

Một chủ nhân không thể thay đổi quyền sở hữu của một tài nguyên mà họ làm chủ. Họ chỉ có thể gán cho người khác hoặc nhóm khác giấy phép lấy quyền sở hữu một tài nguyên. Tính bảo mật của tài nguyên đó vẫn được đảm bảo không bị những người sử dụng khác tạo hoặc sửa các tệp và sau đó tạo ra chúng như là thuộc quyền sở hữu của một người khác.

2.5. So sánh permission cục bộ và trên mạng
Một thư mục hay một tệp có thể chịu hai chế độ giấy phép: một trong chế độ chia sẻ, một trong chế độ bảo mật cục bộ của phân hoạch NTFS. Bảo mật cục bộ ở hệ thống tệp NTFS và có các mức giấy phép truy cập khác nhau. Tổ hợp sẽ lấy mức giấy phép yếu hơn.

2.6. Kết hợp permission chia sẻ và permission NTFS
Permission chia sẻ trên phân hoạch NTFS làm việc theo các tổ hợp giấy phép tệp và thư mục. Để cung cấp cho người sử dụng các quyền truy cậpđược vào các tài nguyên trên đĩa, các thư mục chứa các tài nguyên đó phải được chia sẻ. Một khi các thư mục đã

được chia sẻ, chúng ta có thể bảo vệ nó bằng cách gán permission chia sẻ tới người sử dụng và các nhóm công tác. Tuy nhiên, permission chia sẻ bị hạn chế trong việc bảo mật vì những lí do sau đây:

- Cho người sử dụng cùng một mức truy cậptới tất cả các thư mục bên trong thư mục được chia sẻ
- Không có tác dụng khi một người sử dụng nào đó đã truy cập được vào một tài nguyên một cách cục bộ bằng cách ngồi tại máy có đặt nguồn tài nguyên đó
- Không thể sử dụng để bảo mật các tệp có tính sở hữu riêng

Nếu một thư mục được chia sẻ nằm trên một phân hoạch NTFS thì ta có thể dùng permission NTFS để khoá một cách có hiệu quả hoặc thay đổi một truy cậpnào đó của một người sử dụng nào đó tới các thư mục được chia sẻ. Ta có được tính bảo mật ở mức cao nhất bằng cách kết hợp permission NTFS với permission chia sẻ

3. Mã hoá hệ thống tệp (Encrypting File System - EFS)

Một trong những đặc trưng an toàn cục bộ mới của Windows 2000 là EFS. EFS cho phép người sử dụng cất giữ số liệu an toàn hơn trên máy tính cục bộ của mình bằng cách mã và giải mã số liệu các tệp và thư mục trên NTFS khi cần thiết. EFS được thiết kế để cất giữ thông tin đặc biệt trên máy tính cục bộ và do đó nó không hỗ trợ khả năng chia sẻ các tệp mã hoá. EFS tích hợp vào NTFS làm cho việc quản lý mã hoá dễ dàng và trong suốt với người sử dụng. EFS tự động tạo cặp khoá mã cho người sử dụng nếu cặp khoá này chưa tồn tại. Cặp khoá gồm khoá công khai và khoá mật cho mỗi người sử dụng. Nếu cặp khoá mã cần tạo, và người sử dụng đăng nhập vào mạng theo mô hình miền, thì việc tạo khoá xảy ra tại bộ điều khiển (kiểm soát) miền; còn nếu người sử dụng đăng nhập vào mạng theo mô hình nhóm làm việc, thì việc tạo khoá xảy ra tại máy tính cục bộ. Bộ điều khiển (kiểm soát) miền hoặc máy tính sẽ thực hiện mã hoá kép bằng hai khoá này. Hệ thống máy tính sẽ yêu cầu EFS tạo số giả ngẫu nhiên cho tệp, được gọi là khoá mã tệp (File Encryption Key - FEK). FEK sau đó được dùng để giải mã số liệu tệp. Thuật toán mã hoá DES mở rộng (Exteded Data Encryption Standard - DESX) sử dụng FEK để mã hoá tệp. Các tệp mã hoá được cất giữ trên đĩa cứng. Đến đây, thuật toán mã hoá bằng khoá mật đã xong. Tuy nhiên quá trình này còn nhiều bước nữa. Để đảm bảo hoàn toàn an toàn cho FEK, nó được mã hoá bằng khoá công khai của người sử dụng; bằng cách đó đảm bảo chắc chắn rằng người sử dụng không dùng chung khoá giải mã. FEK sau khi mã được cất giữ cùng với tệp đã được mã hoá. Đến đây cả FEK và tệp được cất giữ an toàn. Các tài khoản của hệ thống khác mặc dù có các permission đối với các tệp mã hoá, ví dụ permission "giành quyền sở hữu", cũng không thể mở được tệp này nếu không có khoá mật của người mã hoá hoặc khoá mật được phục hồi của người đại diện. Tuy nhiên một vài permission khác lại không bị ảnh hưởng. Ví dụ người quản trị có permission xoá bỏ tệp mã hoá thì vẫn còn khả năng đó thậm chí khi anh ta không thể mở và đọc tệp.