Bảo mật trên hệ thống Linux

Security On Linux System

Power by: N.X.Bi O==(=========> ^($)^ Supporter Of VTF)
(E-mail: binhnx2000@yahoo.com | Home: http://www.vieteam.com/)

Mở đầu: Tôi là một Fan của Linux, một người yêu thích Security. Tôi rất thích Linux, đặc biệt là khả năng tuyệt vời của nó. Tôi viết tài liệu này chỉ với mục đích muốn chia sẻ với mọi người một chút hiểu biết ít ỏi của tôi về Security Linux...Không hề có bất cứ mục đích nào khác. Những gì tôi chia sẻ trong tài liệu này đều có nguồn gốc từ các: Magazine, Book, Site, Forum, List...về Linux Security trên thế giới. Những gì tôi cảm thấy hay và thực sự có ích, tôi đã thực hành thử và tìm cách ghi lại một cách ngắn gọn dễ hiểu nhất trong tài liệu này. Thiếu xót là điều không thể tránh khỏi, rất mong nhận được sự góp ý và chỉ bảo thẳng thắn từ phía các bạn. Đây chỉ là Version Demo của tài liệu. Nếu nhận được sự ủng hộ, đón nhận nhiệt tình cũng như sự góp ý và giúp đỡ thẳng thắn từ phía các bạn. Tôi sẽ tiếp tục hoàn thiện tài liệu này để phục vụ mọi người một cách tốt hơn.

Bạn có thể tham gia diễn đàn trao đổi, thảo luận về Unix/Linux với chúng tôi :

http://www.vieteam.com/vtf (Unix/Linux Section)

Lưu ý: Bài viết này chỉ mang tính chất học hỏi và trao đổi kinh nghiệm…Các bạn có thể tự do sử dụng nó, nhưng mong các bạn tôn trọng Copright một chút. Khi cần trích dẫn ở chỗ nào trong tài liệu. Vui lòng ghi rõ nguồn và tên người viết…Rất cảm ơn bạn đã quan tâm đến bài viết của tôi.

1) Về sự phân cấp, quyền hạn, sở hữu cho File

Sự phân cấp, quyền và sự sở hữu rõ ràng đơn giản đã tạo lên sức mạnh bảo mật của Unix/Linux. Vấn đề đầu tiên mà chúng ta cần kiểm tra có lẽ là sự phân cấp, quyền hạn, sở hữu các File trên hệ thống của bạn. Nếu không được cấu hình một cách chính xác điều này hết sức nguy hiểm. Cho lý do này bạn lên thường xuyên kiểm toán hệ thống File trên Server của bạn. Đặc biệt lên chú ý đến ID của root. Có một số chương trình cho phép người sử dụng trên hệ thống của bạn có thể tự do Set UID mà không cần root. Chắc tôi không cần nói, bạn cũng biết là phải làm gì với các chương trình loại này rồi chứ ? Bây giờ chúng ta tìm các File có sự phân cấp, quyền hạn không ổn định trên hệ thống của bạn và sau đó điều chỉnh lại giá trị an toàn cho chúng:

```
root@localhotst# find / -type f -perm +6000 -ls
59520 30 -rwsr-xr-x 1 root root 30560 Apr 15 1999 /usr/bin/chage
59560 16 -r-sr-sr-x 1 root lp 15816 Jan 6 2000 /usr/bin/lpq

root@localhotst# chmod -s /usr/bin/chage /usr/bin/lpq
root@localhotst# ls -l /usr/bin/lpq /usr/bin/chage
-rwxr-xr-x 1 root root 30560 Apr 15 1999 /usr/bin/chage
-r-xr-xr-x 1 root lp 15816 Jan 6 2000 /usr/bin/lpq
```

Các dòng lệnh trên tìm các File có UID root hay tương đương root. Tiếp đó gán thuộc tính chỉ cho phép root mới có quyền thực thi nó.

Chúng ta tiếp tục tìm những File cho phép ghi lại trên hệ thống của bạn. Điều gì sẽ xảy ra nếu kẻ tấn công có thể tự do thay đổi nội dung các File ?

root@localhost# find / -perm -2 ! -type l -ls

Trong các thao tác bình thường việc ghi, thay đổi nội dung File thường được thực hiện ở các thư mục như /**dev** và **/tmp**...Nếu bạn thấy ở các thư mục khác mà các File lại có thể tự do ghi lại được thì có lẽ là có vấn đề nảy sinh rồi đó.

Bạn cũng lên quan tâm đến các File không có chủ sở hữu (không thuộc bất cứ User hay Group nào). Tất nhiên là không ai sở hữu chúng thì kẻ tấn công rất có thể sẽ sở hữu chúng ;-(Để tìm các File không có chủ sở hữu bạn dùng lệnh:

root@localhost# find / -nouser -o -nogroup

Với việc sử dụng lệnh "**lsattr**" và "**chattr**" bạn có thể thay đổi đặc tính cho các File và thư mục dưới cấp độ cao cấp của một quản trị hệ thống như khả năng điều khiển quá trình xoá File, thay đổi File và với những tính năng khác mà lệnh "**chmod**" không thể thực hiện được.

Việc cấp phát quyền hạn sở hữu cho File theo một quy tắc thống nhất, trong suốt, không thay đổi...Tỏ ra có hiệu quả đặc biệt trong việc ngăn chặn quá trình xoá, thay đổi các tập tin Log của kẻ tấn công, hay việc cài đặt Trojan vào những File nhị phân Binnary trên hệ thống của bạn. Lệnh "**chattr**" được sử dụng để gán hay gỡ bỏ quyền hạn sở hữu cho File, thì lệnh "**lsattr**" được sử dụng để liệt kê chúng.

Các File Log cần phải được bảo vệ một cách hợp lý. Khi dữ liệu được ghi vào File Log một lần, nó sẽ không thể được phép chỉnh sửa hay thay đổi. Sở dỹ có nhu cầu này, bởi hiện tại có rất nhiều Script cho phép kẻ tấn công tấn công xoá bỏ, chỉnh sửa nội dung trên File Log. Để xiết chặt hơn an toàn cho File Log chúng ta cần sử dụng lệnh "**chattr**" và **"lsattr**" với một vài đối tượng:

root@localhost# chattr +i /bin/login
root@localhost# chattr +a /var/log/messages
root@localhost# lsattr /bin/login /var/log/messages
----i--- /bin/login
-----a-- /var/log/messages

Tóm lại! sau phần này bạn lên chú ý: Không bao giờ cho phép người sử dụng được phép chạy các chương trình Set UID, hay những chương trình khác có đặc quyền như root trên Home Directory của bạn. Luôn kiểm toán và quan tâm đến hệ thống File trên Server của bạn, đặc biệt là với những loại File có nguy cơ cao đã nêu ở trên.

- Bạn lên sử dụng tuỳ chọn **nouid** trong /**etc/fstab** để cho phép sự chỉnh sửa ghi lại ở các khu vực đã định với từng người sử dụng.
- Tính năng **noexec** và **nodev** cho các File trong Home Directory của người dùng để không cho phép họ tự động thực thi các chương trình hay tạo các thiết bị Block.

2) <u>Vô hiệu hoá các Service không sử dụng</u>

Để tránh tình trạng "đêm dài lắm mộng" bạn lên vô hiệu hoá và gỡ bỏ những chương trình, Service không dùng đến trên hệ thống của mình. Bạn có thể sử dụng các công cụ quản lý để

hiển thị danh sách những gói phần mềm nào đã được cài đặt để thực hiện việc này (Redhat Package Manager - Linux)

Về cơ bản! các Service được định nghĩa hoạt động bởi **inetd** (trên một số hệ thống Linux mới nó có thể là **xinetd**). Nội dung Service được định nghĩa hoạt động bởi **inetd** được chứa ở **/etc/inetd.conf** . Mỗi Service được định nghĩa đằng sau ký tự **"#"**...Bạn có thể vô hiệu hoá Service không sử dụng.

Thư mục **/etc/rc*.d** và **/etc/rc.d/rc*** là nơi chứa các Shell Script và các thông số để điều khiển sự thực hiện của Network và Service trong suốt thời gian nó hoạt động. Bạn có thể xoá bỏ hết những thứ liên quan đến những Service mà bạn không cần sử dụng. Đối với hệ thống Redhat, SuSE, Mandrake...bạn có thể sử dụng lệnh:

root@localhost#chkconfig --list
root@localhost#chkconfig --del <name>

Để hiển thị những Service nào đang hoạt động và xoá bỏ Service nào mà bạn muốn. Bạn muốn kiểm tra xem Service nào đó thực sự đã được gỡ bỏ khỏi hệ thống chưa ?

/bin/netstat -a -p --inet

Trên Redhat, SuSE, Mandrake...chương trình được sử dụng để quản lý các gói phần mềm là **/bin/rpm (Redhat Package Manager)**. Trên Debian là **/usr/bin/dpkg (Debian Package)**. Dưới đây là một số dòng lệnh cơ bản được dùng để quản lý các gói phần mềm. Dòng đầu sẽ là **rpm** và dòng thứ hai sẽ là **dpkg**:

Gỡ bỏ một gói phần mềm:

root@localhost# rpm -e <package-name>
root@localhost# dpkg -r <package-name>

Liệt kê danh sách những gói đã được cài đặt:

root@localhost# rpm -qvl <package-name.rpm>
root@localhost# dpkg -c <package-name.deb>

Liệt kê danh sách những gói đã được cài đặt với thông tin chi tiết cho mỗi gói:

root@localhost# rpm -qvia
root@localhost# dpkg -l

Liệt kê thông tin chính xác các File của gói đã được chỉ định:

root@localhost# rpm -qvpl <package-name.rpm>
root@localhost# dpkg -c <package-name.deb>

Hiển thị thông tin về một gói phần mềm:

root@localhost# rpm -qpi <package-name.rpm>
root@localhost# dpkg -l <package-name.deb>

Kiểm tra tính toàn vẹn cho một gói phần mềm:

root@localhost# rpm -Va
root@localhost# debsums -a

Cài đặt một gói phần mềm mới:

root@localhost# rpm -Uvh <package-name.rpm>
root@localhost# dpkg -i <package-name.deb>

3) Sự kiểm tra tính toàn vẹn của các gói phần mềm

Lệnh "**md5sum**" sự dụng thuật toán 128 bit để xác định chuỗi Finger Print của một gói phần mềm. Với mục đích đảm bảo sự toàn vẹn của các gói phần mềm từ nhà cung cấp đến người sử dụng. Nó có thể cho ta biết về sự thay đổi của các gói phần mềm trên hệ thống của bạn.

root@localhost# md5sum package-name
995d4f40cda13eacd2beaf35c1c4d5c2 package-name

Có lẽ bạn vẫn chưa hiểu được lợi ích thực sự của "**md5sum**" trong thế giới bảo mật. Tôi sẽ lấy một ví dụ đơn giản. Khi kẻ tấn công đã đột nhập được vào hệ thống của bạn, chúng sẽ cài đặt và sử dụng các Rootkit. Thực chất là các chương trình thông dụng của Admin như: netstat, ps, ls...đã được chỉnh sửa để cho ra thông tin sai che mắt bạn. Vậy làm thế nào để biết được điều này ?

Chẳng hạn như chuỗi MD5 mặc định của "**netstat**" khi cài đặt hệ thống SuSE Linux của tôi là "**995d4f40cda13eacd2beaf35c1c4d5c2**"

Bây giờ khi tôi chạy "**md5sum**" với "**netstat**" **:**

root@localhost# md5sum /usr/bin/netstat
995d4f40cda13eacd2beaf35c1c7d8c1 /usr/bin/netstat

Thông tin về chuỗi không khớp nhau, điều gì đã xảy ra vậy ? Câu trả lời này dành cho bạn.

4) Sử dụng Tripwire

Tripwire một chương trình theo dõi nhằm đảm bảo tính toàn vẹn của File bởi việc duy trì sự hoạt động của một cơ sở dữ liệu những File được cài đặt trên hệ thống...Cũng như sẽ cảnh báo khi chúng có sự thay đổi.

Khi cài đặt Tripwire sẽ đọc, thu thập thông tin về trạng thái các File trên hệ thống của bạn và ghi chúng vào một cơ sở dữ liệu. Sau này khi Tripwire chạy nó sẽ đối chiếu các FIle trên hệ thống của bạn với cơ sở dữ liệu chuẩn. Nếu có sự thay đổi nó sẽ thông báo cho bạn.

Có một File chính được sử dụng để cấu hình hoạt động tổng thể cho Tripwire. Thông thường với thông số mặc định nó cũng đã tỏ ra khá hiệu quả. Nếu như bạn không rành về Tripwire, bạn lên sử dụng thông số mặc định của nó. Dưới đây là một số dòng lệnh thông dụng

Tạo File nội quy từ một Text File

root@localhost#: /usr/TSS/bin/twadmin -m P policy.txt

Khởi tạo cơ sở dữ liệu theo File nội quy chính:

root@localhost#: /usr/TSS/bin/tripwire -init

Hiển thị cơ sở dữ liệu:

root@localhost#: /usr/TSS/bin/twprint -m d

Tạo thông báo kết quả theo ngày:

root@localhost#: /usr/TSS/bin/tripwire -m c -t 1 -M

Cập nhật cơ sở dữ liệu theo File nội quy và báo cáo hàng ngày:

**root@localhost#: /usr/TSS/bin/tripwire --update --polfile policy/tw.pol **
--twrfile report/<hostname>-<date>.twr

5) Sử dụng giao thức SSH

Nếu có thể tôi khuyên bạn lên cho Service "**Telnet**" nghỉ hưu và thay vào đó bằng Service "**SSH**". Mặc dù Telnet rất tuyệt nhưng nó lại không cung cấp khả năng mã hoá dữ liệu trên đường truyền, điều gì sẽ xảy ra khi có một Sniffer đặt ở đâu đó trên đường truyền.

Để cài đặt OpenSSH bạn cần Down gói *.rpm từ Site của hãng cung cấp phiển bản Linux mà bạn đang dùng về. Việc cài đặt từ gói *.rpm khá đơn giản, tôi không đề cập đến.

Lưu ý: Nhớ Down và cài thêm OpenSSL, bởi để hoạt động OpenSSH cần một số Lib của OpenSSL.

Chi tiết về việc sử dụng OpenSSH bạn có thể tham khảo bài viết "Open SSH" của tôi ở http://www.polarhome.com/~vicki

Về căn bản OpenSSH sử dụng những Public Key để đảm bảo sự an toàn. Public Key được cấp phát cho bất cứ hệ thống nào mà bạn muốn truyền thông an toàn:

host2$ ssh-keygen
Generating RSA keys: ...ooooooO....oooooooO
Key generation complete.
Enter file in which to save the key (/home/binhnx2000/.ssh/identity):
Created directory '/home/binhnx2000/.ssh'.
Enter passphrase (empty for no passphrase): <passphrase>
Enter same passphrase again: <passphrase>
Your identification has been saved in /home/binhnx2000/.ssh/identity.
Your public key has been saved in /home/binhnx2000/.ssh/identity.pub.
The key fingerprint is:
ac:42:11:c8:0d:b6:7e:b4:06:6a:a3:a7:e8:2c:b0:12 binhnx2000@host2

Tiếp đến Copy các Key để sử dụng:

host2$ mkdir -m 700 ~dave/.ssh
host2$ cp /mnt/floppy/identity.pub ~binhnx2000/.ssh/authorized_keys

Bây giờ từ hệ thống của bạn, nếu muốn Login vào hệ thống này chỉ việc phát lệnh:

root@localhost$ ssh host2
Enter passphrase for RSA key 'binhnx2000@localhost': <passphrase>

Last login: Sat Aug 15 17:13:01 2000 from localhost
No mail.
host2$

Ngoài khả năng cung cấp Shell Login an toàn, OpenSSH còn cung cấp cho bạn công cụ Copy và FTP một cách an toàn. Chẳng khi tôi muốn Copy file từ hệ thống của mình sang một hệ thống khác đã được chấp nhận:

root@localhost$ scp /tmp/file.tar.gz host2:/home/binhnx2000
Enter passphrase for RSA key 'binhnx2000@localhost:
file.tar.gz 100% |***| 98304 00:00**

Nếu có thể lên hướng dẫn và khuyến khích các User trên hệ thống của bạn sử dụng: OpenSSH thay cho Telnet và FTP.

6) Sử dụng TCP Wrappers

Trước khi Server FTP được chạy. Đầu tiên **tcpd** sẽ xác định những địa chỉ nguồn được cho phép, các kết nối sẽ được gửi đến Syslog để đối chiếu sau này. Nếu bạn muốn vô hiệu hoá tất cả các Service, bạn chỉ việc thêm dòng sau vào File **/etc/host.denny**

ALL:ALL

Để gửi E-mail đến nhà quản trị hệ thống và thông báo những lần kết nối bị thất bại, bạn thêm vào các dòng sau:

**ALL: ALL: /bin/mail **
-s "%s connection attempt from %c" admin@mydom.com

Nếu bạn muốn cho phép những địa chỉ tin cậy chạy những dịch vụ mà họ được phép, bạn hãy chỉnh sửa nội dung File /**etc/host.allow**

sshd: magneto.mydom.com, juggernaut.mydom.com
in.ftpd: 192.168.1.

Để đảm bảo an toàn bạn lên kiểm soát và điều khiển quá trình truy nhập một cách cẩn thận hơn. Sử dụng **tcpdchk** để kiểm tra sự truy nhập File, sử dụng Syslog để ghi lại những lần đăng nhập thất bại...Bạn lên điều khiển sự truy nhập cho hệ thống của mình theo nguyên tắc:

Sự truy cập chỉ được thực hiện khi Client/Deadmon có địa chỉ phù hợp với nội dung được cho phép trong /**etc/hosts.allow**

7) Sử dụng chế độ bảo mật mặc định của Kernel

Trong Kernel của một số hệ thống Linux mới hiện giờ có cấu hình sẵn một vài Rules chuẩn với mục đích cung cấp những thông số căn bản nhất để cấu hình cho hệ thống dành cho những Admin không có nhiều kinh nghiệm về bảo mật hệ thống. Các File và thông số đó thường được chứa ở **/proc/sys**. Về căn bản giao thức IPV4, bên trong **/proc/sys/net/ipv4** cung cấp các tính năng căn bản:

icmp_echo_ignore_all: Vô hiệu hoá tất cả các yêu phản hồi ICMP ECHO. Sử dụng tuỳ chọn này nếu như bạn không muốn hệ thống của mình trả lời các yêu cầu Ping.