**Linux Security — Complete Step-by-Step Notes**

**Table of contents**

1. Goals & approach
2. Baseline & inventory
3. Patch management
4. User & authentication hardening (PAM, passwords, sudo)
5. SSH hardening
6. Service hardening & minimization
7. Firewall & network hardening (iptables/nftables & firewalld)
8. Kernel hardening (sysctl)
9. Filesystem hardening, permissions, ACLs, SUID/SGID
10. SELinux / AppArmor
11. Auditing & logging (auditd, rsyslog, journald)
12. Intrusion detection & malware scanning (AIDE, rkhunter, Lynis, OSSEC)
13. Fail2ban & brute-force protection
14. Disk & data encryption (LUKS, GPG, TLS certs)
15. Container security basics
16. Backups & recovery
17. Incident response checklist
18. Verification commands & checklist
19. Common interview Q&A

**1) Goals & Approach**

* **Goal:** Reduce attack surface, ensure accountability (logging/audit), prevent unauthorized access, detect intrusions quickly, and enable recovery.
* **Approach:** Inventory → Patch → Minimize → Harden → Monitor → Respond.

**2) Baseline & Inventory (first thing to do)**

1. List OS & kernel:

cat /etc/os-release

uname -r

1. List installed packages (deb/rpm):

dpkg -l | less # Debian/Ubuntu

rpm -qa | sort > /tmp/packages.txt # RHEL/CentOS

1. Active services & listening ports:

ss -tulnp

systemctl list-unit-files --state=enabled

1. Users and sudoers:

cut -d: -f1,3,4 /etc/passwd

getent group

sudo cat /etc/sudoers /etc/sudoers.d/\*

1. Open network connections:

netstat -tulpen # or ss -tunap

Record this baseline in a secure location. You will compare after changes.

**3) Patch Management (updates & scheduled patching)**

**Step-by-step:**

1. Update package index and check pending updates:

# RHEL/CentOS

sudo yum check-update

# Fedora/RHEL8+

sudo dnf check-update

# Debian/Ubuntu

sudo apt update && apt list --upgradable

1. Test updates on staging first. For production, use rolling patch or maintenance window.
2. Apply updates:

sudo yum update -y # or dnf

sudo apt upgrade -y

1. Kernel updates require reboot. Use needs-restarting -r (yum-utils) or checkrestart.
2. Automate critical security updates (careful):

* Debian/Ubuntu: unattended-upgrades package.
* RHEL: use Satellite or automated patch orchestration.

**Verify:** rpm -q kernel or apt list --installed | grep linux-image then reboot and confirm uname -r.

**4) User & Authentication Hardening (PAM, passwords, sudo)**

**Password policy (PAM and pwquality)**

1. Install pwquality (if needed):

# Debian/Ubuntu

sudo apt install libpam-pwquality

# RHEL

sudo yum install libpwquality

1. Edit /etc/security/pwquality.conf (or /etc/pam.d/common-password snippet) — example settings:

minlen = 12

dcredit = -1 # require digit

ucredit = -1 # require uppercase

lcredit = -1 # require lowercase

ocredit = -1 # require special char

1. Edit PAM password line (Debian example /etc/pam.d/common-password):

password requisite pam\_pwquality.so retry=3

**Password aging and lockouts**

# Set password aging for user

sudo chage -M 90 -m 7 -W 14 username

# Force expire password

sudo chage -d 0 username

# Lock user

sudo passwd -l username

**Sudo hardening**

1. Use visudo to edit /etc/sudoers or better /etc/sudoers.d/ files.
2. Principle of least privilege: give specific commands rather than ALL.
3. Example: allow user to restart apache only:

deploy ALL=(root) NOPASSWD: /bin/systemctl restart httpd

1. Disable !authenticate if you want to require password for sudo. Restrict sudo to groups, not individual users.

**Verify sudo logs:**

ausearch -m USER\_CMD # auditd if configured

grep sudo /var/log/auth.log

**5) SSH Hardening (critical)**

**Steps (edit /etc/ssh/sshd\_config):**

1. Backup config:

sudo cp /etc/ssh/sshd\_config /etc/ssh/sshd\_config.bak

1. Recommended settings (example):

Protocol 2

PermitRootLogin no

PasswordAuthentication no

ChallengeResponseAuthentication no

UsePAM yes

AllowUsers sgharge adminuser # restrict to allowed users

PermitEmptyPasswords no

X11Forwarding no

ClientAliveInterval 300

ClientAliveCountMax 2

LoginGraceTime 30

MaxAuthTries 3

1. Use SSH keys (ed25519 / rsa 4096) and disable password auth.

# Generate key on client

ssh-keygen -t ed25519 -C "sagar@work"

ssh-copy-id -i ~/.ssh/id\_ed25519.pub sgharge@server

1. Optionally, enforce 2FA (Google Authenticator PAM) or hardware tokens (YubiKey).
2. Restart SSH:

sudo systemctl reload sshd # or restart

1. Test in new session before closing existing connection (to avoid lockout):

ssh -v user@host

**Verify:** sshd -T | grep -E 'permitrootlogin|passwordauthentication' and systemctl status sshd.

**6) Service Hardening & Minimization**

1. Disable unused services permanently:

sudo systemctl disable --now cups avahi-daemon bluetooth

sudo systemctl mask some-service

1. Remove unnecessary packages:

sudo apt remove --purge package

sudo yum remove package

1. Check enabled services:

systemctl list-unit-files --state=enabled

1. Limit which users can run binaries — use file permissions and sudo rules.

**7) Firewall & Network Hardening**

**Simple iptables (recommended: use nftables or firewalld in newer distros)**

# Flush existing rules (be careful remotely)

sudo iptables -F

# Allow loopback

sudo iptables -A INPUT -i lo -j ACCEPT

# Allow established connections

sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

# Allow SSH from specific network

sudo iptables -A INPUT -p tcp -s 203.0.113.0/24 --dport 22 -m conntrack --ctstate NEW -j ACCEPT

# Allow HTTP/HTTPS

sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT

sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT

# Drop everything else

sudo iptables -P INPUT DROP

sudo iptables -P FORWARD DROP

sudo iptables -P OUTPUT ACCEPT

# Save rules

sudo iptables-save > /etc/iptables.rules

**firewalld (RHEL/CentOS)**

sudo firewall-cmd --permanent --add-service=ssh

sudo firewall-cmd --permanent --add-service=http

sudo firewall-cmd --reload

**nftables (newer Linux)**

sudo nft add table inet filter

sudo nft add chain inet filter input { type filter hook input priority 0 \; policy drop \; }

sudo nft add rule inet filter input ct state established,related accept

**Network protections (sysctl)**: see Kernel hardening.

**8) Kernel Hardening (/etc/sysctl.conf)**

Add or verify these settings (apply with sysctl -p):

# Disable IP forwarding

net.ipv4.ip\_forward = 0

# Ignore ICMP redirects

net.ipv4.conf.all.accept\_redirects = 0

net.ipv4.conf.default.accept\_redirects = 0

# Ignore source routed packets

net.ipv4.conf.all.accept\_source\_route = 0

net.ipv4.conf.default.accept\_source\_route = 0

# Enable TCP syncookies

net.ipv4.tcp\_syncookies = 1

# Reverse path filtering

net.ipv4.conf.all.rp\_filter = 1

net.ipv4.conf.default.rp\_filter = 1

# Protect against kernel pointer leaks

kernel.kptr\_restrict = 2

# Hardening for symlink/hardlink attacks

fs.protected\_hardlinks = 1

fs.protected\_symlinks = 1

# Reduce core dumps for setuid

fs.suid\_dumpable = 0

Apply:

sudo sysctl -p

**9) Filesystem Hardening, Permissions, ACLs, SUID/SGID**

**File permissions & umask**

* Set a restrictive default umask (e.g., 027 or 077) in /etc/profile or service unit.

umask 027

**Sticky bit on shared dirs**

chmod 1777 /tmp

**Find SUID/SGID files**

find / -perm /6000 -type f -exec ls -ld {} \; 2>/dev/null

Review and remove setuid bits if not necessary:

chmod u-s /path/to/file

**ACLs**

getfacl /data

setfacl -m u:deploy:rwx /data

**Secure /tmp and /var/tmp (mount options)**

Add to /etc/fstab:

tmpfs /tmp tmpfs rw,nosuid,nodev,noexec,mode=1777 0 0

/var/tmp none tmpfs rw,nosuid,nodev,noexec,mode=1777 0 0

(Adding noexec can break some installers—test first.)

**10) SELinux / AppArmor**

**SELinux (RHEL/CentOS)**

* Check status:

getenforce

sestatus

* Keep SELinux Enforcing in production. To change temporary:

sudo setenforce 1 # 1 enforcing, 0 permissive

* Use semanage & restorecon to fix contexts:

sudo semanage fcontext -a -t httpd\_sys\_content\_t '/srv/www(/.\*)?'

sudo restorecon -Rv /srv/www

**AppArmor (Ubuntu)**

* Check status:

aa-status

sudo apparmor\_status

* Enforce profiles: aa-enforce /etc/apparmor.d/\*

**11) Auditing & Logging**

**auditd**

1. Install & enable:

sudo apt install auditd audispd-plugins

sudo systemctl enable --now auditd

1. Add rules (example in /etc/audit/rules.d/audit.rules):

-w /etc/sudoers -p wa -k sudo\_changes

-w /var/log/ -p wa -k log\_modifications

-a always,exit -F arch=b64 -S execve -k exec

1. Search events:

ausearch -k sudo\_changes

aureport --summary

**Central logging (rsyslog/journal forwarding)**

* Configure rsyslog to forward logs to a central log server for analysis.
* Example snippet /etc/rsyslog.d/50-default.conf:

\*.\* @@logs.example.com:514

**12) Intrusion Detection & Malware Scanning**

**AIDE (File integrity)**

sudo apt install aide

sudo aideinit

sudo cp /var/lib/aide/aide.db.new /var/lib/aide/aide.db

# Check

sudo aide --check

Schedule a daily check via cron/systemd timer.

**rkhunter / chkrootkit**

sudo apt install rkhunter chkrootkit

sudo rkhunter --update

sudo rkhunter --checkall

sudo chkrootkit

**Lynis (security audit)**

wget https://packages.cisofy.com/lynis/lynis-3.0.tar.gz

sudo ./lynis audit system

**13) Fail2ban & Brute-force Protection**

1. Install:

sudo apt install fail2ban

1. Create /etc/fail2ban/jail.local sample:

[sshd]

enabled = true

port = ssh

filter = sshd

logpath = /var/log/auth.log

maxretry = 5

bantime = 3600

1. Restart and verify:

sudo systemctl restart fail2ban

sudo fail2ban-client status sshd

**14) Disk & Data Encryption**

**LUKS Full-disk encryption (example)**

# WARNING: DESTROYS DATA

sudo cryptsetup luksFormat /dev/sdb1

sudo cryptsetup luksOpen /dev/sdb1 cryptdata

sudo mkfs.ext4 /dev/mapper/cryptdata

sudo mount /dev/mapper/cryptdata /data

Add to /etc/crypttab for auto-open at boot and add mapping to /etc/fstab.

**GPG for files**

gpg --gen-key

gpg -e -r recipient file.txt

**TLS certificates (Let's Encrypt)**

sudo apt install certbot

sudo certbot certonly --standalone -d example.com

**15) Container Security Basics**

* Do not run containers as root. Use user namespaces.
* Limit capabilities: --cap-drop=ALL --cap-add=NET\_BIND\_SERVICE.
* Use read-only filesystem --read-only and mount only required volumes.
* Scan images (Clair, trivy) and use minimal base images.

**16) Backups & Recovery**

* Regular, tested backups. Use rsync -aHAX, snapshots, or block-level backups.
* Keep at least 3-2-1 rule: 3 copies, 2 media types, 1 offsite.
* Encrypt backups and test restores regularly.

**17) Incident Response Checklist (first 24 hours)**

1. Isolate host (remove from network or firewall rules).
2. Preserve volatile data (running processes, network connections):

ps aux > /tmp/ps.aux

ss -tunap > /tmp/ss.txt

netstat -rn > /tmp/netstat.txt

1. Collect logs: /var/log/, journalctl -k, ausearch.
2. Create disk image (for forensic): dd if=/dev/sda of=/mnt/forensics/sda.img bs=4M conv=sync,noerror.
3. Change passwords (after containment) and check sudoers.
4. Reinstall from trusted media if integrity is in doubt.

**18) Verification Commands & Quick Checklist**

* Is SSH key-only enabled?

sshd -T | grep -i passwordauthentication

* Is root login disabled?

sshd -T | grep -i permitrootlogin

* Check for SUID files:

find / -perm /6000 -type f -ls

* Auditd running?

systemctl status auditd

ausearch -m USER\_LOGIN --start today

* SELinux enabled?

getenforce

sestatus

* Firewall active?

sudo iptables -L -n -v

sudo firewall-cmd --state

**19) Common Interview Q&A (short answers)**

**Q1: How do you harden SSH?**

* Disable root login, disable password auth, use keys, limit users, reduce MaxAuthTries, enable 2FA.

**Q2: What is SELinux and why use it?**

* Mandatory access control system that confines processes and limits damage from compromised services.

**Q3: How to detect rootkits?**

* Use rkhunter, chkrootkit, file integrity checks (AIDE), monitor unusual outbound connections.

**Q4: What is auditd used for?**

* Records security-relevant events (execve, file changes, sudo use). Useful for forensic and compliance.

**Q5: How to securely manage secrets?**

* Use a secrets manager (Vault), avoid storing credentials in plain files, use restricted ACLs and encryption.

**Q6: How to prevent privilege escalation?**

* Minimize SUID binaries, keep software patched, use least privilege sudo, enable SELinux/AppArmor.

**Q7: What is a good umask?**

* 027 or 077 depending on team needs (027 allows group read/execute).

**Q8: How to secure web server files?**

* Run as dedicated user, correct SELinux context, minimal modules enabled, TLS, WAF if available.

**Q9: How to secure Docker containers?**

* Use non-root containers, limit capabilities, scan images, use user namespaces.

**Q10: What to do immediately after a suspected compromise?**

* Isolate host, collect evidence, preserve logs, take disk image, notify stakeholders, investigate.

**Final notes**

* Hardening is iterative — apply changes to staging first.
* Logging and monitoring are as important as prevention.
* Automate checks (Lynis, CIS benchmarks, OpenSCAP) and schedule periodic audits.