TOC

1. Setup & Backup
2. User Account Management
3. Service Hardening (Samba, FTP, SSH, Telnet, Mail, Print, MySQL, HTTP, DNS)
4. Media File Removal
5. Hacking Tool Removal
6. Password Policy & PAM Security
7. Firewall & Network Security
8. Crontab & Startup Security
9. System Updates & Repository Fixes
10. System Logging and Verification

1 Setup & Backup

1. Become Root

What it does: Opens a root shell so you don’t need sudo for every command.

Command:

sudo -i

2. Create a Script Log File

What it does: Creates and clears the log file that will record timestamps and actions.

Command:

touch ~/Desktop/Script.log && : > ~/Desktop/Script.log

Secure it (recommended):

chmod 600 ~/Desktop/Script.log

Undo (restore less secure world-writable permissions):

chmod 777 ~/Desktop/Script.log

3. Create a Backups Folder

What it does: Makes a folder to store backup configuration files.

Command:

mkdir -p ~/Desktop/backups

Secure it (recommended):

chmod 700 ~/Desktop/backups

Undo (restore world-writable permissions):

chmod 777 ~/Desktop/backups

4. Backup /etc/passwd and /etc/group

What it does: Saves the system’s user and group configuration files for recovery if something breaks.

Commands:

cp /etc/passwd ~/Desktop/backups/passwd

cp /etc/group ~/Desktop/backups/group

chmod 600 ~/Desktop/backups/passwd ~/Desktop/backups/group

Undo (restore world-writable permissions):

chmod 777 ~/Desktop/backups/passwd ~/Desktop/backups/group

5. (Optional, Recommended) Backup the Entire /etc Directory

What it does: Creates a compressed, timestamped archive of /etc so you can fully restore configurations later.

Command:

BACKUP=/root/etc-backup-$(date +%Y%m%d-%H%M%S).tar.gz

tar -czf "$BACKUP" /etc

chmod 600 "$BACKUP"

echo "Saved /etc backup to $BACKUP"

Undo (to restore the backup later):

tar -xzf /root/etc-backup-YYYYMMDD-HHMMSS.tar.gz -C /

(Replace the filename with your actual backup.)

6. Save a List of Installed Packages

What it does: Records all installed packages and manually installed programs. Useful for restoration and scoring reports.

Commands:

dpkg --get-selections > ~/Desktop/backups/dpkg-selections.txt

apt-mark showmanual > ~/Desktop/backups/manuallyinstalled-packages.txt

chmod 600 ~/Desktop/backups/dpkg-selections.txt ~/Desktop/backups/manuallyinstalled-packages.txt

Undo (to restore package selections):

dpkg --set-selections < ~/Desktop/backups/dpkg-selections.txt

apt-get -u dselect-upgrade

7. Confirm You’re Root

What it does: Verifies that you are running as root before making system changes.

Command:

if [ "$EUID" -ne 0 ]; then echo "You must run this as root (use sudo -i)."; else echo "Running as root."; fi

8. Create a Logs Folder

What it does: Sets up a directory for saving logs, system output, and screenshots.

Command:

mkdir -p ~/Desktop/logs

chmod 700 ~/Desktop/logs

Undo (restore world-writable permissions):

chmod 777 ~/Desktop/logs

9. (Optional) Protect Backups from Tampering

What it does: Makes backup files immutable so they can’t be deleted or modified without explicit permission.

Command:

chattr +i ~/Desktop/backups/group ~/Desktop/backups/passwd ~/Desktop/backups/dpkg-selections.txt 2>/dev/null || true

Undo (allow editing again):

chattr -i ~/Desktop/backups/group ~/Desktop/backups/passwd ~/Desktop/backups/dpkg-selections.txt 2>/dev/null || true

10. Record Completion in Script Log

What it does: Logs a timestamp marking completion of Step 1.

Command:

echo "$(date -u +"%Y-%m-%d %H:%M:%SZ") -- Step 1 (Setup & Backup) completed" >> ~/Desktop/Script.log

2 User Account Management

1. list all user accounts

command:

cut -d: -f1 /etc/passwd

2. list all groups

command:

cut -d: -f1 /etc/group

3. check for users with sudo or admin privileges

commands:

getent group sudo

getent group adm

4. remove unauthorized users

command:

deluser <username>

undo:

adduser <username>

check for files owned by a user before deleting:

find / -user <username> 2>/dev/null

5. lock accounts that shouldn’t log in

command:

passwd -l <username>

undo:

passwd -u <username>

6. force all users to change passwords

command:

awk -F: '$3 >= 1000 {print $1}' /etc/passwd | xargs -n1 chage -d 0

7. disable guest account

commands:

sudo sh -c 'echo "[SeatDefaults]" > /etc/lightdm/lightdm.conf'

sudo sh -c 'echo "allow-guest=false" >> /etc/lightdm/lightdm.conf'

undo:

sudo sed -i '/allow-guest/d' /etc/lightdm/lightdm.conf

8. check for users with empty or weak passwords

command:

awk -F: '($2=="!" || $2=="\*"){print $1}' /etc/shadow

if any are listed, set new passwords:

passwd <username>

9. verify only root has uid 0

command:

awk -F: '($3 == 0) {print $1}' /etc/passwd

if others are listed, change uid:

usermod -u 1001 <username>

10. check for users without password expiration

command:

awk -F: '($2 != "\*" && $2 != "!" && $2 != "" && $3 >= 1000) {print $1}' /etc/passwd | xargs -n1 chage -l

set password expiration manually if needed:

chage -M 90 -m 7 -W 14 <username>

11. lock system accounts that shouldn’t log in

command:

for u in $(awk -F: '($3 < 1000) {print $1}' /etc/passwd); do usermod -s /usr/sbin/nologin "$u"; done

undo:

usermod -s /bin/bash <username>

12. ensure root has a strong password

command:

passwd root

13. check and fix home directory permissions

command:

ls -ld /home/\*

secure homes (recommended):

chmod 700 /home/\*

undo:

chmod 755 /home/\*

14. list hidden or service users

command:

awk -F: '$7!="/bin/bash"{print $1, $7}' /etc/passwd

15. log completion of this step

command:

echo "$(date -u +"%Y-%m-%d %H:%M:%SZ") -- step 2 (user account management) completed" >> ~/Desktop/Script.log

3 Service Hardening

1. check active network services

command:

ss -tulpn

or

netstat -tulpn

2. stop and disable telnet

commands:

systemctl stop inetd.service 2>/dev/null

systemctl disable inetd.service 2>/dev/null

systemctl stop telnet.socket 2>/dev/null

systemctl disable telnet.socket 2>/dev/null

remove telnet package if installed:

apt remove -y telnet

undo (not recommended):

apt install -y telnet

3. stop and disable ftp services

commands:

systemctl stop vsftpd 2>/dev/null

systemctl disable vsftpd 2>/dev/null

apt remove -y vsftpd proftpd-\* pure-ftpd-\* 2>/dev/null

undo:

apt install -y vsftpd

4. stop and disable samba (windows file sharing)

commands:

systemctl stop smbd nmbd winbind 2>/dev/null

systemctl disable smbd nmbd winbind 2>/dev/null

apt remove -y samba samba-common samba-common-bin 2>/dev/null

undo:

apt install -y samba

5. stop and disable apache/httpd web server

commands:

systemctl stop apache2 2>/dev/null

systemctl disable apache2 2>/dev/null

apt remove -y apache2 2>/dev/null

undo:

apt install -y apache2

systemctl enable apache2

systemctl start apache2

6. stop and disable nginx (if installed)

commands:

systemctl stop nginx 2>/dev/null

systemctl disable nginx 2>/dev/null

apt remove -y nginx 2>/dev/null

undo:

apt install -y nginx

systemctl enable nginx

systemctl start nginx

7. stop and disable mysql or mariadb

commands:

systemctl stop mysql mariadb 2>/dev/null

systemctl disable mysql mariadb 2>/dev/null

apt remove -y mysql-server mariadb-server 2>/dev/null

undo:

apt install -y mysql-server

systemctl enable mysql

systemctl start mysql

8. stop and disable mail services (postfix, exim, sendmail)

commands:

systemctl stop postfix exim4 sendmail 2>/dev/null

systemctl disable postfix exim4 sendmail 2>/dev/null

apt remove -y postfix exim4 sendmail 2>/dev/null

undo:

apt install -y postfix

systemctl enable postfix

systemctl start postfix

9. stop and disable printing services (cups)

commands:

systemctl stop cups 2>/dev/null

systemctl disable cups 2>/dev/null

apt remove -y cups 2>/dev/null

undo:

apt install -y cups

systemctl enable cups

systemctl start cups

10. stop and disable dns servers (bind9, dnsmasq)

commands:

systemctl stop bind9 dnsmasq 2>/dev/null

systemctl disable bind9 dnsmasq 2>/dev/null

apt remove -y bind9 dnsmasq 2>/dev/null

undo:

apt install -y bind9

systemctl enable bind9

systemctl start bind9

11. secure ssh if needed

check if ssh is installed:

dpkg -l | grep ssh

if you need ssh for remote scoring, secure it instead of removing:

nano /etc/ssh/sshd\_config

make sure these lines exist:

PermitRootLogin no

PasswordAuthentication yes

AllowUsers <your\_user>

then restart ssh:

systemctl restart ssh

to disable ssh completely:

systemctl stop ssh

systemctl disable ssh

undo (to re-enable):

systemctl enable ssh

systemctl start ssh

12. remove anonymous or guest ftp and samba shares (if still present)

check for shares:

cat /etc/samba/smb.conf | grep -i guest

if guest ok = yes or public = yes appears, edit smb.conf:

nano /etc/samba/smb.conf

set guest ok = no

set browseable = no

save and restart samba:

systemctl restart smbd

13. check and disable unnecessary startup services

commands:

systemctl list-unit-files --type=service | grep enabled

for each unneeded service:

systemctl disable <service\_name>

systemctl stop <service\_name>

14. check cron jobs for backdoors

commands:

ls -la /etc/cron\*

cat /etc/crontab

remove suspicious entries with nano /etc/crontab

undo (restore from backup if needed):

cp ~/Desktop/backups/crontab /etc/crontab

15. log completion of this step

command:

echo "$(date -u +"%Y-%m-%d %H:%M:%SZ") -- step 3 (service hardening) completed" >> ~/Desktop/Script.log

4 Media File Removal

1. locate all music files

command:

find /home -type f ( -iname "\*.mp3" -o -iname "\*.wav" -o -iname "\*.flac" -o -iname "\*.m4a" -o -iname "\*.aac" -o -iname "\*.ogg" ) 2>/dev/null

2. locate all video files

command:

find /home -type f ( -iname "\*.mp4" -o -iname "\*.mov" -o -iname "\*.avi" -o -iname "\*.mkv" -o -iname "\*.wmv" -o -iname "\*.flv" -o -iname "\*.webm" ) 2>/dev/null

3. locate all image files

command:

find /home -type f ( -iname "\*.jpg" -o -iname "\*.jpeg" -o -iname "\*.png" -o -iname "\*.gif" -o -iname "\*.bmp" -o -iname "\*.tiff" -o -iname "\*.svg" ) 2>/dev/null

4. locate all document files

command:

find /home -type f ( -iname "\*.doc" -o -iname "\*.docx" -o -iname "\*.xls" -o -iname "\*.xlsx" -o -iname "\*.ppt" -o -iname "\*.pptx" -o -iname "\*.pdf" -o -iname "\*.txt" -o -iname "\*.rtf" -o -iname "\*.odt" -o -iname "\*.ods" -o -iname "\*.odp" ) 2>/dev/null

5. delete found media files (only after confirming)

command:

read -p "delete all found media files? (y/n): " ans; if [ "$ans" = "y" ]; then find /home -type f ( -iname "\*.mp3" -o -iname "\*.wav" -o -iname "\*.flac" -o -iname "\*.m4a" -o -iname "\*.aac" -o -iname "\*.ogg" -o -iname "\*.mp4" -o -iname "\*.mov" -o -iname "\*.avi" -o -iname "\*.mkv" -o -iname "\*.wmv" -o -iname "\*.flv" -o -iname "\*.webm" -o -iname "\*.jpg" -o -iname "\*.jpeg" -o -iname "\*.png" -o -iname "\*.gif" -o -iname "\*.bmp" -o -iname "\*.tiff" -o -iname "\*.svg" -o -iname "\*.doc" -o -iname "\*.docx" -o -iname "\*.xls" -o -iname "\*.xlsx" -o -iname "\*.ppt" -o -iname "\*.pptx" -o -iname "\*.pdf" -o -iname "\*.txt" -o -iname "\*.rtf" -o -iname "\*.odt" -o -iname "\*.ods" -o -iname "\*.odp" ) -delete; fi

undo:

restore from backup if taken earlier. for example, if /home was backed up:

tar -xzf ~/Desktop/backups/home-backup.tar.gz -C /

6. check for downloads or desktop folders with suspicious content

commands:

ls -la /home/\*/Downloads

ls -la /home/\*/Desktop

review and delete suspicious files manually.

7. clear trash for all users

commands:

rm -rf /home/\*/.local/share/Trash/\*

rm -rf /root/.local/share/Trash/\*

undo:

none (deleted files can’t be recovered unless backed up)

8. remove sample media files from system directories

commands:

rm -rf /usr/share/example-content

rm -rf /usr/share/sounds/\*

rm -rf /usr/share/backgrounds/\*

undo:

reinstall package providing default content if needed, for example:

apt install --reinstall ubuntu-wallpapers gnome-backgrounds

9. find and delete iso or disk image files

command:

find /home -type f ( -iname "\*.iso" -o -iname "\*.img" ) -delete

undo:

restore from backup

10. log completion of this step

command:

echo "$(date -u +"%Y-%m-%d %H:%M:%SZ") -- step 4 (media file removal) completed" >> ~/Desktop/Script.log

5 Hacking Tool Removal

1. check for common hacking tool directories

commands:

ls -la /opt

ls -la /usr/share

ls -la /home

look for folders named like metasploit, beef, nmap, aircrack, john, hydra, or sqlmap.

remove any suspicious ones:

rm -rf /opt/<foldername>

rm -rf /usr/share/<foldername>

rm -rf /home/\*/<foldername>

2. uninstall metasploit framework

command:

apt remove -y metasploit-framework

undo:

apt install -y metasploit-framework

3. uninstall nmap and zenmap

commands:

apt remove -y nmap zenmap

undo:

apt install -y nmap zenmap

4. uninstall john the ripper

command:

apt remove -y john john-data

undo:

apt install -y john

5. uninstall hydra and xhydra

commands:

apt remove -y hydra hydra-gtk

undo:

apt install -y hydra

6. uninstall aircrack-ng and reaver

commands:

apt remove -y aircrack-ng reaver

undo:

apt install -y aircrack-ng

7. uninstall wireshark and tcpdump

commands:

apt remove -y wireshark tcpdump

undo:

apt install -y wireshark tcpdump

8. uninstall netcat and variants

commands:

apt remove -y netcat netcat-traditional netcat-openbsd ncat

undo:

apt install -y netcat-openbsd

9. uninstall sqlmap

command:

apt remove -y sqlmap

undo:

apt install -y sqlmap

10. uninstall nikto and dirbuster

commands:

apt remove -y nikto dirbuster

undo:

apt install -y nikto dirbuster

11. uninstall ettercap and dsniff

commands:

apt remove -y ettercap-graphical ettercap-text-only dsniff

undo:

apt install -y ettercap-graphical

12. uninstall netdiscover and arp-scan

commands:

apt remove -y netdiscover arp-scan

undo:

apt install -y netdiscover arp-scan

13. uninstall social engineering toolkit (setoolkit)

command:

apt remove -y set

undo:

apt install -y set

14. uninstall hping3

command:

apt remove -y hping3

undo:

apt install -y hping3

15. uninstall dnsenum, dnsmap, and dnsrecon

commands:

apt remove -y dnsenum dnsmap dnsrecon

undo:

apt install -y dnsenum dnsrecon

16. uninstall crunch and wordlist generators

commands:

apt remove -y crunch

rm -rf /usr/share/wordlists

undo:

apt install -y crunch

17. uninstall beef (browser exploitation framework)

command:

apt remove -y beef-xss

undo:

apt install -y beef-xss

18. uninstall nikto, wpscan, and other web scanners

commands:

apt remove -y wpscan nikto skipfish

undo:

apt install -y wpscan

19. uninstall net-tools replacements if unnecessary

command:

apt remove -y hping3 nmap socat

undo:

apt install -y socat

20. check for manually installed hacking tools

command:

grep -iE "metasploit|hydra|aircrack|beef|nmap|wireshark|john|sqlmap|ettercap|setoolkit|crunch" ~/.bash\_history /root/.bash\_history 2>/dev/null

manually inspect output and remove any custom-installed tools or scripts.

21. clear downloaded hacking scripts

commands:

find /home -type f ( -iname "\*exploit\*" -o -iname "\*payload\*" -o -iname "\*hacking\*" -o -iname "\*scanner\*" -o -iname "\*.py" -o -iname "\*.sh" ) | grep -E "metasploit|hydra|sqlmap|aircrack|john|beef|exploit"

delete confirmed malicious or hacking-related scripts manually.

22. check and clean /usr/local/bin for unsafe binaries

command:

ls -la /usr/local/bin

remove suspicious executables:

rm -f /usr/local/bin/<filename>

23. verify system is clean

commands:

dpkg -l | grep -E "metasploit|hydra|aircrack|john|sqlmap|wireshark|nmap|ettercap|beef|setoolkit|nikto|wpscan"

if any appear, remove again using apt remove -y <package\_name>

24. clean apt cache to ensure removed tools don’t remain locally

commands:

apt autoremove -y

apt clean

25. log completion of this step

command:

echo "$(date -u +"%Y-%m-%d %H:%M:%SZ") -- step 5 (hacking tool removal) completed" >> ~/Desktop/Script.log

6 Password Policy & PAM Security

1. back up current files (required)

what it does: saves originals so you can restore them if needed

command:

cp /etc/login.defs ~/Desktop/backups/login.defs.bak

cp /etc/pam.d/common-auth ~/Desktop/backups/common-auth.bak

cp /etc/pam.d/common-password ~/Desktop/backups/common-password.bak

2. install required packages

what it does: ensures pam cracklib (or pam pwquality) is available

command:

apt-get update -qq && apt-get install -y libpam-cracklib libpam-modules libpam-modules-bin

undo:

apt-get purge -y libpam-cracklib

3. set password aging defaults in /etc/login.defs

what it does: sets max days, min days, min length, and warning age system-wide

command:

sed -i.bak -E 's/^(PASS\_MAX\_DAYS)[[:space:]]+.\*/\1 30/' /etc/login.defs

sed -i -E 's/^(PASS\_MIN\_DAYS)[[:space:]]+.\*/\1 3/' /etc/login.defs

sed -i -E 's/^(PASS\_MIN\_LEN)[[:space:]]+.\*/\1 8/' /etc/login.defs

sed -i -E 's/^(PASS\_WARN\_AGE)[[:space:]]+.\*/\1 7/' /etc/login.defs

undo:

cp ~/Desktop/backups/login.defs.bak /etc/login.defs

note: the first sed creates /etc/login.defs.bak as an extra backup. restoring the user backup will revert everything.

4. ensure password hashing uses sha512 (common on modern systems)

what it does: ensures passwords are stored with SHA-512 hashing

command:

grep -q "pam\_unix.so" /etc/pam.d/common-password && sed -i.bak -r 's/(pam\_unix.so.\*)(sha512)?(.\*)/\1 sha512\3/' /etc/pam.d/common-password || echo "password [success=1 default=ignore] pam\_unix.so sha512" >> /etc/pam.d/common-password

undo:

cp ~/Desktop/backups/common-password.bak /etc/pam.d/common-password

5. add or ensure account lockout on repeated failures (pam\_tally / pam\_tally2 style)

what it does: locks accounts after repeated failed auth attempts to slow brute force

safety note: this can lock accounts (including admin) if misconfigured. back up first and test with a non-root account.

command:

grep -q "pam\_tally.so" /etc/pam.d/common-auth || echo "auth required pam\_tally.so deny=5 unlock\_time=900 onerr=fail even\_deny\_root\_account audit silent" >> /etc/pam.d/common-auth

grep -q "pam\_tally.so" /etc/pam.d/common-account || echo "account required pam\_tally.so" >> /etc/pam.d/common-account 2>/dev/null || true

undo:

cp ~/Desktop/backups/common-auth.bak /etc/pam.d/common-auth

cp ~/Desktop/backups/common-password.bak /etc/pam.d/common-password

alternative (faillock) for newer distributions:

you can use pam\_faillock instead if present; skip above and run:

apt-get install -y libpam-modules-bin

pam-auth-update --package

6. enforce password quality and history

what it does: enforces complexity, minimum length, and remembers previous passwords

command:

grep -q "pam\_cracklib.so" /etc/pam.d/common-password || echo "password requisite pam\_cracklib.so retry=3 minlen=8 difok=3 reject\_username minclass=3 maxrepeat=2 dcredit=1 ucredit=1 lcredit=1 ocredit=1" >> /etc/pam.d/common-password

grep -q "pam\_pwhistory.so" /etc/pam.d/common-password || echo "password requisite pam\_pwhistory.so use\_authtok remember=24 enforce\_for\_root" >> /etc/pam.d/common-password

undo:

cp ~/Desktop/backups/common-password.bak /etc/pam.d/common-password

7. add lockout-check to pam (prevent bypass)

what it does: ensure the auth stack checks the tally before granting access

command:

grep -q "pam\_tally.so" /etc/pam.d/common-auth || sed -i -e '1ipam\_tally\_placeholder' /etc/pam.d/common-auth && sed -i '1s/pam\_tally\_placeholder/auth required pam\_tally.so deny=5 unlock\_time=900 onerr=fail audit even\_deny\_root\_account silent/' /etc/pam.d/common-auth

undo:

cp ~/Desktop/backups/common-auth.bak /etc/pam.d/common-auth

8. apply password aging to all regular users (UID >= 1000)

what it does: sets max/min/warning days per user

command:

awk -F: '$3 >= 1000 {print $1}' /etc/passwd | xargs -r -n1 chage -M 30 -m 3 -W 7

undo:

# there is no single undo; restore from login.defs backup then individually adjust:

cp ~/Desktop/backups/login.defs.bak /etc/login.defs

# to clear per-user aging you can run for a user:

# chage -M -1 -m -1 -W -1 <username>

9. test authentication before logging out (do not close current root session)

what it does: tests non-root account login and lockout behavior so you don’t get locked out of admin accounts

command:

# open a second terminal or use su - <non-root-user> to test

su - <non-root-user> -c "echo testlogin"

note: do not log out of your root/admin shell until you confirm logins work as expected.

10. audit pam changes for syntax errors

what it does: quick sanity check for common errors that would prevent logins

command:

pamtester --help >/dev/null 2>&1 || apt-get install -y pamtester

pamtester login <your-user> authenticate

undo:

apt-get purge -y pamtester

11. record completion

what it does: log step completion in Script.log

command:

echo "$(date -u +"%Y-%m-%d %H:%M:%SZ") -- step 6 (password policy & pam security) completed" >> ~/Desktop/Script.log

\*\*pam hardening checklist\*\*

1. \*\*backup existing files\*\*

what it does: ensures you can undo changes if login breaks

commands:

cp /etc/pam.d/common-auth ~/Desktop/backups/common-auth.bak

cp /etc/pam.d/common-password ~/Desktop/backups/common-password.bak

cp /etc/pam.d/common-account ~/Desktop/backups/common-account.bak

---

2. \*\*lock accounts after repeated failed logins\*\*

file: `/etc/pam.d/common-auth`

line to add \*\*at the top of the file\*\* (before other `auth` lines):

auth required pam\_tally.so deny=5 unlock\_time=900 onerr=fail even\_deny\_root\_account audit silent

undo: restore from backup:

cp ~/Desktop/backups/common-auth.bak /etc/pam.d/common-auth

---

3. \*\*ensure account stack includes tally\*\*

file: `/etc/pam.d/common-account`

line to add \*\*anywhere in the file\*\* (after `account` lines):

account required pam\_tally.so

undo: restore backup:

cp ~/Desktop/backups/common-account.bak /etc/pam.d/common-account

---

4. \*\*enforce password complexity\*\*

file: `/etc/pam.d/common-password`

line to add \*\*after `password` lines\*\* (or at the end if none exist):

password requisite pam\_cracklib.so retry=3 minlen=8 difok=3 reject\_username minclass=3 maxrepeat=2 dcredit=1 ucredit=1 lcredit=1 ocredit=1

undo: restore backup:

cp ~/Desktop/backups/common-password.bak /etc/pam.d/common-password

---

5. \*\*enforce password history\*\*

file: `/etc/pam.d/common-password`

line to add \*\*after the pam\_cracklib line\*\*:

password requisite pam\_pwhistory.so use\_authtok remember=24 enforce\_for\_root

undo: restore backup:

cp ~/Desktop/backups/common-password.bak /etc/pam.d/common-password

---

6. \*\*ensure SHA-512 hashing for passwords\*\*

file: `/etc/pam.d/common-password`

line to edit: find the line starting with `password [success=1 default=ignore] pam\_unix.so` and ensure `sha512` is included:

password [success=1 default=ignore] pam\_unix.so sha512

undo: restore backup:

cp ~/Desktop/backups/common-password.bak /etc/pam.d/common-password

---

7. \*\*apply changes to all users\*\*

command: set password aging and warnings for all users (UID >= 1000):

awk -F: '$3 >= 1000 {print $1}' /etc/passwd | xargs -r -n1 chage -M 30 -m 3 -W 7

undo: manually reset per-user aging using:

chage -M -1 -m -1 -W -1 <username>

---

8. \*\*audit and test changes\*\*

commands:

pamtester login <non-root-user> authenticate

su - <non-root-user> -c "echo testlogin"

note: keep one root shell open until you confirm logins work.

---

9. \*\*log completion\*\*

command:

echo "$(date -u +"%Y-%m-%d %H:%M:%SZ") -- pam hardening completed" >> ~/Desktop/Script.log

7 Firewall & Network Security

1. back up current network and firewall configs

what it does: lets you restore prior state if something breaks

commands:

cp /etc/ufw/ufw.conf ~/Desktop/backups/ufw.conf.bak 2>/dev/null || true

cp /etc/ufw/before.rules ~/Desktop/backups/before.rules.bak 2>/dev/null || true

cp /etc/ufw/after.rules ~/Desktop/backups/after.rules.bak 2>/dev/null || true

cp /etc/sysctl.conf ~/Desktop/backups/sysctl.conf.bak 2>/dev/null || true

undo:

cp ~/Desktop/backups/ufw.conf.bak /etc/ufw/ufw.conf 2>/dev/null || true

cp ~/Desktop/backups/before.rules.bak /etc/ufw/before.rules 2>/dev/null || true

cp ~/Desktop/backups/after.rules.bak /etc/ufw/after.rules 2>/dev/null || true

cp ~/Desktop/backups/sysctl.conf.bak /etc/sysctl.conf 2>/dev/null || true

2. install and enable UFW (uncomplicated firewall)

what it does: provides a simple front-end for iptables and a standard firewall policy

commands:

apt-get update -qq

apt-get install -y ufw

ufw default deny incoming

ufw default allow outgoing

ufw logging on

ufw enable

undo (disable UFW):

ufw disable

3. set recommended UFW defaults (deny new inbound, allow related/established)

what it does: enforces a least-privilege default policy

commands:

ufw default deny incoming

ufw default allow outgoing

ufw reload

undo:

ufw default allow incoming && ufw reload

4. explicitly allow only required services (example set)

what it does: open only the services you need; replace or remove lines for services you will not use

commands (adjust to your needs):

ufw allow 22/tcp # SSH (only if you need remote shell)

ufw allow 80/tcp # HTTP (if web server required)

ufw allow 443/tcp # HTTPS (if web server required)

ufw allow 53/udp # DNS (if this host should resolve for others)

ufw allow 53/tcp # DNS over TCP

ufw allow 3306/tcp # MySQL (only if required)

ufw allow 137:139/tcp # Samba (only if required)

ufw allow 137:139/udp # Samba (only if required)

ufw reload

undo (remove an allow):

ufw delete allow 22/tcp # replace 22/tcp with the rule you want removed

5. deny specific risky ports used by the original script

what it does: explicitly denies a port shown in the original script and any other known risky ports

commands:

ufw deny 1337 # example from original script

ufw deny telnet # deny telnet service by name if available

ufw reload

undo:

ufw delete deny 1337

ufw delete deny telnet

6. block forged loopback and spoofed packets (iptables)

what it does: deny any packet claiming to be from a local address that doesn’t come via the loopback interface (prevents IP spoofing using 127.0.0.0/8)

commands:

iptables -I INPUT 1 -s 127.0.0.0/8 ! -i lo -j DROP

iptables -I INPUT 1 -s 0.0.0.0/8 -j DROP

iptables -I INPUT 1 -s 169.254.0.0/16 -j DROP

undo:

iptables -D INPUT -s 127.0.0.0/8 ! -i lo -j DROP

iptables -D INPUT -s 0.0.0.0/8 -j DROP

iptables -D INPUT -s 169.254.0.0/16 -j DROP

7. make the iptables rule persistent (so it survives reboots)

what it does: saves iptables rules to be loaded on boot

commands:

apt-get install -y iptables-persistent netfilter-persistent

netfilter-persistent save

undo:

apt-get remove -y iptables-persistent netfilter-persistent

8. harden kernel network settings via sysctl (recommended)

what it does: tunes kernel to ignore redirects, disable source routing, enable SYN cookies, and disable IP forwarding

commands:

cp /etc/sysctl.conf ~/Desktop/backups/sysctl.conf.before\_hardening.bak 2>/dev/null || true

cat >> /etc/sysctl.conf <<'EOF'

# network hardening for CyberPatriot

net.ipv4.ip\_forward = 0

net.ipv6.conf.all.forwarding = 0

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

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

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

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

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

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

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

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

net.ipv4.tcp\_syncookies = 1

net.ipv4.icmp\_echo\_ignore\_broadcasts = 1

net.ipv4.conf.all.log\_martians = 1

EOF

sysctl -p

undo:

cp ~/Desktop/backups/sysctl.conf.before\_hardening.bak /etc/sysctl.conf && sysctl -p

9. disable IPv6 (optional — only if you do not need IPv6)

what it does: turns IPv6 off to avoid misconfigured IPv6 services being exposed

commands (enable only if you are sure):

echo "net.ipv6.conf.all.disable\_ipv6 = 1" >> /etc/sysctl.conf

echo "net.ipv6.conf.default.disable\_ipv6 = 1" >> /etc/sysctl.conf

sysctl -p

undo:

sed -i '/disable\_ipv6/d' /etc/sysctl.conf && sysctl -p

10. drop invalid packets and limit new connections per IP (iptables)

what it does: reduces certain scan/DoS vectors and drops suspicious packets

commands:

iptables -N LIMIT\_NEW 2>/dev/null || true

iptables -F LIMIT\_NEW 2>/dev/null || true

iptables -A LIMIT\_NEW -m conntrack --ctstate NEW -m limit --limit 25/minute --limit-burst 100 -j RETURN

iptables -A LIMIT\_NEW -j LOG --log-prefix "LIMIT\_NEW\_DROP: " --log-level 4

iptables -A LIMIT\_NEW -j DROP

iptables -I INPUT -m conntrack --ctstate NEW -j LIMIT\_NEW

undo:

iptables -D INPUT -m conntrack --ctstate NEW -j LIMIT\_NEW

iptables -F LIMIT\_NEW

iptables -X LIMIT\_NEW

11. enable logging and inspect logs for blocked traffic

what it does: turns on logging so you can detect blocked or suspicious attempts

commands:

ufw logging on

dmesg | tail -n 50

journalctl -u ufw -n 200 --no-pager 2>/dev/null || tail -n 200 /var/log/ufw.log 2>/dev/null

undo:

ufw logging off

12. verify listening ports and services after firewall changes

what it does: confirms only expected services are listening and accessible

commands:

ss -tulpn | sed -n '1,200p'

or

netstat -tulpn 2>/dev/null | sed -n '1,200p'

13. block outgoing connections for specific risky apps (optional)

what it does: prevents known risky tools from calling home (example blocks)

commands:

ufw deny out 1337/tcp

ufw deny out 1337/udp

undo:

ufw delete deny out 1337/tcp

ufw delete deny out 1337/udp

14. check UFW status and ruleset for audit / documentation

what it does: produce an exportable snapshot you can put in /Desktop/logs

commands:

ufw status verbose > ~/Desktop/logs/ufw-status.txt

iptables-save > ~/Desktop/logs/iptables-save.txt

15. save a short revert helper (recommended)

what it does: creates a script you can run to quickly revert the hardening changes from this step

commands:

cat > ~/Desktop/network-revert.sh <<'EOF'

#!/bin/bash

# revert UFW and iptables changes from step 7

ufw disable

iptables -F

iptables -X

if [ -f ~/Desktop/backups/sysctl.conf.before\_hardening.bak ]; then cp ~/Desktop/backups/sysctl.conf.before\_hardening.bak /etc/sysctl.conf && sysctl -p; fi

echo "Revert complete. UFW disabled and iptables flushed. sysctl restored if backup found."

EOF

chmod +x ~/Desktop/network-revert.sh

undo (run the revert script):

~/Desktop/network-revert.sh

16. log completion of this step

command:

echo "$(date -u +"%Y-%m-%d %H:%M:%SZ") -- step 7 (firewall & network security) completed" >> ~/Desktop/Script.log

8 Crontab & Startup Security

1. \*\*backup package and service info\*\*

what it does: allows you to restore previous state if something breaks

commands:

dpkg --get-selections > ~/Desktop/backups/packages-list.bak

systemctl list-unit-files > ~/Desktop/backups/services-list.bak

undo:

dpkg --set-selections < ~/Desktop/backups/packages-list.bak

apt-get dselect-upgrade -y

---

2. \*\*update and upgrade system packages\*\*

what it does: ensures all installed software has latest security updates

commands:

apt-get update -qq

apt-get upgrade -y -qq

apt-get dist-upgrade -y -qq

undo: cannot “downgrade” automatically; revert by restoring from a system snapshot or backup

---

3. \*\*remove dangerous or hacking tools\*\*

what it does: removes common pentesting/cracking tools from the system

commands:

apt-get purge -y -qq john john-data hydra hydra-gtk aircrack-ng fcrackzip lcrack ophcrack ophcrack-cli pdfcrack pyrit rarcrack sipcrack irpas netcat netcat-openbsd netcat-traditional ncat pnetcat socat sock socket sbd

undo: reinstall packages if needed:

apt-get install -y <package\_name>

---

4. \*\*remove unnecessary media files if policy requires\*\*

what it does: deletes audio, video, and image files to reduce storage of potentially unsafe content

commands:

find / -type f ( -iname "\*.mp3" -o -iname "\*.wav" -o -iname "\*.avi" -o -iname "\*.mp4" -o -iname "\*.mkv" -o -iname "\*.flv" -o -iname "\*.mov" -o -iname "\*.jpg" -o -iname "\*.png" -o -iname "\*.gif" ) -delete

undo: cannot undo; must restore from backup

---

5. \*\*remove unnecessary services\*\*

what it does: disables services that are not required based on your system role

commands:

systemctl disable samba.service 2>/dev/null

systemctl disable vsftpd.service 2>/dev/null

systemctl disable apache2.service 2>/dev/null

systemctl disable bind9.service 2>/dev/null

systemctl disable telnet.service 2>/dev/null

systemctl disable ssh.service 2>/dev/null (only if SSH not required)

systemctl disable cups.service 2>/dev/null

undo: enable again:

systemctl enable <service\_name>

---

6. \*\*stop unnecessary running services immediately\*\*

what it does: ensures disabled services are not active now

commands:

systemctl stop samba.service 2>/dev/null

systemctl stop vsftpd.service 2>/dev/null

systemctl stop apache2.service 2>/dev/null

systemctl stop bind9.service 2>/dev/null

systemctl stop telnet.service 2>/dev/null

systemctl stop ssh.service 2>/dev/null

systemctl stop cups.service 2>/dev/null

undo: start again:

systemctl start <service\_name>

---

7. \*\*check for and remove auto-start scripts\*\*

what it does: prevents risky scripts from running on boot

commands:

ls /etc/rc.local

cp /etc/rc.local ~/Desktop/backups/rc.local.bak

echo -e "#!/bin/bash\nexit 0" > /etc/rc.local

chmod +x /etc/rc.local

undo: restore backup:

cp ~/Desktop/backups/rc.local.bak /etc/rc.local

---

8. \*\*remove unnecessary cron jobs\*\*

what it does: removes scheduled tasks for non-root users to reduce attack surface

commands:

crontab -l > ~/Desktop/backups/crontab-root.bak

crontab -r

undo: restore backup:

crontab ~/Desktop/backups/crontab-root.bak

---

9. \*\*verify running services\*\*

what it does: ensures only expected services are running

commands:

systemctl list-units --type=service --state=running

---

10. \*\*log completion\*\*

commands:

echo "$(date -u +"%Y-%m-%d %H:%M:%SZ") -- step 8 (services and packages hardening) completed" >> ~/Desktop/Script.log

---

9 System Updates & Repository Fixes

create backup folders

what it does: stores critical files and logs before changes

commands:

mkdir -p ~/Desktop/backups

chmod 777 ~/Desktop/backups

undo: remove backup folder (optional):

rm -rf ~/Desktop/backups

backup important system files

what it does: allows restoring files if something breaks

commands:

cp /etc/hosts ~/Desktop/backups/hosts.bak

cp /etc/mdm/mdm.conf ~/Desktop/backups/mdm.conf.bak

cp /etc/login.defs ~/Desktop/backups/login.defs.bak

cp /etc/pam.d/common-auth ~/Desktop/backups/common-auth.bak

cp /etc/pam.d/common-password ~/Desktop/backups/common-password.bak

cp /etc/apt/sources.list ~/Desktop/backups/sources.list.bak

cp /etc/init/control-alt-delete.conf ~/Desktop/backups/control-alt-delete.conf.bak

undo: restore backups:

cp ~/Desktop/backups/.bak

set bash history permissions

what it does: prevents unauthorized users from reading command history

commands:

chmod 640 ~/.bash\_history

undo: reset default:

chmod 600 ~/.bash\_history

secure shadow file permissions

what it does: ensures only root can read/write hashed passwords

commands:

chmod 604 /etc/shadow

undo: restore default:

chmod 000 /etc/shadow (or chmod 640 /etc/shadow depending on distro)

check user home folders

what it does: identifies orphaned or incorrectly owned home directories

commands:

ls -la /home/ >> ~/Desktop/Script.log

undo: no change; just logging

check sudoers directory

what it does: identifies files granting admin rights that may be unnecessary

commands:

ls -la /etc/sudoers.d >> ~/Desktop/Script.log

undo: no change; just logging

secure MDM login manager

what it does: prevents root login via GUI and sets auto-login to authorized user

commands:

echo -e "[daemon]\nAutomaticLoginEnable=true\nAutomaticLogin=$USER\nTimedLoginEnable=true\nTimedLogin=$USER\nTimedLoginDelay=10\n\n[security]\nAllowRoot=false" > /etc/mdm/mdm.conf

undo: restore backup:

cp ~/Desktop/backups/mdm.conf.bak /etc/mdm/mdm.conf

secure /etc/hosts file

what it does: prevents unauthorized modifications and restores default entries

commands:

chmod 777 /etc/hosts

cp /etc/hosts ~/Desktop/backups/hosts.bak

echo -e "127.0.0.1 localhost\n127.0.1.1 $USER\n::1 ip6-localhost ip6-loopback\nfe00::0 ip6-localnet\nff00::0 ip6-mcastprefix\nff02::1 ip6-allnodes\nff02::2 ip6-allrouters" > /etc/hosts

chmod 644 /etc/hosts

undo: restore backup:

cp ~/Desktop/backups/hosts.bak /etc/hosts

disable Ctrl-Alt-Delete reboot

what it does: prevents accidental or unauthorized system reboot

commands:

sed -i '/^exec/ c\exec false' /etc/init/control-alt-delete.conf

undo: restore backup:

cp ~/Desktop/backups/control-alt-delete.conf.bak /etc/init/control-alt-delete.conf

set root account locked (already done in step 8, repeat for safety)

commands:

usermod -L root

undo:

usermod -U root

secure crontab access

what it does: allows only root to schedule jobs

commands:

cd /etc

/bin/rm -f cron.deny at.deny

echo root > cron.allow

echo root > at.allow

chown root:root cron.allow at.allow

chmod 400 cron.allow at.allow

cd -

undo: remove restrictions (less secure):

rm /etc/cron.allow /etc/at.allow

backup logs for auditing

what it does: preserves system and auth logs for inspection

commands:

mkdir -p ~/Desktop/logs

chmod 777 ~/Desktop/logs

cp /var/log/auth.log ~/Desktop/logs/auth.log

cp /var/log/syslog ~/Desktop/logs/syslog.log

cp /etc/services ~/Desktop/logs/allports.log

dpkg -l > ~/Desktop/logs/packages.log

apt-mark showmanual > ~/Desktop/logs/manuallyinstalled.log

service --status-all > ~/Desktop/logs/allservices.txt

ps ax > ~/Desktop/logs/processes.log

ss -l > ~/Desktop/logs/socketconnections.log

netstat -tlnp > ~/Desktop/logs/listeningports.log

undo: no undo needed; logs are backups

log completion

commands:

echo "$(date -u +"%Y-%m-%d %H:%M:%SZ") -- step 9 (filesystem & logs hardening) completed" >> ~/Desktop/Script.log

10 System Logging and Verification

1. \*\*install and enable UFW firewall\*\*

what it does: ensures firewall is active and ready to block/allow traffic

commands:

apt-get install ufw -y -qq

ufw enable

undo: disable firewall:

ufw disable

---

2. \*\*block a specific risky port (example: 1337)\*\*

what it does: prevents access to a port known to be unused or risky

commands:

ufw deny 1337

undo: allow port again:

ufw allow 1337

---

3. \*\*configure firewall for common services based on system role\*\*

\* \*\*Samba\*\*

if not needed:

ufw deny samba

apt-get purge -y samba samba-common samba-common-bin samba4

if needed:

ufw allow samba

undo: reverse deny/allow using the opposite command

\* \*\*FTP\*\*

if not needed:

ufw deny ftp

ufw deny sftp

ufw deny saft

ufw deny ftps-data

ufw deny ftps

apt-get purge -y vsftpd

if needed:

ufw allow ftp

ufw allow sftp

ufw allow saft

ufw allow ftps-data

ufw allow ftps

service vsftpd restart

undo: reverse deny/allow using the opposite command; reinstall service if purged

\* \*\*SSH\*\*

if not needed:

ufw deny ssh

apt-get purge -y openssh-server

if needed:

ufw allow ssh

backup config: cp /etc/ssh/sshd\_config ~/Desktop/backups/sshd\_config.bak

ensure security settings:

\* `PermitRootLogin no`

\* `Protocol 2`

\* `X11Forwarding no`

\* `PermitEmptyPasswords no`

service ssh restart

undo: restore backup or reinstall SSH if purged

\* \*\*Telnet\*\*

if not needed:

ufw deny telnet

apt-get purge -y telnet telnetd inetutils-telnetd telnetd-ssl

if needed:

ufw allow telnet

undo: reinstall service if purged

\* \*\*Mail (SMTP/POP/IMAP)\*\*

if not needed:

ufw deny smtp

ufw deny pop2

ufw deny pop3

ufw deny imap2

ufw deny imaps

ufw deny pop3s

if needed:

ufw allow smtp

ufw allow pop2

ufw allow pop3

ufw allow imap2

ufw allow imaps

ufw allow pop3s

undo: reverse deny/allow using opposite commands

\* \*\*Printing (CUPS/IPP)\*\*

if not needed:

ufw deny ipp

ufw deny printer

ufw deny cups

if needed:

ufw allow ipp

ufw allow printer

ufw allow cups

undo: reverse deny/allow using opposite commands

\* \*\*MySQL/Database ports\*\*

if not needed:

ufw deny mysql

ufw deny mysql-proxy

ufw deny ms-sql-s

ufw deny ms-sql-m

apt-get purge -y mysql mysql-client-core-5.5 mysql-server mysql-server-5.5 mysql-client-5.5

if needed:

ufw allow mysql

ufw allow mysql-proxy

ufw allow ms-sql-s

ufw allow ms-sql-m

backup config files:

cp /etc/my.cnf ~/Desktop/backups/

cp /etc/mysql/my.cnf ~/Desktop/backups/

service mysql restart

undo: reinstall database or reverse allow/deny commands

\* \*\*Web server (HTTP/HTTPS)\*\*

if not needed:

ufw deny http

ufw deny https

apt-get purge -y apache2

rm -rf /var/www/\*

if needed:

ufw allow http

ufw allow https

backup config: cp /etc/apache2/apache2.conf ~/Desktop/backups/apache2.conf.bak

restrict directory:

echo -e '<Directory >\n\t AllowOverride None\n\t Order Deny,Allow\n\t Deny from all\n<Directory />\nUserDir disabled root' >> /etc/apache2/apache2.conf

chown -R root:root /etc/apache2

undo: reinstall apache2 if purged; restore backup if needed

\* \*\*DNS (BIND9)\*\*

if not needed:

ufw deny domain

apt-get purge -y bind9

if needed:

ufw allow domain

undo: reinstall bind9 if purged; reverse allow/deny

---

4. \*\*log completion\*\*

commands:

echo "$(date -u +"%Y-%m-%d %H:%M:%SZ") -- step 10 (network hardening) completed" >> ~/Desktop/Script.log

---

11 Final Cleanup, Auditing, and Verification

1. \*\*remove all temporary files\*\*

what it does: cleans up leftover temp files that may contain sensitive info

commands:

rm -rf /tmp/\* /var/tmp/\*

undo: cannot undo; only restore from backup if needed

---

2. \*\*remove unnecessary media files if policy requires (repeat for safety)\*\*

what it does: ensures no audio, video, or image files remain

commands:

find / -type f ( -iname "\*.mp3" -o -iname "\*.wav" -o -iname "\*.avi" -o -iname "\*.mp4" -o -iname "\*.mkv" -o -iname "\*.flv" -o -iname "\*.mov" -o -iname "\*.jpg" -o -iname "\*.png" -o -iname "\*.gif" ) -delete

undo: cannot undo; restore from backup

---

3. \*\*verify system logs\*\*

what it does: ensures logs are intact and permissions are secure

commands:

ls -la ~/Desktop/logs

chmod 600 ~/Desktop/logs/\*

undo: adjust permissions if needed, e.g., chmod 777 ~/Desktop/logs/\*

---

4. \*\*verify running services\*\*

what it does: ensures only expected services are active

commands:

systemctl list-units --type=service --state=running

ss -l

netstat -tlnp

undo: start/stop services as required

---

5. \*\*verify user accounts\*\*

what it does: ensures no unexpected users exist and privileges are correct

commands:

cat /etc/passwd | grep -v 'nologin'

grep '^sudo:.\*$' /etc/group

grep '^adm:.\*$' /etc/group

grep '^lpadmin:.\*$' /etc/group

grep '^sambashare:.\*$' /etc/group

undo: usermod/gpasswd to add/remove users from groups

---

6. \*\*verify crontab\*\*

what it does: ensures no non-root scheduled jobs remain

commands:

ls -la /etc/cron\*

crontab -l

undo: restore from backup if needed:

crontab ~/Desktop/backups/crontab-root.bak

---

7. \*\*verify firewall rules\*\*

what it does: ensures only allowed ports are open

commands:

ufw status verbose

undo: reverse allow/deny commands as needed

---

8. \*\*verify file permissions\*\*

what it does: ensures key files have secure permissions

commands:

ls -la /etc/shadow

ls -la ~/.bash\_history

ls -la /etc/apt/sources.list

ls -la /etc/hosts

ls -la /etc/mdm/mdm.conf

undo: chmod/chown as needed

---

9. \*\*final update and cleanup\*\*

what it does: ensures all packages are up to date and system is clean

commands:

apt-get update -qq

apt-get upgrade -y -qq

apt-get dist-upgrade -y -qq

apt-get autoremove -y -qq

apt-get autoclean -y -qq

apt-get clean -y -qq

undo: cannot easily undo; restore from backup if needed

---

10. \*\*create final log of system state\*\*

what it does: saves an audit trail for verification

commands:

dpkg -l > ~/Desktop/logs/final-packages.log

service --status-all > ~/Desktop/logs/final-services.log

ps ax > ~/Desktop/logs/final-processes.log

ss -l > ~/Desktop/logs/final-socketconnections.log

netstat -tlnp > ~/Desktop/logs/final-listeningports.log

echo "$(date -u +"%Y-%m-%d %H:%M:%SZ") -- step 11 (final cleanup & audit) completed" >> ~/Desktop/Script.log