Linux Security Ideas and Tips

Hugh Brown
Sr. Systems Administrator
ITS Enterprise Infrastructure

University of Iowa

October 8, 2014
Introduction

Who am I?
- Linux User and Sysadmin for 15+ years
- Worked in big business, small business and university environments
- Taught Linux courses for the UI

My assumptions:
- You are running Linux in a server capacity
- You are only running a handful of systems
- You aren’t managing 100s of Linux systems
- The command line isn’t scary
- Specifics may be Redhat-centric
Overview

- Where to get Linux
- Minimalism
- Baseline
- Patching
- Least privilege
- User management
- Remote access and ssh hardening
- Packet Filtering
- Logging
- AV protection
- Secure applications
- Further reading
Where to get Linux

- Redhat: https://helpdesk.its.uiowa.edu/software/signin.htm
- CentOS: http://www.centos.org/
- Fedora: http://fedoraproject.org
- SuSE: $$ https://www.suse.com
- openSuSE: http://www.opensuse.org
- Ubuntu: http://www.ubuntu.com
- Debian: http://www.debian.org
Minimalism

- Install the minimum you need, add software/packages as necessary
- Use package manager to remove any extras that you don’t need
- Fewer packages/applications means less to patch later
Minimalism: Disabling unneeded services

Disable unnecessary services - Don’t install them in the first place

How to determine what’s listening:

```
netstat -vannp | grep LISTEN
```

<table>
<thead>
<tr>
<th>Proto</th>
<th>Recv-Q</th>
<th>Send-Q</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>State</th>
<th>PID/Program name</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:5037</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>5464/adb</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>0.0.0.0:111</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>2017/rpcbind</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>0.0.0.0:22</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>10260/sshd</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>0.0.0.0:49218</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>9997/rpc.statd</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>0.0.0.0:24800</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>11677/synergys</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>0.0.0.0:40</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>10260/sshd</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:631</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>10073/cupsd</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>10349/master</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>::1:111</td>
<td>:::*</td>
<td>LISTEN</td>
<td>2017/rpcbind</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>::50162</td>
<td>:::*</td>
<td>LISTEN</td>
<td>9997/rpc.statd</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>::22</td>
<td>:::*</td>
<td>LISTEN</td>
<td>10260/sshd</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>::44</td>
<td>:::*</td>
<td>LISTEN</td>
<td>10260/sshd</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>::1:631</td>
<td>:::*</td>
<td>LISTEN</td>
<td>10073/cupsd</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>::1:25</td>
<td>:::*</td>
<td>LISTEN</td>
<td>10349/master</td>
</tr>
</tbody>
</table>

What/where is that program:

```
man -k <program>  ls -l /proc/<pid>/exe
```
After install, and initial configuration: determine what “normal” is

- Which processes are running as which users (ps)?
- Which ports are they listening on (netstat/fuser/lsof)?
- Scheduled tasks (cron/at)
- Disk utilization (df)
- Log entries (logwatch, checking logs manually)
- Memory usage (top, vmstat)
- CPU usage (top, mpstat)
- Accounts on the system (/etc/passwd)
- Account access (who is “normally” logged in: w, last)
One of the foremost ways to keep your system secure is to patch regularly.

- Schedule a time to do patching (monthly or better)
- Allow for emergency patching for critical vulnerabilities
- Evaluate patches: local exploit vs. remote exploit
- Software inventory on a system (what to patch)
- Subscribe to appropriate information sources to find out when patches are available
Patching: Post Patching

Linux won’t (usually) prompt you to reboot

- If you updated glibc or the kernel, you’ll need to reboot
- Otherwise, restart services as needed.

How do you find out which services need to be restarted:

```
needs-restarting # part of yum-utils package
lsof | grep inode | grep lib # RHEL5
lsof | grep DEL | grep lib    # RHEL6
```

will output a list of all processes which have deleted files with the name `lib` in them.

```
COMMAND    PID  USER FD  TYPE DEVICE SIZE/OFF NODE NAME
sudo       664  root  DEL REG 253,0 1742367 /usr/lib64/libnssutil3.so
sudo       664  root  DEL REG 253,0 1742397 /usr/lib64/libnss3.so
firefox 3528 hbrown  DEL REG 253,0 1742367 /usr/lib64/libnssutil3.so
firefox 3528 hbrown  DEL REG 253,0 1742397 /usr/lib64/libnss3.so
cupsd 10073 root  DEL REG 253,0 1742367 /usr/lib64/libnssutil3.so
cupsd 10073 root  DEL REG 253,0 1742397 /usr/lib64/libnss3.so
sshd 10260 root  DEL REG 253,0 1742367 /usr/lib64/libnssutil3.so
sshd 10260 root  DEL REG 253,0 1742397 /usr/lib64/libnss3.so
qmqr 10357 postfix DEL REG 253,0 1742367 /usr/lib64/libnssutil3.so
qmqr 10357 postfix DEL REG 253,0 1742397 /usr/lib64/libnss3.so
```
Patching: Campus Resources

- Satellite server: Redhat products
  https://rhnsat.uiowa.edu
  Details:
  http://its.uiowa.edu/campus-software-program/red-hat-linux-campus

- Spacewalk server: Fedora and CentOS
  http://spacewalk.its.uiowa.edu
Least Privilege

Only give an application/user the minimally needed permissions to get the job done

- Most processes don’t need to run as root, don’t let them (when possible)
- Binding to a port between 1 and 1024 is root only
- Don’t give out the root password
- Don’t ever do `chmod 777` (unless you really mean it)
- Evaluate Setuid programs and determine if any user will need to run them
- Limit sudo privileges to just the needed commands
The setuid permission coupled with root ownership allows any user to run that command with root privileges.

Finding setuid programs:
/usr/bin/find / -user root -perm -4000 -print

Remove:
chmod u-s /path/to/binary
Potentially valid uses for setuid programs

- Reading files that would otherwise be protected
  /bin/sudo needs to read /etc/sudoers to validate commands
- Writing files to locations that are protected
  /usr/bin/crontab stores a user’s crontab in /var/spool/cron
- Creating a raw socket (ping/traceroute)
Sudo can be used to give selective permission for performing a limited number of tasks as root

- Using `hbrown ALL=(ALL) ALL` in sudoers is full root access, be selective.
- Allowing `vi /my/config` via sudo is full root access. Allow `sudoedit /my/config` instead.
- Using sudo for select functions is a good thing.
Permissions

Tips for troubleshooting permissions:

- Check to see if SELinux is enabled: `getenforce`
- `cat /proc/<pid>/status` and look for Uid/Gid to see which user a process is running as
- Check from `/` down to wherever it is the app is trying to write instead of the other way round
- Use `strace` to do a system trace
  `strace -ffvvvto output.strace -s 2048 -u <username> <cmd>`
Avoid shared/service accounts with passwords that everyone logs in with
  - Use sudo access to a service account when needed
Use directory based user accounts (for built-in account expiration/password rules)
Collect logs of user account activity on a different system (wtmp, /var/log/secure)
Remove .ssh/authorized_keys for users that are gone/disabled.
Authorization against AD is an on-going project. For now, you’ll need to populate the /etc/passwd file with an entry for each user.
Authentication is possible now
http://its.uiowa.edu/support/article/100409
ssh is a wonderful tool, hackers think so too. Ideas for securing it:

- **Edit your `/etc/ssh/sshd_config**
  - Disable root login (`PermitRootLogin no`)
  - Disable Password authentication and only use keys (`PasswordAuthentication no`)
  - Leave X11Forwarding turned off (`X11Forwarding no`)

- **Edit your `/etc/ssh/ssh_config**
  - Enable hashed known_hosts files (`HashKnownHosts yes`)
  - Disable X11Trusted (`ForwardX11Trusted no`)

- Run it on a different port than 22
- Use iptables to rate limit access and only allow trusted IPs/nets.
An easy way to keep the bad actors out is to refuse to respond
- Use host based firewalls - iptables/ip6tables
- Use TCP Wrappers

Scan your system regularly to make sure those filters are working.
Use a local host based firewall

- Make sure you have ip6tables configured if you are using IPv6
- Remember that iptables uses a first match paradigm
- Make sure you have a line that rejects/drops traffic at the end
- Test your rules with nmap from both on and off campus
TCP Wrappers is a classic lightweight application firewall. Rules are in /etc/hosts.allow and /etc/hosts.deny

- Make use of it to fine tune what you protect
- It’s a second layer of defense behind your host based firewall (ip*tables)
- Use ldd <path/to/binary> | grep libwrap to see if your application supports it
- The daemon name for the hosts.allow file is usually the name of the binary. Check documentation to be sure.
Packet Filtering: Scanning

- Scan your systems regularly to make sure the view from the outside is “normal”
- Request a scan from the Security Office
  http://itsecurity.uiowa.edu/scan/networkscan-form.shtml
- Use nmap judiciously against your own systems
There’s lots of useful information to be had in /var/log. Attackers know that and clean it out to hide what they’ve done

- Use a central logging host to gather that information
- Use some sort of log analysis suite (or logwatch) to help you wade through it
- Use logging to help you determine “normal”
AV? We don’t need no AV on Linux!

- Sometimes your Linux server is serving up files to MS Windows hosts.
- Running AV on those Linux systems is a must
- ClamAV http://www.clamav.net
Look for best practice guides for any application you deploy

An Example: Apache Host Access to protect phpMyAdmin

```plaintext
<Directory "/usr/share/phpMyAdmin”>
  Order deny, allow
  Deny from all
  Allow from uiowa.edu
  Allow from 128.255.
</Directory>
```
Secure Your Applications: Apache Allow/Deny Ordering

http://httpd.apache.org/docs/2.2/mod/mod_authz_host.html#order

<table>
<thead>
<tr>
<th>Match</th>
<th>Allow,Deny result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match Allow only</td>
<td>Request allowed</td>
</tr>
<tr>
<td>Match Deny only</td>
<td>Request denied</td>
</tr>
<tr>
<td>No match</td>
<td>Default to second directive: Denied</td>
</tr>
<tr>
<td>Match both Allow &amp; Deny</td>
<td>Final match controls: Denied</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Match</th>
<th>Deny,Allow result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match Allow only</td>
<td>Request allowed</td>
</tr>
<tr>
<td>Match Deny only</td>
<td>Request denied</td>
</tr>
<tr>
<td>No match</td>
<td>Default to second directive: Allowed</td>
</tr>
<tr>
<td>Match both Allow &amp; Deny</td>
<td>Final match controls: Allowed</td>
</tr>
</tbody>
</table>
Further reading

UNIX and Linux System Administration Handbook
http://www.admin.com

The Linux Command Line
http://linuxcommand.org/tlcl.php