

Linux Hardening Techniques

Vasudev Baldwa

UBNetDef, Spring 2021



Agenda

1. What is Systems Hardening?
2. Basic Principles
3. Updates & Encryption
4. Monitoring
5. Services
6. Firewalls
7. Logging

What is System Hardening?

- ⬡ A collection of tools, techniques, and best practices to reduce vulnerability in technology applications, systems, infrastructure, firmware, and other areas
- ⬡ 3 major areas: OS vs Software vs Network
 - ⬡ When have we done hardening in this class before?
 - ⬡ This lecture is focusing mostly on OS and software level

Why Harden?

- Hexagon icon: Firewalls can only get us so far, what happens when an attack is inside the network?
 - Triangle icon: If you have nothing protecting your systems you are in trouble
- Hexagon icon: We want some kind of secondary protection



News | **Opinion** | **Sport** | **Culture** | **Lifestyle**

Australia | Coronavirus | World | AU politics | Environment | Football | Indigenous Australia

New South Wales

🕒 This article is more than **7 months old**

Service NSW hack could have been prevented with simple security update

Cybersecurity experts say updating to the newest security patch could have protected against the majority of incidents within the NSW government last year

A Few Cybersecurity Principles

- ⬡ Zero Trust Security
 - ⬡ Instead of assuming everything behind the firewall is safe, Zero Trust verifies each request as though it originates from an unsecure network
- ⬡ Principle of Least Privilege
 - ⬡ Only privileges needed to complete a task should be allowed
 - ⬡ Users should not have domain administrator/root privileges
- ⬡ Principle of Least Common Mechanism
 - ⬡ Mechanisms used to access resources should not be shared in order to avoid the transmission of data.
 - ⬡ Shared resources should not be used to access resources

The Threat Model

- ⬡ A process by which potential threats can be identified and prioritized.
 - ⬡ If you have a web server that feeds input to a mysql database, then protecting against mysql injections would be prioritized in your model.



2 considerations

- ⬡ *nix like is a very broad grouping of systems
 - ⬡ Some of this might not translate 1:1, but the theory remains the same
- ⬡ IR and Hardening are 2 sides of the same coin
 - ⬡ Usually the Exploitation/Installation signals the transition into IR

Systems Updates

- Running an outdated system is a bad idea
- Constantly upgrading to the bleeding edge might also cause issues
 - What issues can we think off?
- This is why we use a LTS based update system
 - Recall from the linux lecture we configured the machines to only do security updates

Hard Disk Encryption

- What is Encryption?
 - Turns readable text into cipher text
 - GPG is a common tool
- Full Disk Encryption might not work for all cases

System Monitoring

- Recall from the Services Lecture that linux had process and services
 - A process is an instance of a particular executable
 - A service is a process which runs in the background (daemons)
 - firewalld, named, systemd

System Monitoring

- systemd
 - system and service manager
- systemctl
 - control the state of the systemd system and service manager
 - systemctl enable, disable, start, stop <service>
- journalctl

System Monitoring

⬡ ps aux

- ⬡ Very common command
- ⬡ ps → process status
- ⬡ a → running processes from all users
- ⬡ u → user or owner column in output
- ⬡ x → prints the processes those have not been executed from the terminal

⬡ htop

- ⬡ A process viewer

System Monitoring

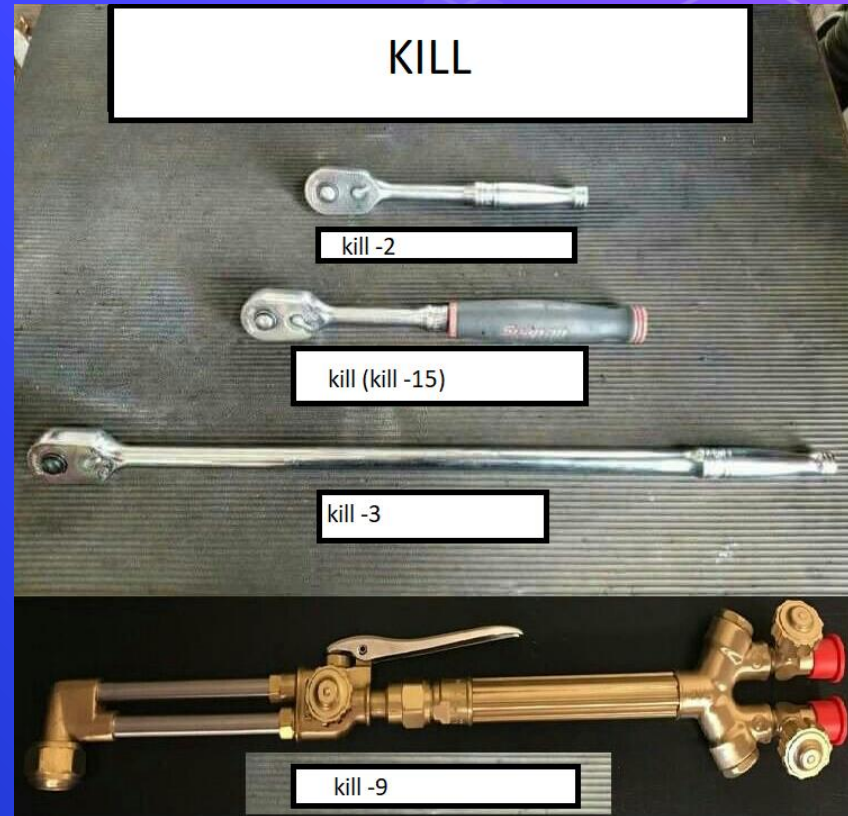
📡 lsof - “list open files”

- 📡 Because everything in linux is a byte stream, we can treat each stream as a file
- 📡 Lets us monitor what each process is doing on our system

```
apache2 14746 www-data mem REG 8,5 229248 131156 /usr/lib/x86_64-linux-gnu/libapr-1.so.0.6.
apache2 14746 www-data mem REG 8,5 184152 133833 /usr/lib/x86_64-linux-gnu/libaprutil-1.so.
apache2 14746 www-data mem REG 8,5 465008 139929 /usr/lib/x86_64-linux-gnu/libpcre.so.3.13.
apache2 14746 www-data mem REG 8,5 191472 138874 /usr/lib/x86_64-linux-gnu/ld-2.31.so
apache2 14746 www-data 0r CHR 1,3 0t0 6 /dev/null
apache2 14746 www-data 1w CHR 1,3 0t0 6 /dev/null
apache2 14746 www-data 2w REG 8,5 239 675232 /var/log/apache2/error.log
apache2 14746 www-data 3u sock 0,8 0t0 25895 protocol: TCP
apache2 14746 www-data 4u IPv6 25896 0t0 TCP *:http (LISTEN)
apache2 14746 www-data 5r FIFO 0,12 0t0 316658 pipe
apache2 14746 www-data 6w FIFO 0,12 0t0 316658 pipe
apache2 14746 www-data 7w REG 8,5 0 656971 /var/log/apache2/other_vhosts_access.log
apache2 14746 www-data 8w REG 8,5 0 661337 /var/log/apache2/access.log
apache2 14746 www-data 9u REG 8,5 0 1082820 /tmp/ZendSem_xc608b (deleted)
```

Finding & Killing processes

- ⬡ pgrep vs grep
 - ⬡ ps aux | grep bash
- ⬡ pkill vs kill
 - ⬡ SIGKILL (kill -9)
- ⬡ pkill / pgrep just kill by process name
 - ⬡ kill 3782
 - ⬡ pkill nautilus



In Class Activity

- Fun with killing and finding commands
- Going to be using ss and kill to find and kill a process





Break Time!

Please return in 10 mins

More useful commands

- ⬡ `cat /proc/*/stat | awk '{print $1,$2}'`
 - ⬡ query's the /proc filesystem, which contains everything the kernel knows about processes that are running

- ⬡ `iostat`
 - ⬡ See what network inputs/outputs we've got

- ⬡ `df -h`
 - ⬡ How much disk space is taken up

- ⬡ `w`
 - ⬡ Who is on the system

```
sysadmin@batkaikaldev:~$ cat /proc/*/stat | awk '{print$1, $2}'
100 (irq/30-pciehp)
101 (irq/31-pciehp)
102 (irq/32-pciehp)
103 (irq/33-pciehp)
104 (irq/34-pciehp)
105 (irq/35-pciehp)
106 (irq/36-pciehp)
107 (irq/37-pciehp)
108 (irq/38-pciehp)
109 (irq/39-pciehp)
10 (rcu_sched)
110 (irq/40-pciehp)
1117 (upowerd)
111 (irq/41-pciehp)
112 (irq/42-pciehp)
113 (irq/43-pciehp)
114 (irq/44-pciehp)
```

A quick review of the file system

⬡ /var

- ⬡ data that is changed when the system is running normally

⬡ /etc

- ⬡ Lots of files. Many config files are located here.

⬡ /etc/passwd

- ⬡ All the users and passwords

⬡ /etc/group

- ⬡ Same as passwd, but with groups instead

⬡ /etc/motd

- ⬡ Contains stuff that is run whenever a user logs in

A quick review of the file system

⬡ /proc

- ⬡ “Fake filesystem”
- ⬡ provide information about the system, created by the kernel
- ⬡ `cat /proc/*/stat | awk '{print $1,$2}'`

Securing Services

- Recall from the pentesting lecture: The Cron Service
 - time-based job scheduler
 - Remember the Cybersecurity Principles we talked about
 - `/etc/cron.allow` - If this file exists, it must contain the user's name for that user to be allowed to use cron jobs.
 - `/etc/cron.deny` - If the `cron.allow` file does not exist but the `/etc/cron.deny` file does exist then, to use cron jobs, users must not be listed in the `/etc/cron.deny` file.
- Used by attacker, therefore should be considered when defending a system
 - `echo ALL >>/etc/cron.deny`
 - Disables all users from using cron

Securing Services

- Some services come with secure installation scripts
 - Recall from the Services lecture: `mysql_secure_installation`
- Other services have config files
 - `/etc/ssh/sshd_config`
 - `PermitRootLogin`
 - `AllowList`
- Services should handle unexpected input gracefully
 - SQL injections, Heartbleed, XSS

Securing Services

- ⬡ The User Service
- ⬡ Recall from the Linux HW: Linux Pluggable Authentication Modules (PAM)
 - ⬡ configure methods to authenticate users
 - ⬡ Allows authentication with LDAP
 - ⬡ Why might we find this useful?
- ⬡ If you don't want to do that you can use PAM configure other password policies

Securing Services

- ⬡ Access Control Lists
 - ⬡ a list of permissions associated with a system resource
- ⬡ SELinux
 - ⬡ set of kernel modifications and user-space tools
 - ⬡ enforces mandatory access control policies that confine user programs and system services, as well as access to files and network resources
 - ⬡ Recall the Principle of Least Privilege

In Class Activity

- Secure a SSH Server



Local firewalls

- ⬡ When have we used these before?
 - ⬡ The Services Lecture
- ⬡ Recall the Principle of Least Common Mechanism
 - ⬡ We don't want to share anything except 1 service
 - ⬡ Default deny all is great here

Local Firewalls

- ⬡ How do we see what our machine is talking to on the network?
 - ⬡ ss/netstat(deprecated)
 - ⬡ What are the security considerations of using deprecated software on systems?
- ⬡ PortSentry
 - ⬡ Daemon that will watch unused ports for activity
 - ⬡ What phases of the kill chain does this help us defend against?
- ⬡ Fail2Ban

Logging

- ⬡ Which logs have we looked at before?
 - ⬡ Apache logs in Services Lecture
- ⬡ Located in `/var/log`
 - ⬡ `auth.log (deb)` or `secure (rehl)` for logins
 - ⬡ `faillog` for failed logins
 - ⬡ cron keeps a log of cronjobs
 - ⬡ Other service will have their own logs

Logging

- ⬡ We can also log changes to files
 - ⬡ When would this be useful?
- ⬡ This is where file integrity tools come in handy
 - ⬡ Tripwire
 - ⬡ OSSEC







Quick and Dirty 5 min security plan

1. Change your password, change root password
2. Edit firewall rules,
 1. Allow services you need, block everything else
3. Services
 1. `ps aux | less`
 2. `ss -tulpan`
 3. `kill -9` anything that looks funny
 4. `grep CRON /var/log/syslog`
4. Disable users
 1. `cat /etc/passwd | grep /bin/bash`

Homework

- 4 Guided Activities
- On the LSHHW VM find and document 3 things that you have hardened
 - Document the issue
 - Document how you fixed it
 - Do a simple risk analysis on the issue
- I will hijack Phil's office hours next week (And possible some others, stay tuned in mattermost)

Further reading

-  [Saltzer and Schroeder's Design Principles](#)
-  [CIS Security Benchmarks](#)
-  [SELinux Docs](#)
-  [LSOF cheat sheet](#)