Securing Linux

John Kristoff

jtk@depaul.edu
http://condor.depaul.edu/~jkrystof/
+1 312 362-5878
DePaul University
Chicago, IL 60604
Starting comments

- A mish-mash of mostly Linux-specific security tips
- This is NOT a complete survey of all there is to do
- Distro agnostic, but speaker mostly uses Debian
- Maintain installation/change documentation offline
Why or why not Linux?

- Learning curve is usually high
- With a good initial install, re-installs should be rare
- Reboots due to OS crashes should be infrequent
- Preventing a remote compromise is usually easy
- Preventing local attacks can be nearly impossible
Installation decisions

• Server, client, multiuser system, a mix?
• Which distribution?
• Will this be a multi-boot system?
• What remote services will be available, if any?
Things to have before an install

- Hardware details in hard copy format
- IP addressing and DNS naming requirements
- Installation media or trusted remote sites
Securing the hardware

- Physical security is often the weakest link
- Limit physical access to the hardware
- Use BIOS passwords
  - Suggestion: use hardware code + secret key
- After install, set hard drive to first boot device
- Comment out the 'ca:ctrlaltdel' line in /etc/inittab
Partitioning strategy

- By default, some distros put everything under '/'
- It's probably better not to, I recommend at least:
  - /
  - /usr
  - /home
  - /var
  - /tmp
  - swap
Partition mounts

• Options help limit unauthorized system abuse
• Options configured in /etc/fstab, for example:

/dev/hda    /     ext3  errors=remount-ro
/dev/hda2   /usr   ext3  defaults,ro,nodev
/dev/hda3   /home  ext3  defaults,nodev,nosuid
/dev/hda5   /var   ext3  defaults,nodev,nosuid,noexec
/dev/hda6   /tmp   ext3  defaults,nodev,nosuid,noexec
The LILO bootloader

- Configuration options set in /etc/lilo.conf
- File should be read/write only by root
- Some recommended options:

  delay=<x>  # set to 0 if no other OSes exist
  restricted  # boot-time options require password
  password=<x>  # password for non-default boot
The GRUB bootloader

- Configuration file is /boot/grub/menu.lst
- Can be read by all, if paranoid restrict to root
- Some recommended options:

  password --md5 <pw>  # boot options require password
  timeout <x>          # boot delay
  lock                 # password protect insecure OS
Startup scripts

• Found in /etc/rc.d/init.d (/etc/init.d in Debian)
• Links to init.d scripts found in /etc/rc<0-6>.d
• /etc/inittab sets run level and startup scripts to run
• Know your run level and which scripts get loaded
• Many scripts start network listening services
• For security, the fewer services enabled the better
Managing startup scripts

- Use distro tools (chkconfig, update-rc.d)
- Remove unnecessary packages/software completely
- Delete startup scripts/links
- Rename links of startup scripts in your run level
- Example startup scripts:

  /etc/rc5.d/S80sendmail -> ../init.d/sendmail
  /etc/rc0.d/K20inetd -> ../init.d/inetd
  /etc/rc3.d/.s20apache -> ../init.d/apache
Services to consider disabling

-amd/autofs
- apache/httpd
- inetd/xinetd
- linuxconf
- lpd
- named
- netfs
- nfs
- nfslock
- portmap
- routed
- rstat/ruser/rwall/rwho
- sendmail
- smbd
- snmpd
- yp*
Examing listeners with netstat

- `netstat -tuna`

<table>
<thead>
<tr>
<th>Proto</th>
<th>Local Address</th>
<th>Foreign Addr</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp</td>
<td>0.0.0.0:22</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>192.0.2.1:80</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>192.0.2.1:22</td>
<td>192.0.2.2:1024</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>udp</td>
<td>192.0.2.1:123</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
</tr>
</tbody>
</table>
Examining listeners with lsof

- `lsof -ni +M`

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>TYPE</th>
<th>NODE</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntpd</td>
<td>IPv4</td>
<td>UDP</td>
<td>*:ntp</td>
</tr>
<tr>
<td>ntpd</td>
<td>IPv4</td>
<td>UDP</td>
<td>127.0.0.1:ntp</td>
</tr>
<tr>
<td>ntpd</td>
<td>IPv4</td>
<td>UDP</td>
<td>192.0.2.1:ntp</td>
</tr>
<tr>
<td>ntpd</td>
<td>IPv4</td>
<td>UDP</td>
<td>127.0.0.1:1024-&gt;127.0.0.1:ntp</td>
</tr>
</tbody>
</table>
TCP Wrappers

- Access control and logging for network services
- Use `/etc/hosts.allow` to permit services/hosts
- Use `/etc/hosts.deny` to prohibit services/hosts
- Common services protected by `tcp_wrappers`:
  - `ftp`, `imap`, `pop`, `ssh`, `telnet`, `tftp`
TFTP with TCP Wrappers

# /etc/inetd.conf
tftp dgram udp wait root /usr/sbin/tcpd \
in.tftpd -s /tftpboot

# /etc/hosts.allow
in.tftpd: 192.0.2.0/255.255.255.0

# /etc/hosts.deny
ALL: ALL
Logging and syslog

- Logs found in /var/log
- /etc/syslog.conf used to configure various options
- /etc/logrotate.conf configures log rotation
- tail -f /var/log/<logfile> to watch a log in realtime
- Get familiar with what are normal log messages
- Use a remote logging host if possible (syslog.conf)

  * .debug @loghost.example.com
Time synchronization

- Use NTP to maintain precise timestamps
- Example /etc/ntp.conf configuration:

```plaintext
restrict default notrust nomodify noquery notrap nopeer ignore
server ntp1.example.com
server ntp2.example.com
server ntp3.example.com
restrict 192.0.2.0 mask 255.255.255.0 nomodify noquery notrap nopeer
```
User account security

- Always use shadow passwords and MD5 hashing
- Avoid root, use groups and sudo where appropriate
- Disable unnecessary user accounts (e.g. uucp)
- Use long and strong passwords
- Example password creation strategy:

  4 score & 7 years ago our fathers brought 4th, upon this continent, a new nation, conceived in liberty, & dedicated 2 the proposition
User command line history

- Setup /etc/profile to make .bash_history permanent:

  HISTFILE=~/bash_history
  HISTSIZE=1000000000000000
  HISTFILESIZE=10000000000000000
  readonly=HISTFILE
  readonly=HISTSIZE
  readonly=HISTFILESIZE
  export HISTFILE HISTSIZE HISTFILESIZE
File permission suggestions

- Set umask in `.bash_profile` to 0037 or 0077
- Restrict read/write access to system files
- Know the suid/sgid permissions on your system
  - `find / -perm +4000`
  - `find / -perm +2000`
- Use file attributes to your advantage, for example:
  - `chattr +a /home/<user>/.bash_history`
  - `chattr +i /etc/inetd.conf`
Tripwire

- File system integrity and auditing tool
- Config/database tends to be customization-heavy
- Run from a remote system or read-only media
  - See security.uchicago.edu's sshtrip tool
- Example config file entries for 1.x version:

```
/var      R      # default monitoring flags
/var/log  L-i    # for files that change often
```
AIDE

- File system integrity and auditing tool like Tripwire
- Adds powerful regex capability for filespec
- Example config file entries:

```
/var R # default flags
/var/log/.*\log p+n+u+g # for log files
/var/log/.*\.log\.[0-9] # for archived log files
```
rpm -Va

- Compare changes from package install time
- Examines size, MD5, ownership, timestamp, etc.

missing /root/.bash_profile
S.5....T c /etc/logrotate.conf
..?..... c /etc/sudoers
Update system and software

- RedHat has up2date
- Debian has apt
- Some prefer to build from source
- Get on *-announce mailing lists for distro and apps
Verifying software

- Almost no one verifies downloaded software
- A few distros do some automated validation
- To validate MD5 hashes and PGP signatures:

  md5sum <filename>
  gpg --key-server pgp.mit.edu --recv-key <keyid>
  gpg --verify <signature-file>
Firewalling and packet filtering

- Used to provide low level packet access control
- Can ensure unauthorized services are inaccessible
- All hosts should probably do some filtering
- Example iptables config to block < 1024 ports:

  ```
ip tables -A INPUT -p tcp --dport 0:1023 -j DROP
ip tables -A INPUT -p udp --dport 0:1023 -j DROP
ip tables -A INPUT -j ACCEPT
```
Use SSH for remote access

- Eliminates plain text from the network
- Use only SSH version 2
- Can be used with TCP Wrappers
- Replaces remote terminal access (TELNET)
- Replaces file transfer and remote copy (ftp, rcp)
- Tunnel insecure protocols over an SSH connection
  - e.g. pop3, smtp, nfs, telnet, ftp
OpenSSH server

- Requires OpenSSL
- Install OpenSSH using privilege separation
- Some recommended sshd_config config settings:

Protocol 2
PermitRootLogin no
AllowUsers <user1> <user2> <usern>
AllowGroup <group1> <group2> <groupn>
Banner /etc/motd
Miscellaneous thoughts

- For multiuser systems, consider a restricted shell
- Use chroot where possible
- Have nmap/nessus audits done
- If someone really wants to get in, they will
References

- SANS Securing Linux Step-by-Step
  - Somewhat dated
- Linux Administration Handbook
  - Prentice Hall ISBN: 0130084662
- Linux System Security
  - Prentice Hall ISBN: 0130470112
- Securing Debian Manual:
  www.debian.org/doc/manuals/securing-debian-howto/