CYBERPATRIOT LINUX (UBUNTU / MINT) ULTRA–DETAILED CHECKLIST
Copy/paste this whole box into Docs. Work top-to-bottom each round.

==========
0. BEFORE YOU TOUCH ANYTHING IN THE VM
==========

0.1 Host machine + VM prep (before the round starts)
- Verify:
  - VMware / Hyper-V / Player is installed and working.
  - You have enough RAM and disk space (at least 4 GB RAM per Linux VM is recommended).
[oai_citation:0‡Basic-Linux-Security-Reading.pdf](sediment://file_00000000b2c871f581527f7efffe3079)
  - Images are extracted and bootable.
  - You know the team ID and extraction password.
[oai_citation:1‡Basic-Linux-Security-Reading.pdf](sediment://file_00000000b2c871f581527f7efffe3079)
- Open:
  - A text editor on the host (Notepad/Docs) to log every change and its approximate score impact.
  - Scoreboard in a browser (host, not in VM).

0.2 Immediately after boot
- Give VM enough RAM if you can (around 4 GB).
[oai_citation:2‡Basic-Linux-Security-Reading.pdf](sediment://file_00000000b2c871f581527f7efffe3079)
- Log in with credentials from the README.
- Confirm distro:
  - Ubuntu 22 / Mint 21 / Mint 22, etc. (affects some GUI locations).
- Open:
  - Terminal (pin to panel if needed).
  - File manager.
  - System Settings.

0.3 Absolute order of operations (DO NOT BREAK THIS)
1) READ THE README COMPLETELY (twice).
2) Open and READ all Forensics Question files on the Desktop.
3) Answer forensics that could be broken by system changes (hashes, deleted files, network captures, etc.).
[oai_citation:3‡Basic-Linux-Security-Reading.pdf](sediment://file_00000000b2c871f581527f7efffe3079)
4) THEN start hardening: users → passwords → local policy → services → firewall → updates → browser → prohibited software/media.

[oai_citation:4‡Basic-Linux-Security-Reading.pdf](sediment://file_00000000b2c871f581527f7efffe3079)

0.4 Notes discipline
- Every time you change something, note:
  - Time, command or GUI steps.
  - Which category you think it affects (User auditing, Account policy, Prohibited files, etc.).
[oai_citation:5‡CP 18 r1 Img Ans and Vulns.pdf](sediment://file_00000000f4e0722fb7fc097d31b24d68)
- If you reboot or VM crashes, this lets you quickly redo critical fixes.

==========
1. FORENSICS WORKFLOW (DO FIRST)
==========

1.1 General approach
- On Desktop, open each “Forensics Question X” file.
- Before changing the system:
  - If it asks you to find evidence in Trash, specific dirs, or pcap files, do that **before** deleting or cleaning anything.
[oai_citation:6‡CP-Ubuntu22-Training2-Answer-Key.pdf](sediment://file_00000000683c71fb943e9a804acaaa0c)
  - If it asks for a hash, compute it before editing the file.
[oai_citation:7‡Basic-Linux-Security-Reading.pdf](sediment://file_00000000b2c871f581527f7efffe3079)

1.2 Common commands you will likely need
- Compute an MD5 (or other) checksum:
  - `md5sum file.txt`
[oai_citation:8‡Basic-Linux-Security-Reading.pdf](sediment://file_00000000b2c871f581527f7efffe3079)
- Decode base64 text (if given a base64 blob):
  - Option 1 (CLI): `echo "BASE64_STRING" | base64 -d`
  - Option 2 (as in Ubuntu22 Training 2): copy text into an online base64 decoder in the VM browser; record the decoded instructions and then follow them.
[oai_citation:9‡CP-Ubuntu22-Training2-Answer-Key.pdf](sediment://file_00000000683c71fb943e9a804acaaa0c)
- Search for files by extension:
  - Update database: `sudo updatedb` (if `locate` is available).
[oai_citation:10‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
  - Then: `locate '*.mp3'`, `locate '*.mp4'`, `locate '*.ogg'`, etc.
[oai_citation:11‡CP-Ubuntu22-Training2-Answer-Key.pdf](sediment://file_00000000683c71fb943e9a804acaaa0c)

1.3 PCAP / network forensics (Mint Training 2 style)
- If you see a .pcap on the Desktop:
  - Install tshark if needed: `sudo apt install tshark`
[oai_citation:12‡CP-Mint21-Training2-Answer-Key.pdf](sediment://file_00000000269c722faa8688903b5b2a62)
  - `cd ~/Desktop`
  - To follow TCP streams and look for passwords:
    - `tshark -q -z follow,tcp,ascii,0 -r capture.pcap | grep PASS`
    - Change the stream index (0, 1, 2, …, 21) until you find the credentials.
[oai_citation:13‡CP-Mint21-Training2-Answer-Key.pdf](sediment://file_00000000269c722faa8688903b5b2a62)

1.4 Finding backdoors related to forensics
- If question hints at a Python backdoor:
  - `ss -tlnp` → find suspicious listening ports.
[oai_citation:14‡CP-Mint21-Training2-Answer-Key.pdf](sediment://file_00000000269c722faa8688903b5b2a62)
  - `ps -ef | grep python` → see which script is bound to that port.
[oai_citation:15‡CP-Mint21-Training2-Answer-Key.pdf](sediment://file_00000000269c722faa8688903b5b2a62)
  - Take note of the directory (e.g., `/usr/share/zod/`). Answer the question; later you'll remove/fix it to close the vulnerability.

1.5 Deleted files (Trash)
- If the question mentions deleted files:
  - Open Trash via file manager.
  - Look for suspicious images/documents (e.g., `l.jpg`) and open them.
[oai_citation:16‡CP-Ubuntu22-Training2-Answer-Key.pdf](sediment://file_00000000683c71fb943e9a804acaaa0c)

1.6 Do NOT:
- Don't delete or clean up anything related to a forensic question until:
  - You've answered it AND
  - You've written the answer in the scoring client / README as required.

==========
2. USER & GROUP AUDIT (USER AUDITING CATEGORY)
==========

2.1 List all human accounts
- In terminal:
  - `cut -d: -f1 /etc/passwd` (quick list)
  - Or: `getent passwd` (full info).

- Compare against:
  - Users listed as allowed/required in the README.
  - Answer keys from past rounds show expected pattern:
    - Remove users like `ttanner`, `cdennis`, `dhardman`, etc. when they're not supposed to exist. [oai_citation:17‡CP 18 r1 Img Ans and Vulns.pdf](sediment://file_00000000f4e0722fb7fc097d31b24d68)
    - Similar unauthorized names show up in new rounds; pattern is the same.

2.2 Remove unauthorized users
- For each user not in README:
  - `sudo deluser --remove-home baduser` (deletes account and home). [oai_citation:18‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
- Confirm removal:
  - `id baduser` should fail.
- Typical scoring events: "Removed unauthorized user <name>". [oai_citation:19‡CP 18 r2 Img Ans & Vulns.pdf](sediment://file_00000000c3f871f599b22dd45098cacd)

2.3 Add missing required users
- If README says a user must exist but doesn't:
  - `sudo adduser username`
    - Set a secure password following policy (see section 3).
  - If they must be administrator:
    - `sudo usermod -aG sudo username` (or GUI Users & Groups). [oai_citation:20‡Basic-Linux-Checklist.pdf](sediment://file_000000008a3471f5bf9ae4604f17f3b1)

2.4 Correct group membership (admin vs standard)
- Check groups:
  - `getent group`
  - Focus on `sudo`, `adm`, and any custom groups described in README.
- For each user:
  - `id username`
    - If the user should be admin but isn't in `sudo`, add them.
    - If the user is admin but should be standard, remove them from `sudo`:
      - `sudo gpasswd -d username sudo`
- Examples from CP rounds:
  - "User dscott is not an administrator" (remove from sudo). [oai_citation:21‡CP 18 r2 Img Ans & Vulns.pdf](sediment://file_00000000c3f871f599b22dd45098cacd)
  - "User mross must change password at next login" (see 3.4). [oai_citation:22‡CP 18 r1 Img Ans and Vulns.pdf](sediment://file_00000000f4e0722fb7fc097d31b24d68)

2.5 Disable/lock accounts that must exist but be inactive
- If README says user exists but should be disabled:

  - `sudo passwd -l username` (locks the account).
[oai_citation:23‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
- Always lock the `root` login (but DO NOT delete root):
  - `sudo passwd -l root` (after ensuring you don't need direct root login).
[oai_citation:24‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)

==========
3. PASSWORDS & ACCOUNT POLICY (ACCOUNT POLICY CATEGORY)
==========

3.1 Change insecure or blank passwords
- From README or user list, identify:
  - Users with weak/known passwords.
  - Users with no password (blank).
- Commands:
  - `sudo passwd username`
    - Choose a strong password: length ≥ 12, mix of upper/lowercase, numbers, symbols.
- Past scoring examples:
  - "Changed insecure password for user llitt" [oai_citation:25‡CP 18 r2 Img Ans & Vulns.pdf](sediment://file_00000000c3f871f599b22dd45098cacd)
  - "User sbandaru has a password" (fix blank password). [oai_citation:26‡CP 18 r2 Img Ans & Vulns.pdf](sediment://file_00000000c3f871f599b22dd45098cacd)

3.2 System-wide password aging (login.defs)
- Edit `/etc/login.defs` with gedit or nano:
  - `sudo gedit /etc/login.defs` or `sudo nano /etc/login.defs`
[oai_citation:27‡Basic-Linux-Checklist.pdf](sediment://file_000000008a3471f5bf9ae4604f17f3b1)
- Set reasonable values (unless README specifies otherwise):
  - `PASS_MAX_DAYS 90`   (force password change at least every 90 days)
  - `PASS_MIN_DAYS 10`   (prevent rapid reuse)
  - `PASS_WARN_AGE 7`   (warn 7 days before expiry).
- This is the key place CP checks for "default minimum password age is set" etc.
[oai_citation:28‡CP 18 r1 Img Ans and Vulns.pdf](sediment://file_00000000f4e0722fb7fc097d31b24d68)

3.3 Apply password aging to existing users
- For each real human user:
  - `sudo chage -M 90 -m 10 -W 7 username`
- Verify:
  - `chage -l username` should show correct max/min/warn settings.

3.4 Enforce “must change at next login” where required
- If README or scoring hints say:
  - “User <name> must change password at next login”
- Command:
  - `sudo chage -d 0 username` (forces change next login).
- CP example: “User mross must change password at next login”, “User edarby must change password at next login”.

3.5 Password complexity & history (PAM)
- Install password quality module if needed:
  - `sudo apt-get install libpam-cracklib` (or pam_pwquality on newer systems).
[oai_citation:29‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
- Edit `/etc/pam.d/common-password`:
  - `sudo gedit /etc/pam.d/common-password`
[oai_citation:30‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
- On the `pam_unix.so` line, add:
  - `remember=5` (prevents reuse of last 5 passwords).
[oai_citation:31‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
- On the `pam_cracklib.so` or `pam_pwquality.so` line, add options such as:
  - `ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1 minlen=12`
  - This enforces at least one uppercase, lowercase, digit, and other symbol, plus minimum length.
- Save and close.

3.6 Account lockout policy (high-level approach)
- Look for a GUI or PAM-based setting to:
  - Lock account after several failed logons.
- On systems using `faillock`:
  - Configure `/etc/security/faillock.conf` (or equivalent) to:
    - `deny = 5` (lock after 5 bad attempts)
    - `lock_time = 900` (lock for 15 minutes)
- This corresponds to “A secure lockout threshold exists” or “secure account lockout duration exists” on score sheets.

==========
4. LOGIN / DISPLAY MANAGER SECURITY (GUEST, AUTOLOGIN)
==========

4.1 Disable guest account & autologin (LightDM)
- Edit LightDM config:

  - `sudo gedit /etc/lightdm/lightdm.conf`
[oai_citation:32‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
- Make sure:
  - Remove or comment out any line with `autologin-user=...`
  - Add / ensure:
    - `allow-guest=false` or `allow_guest=false` (depending on version).
- This fixes:
  - “Guest account is disabled”
  - “Do not allow automatic login”.

4.2 Limit empty-password logins
- Many rounds check “Limit local use of blank passwords to console only [enabled]” on Windows; equivalent on Linux:
  - Ensure no user has empty password.
  - Lock or delete any account with no password set (section 3.1).
  - For SSH, require passwords or keys (see 6.3).

==========
5. PROGRAMS, SERVICES & DAEMONS (SERVICE AUDITING)
==========

5.1 Enumerate running services
- Terminal:
  - `sudo service --status-all` (SysV style listing).
[oai_citation:33‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
  - `systemctl list-units --type=service --state=running`
- Look for:
  - Services explicitly listed as **critical** in README (eg, OpenSSH, FTP servers in some images).
  - Unnecessary or prohibited services (telnet, vnc, nfs, apache if not required, etc.).
[oai_citation:34‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)

5.2 Removing prohibited services
- For each prohibited service (telnet, vnc, etc.):
  - `sudo apt-get purge <servicename>`
  - Example: `sudo apt-get purge telnet`
[oai_citation:35‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
- Confirm they are gone:
  - `sudo service <service> status` (should fail or show not found).
- Corresponds to vulnerabilities like:

- “Apache2 service has been disabled or removed”, “prohibited software Wireshark removed”, etc.

5.3 Critical services that must remain but be secured
- READ README CAREFULLY. Some services **must stay installed and running**, but must be configured securely.
  - Example from Ubuntu 22 Training:
    - OpenSSH is a critical service; disabling it gives a penalty.
[oai_citation:36‡CP-Ubuntu22-Training2-Answer-Key.pdf](sediment://file_00000000683c71fb943e9a804acaaa0c)
  - Example from Mint practice:
    - Vsftpd must remain but with correct permissions & SSL.

- Do NOT:
  - Remove Firefox, Thunderbird, Perl, or OpenSSH if README calls them required. Penalties exist for that.
[oai_citation:37‡CP-Ubuntu22-Training2-Answer-Key.pdf](sediment://file_00000000683c71fb943e9a804acaaa0c)

5.4 Boot-time startup & cron
- Check startup scripts:
  - `/etc/init.d/` (older) and `systemctl` services.
  - `sudo crontab -e` and `/etc/cron.d/` for scheduled tasks.
- Remove malicious or unnecessary entries:
  - Look for `nc.traditional` or strange scripts in crontab; remove the line.
[oai_citation:38‡CP-Ubuntu22-Training2-Answer-Key.pdf](sediment://file_00000000683c71fb943e9a804acaaa0c)
  - Then kill and delete the binary:
    - `sudo pkill -f nc.traditional`
    - `which nc.traditional`
    - `sudo rm /usr/bin/nc.traditional`
[oai_citation:39‡CP-Ubuntu22-Training2-Answer-Key.pdf](sediment://file_00000000683c71fb943e9a804acaaa0c)

==========
6. SSH / REMOTE ACCESS HARDENING
==========

6.1 Find SSH config
- Typical location: `/etc/ssh/sshd_config`
- Edit with:
  - `sudo gedit /etc/ssh/sshd_config` or

  - `gedit admin:///etc/ssh/sshd_config` (GNOME admin method).
[oai_citation:40‡CP-Ubuntu22-Training2-Answer-Key.pdf](sediment://file_00000000683c71fb943e9a804acaaa0c)

6.2 Disable root SSH login
- In sshd_config:
  - Find `PermitRootLogin` line.
  - Set:
    - `PermitRootLogin no`
- Save and restart SSH:
  - `sudo systemctl restart sshd` or `sudo systemctl restart ssh`

6.3 Additional SSH tightening (only if not breaking functionality)
- Consider:
  - `PasswordAuthentication yes` if keys not configured, but ensure strong passwords.
  - Restrict which users can SSH in with `AllowUsers` or `AllowGroups` if README suggests secure remote use.
- DO NOT disable SSH entirely if README lists it as a critical service (there's a penalty in some practice images).
[oai_citation:41‡CP-Ubuntu22-Training2-Answer-Key.pdf](sediment://file_00000000683c71fb943e9a804acaaa0c)

==========
7. FIREWALL & NETWORK HARDENING (DEFENSIVE COUNTERMEASURES)
==========

7.1 Enable UFW firewall
- Basic commands:
  - `sudo ufw status` (check current)
  - `sudo ufw enable`
  - (If you break SSH/critical service, adjust rules accordingly.)

7.2 Configure firewall rules (read README)
- Allow only necessary ports:
  - For SSH: `sudo ufw allow 22/tcp`
  - For FTP (if required): `sudo ufw allow 21/tcp`
- Deny everything else by default (UFW does this when enabled unless changed).
- Verify:
  - `sudo ufw status verbose` (check logging and default deny).
[oai_citation:42‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
- CP scoring items:
  - "Uncomplicated Firewall (UFW) protection has been enabled".

7.3 TCP SYN cookies & basic sysctl hardening
- Edit `/etc/sysctl.conf`:
  - `sudo gedit /etc/sysctl.conf`
[oai_citation:43‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
- Ensure:
  - `net.ipv4.tcp_syncookies = 1` (protects against some SYN flood attacks).
- Apply:
  - `sudo sysctl -p`

==========
8. PROHIBITED SOFTWARE & MALICIOUS PROGRAMS
==========

8.1 List installed packages
- `dpkg --get-selections` or `apt list --installed`
- Pipe to `grep` for suspicious software:
  - `| grep wireshark`
  - `| grep amule`
  - `| grep ophcrack`
  - `| grep zod`
  - etc.

8.2 Remove prohibited applications
- For each prohibited app (from README or known CP rounds):
  - `sudo apt-get purge <APP>`
- Examples CP has removed:
  - Wireshark, aMule, Zangband, aisleriot, ophcrack.

8.3 Hunt for backdoors and network listeners
- Check open ports:
  - `sudo ss -tulnp` (preferred over netstat on newer distros).
- Identify suspicious processes:
  - `ps -ef | grep <name>` (for python scripts, netcat, weird daemons).
- If you see something like `nc.traditional` or weird scripts:
  - Remove associated cron/systemd entries (see 5.4).
  - Kill the process & delete the binary if not required.
[oai_citation:44‡CP-Ubuntu22-Training2-Answer-Key.pdf](sediment://file_00000000683c71fb943e9a804acaaa0c)

==========
9. PROHIBITED MEDIA & FILE PERMISSIONS
==========

9.1 Find prohibited media files
- Update locate database:
  - `sudo updatedb`
[oai_citation:45‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
- Search common extensions:
  - `locate '*.mp3'`
  - `locate '*.mp4'`
  - `locate '*.avi'`
  - `locate '*.ogg'`
- Confirm they are not required by README (for training or mission).
- Delete non-work-related music/videos (e.g., in `~/Music` or `Downloads`).

9.2 Fix insecure directory/file permissions (esp. FTP root)
- If there is an FTP server (vsftpd):
  - Check root directory ownership/permissions (often `/srv/ftp` or `/var/ftp`):
    - `ls -ld /path/to/ftp/root`
  - Ensure:
    - Owned by intended FTP user or root.
    - Mode about `755` for directories and `644` for files (unless README-specific).
- CP Mint21 / Round 2 includes “Insecure permissions on FTP root directory fixed”.
[oai_citation:46‡CP 18 r2 Img Ans & Vulns.pdf](sediment://file_00000000c3f871f599b22dd45098cacd)

==========
10. ANTIVIRUS & MALWARE SCAN (CLAMAV)
==========

10.1 Install/verify ClamAV
- Commands:
  - `sudo apt-get update`
  - `sudo apt-get install clamav`
  - `sudo freshclam` (update virus DB)

10.2 Run system scan
- Full recursive scan (takes long, run in separate terminal):
  - `sudo clamscan -i -r --remove=yes /`
- You can prioritize scanning user directories first:
  - `sudo clamscan -i -r --remove=yes /home`
- Log what gets removed in your notes.

==========
11. SYSTEM & APPLICATION UPDATES (OS + APPS)
==========

11.1 Automatic updates (APT periodic config)
- Edit periodic config:
  - `sudo gedit /etc/apt/apt.conf.d/10periodic`
- Set:
  - `APT::Periodic::Update-Package-Lists "1";`
  - Optionally configure unattended upgrade if appropriate.

11.2 OS updates
- CLI:
  - `sudo apt-get update`
  - `sudo apt-get dist-upgrade` (or `sudo apt-get upgrade` depending on scenario).
[oai_citation:47‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
- When prompted about config files:
  - Unless README says otherwise, keep the current version of config files (avoid accidentally undoing your hardening).
[oai_citation:48‡Basic-Linux-Security-Checklist.pdf](sediment://file_00000000778c722f93e1da8088b9f194)
- GUI (Mint/Ubuntu Update Manager):
  - Open Update Manager icon.
  - Make sure it:
    - Automatically refreshes update list.
    - Installs important security updates automatically.
  - CP scoring items: “The system refreshes the list of updates automatically”, “Install updates from important security updates”.

11.3 Application updates
- From CP answers:
  - Update apps like:
    - Chromium
    - Systemd
    - Vsftpd
  - Use GUI Software Manager or `sudo apt-get install --only-upgrade <package>`.

==========
12. BROWSER SECURITY
==========

12.1 Secure default browser (Firefox/Chromium)
- Open the browser and adjust:
  - Disable or reduce:
    - Password auto-save or at least require master password.
    - Pop-ups & insecure add-ons.

  - Increase:
    - Use “Always use HTTPS” / HTTPS-Only mode if available.
- Basic Ubuntu Security reading recommends making browser less exploitable and turning off risky defaults.
[oai_citation:49‡Basic-Linux-Security-Reading.pdf](sediment://file_00000000b2c871f581527f7effe3079)

12.2 Remove malicious or suspicious extensions
- Check browser extension/add-on manager:
  - Remove any non-work extensions (torrent helpers, unknown ad-ons, etc.), unless README says they’re needed.

==========
13. LOGGING, CHECKSUMS & GENERAL FORENSIC READINESS
==========

13.1 Check for log tampering (lightweight)
- Look quickly at:
  - `/var/log/auth.log` or `/var/log/secure` (distro-specific).
  - `/var/log/syslog`
- You’re mostly looking for:
  - Repeated failed SSH logins.
  - Strange cron or script executions.
- This helps explain suspicious users or backdoors you found earlier.

13.2 Hash verification (if asked again mid-competition)
- If asked to verify if a file changed:
  - `md5sum` or `sha256sum` on the file; compare with expected hash from question text.
[oai_citation:50‡Basic-Linux-Security-Reading.pdf](sediment://file_00000000b2c871f581527f7effe3079)

==========
14. EXTRA (CIS-STYLE) HARDENING – ONLY IF TIME & SAFE
==========

(These are inspired by CIS benchmarks; only do them if you’re sure they won’t break required functionality.)

14.1 File system mount options (high-level)
- Ensure `/tmp` and `/dev/shm` (often tmpfs) have:
  - `nodev`, `nosuid`, `noexec` where safe.
[oai_citation:51‡CIS-Linux-Mint22-Benchmark-1.0.0.pdf](sediment://file_0000000086f471f5ab00589cabe5d280)
- This helps contain exploits using temporary files.

14.2 Disable unused, exotic filesystems
- Add to `/etc/modprobe.d/blacklist.conf`:
  - `install cramfs /bin/true`
  - `install freevxfs /bin/true`
  - `install hfs /bin/true`
  - etc. (only if those FS types are not actually in use).
[oai_citation:52‡CIS-Linux-Mint22-Benchmark-1.0.0.pdf](sediment://file_0000000086f471f5ab00589cabe5d280)

14.3 Tighten world-writable directories
- Ensure no unnecessary world-writable files with sticky bit missing:
  - `find / -xdev -type d -perm -0002 ! -perm -1000 -ls`
  - Add sticky bit (`+t`) or restrict ACL if you know what you're doing.

==========
15. END-OF-ROUND VERIFICATION PASS
==========

15.1 Re-check critical categories (compare with CP vuln summaries)
- User auditing:
  - All unauthorized users removed.
  - Required users present.
  - Admin vs standard rights match README. [oai_citation:53‡CP 18 r1 Img Ans and Vulns.pdf](sediment://file_00000000f4e0722fb7fc097d31b24d68)
- Account policy:
  - Password aging configured.
  - Password complexity enforced.
  - Lockout threshold & duration set.

- Defensive countermeasures:
  - UFW enabled with sensible rules.
  - TCP SYN cookies turned on.
  - ClamAV installed and ran.

- Service auditing:
  - No prohibited services (telnet, unnecessary FTP, Apache when not required).
  - Critical services still present and running.

- Application security & updates:
  - Required apps present and up to date.
  - Prohibited apps removed.

- Prohibited files:

  - No MP3/video junk outside what README allows.

15.2 Quick sanity checks before you stop
- Reboot (if time):
  - Confirm everything still works after reboot:
    - No boot failures.
    - Critical services still start.
- Scoreboard:
  - If score dropped after a change, investigate:
    - Did you remove a required package?
    - Did you disable a critical service?
- Final sweep:
  - Re-run `ss -tulnp` and visually confirm only expected listeners.
  - Glance at `/home` directories for leftover weird files or scripts.

==========
16. MENTAL MODEL / STRATEGY (TO INTERNALIZE)
==========

- Think in **categories**, not random commands:
  - Forensics → Users → Passwords → Local policy → Services → Firewall → Updates → Browser → Prohibited stuff.
- Every action should map to at least one CyberPatriot scoring category (user auditing, account policy, defensive countermeasures, etc.). [oai_citation:54‡CP 18 r1 Img Ans and Vulns.pdf](sediment://file_00000000f4e0722fb7fc097d31b24d68)
- When in doubt:
  - Check the README.
  - Check man pages (`man <command>`).
  - Avoid deleting anything listed as “required”.

If you run this checklist in that order and keep your notes clean, you’ll be covering essentially all Linux vulnerabilities CyberPatriot tends to score, plus some extra hardening that’ll help you reason about any weird twist they throw at you.