

Securing Linux System

Arnon Rungsawang

fenganr@ku.ac.th

Massive Information & Knowledge Engineering

Department of Computer Engineering

Faculty of Engineering

Kasetsart University, Bangkok, Thailand.

Outline

- What is Linux?
- Some good reasons to use Linux!
- Uncertainty, doubt about Linux?
- Linux general security issue.
 - ...
 - ...



What is Linux?

- Small UNIX operating system that can be run on less expensive PC.
- Developed by Linus Trovalds at the U. of Helsinki in Finland since 1991.
- Under GNU GPL, its source code is free for everyone to download and use.

Some good reasons to use...

- No royalty or licensing fee, source code can be modified to fit your need.
- Open kernel source code makes it quite portable.
- Running on any kind of computer, aging from an old x386 with limited amount of RAM.
- True multi-tasking operating system.
- Being immunized against all kinds of viruses.

Uncertainty, doubt about...

- It's a toy operating system?
 - Used by Fortune500 companies, IBM, AMTRAK, NASA,...
- There's no support!
 - Coming with more than 12,000 pages of documentation.
 - Some commercial distributors provide 24/7 helpdesk support.
 - Online Linux community fixes many serious bugs within hours.

Linux general security issue

- Why we need to know?
- A secure server depends on how the administrator makes it.
- BIOS security
 - Bios security set a boot password.
 - Disallow booting from floppy and cdrom drives.
- Security policy...

Security Policy

- A list of what you consider allowable and what you do not consider allowable upon which to base any decisions regarding security.
- Security policy depends entirely on your definition of security!
 - What needs to be protected, and from whom?
 - Providing a balance between allowing your groups reasonable access to the information they require to do their jobs, and totally disallowing access to your information.

Some general considerations...

- How do you classify confidential or sensitive information?
- Exactly whom do you want to guard against?
- Do remote groups really need access to your system?
- Do you need access to the Internet?
- How much access do you want to allow to your system from the Internet?
- What action will you take if you discover a breach in your security?
- ...

Choosing the right password

- An unbreakable password does not exist.
- Given time and resources all passwords can be guessed either by **social engineering** or by **brute force**.
- Some rules to make passwords effective:
 - Preferably at least 8 chars including at least one numeral or special character.
 - Not be trivial, i.e. that are easy to guess or based on the groups' name, family,...
 - Should have an aging period, requiring to change within a specific time frame.
 - Should be revoked or reset after a limited number of concurrent incorrect retries.

The password length

■ Edit the /etc/login.defs

```
# *REQUIRED*
# Directory where mailboxes reside, _or_ name of file, relative to the
# home directory. If you _do_ define both, MAIL_DIR takes precedence.
# QMAIL_DIR is for Qmail
#
#QMAIL_DIR Maildir
MAIL_DIR /var/spool/mail
#MAIL_FILE .mail
# Password aging controls:
# PASS_MAX_DAYS Maximum number of days a password may be used.
# PASS_MIN_DAYS Minimum number of days allowed between password changes.
# PASS_MIN_LEN Minimum acceptable password length.
# PASS_WARN_AGE Number of days warning given before a password expires.
#
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_MIN_LEN 8
PASS_WARN_AGE 7
```

The root account

- The most privileged account on a Unix system.
- Has no security restriction imposed upon.
- Never log in on your server as “root” unless it is absolutely an instance that necessitates root access.
- Never sign in and leave yourself as “root”.

Set login time out for the root account

■ Edit the /etc/profile

```
# /etc/profile
# System wide environment and startup programs
# Functions and aliases go in /etc/bashrc
```

```
TMOUT=3600
```

```
PATH="$PATH:/usr/X11R6/bin"
ulimit -c 1000000
if [ `id -gn` = `id -un` -a `id -u` -gt 14 ]; then
    umask 002
else
    umask 022
fi
```

```
group=`id -un`
LOGNAME=$group
MAIL="/var/spool/mail/$group"
```

NFS service

- Be sure to configure the NFS export files (/etc/exports) with the most restrictive access possible, i.e. not using wildcards, not allowing root write access, and mounting read-only wherever possible, etc.

```
#!/etc/exports
/godzilla01 godzilla(ro,root_squash)
/godzilla01 mammoth(ro,root_squash)
/godzilla01 gargantua(ro,root_squash)
/godzilla01 merlin(ro,root_squash)
/godzilla01 toto(ro,root_squash)
/godzilla01 thor(ro,root_squash)
/godzilla01 trex(ro,root_squash)
/godzilla01 louis(ro,root_squash)
```

- For this change to take effect, you will need to run the following command on your terminal:

```
[root@godzilla]# /usr/sbin/exportfs -a
```

Disable console program access

- Disable all console-equivalent access to programs like `shutdown`, `reboot`, and `halt` for regular groups on your server.

```
[root@godzilla]# rm -f /etc/security/console.apps/halt
[root@godzilla]# rm -f /etc/security/console.apps/poweroff
[root@godzilla]# rm -f /etc/security/console.apps/reboot
[root@godzilla]# rm -f /etc/security/console.apps/shutdown
[root@godzilla]# rm -f /etc/security/console.apps/xserver
```

- Disable all PAM console access by commenting out all lines that refer to `pam_console.so` in the `/etc/pam.d` directory.

```
#/etc/pam.d/reboot
#%PAM-1.0
auth      sufficient /lib/security/pam_rootok.so
#auth    required /lib/security/pam_console.so
auth      required   /lib/security/pam_pwdb.so
account   required   /lib/security/pam_permit.so
```

Inetd super sever config

- Disable all unneeded services in `/etc/inetd.conf` and uninstall the packages to let attackers have one less place to look for a hole.

- Step 1: change the permission on this file to 600.

```
[root@godzilla]# chmod 600 /etc/inetd.conf
```

- Step 2: ensure that the owner is **root**.

```
[root@godzilla]# stat /etc/inetd.conf
```

```
File: "/etc/inetd.conf"  
Size: 3135                               Filetype: Regular File  
Mode: (0600/-rw-----)                 Uid: (  0/  root) Gid: (  0/  root)  
Device: 3,2  Inode: 194572  Links: 1  
Access: Tue Feb 19 07:47:03 2002(00002.13:31:10)  
Modify: Sat Oct 27 00:55:15 2001(00117.20:22:58)  
Change: Thu Feb 21 21:17:53 2002(00000.00:00:20)  
[root@godzilla]#
```

Inetd super sever config (2)

- Step 3: disable unneeded services in /etc/inetd.conf

```
#!/etc/inetd.conf
# <service_name> <sock_type> <proto> <flags> <group> <server_path> <args>
# Echo, discard, daytime, and chargen are used primarily for testing.
# To re-read this file after changes, just do a 'killall -HUP inetd'
#
#echo    stream  tcp    nowait  root    internal
#echo    dgram   udp     wait    root    internal
#discard stream  tcp    nowait  root    internal
#discard dgram   udp     wait    root    internal
#daytime stream  tcp    nowait  root    internal
#daytime dgram   udp     wait    root    internal
#chargen stream  tcp    nowait  root    internal
#chargen dgram   udp     wait    root    internal
#time    stream  tcp    nowait  root    internal
#time    dgram   udp     wait    root    internal
#
# These are standard services.
ftp      stream  tcp    nowait  root    /usr/sbin/tcpd  in.ftpd -l -a
telnet   stream  tcp    nowait  root    /usr/sbin/tcpd  in.telnetd
```

Inetd super sever config (3)

- Step 4: reread the /etc/inetd.conf

```
[root@godzilla]# killall -HUP inetd
```

or

```
[root@godzilla]# /etc/rc.d/init.d/inetd restart
```

- Step 5: set immutable flag for /etc/inetd.conf

```
[root@godzilla]# chattr +i /etc/inetd.conf
```

xinetd super sever in RH7.2

```
[root@tutu]# vi /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
# unencrypted groupname/password pairs for authentication.
service telnet
{
    disable                = no
    only_from              = 192.168.33.*
    flags                  = REUSE
    socket_type            = stream
    wait                   = no
    group                  = root
    server                 = /usr/sbin/in.telnetd
    log_on_failure        += groupID
}
```

TCP_wrappers

- Using TCP_wrappers to make securing your servers against outside intrusion is a lot simpler and painless than you would expect!
- TCP_wrappers is controlled from 2 files, the search stops at the first match.
 - Grant access when a (daemon, client) pair matches an entry in the /etc/hosts.allow file.

```
[root@godzilla]# vi /etc/hosts.allow  
sshd:208.164.186.1 gate.openna.com
```

- Otherwise, deny when a (...) pair matches an entry in the /etc/hosts.deny file.

```
[root@godzilla]# vi /etc/hosts.deny  
#Deny access to everyone.  
ALL:ALL@ALL,PARANOID
```

- Otherwise, access will be granted.

The /etc/host.conf file

- Tell the resolver library what services to use, and in what order, to resolve the hostname to IP address.

```
[root@godzilla]# vi /etc/host.conf
#Lookup names via DNS first then fall back to /etc/hosts
order bind,hosts
#If you have a machine with multiple IP addresses, turn this on
multi on
#Check for IP address spoofing
nospoof on
```

The /etc/services and /etc/securetty files

- /etc/services file enables server and client programs to convert service names to corresponding standard port numbers.
- Immunize the /etc/services file to improve security.

```
[root@godzilla]# chattr +i /etc/services
```

```
[root@godzilla]# less /etc/services
```

```
# /etc/services:
```

```
# $Id: services,v 1.4 2000/01/23 21:03:36 notting Exp $
```

```
# Network services, Internet style
```

```
# Note that it is presently the policy of IANA to assign a single well-known
```

```
# port number for both TCP and UDP; hence, most entries here have two entries
```

```
# even if the protocol doesn't support UDP operations.
```

```
# Updated from RFC 1700, ``Assigned Numbers'' (October 1994). Not all ports
```

```
# are included, only the more common ones.
```

```
tcpmux    1/tcp                # TCP port service multiplexer
```

```
echo      7/tcp
```

```
echo      7/udp
```

```
discard   9/tcp    sink null
```

```
discard   9/udp    sink null
```

The /etc/securetty files

- The /etc/securetty file specifies which tty devices the "root" group is allowed to login on.
- We recommend to allow "root" to log in only on one tty device and use the "su" command to switch to "root" if you need more.

```
[root@godzilla]# vi /etc/securetty
```

```
tty1
```

```
#tty2
```

```
#tty3
```

```
#tty4
```

Disable all default vendor accounts

- Check after each upgrade or new software installation, as Linux provides those accounts for various system activities. The more unused accounts you have, the easier it is to access to your system.

- Step 1: delete groups listed below.

```
[root@godzilla]# groupdel adm  
[root@godzilla]# groupdel lp  
[root@godzilla]# groupdel sync  
[root@godzilla]# groupdel shutdown  
[root@godzilla]# groupdel halt  
[root@godzilla]# groupdel news  
[root@godzilla]# groupdel uucp  
[root@godzilla]# groupdel operator  
[root@godzilla]# groupdel games  
[root@godzilla]# groupdel gopher  
[root@godzilla]# groupdel ftp
```

Disable all default vendor accounts (2)

- Step 2: delete groupgroups listed below.

```
[root@godzilla]# groupdel adm  
[root@godzilla]# groupdel lp  
[root@godzilla]# groupdel news  
[root@godzilla]# groupdel uucp  
[root@godzilla]# groupdel games  
[root@godzilla]# groupdel dip  
[root@godzilla]# groupdel pppusers  
[root@godzilla]# groupdel popusers  
[root@godzilla]# groupdel slipusers
```

- Step 3: set immutable bit on these files.

```
[root@godzilla]# chattr +i /etc/passwd  
[root@godzilla]# chattr +i /etc/shadow  
[root@godzilla]# chattr +i /etc/group  
[root@godzilla]# chattr +i /etc/gshadow
```

Blocking anyone to su to root

- Restrict "su" command to certain users.
 - Step 1: add the following 2 lines to the top of your su configuration file in /etc/pam.d/ directory.

```
#%PAM-1.0
```

```
auth sufficient /lib/security/pam_rootok.so debug
```

```
auth required /lib/security/pam_wheel.so group=wheel
```

```
auth required /lib/security/pam_pwdb.so shadow nullok
```

```
account required /lib/security/pam_pwdb.so
```

```
password required /lib/security/pam_cracklib.so
```

```
password required /lib/security/pam_pwdb.so shadow use_authtok nullok
```

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

```
session optional /lib/security/pam_xauth.so
```

- Step 2: add some users allowed to "su" to root account by making them to be a member of the "wheel" group.

```
[root@godzilla]# usermod -G10 toto
```

Resource limits

- Use limits.conf file located under /etc/security directory to control and limit resources so that users can't perform denial of service attacks (number of processes, amount of memory, etc).

- Step 1: edit the limits.conf to be contained by these lines

- * hard core 0
- * hard rss 5000
- * hard nproc 20

- Step 2: edit /etc/pam.d/login file

```
#%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  required /lib/security/pam_limits.so
#session  optional /lib/security/pam_console.so
```

More control on mounting a file system

- Information related to security options in the `/etc/fstab`

`defaults` allow everything (quota, read-write, and `suid`) on this partition.

`noquota` do not set users quotas on this partition.

`nosuid` do not set SUID/SGID access on this partition.

`nodev` do not set character or special devices access on this partition.

`noexec` do not set execution of any binaries on this partition.

`quota` allow users quota on this partition.

`ro` allow read-only on this partition.

`rw` allow read-write on this partition.

`suid` allow SUIF/SGID access on this partion.

- Edit `/etc/fstab` depending on your needs. For example:

```
/dev/sda11      /tmp      ext2 defaults,rw,nosuid,nodev,noexec 1 2
/dev/sda6       /home     ext2,defaults,rw,nosuid,nodev      1 2
```

Change binary RPM default permission

- Change the default permission of the “rpm” binary from 755 to 700 so that non-root users can't use rpm program to query, install etc.

```
[root@godzilla]# chmod 700 /bin/rpm
```

Shell logging

- Let a cracker have less chance to find some password typed by mistake in plain text in the “~/.bash_history” file.
 - Step1 : edit /etc/profile to have these lines
`HISTFILESIZE=20`
`HISTSIZ=20`
 - Step2 : add into the /etc/skel/.bash_logout the following
`rm -f $HOME/.bash_history`

The /etc/lilo.conf file

- LILO acts as a boot manager that manages the boot process. It must be protected against single mode boot.
 - Step 1: edit /etc/lilo.conf to have these 3 lines

```
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
timeout=0
default=linux
restricted
password=<password>
image=/boot/vmlinuz-2.2.12-10
    lable=linux
    initrd=/boot/initrd-2,2,12-10.img
    root/dev/sda6
    read-only
```

The /etc/lilo.conf file (2)

- Step 2: let /etc/lilo.conf to be able to be read only by root.
[root@godzilla]# **chmod 600 /etc/lilo.conf**
- Step 3: update the /etc/lilo.conf configuration file.
[root@godzilla]# **/sbin/lilo -v**
- Step 4: set the immutable flag of the /etc/lilo.conf file.
[root@godzilla]# **chattr +i /etc/lilo.conf**

Disable the Control-Alt-Delete keyboard shutdown command

- Edit the `/etc/inittab` and change the line:

```
# Trap CTRL-ALT-DELETE  
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

to read:

```
# Trap CTRL-ALT-DELETE  
#ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

- Let init process to know the change in the `/etc/inittab` file.

```
[root@godzilla]# /sbin/init q
```

Keep another copy of all-important logs

- Log of all telnet, mail, boot messages and ssh connections from your server to the attached printer.
 - Add this line at the end of /etc/syslog.conf file.

```
authpriv.*;mail.*;local7.*;auth.*;daemon.info /dev/lp0
```

- Keep those logs into a remote machine.
 - Step 1: Enable on the remote server the facility to receive messages from the network by editing the /etc/rc.d/init.d/syslog daemon to read

```
daemon syslogd -r -m 0
```

- Step 2: Restart your syslog daemon on the remote host.

```
[root@tutu]# /etc/rc.d/init.d/syslog restart
```

Keep another copy of all-important logs (2)

- Step 3: If we have a firewall on the remote server, we must add or verify the existence of the following lines.

```
ipchain -A input -i $EXT_INTERFACE -p udp \  
-s $SYSLOG_CLIENT -d $IPADDR 514 -j ACCEPT
```

where EXT_INTERFACE="eth0" in the firewall file.
IPADDR=IP address of the remote server.
SYSLOG_CLIENT=IP address of the sending log client.

- Step 4: Restart the firewall on the remote machine.

```
[root@tutu]# /etc/rc.d/init.d/firewall restart
```

- Step 5: Add this line to the local server at the end of /etc/syslog.conf file.

```
authpriv.*;mail.*;local7.*;auth.*;daemon.info @tutu
```

Keep another copy of all-important logs (3)

- Step 6: Restart your syslog daemon on the local host.

```
[root@godzilla]# /etc/rc.d/init.d/syslog restart
```

- Step 7: Same as the remote host, we must verify the existence of the following line in the firewall script of the local host.

```
ipchain -A output -i $EXT_INTERFACE -p udp \  
-s $IPADDR 514 -d $SYSLOG_SERVER 514 -j ACCEPT
```

where EXT_INTERFACE="eth0" in the firewall file.
IPADDR=IP address of the local server.
SYSLOG_SERVER=IP address of the remote host.

- Step 8: Restart the firewall on the local machine.

```
[root@godzilla]# /etc/rc.d/init.d/firewall restart
```

The permission of the `/etc/rc.d/init.d/*` files

- Allow only root to read, write, execute the script files that are responsible for starting and stopping all your normal processes that need to run at boot time.

```
[root@godzilla]# chmod -R 700 /etc/rc.d/init.d/*
```

- If you install a new or update a program that use the init system V script located under `/etc/rc.d/init.d/` directory, don't forget to change or verify the permission of that script file again.

Remove the distribution's banner

- Information about Linux distribution name, version, kernel version, and the name of the server can be exploited by any hacker.
- We'd rather just prompt users with a "login:" prompt.
 - Step 1: Edit /etc/rc.d/rc.local file and place "#" in front of the following lines as shown.

```
# This will overwrite /etc/issue at every boot. So, make any changes you  
# want to make to /etc/issue here or you will lose them when you reboot.  
#echo "" > /etc/issue  
#echo "$R" >> /etc/issue  
#echo "Kernel $(uname -r) on $a $SMP$(uname -m)" >> /etc/issue  
  
#cp -f /etc/issue /etc/issue.net  
#echo >> /etc/issue
```
 - Step2: Remove "issue.net" and "issue" in /etc directory.

```
[root@godzilla]# rm -f /etc/issue /etc/issue.net
```

Bits from root-owned programs

- A regular user can run a program as he is root if that program has the SUID (-rwsr-xr-x) or SGID (-rwxr-sr-x) bit enable.
- It's important to remove the 's' bits from root-owned programs that won't absolutely require such privilege for daily normal usage.

- Step 1: Find all root-owned "s" bit enabled files.

```
[root@godzilla]# find / -type f \( -perm -04000 -o -perm -02000 \)
\ -exec ls -lg {} \;
```

```
-rwsr-xr-x 1 root root 14188 Mar 7 2000 /bin/su
-rwsr-xr-x 1 root root 17968 Mar 6 2000 /bin/ping
-rwsr-xr-x 1 root root 56208 Feb 3 2000 /bin/mount
-rwsr-xr-x 1 root root 26608 Feb 3 2000 /bin/umount
-rwxr-sr-x 1 root root 3860 Mar 8 2000 /sbin/netreport
-r-sr-xr-x 1 root root 26126 Feb 6 2000 /sbin/pwdb_chkpwd
-r-sr-xr-x 1 root root 27114 Feb 6 2000 /sbin/unix_chkpwd
```

- Step 2: Disable the suid bits on selected programs.

```
[root@godzilla]# chmod a-s /usr/ping
```

Kernel tunable parameters

- With newer version of RH Linux, such as 6.2, all kernel parameters available under the /proc/sys subdirectory can be configured at runtime.
- We can use the /etc/sysctl.conf file to modify and set some kernel parameters concerning **network security** at runtime.
- Prevent system from responding to ping request.
 - Step 1: Edit the /etc/sysctl.conf and add the following line.
#Enable Ignoring Ping Request
net.ipv4.icmp_echo_ignore_all=1
 - Step 2: Restart all network devices.
[root@godzilla]# **/etc/rc.d/init.d/network restart**
- Prevent system from responding to broadcast request.
 - Step 1: Edit the /etc/sysctl.conf and add the following line.
#Enable Ignoring Ping Broadcast
net.ipv4.icmp_echo_ignore_broadcasts=1
 - Step 2: Restart all network devices.
[root@godzilla]# **/etc/rc.d/init.d/network restart**

Kernel tunable parameters (2)

- Prevent system from IP routing protocol hole
 - Step 1: Edit the /etc/sysctl.conf and add the following line.
#Disable IP Source Routing
net.ipv4.conf.all.accept_source_route=0
 - Step 2: Restart all network devices.
[root@godzilla]# **/etc/rc.d/init.d/network restart**
- Enable TCP SYN Cookie Protection
 - Step 1: Edit the /etc/sysctl.conf and add the following line.
#Enable TCP SYN Cookie Protection
net.ipv4.tcp_syncookies=1
 - Step 2: Restart all network devices.
[root@godzilla]# **/etc/rc.d/init.d/network restart**
- Disable ICMP redirect acceptance
 - Step 1: Edit the /etc/sysctl.conf and add the following line.
#Disable ICMP Redirect Acceptance
net.ipv4.conf.all.accept_redirects=0
 - Step 2: Restart all network devices.
[root@godzilla]# **/etc/rc.d/init.d/network restart**

Kernel tunable parameters (3)

- Enable always defragging protection
 - Step 1: Edit the /etc/sysctl.conf and add the following line.
`#Enable Always Defragging Protection`
`net.ipv4.ip_always_defrag=1`
 - Step 2: Restart all network devices.
`[root@godzilla]# /etc/rc.d/init.d/network restart`
- Enable bad error message protection
 - Step 1: Edit the /etc/sysctl.conf and add the following line.
`#Enable Bad Error Message Protection`
`net.ipv4.icmp_ignore_bogus_error_responses=1`
 - Step 2: Restart all network devices.
`[root@godzilla]# /etc/rc.d/init.d/network restart`
- Enable IP spoofing protection
 - Step 1: Edit the /etc/sysctl.conf and add the following line.
`#Enable IP Spoofing Protection`
`net.ipv4.conf.all.rp_filter=1`
 - Step 2: Restart all network devices.
`[root@godzilla]# /etc/rc.d/init.d/network restart`

Unusual or hidden files

- Look everywhere on the system for unusual or hidden files (files that start with a period, such as ``....'``, ``...'`` and are normally not shown by the `"ls"` command), as they can be used to hide tools and information (password cracking programs, password files from other systems, etc.).

```
[root@godzilla]# find / -name "." -print -xdev  
[root@godzilla]# find / -name ".*" -print -xdev | cat -v
```

All SUID/SGID bit enabled files

- A favorite trick of crackers is to exploit SUID “root” programs, and leave it as a backdoor to get in the next time.
- Find all SUID and SGID programs on your system, and keep track of what they are so that you are aware of any changes, which could indicate a potential intruder.

```
[root@godzilla]# find / -type f \( -perm -04000 -o -perm -02000 \)  
                \-exec ls -lg {} \;
```

Group and world writable files and directories

- Group and world writable files and directories can be security hole, as hackers can add or delete files at will.

- Step 1: Locate all group&world writable file.

```
[root@godzilla]# find / -type f \( -perm -2 -o -perm -02 \  
                \-exec ls -lg {} \;
```

- Step 2: Locate all group&world writable directories.

```
[root@godzilla]# find / -type d \( -perm -2 -o -perm -02 \  
                \-exec ls -ldg {} \;
```

Unowned files

- Unowned files may also be an indication that an intruder has accessed your system.
- Sometime you may uninstall a program and get an unowned file or directory related to that software.

```
[root@godzilla]# find / -nouser -o nogroup
```

Finding .rhosts files

- Remember that a cracker only needs one insecure account to potentially gain access to your entire network using .rhosts files.
 - Locate all .rhosts files on your system.
`[root@godzilla]# find /home -name .rhosts`
 - Use a cron job to periodically check and report via email all .rhosts files.
 - Step 1: Create a find_rhosts_files script under /etc/cron.daily directory.

```
#!/bin/sh
/usr/bin/find /home -name .rhosts | (cat << EOF
This is an automated report of possible existent .rhosts files on the server.
New detected .rhosts files under /home directory include:
EOF
cat
) | /bin/mail -s "Content of .rhosts file audit report" root
```
 - Step 2: Make the above script file executable and owned by root.
`[root@godzilla]# chmod 755 /etc/cron.daily/find_rhosts_files`
`[root@godzilla]# chown 0.0 /etc/cron.daily/find_rhosts_files`

Linux general security issue