UBUNTU 24.04 LTS CYBERPATRIOT SCRIPT CHECKLIST
(Focus: CP image hardening + automation; adapt to README requirements)

==
0. SCRIPT SAFETY / PRE-CHECKS
==

0.1 Detect environment
  - Script logic:
    - Confirm Ubuntu 24.04:
      - Parse /etc/os-release for VERSION_ID="24.04".
    - Record hostname, date/time, and current user.
    - Print a big warning header: "READ README BEFORE RUNNING".
    - Optionally require a confirmation flag (e.g., --i-know-what-im-doing).

0.2 Logging for your script
  - Create /root/cp-hardening-logs/ if not exists.
  - Log:
    - Start time, end time.
    - Every change made (file edited, user removed, service stopped, etc.).
    - stdout/stderr for risky commands.
  - Use tee or redirect all output to a timestamped log:
    - /root/cp-hardening-logs/hardening-$(date +%F-%H%M%S).log.

0.3 Read README + forensics before changes
  - Script should:
    - Prompt user to confirm they've answered forensics questions.
    - Optionally copy README and Forensics files to /root/cp-backup/ for reference.
  - DO NOT auto-delete anything mentioned in README as needed.

0.4 Backup critical config files before edits
  - For each file you touch, create .bak copies once:
    - /etc/passwd, /etc/shadow, /etc/group, /etc/gshadow
    - /etc/sudoers, /etc/sudoers.d/*
    - /etc/login.defs
    - /etc/pam.d/*
    - /etc/ssh/sshd_config
    - /etc/hosts.allow, /etc/hosts.deny
    - /etc/sysctl.conf and /etc/sysctl.d/*
    - /etc/fstab
    - /etc/systemd/logind.conf
    - /etc/rsyslog.conf and /etc/rsyslog.d/*
    - /etc/cron.*, /var/spool/cron/crontabs/*
    - Any service-specific configs (apache2, vsftpd, proftpd, vsftpd.conf, etc.)

```
  - Convention:
    - cp yourfile yourfile.cpbackup-<DATE> if not already backed up.

==============================================================================
1. USER, GROUP, AND ACCOUNT HARDENING
==============================================================================

 1.1 Enumerate all local users and groups
  - Script actions:
    - List users: getent passwd
    - List groups: getent group
    - Save to /root/cp-hardening-logs/users-groups-<DATE>.txt.
  - Mark suspicious users:
    - Any not in README users list.
    - Any with login shells who shouldn't log in (/bin/bash, /bin/sh, /bin/zsh, etc.).
    - Any with UID < 1000 that aren't standard system accounts.
    - Any duplicate UID or GID.

 1.2 Remove unauthorized users
  - Read authorized users from:
    - README or a file you create manually (e.g., /root/authorized_users.txt).
  - Script logic:
    - For each non-system user (UID ≥ 1000) not in authorized list:
      - Lock account (passwd -l USER) OR remove (deluser --remove-home USER),
        depending on how aggressive you want to be.
    - Always log removals.
  - Do NOT delete obvious system accounts (e.g., root, daemon, sys, sync, nobody).

 1.3 Ensure only root has UID 0
  - Script checks:
    - awk -F: '$3 == 0 {print $1}' /etc/passwd
  - If any UID 0 accounts besides root:
    - Change UID to safe value or lock them.
    - Document in /root/cp-hardening-logs.

 1.4 Fix shells for system/service accounts
  - Script checks:
    - For system accounts (UID < 1000) that don't need login:
      - Ensure their shells are /usr/sbin/nologin or /bin/false.
    - For human users:
      - Ensure they have a real interactive shell (e.g., /bin/bash) only if allowed by README.
  - Use usermod -s.

 1.5 Group membership and sudo rights
```

- Script logic:
  - Determine admin group: "sudo" is typical on Ubuntu.
  - List members of sudo group: getent group sudo.
  - Cross-check with README:
    - Remove unauthorized admins: gpasswd -d USER sudo.
    - Add authorized admins: usermod -aG sudo USER.
  - Ensure regular users are not in privileged groups:
    - Groups to inspect: sudo, adm, lpadmin, docker, lxd, sambashare, etc.
  - Log all membership changes.

1.6 Ensure all human accounts have passwords

- For each human user (UID ≥ 1000 or specified by README):
  - Check for locked or empty passwords:
    - sudo passwd -S USER (or inspect /etc/shadow).
  - If password is empty or locked but account should be usable:
    - Set a strong password (e.g., standard team password like CyberPatriot1! if allowed).
- Script can force:
  - passwd --delete is dangerous; instead:
    - echo "USER:CyberPatriot1!" | chpasswd
    - chage -d 0 USER (force password change at next login) – only if not disruptive to scoring.

===

2. PASSWORD AND AUTHENTICATION POLICY (login.defs + PAM)

===

2.1 Configure global password aging in /etc/login.defs

- Script edits (using sed or line replacement):
  - PASS_MAX_DAYS 90
  - PASS_MIN_DAYS 7
  - PASS_WARN_AGE 14
- Ensure no conflicting duplicate lines.
- For existing users:
  - chage -M 90 -m 7 -W 14 USER for each human account.

2.2 Ensure account lockout for brute-force protection (PAM)

- Identify PAM stack for auth:
  - For Ubuntu 24.04, typically use /etc/pam.d/common-auth and pam_faillock or pam_tally2 depending on configuration.
- Script recommendations (adjust if image already uses pam_faillock):
  - Install pam_faillock if missing:
    - apt-get install -y libpam-modules libpam-pwquality (if allowed by README).
  - Configure pam_faillock in /etc/pam.d/common-auth and /etc/pam.d/common-account:
    - Deny logins after N bad attempts (e.g., 5).
    - Lockout time (e.g., 900 seconds).

   - Important: Keep defaults moderate to avoid locking yourself out.

 2.3 Password quality controls (PAM pwquality)
   - Ensure libpam-pwquality is installed.
   - In /etc/pam.d/common-password:
     - Reference pam_pwquality.so or pam_cracklib equivalent.
     - Set minimum length (e.g., 12), complexity parameters, and history (remember=N).
   - Avoid insane settings that block normal password changes.

 2.4 Force password policies on all human users
   - For each non-system user:
     - chage -l USER to verify.
     - If out of compliance:
       - chage -M 90 -m 7 -W 14 USER.

==
3. FILESYSTEM, PARTITIONS, AND MOUNT OPTIONS (CIS-INSPIRED)
==

Note: Many of these changes can break software if applied blindly. For a CP script:
- Implement checks.
- Print what would be changed.
- Optionally perform changes only if a --cis-hardening flag is provided.

 3.1 Confirm separate or tmpfs partitions for key directories when present
   - Use findmnt or lsblk:
     - /tmp
     - /dev/shm
     - /home
     - /var
     - /var/tmp
     - /var/log
     - /var/log/audit (if exists)
   - If they're separate partitions or tmpfs:
     - Add secure mount options (nodev, nosuid, noexec where appropriate) in /etc/fstab.
   - Do not invent extra partitions if filesystem layout is simple; just document that step as manual.

 3.2 Harden /tmp mount options if /tmp is separate or tmpfs
   - In /etc/fstab for /tmp:
     - Ensure at least:
       - nodev (no device files),
       - nosuid (no setuid),
       - noexec (no binaries executed).

- Example options: defaults,rw,nosuid,nodev,noexec,relatime
- After editing:
  - mount -o remount /tmp
- If /tmp not separate:
  - Script can print a warning and optionally create tmp.mount using systemd tmp.mount with tmpfs,
    but do this only if safe and tested.

3.3 Harden /var/tmp mount options if separate
- Similar to /tmp:
  - nodev, nosuid, noexec in fstab.
  - Remount with new options.

3.4 Harden /dev/shm if separate
- For /dev/shm:
  - nodev, nosuid, noexec.
- Remount after changes.

3.5 Optionally secure /home and /var partitions
- If /home has a separate partition:
  - At least nodev.
- If /var has a separate partition:
  - nodev, possibly nosuid (but be careful with some services).
- Again, script should default to audit/print and only auto-change with a serious hardening flag.

3.6 Disable unused or legacy filesystem kernel modules (careful)
- For a CP script, usually:
  - Optionally add blacklist lines to /etc/modprobe.d/:
    - e.g., blacklist cramfs, freevxfs, hfs, hfsplus, jffs2, udf, etc., only if not required by README.
- Always check:
  - lsmod or modinfo before blacklisting.
  - Comment all changes in config files.

==
4. SYSTEM UPDATES AND PACKAGES
==

4.1 Update package lists and installed packages
- Script actions:
  - apt-get update
  - apt-get dist-upgrade -y (or apt-get upgrade -y if you want lower risk).
- Respect README if it says not to upgrade certain packages.

- Log installed/updated packages:
  - Save dpkg -l output to /root/cp-hardening-logs/dpkg-before.txt before updates and after.

4.2 Configure automatic updates (if not already set)

- Check:
  - /etc/apt/apt.conf.d/20auto-upgrades
- Ensure:
  - APT::Periodic::Update-Package-Lists "1";
  - APT::Periodic::Unattended-Upgrade "1";
- For CP, enabling automatic security updates usually earns points.

4.3 Remove dangerous or unnecessary packages

- Script should look for and purge:
  - telnet, rlogin, rsh-client, tftp, ftp, vsftpd, proftpd, pure-ftpd (unless README requires).
  - netcat, nmap, zenmap, aircrack-ng, john/johnny, hydra, ettercap, wireshark (if prohibited).
  - bittorrent clients, P2P: transmission, amule, etc.
  - remote admin: vnc4server, tightvncserver, xrdp (if not required).
  - compilers and dev tools (optional CIS-like): gcc, g++, clang, make, etc., if not needed by scenario.
- Use: apt-get purge -y PACKAGE && apt-get autoremove -y.

4.4 Verify no prohibited media or files

- Script can:
  - Scan user home directories for patterns such as:
    - *.mp3, *.mp4, *.avi, *.torrent, etc.
  - Log all findings.
  - Optionally delete only if README explicitly says to remove.

==

5. SERVICES, DAEMONS, AND FIREWALL

==

5.1 Enumerate all services

- Use systemctl:
  - systemctl list-unit-files --type=service
  - systemctl --type=service --state=running
- Save to /root/cp-hardening-logs/services-<DATE>.txt.

5.2 Disable unneeded or dangerous network services

- For each service:
  - If not in README as required:
    - Stop and disable:
      - systemctl stop SERVICE
      - systemctl disable SERVICE

  - Pay attention to:
    - apache2, nginx, lighttpd
    - vsftpd, proftpd, pure-ftpd
    - openssh-server (sometimes must remain enabled but locked down)
    - mysql, mariadb, postgresql
    - rpcbind, nfs-server, cups, avahi-daemon, samba (smbd, nmbd)
    - telnet, tftp, rsh, rlogin-related services.

5.3 Check listening ports and processes
  - Script uses:
    - ss -tulnp or netstat -tulnp.
  - For each listening socket:
    - Map port → process.
    - If service is not required:
      - Stop/disable the service.
  - Log all decisions: which port, which process, action taken.

5.4 Configure UFW (Uncomplicated Firewall)
  - Enable UFW:
    - ufw enable
  - Default policy:
    - ufw default deny incoming
    - ufw default allow outgoing
  - Allow only necessary services (e.g., SSH if remote login is required):
    - ufw allow OpenSSH or ufw allow 22/tcp if required.
  - Optional: enable logging:
    - ufw logging on
  - Note: On CP images, enabling UFW is usually a direct scoring item.

5.5 Enable TCP SYN cookie protection and basic network sysctl hardening
  - In /etc/sysctl.conf or /etc/sysctl.d/99-sysctl-hardening.conf:
    - net.ipv4.tcp_syncookies = 1
    - net.ipv4.conf.all.rp_filter = 1
    - net.ipv4.conf.default.rp_filter = 1
    - net.ipv4.conf.all.accept_redirects = 0
    - net.ipv4.conf.default.accept_redirects = 0
    - net.ipv4.conf.all.accept_source_route = 0
    - net.ipv4.conf.default.accept_source_route = 0
    - net.ipv4.icmp_echo_ignore_broadcasts = 1
    - net.ipv4.icmp_ignore_bogus_error_responses = 1
  - Apply:
    - sysctl -p or sysctl --system.

==

6. SSH HARDENING (IF OPENSSH-SERVER INSTALLED)
==

6.1 Confirm SSH usage
  - If README says SSH must be enabled:
    - Keep service active but hardened.
  - If not required:
    - systemctl stop ssh
    - systemctl disable ssh
    - Optionally apt-get remove --purge openssh-server.

6.2 Basic sshd_config controls
  - File: /etc/ssh/sshd_config (and possibly /etc/ssh/sshd_config.d/*).
  - Ensure:
    - Protocol 2 (if present).
    - PermitRootLogin no or prohibit-password (depending on scenario).
    - PasswordAuthentication yes/no according to README (often yes in CP).
    - PermitEmptyPasswords no
    - X11Forwarding no
    - AllowTcpForwarding no (unless needed)
    - AllowAgentForwarding no
    - UsePAM yes
    - ClientAliveInterval 300
    - ClientAliveCountMax 0 or small
    - MaxAuthTries 4 (or similar)
    - LoginGraceTime 30
    - MaxSessions small number (e.g., 10 or less)
    - MaxStartups limited (e.g., 10:30:60)
  - Optionally:
    - Banner /etc/issue.net with proper legal warning text.

6.3 Restrict which users can SSH
  - Use AllowUsers or AllowGroups in sshd_config to restrict remote login to:
    - Known admins or remote users from README.
  - Reload SSH after changes:
    - systemctl reload ssh.

==
7. LOGGING, AUDIT, AND TIME
==

7.1 Ensure a single logging system in use (rsyslog or systemd-journald)
  - On Ubuntu, journald is always present; rsyslog is usual.
  - Script should:

```
    - Check if rsyslog is installed:
      - systemctl status rsyslog
    - If present, ensure:
      - systemctl enable rsyslog
      - systemctl start rsyslog
    - If other loggers exist, ensure they’re not fighting each other.

 7.2 Secure log file permissions
   - Ensure /var/log and subdirectories:
     - Owned by root (or appropriate service account).
     - Not world-writable.
   - Specific checks:
     - chmod 640 /var/log/*.log
     - chmod 600 sensitive logs (auth.log, secure equivalents, etc.).
     - chown root:adm or root:syslog as appropriate.

 7.3 Log rotation
   - Confirm logrotate is installed and configured:
     - /etc/logrotate.conf
     - /etc/logrotate.d/*
   - Ensure:
     - Reasonable rotation frequency.
     - Use of "compress", "missingok", "notifempty", etc.

 7.4 Auditd (optional but CIS-like)
   - If auditd available on the image and not prohibited:
     - apt-get install auditd audispd-plugins
     - enable and start:
       - systemctl enable auditd
       - systemctl start auditd
     - Ensure basic rules exist or at least service is running.
   - CP images don’t always require this; treat as optional.

 7.5 Time synchronization
   - Ensure timesyncd or chrony is active (depending on default for Ubuntu 24):
     - systemctl status systemd-timesyncd or chronyd.
   - Enable and start whichever is used by default.
   - Consistent time helps logs and sometimes scoring.

================================================================
8. CRON, AT, AND SCHEDULED TASKS
================================================================

 8.1 Verify cron service is enabled
```

```
  - Ensure cron is active:
    - systemctl enable cron
    - systemctl start cron

 8.2 Audit cron/at jobs
  - Inspect:
    - /etc/crontab
    - /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly, /etc/cron.monthly
    - /etc/cron.d/*
    - per-user cron: crontab -l -u USER
    - at jobs: atq
  - Remove any malicious or suspicious jobs:
    - Unknown scripts in /tmp, /home/*/.local, etc.
    - Network backdoors, Bitcoin miners, weird Python/Perl one-liners.

 8.3 Lock down cron and at access
  - Use /etc/cron.allow, /etc/cron.deny:
    - Typically:
      - Create /etc/cron.allow with only specific users if needed (e.g., root).
      - Empty or remove /etc/cron.deny.
  - Similarly for /etc/at.allow and /etc/at.deny.
  - For CP, just ensure no obviously insecure wildcard access.

======================================================================
9. FILE PERMISSIONS, SUID/SGID, WORLD-WRITABLE
======================================================================

 9.1 Critical system files permissions
  - Script should enforce:
    - /etc/passwd: root:root, mode 644
    - /etc/shadow: root:shadow or root:root, mode 640 or 600
    - /etc/group: root:root, mode 644
    - /etc/gshadow: root:shadow or root:root, mode 640 or 600
    - /etc/sudoers: root:root, mode 440
    - /etc/ssh/sshd_config: root:root, mode 600 or 640
    - /boot/grub/grub.cfg (or equivalent): root:root, 600 or 640
  - Fix with chown/chmod as needed.

 9.2 Find and review SUID/SGID binaries
  - Commands:
    - find / -xdev -type f -perm -4000 -o -perm -2000 2>/dev/null
  - For each SUID/SGID binary:
    - Decide if it's required (ping, sudo, su, passwd, etc. usually needed).
    - For obviously unnecessary or risky ones:
      - chmod u-s or g-s to remove SUID/SGID.
    - Log all modifications.

9.3 Find world-writable files and directories
  - Directories:
    - find / -xdev -type d -perm -0002 2>/dev/null
    - Exclude known safe locations (/tmp, /var/tmp, /dev/shm) but check they have the sticky bit:
      - chmod +t /tmp /var/tmp /dev/shm
  - Files:
    - find / -xdev -type f -perm -0002 2>/dev/null
    - Remove world-write where not needed:
      - chmod o-w FILE
  - Log all changes, especially in /home.

9.4 Local interactive user dotfiles
  - For each human user:
    - Inspect ~/.bashrc, ~/.profile, ~/.bash_profile, ~/.ssh/authorized_keys.
    - Ensure dot files are not world-writable.
    - Remove suspicious aliases or PATH modifications that pull from /tmp or untrusted dirs.
  - Script can:
    - chmod go-rwx ~USER/.ssh/* where appropriate.
    - Ensure ~/.ssh directory is 700, authorized_keys 600.

==========================================================================
10. BOOTLOADER AND PHYSICAL SECURITY (OPTIONAL)
==========================================================================

10.1 Secure GRUB configuration (only if safe in CP)
  - Check /boot/grub/grub.cfg (or /boot/grub/grub.cfg.d).
  - Optionally:
    - Set a GRUB password:
      - Use grub-mkpasswd-pbkdf2 and configure in /etc/grub.d/40_custom.
    - Regenerate GRUB config with update-grub.
  - This is advanced; often optional for CyberPatriot. Consider leaving as manual step.

==========================================================================
11. DESKTOP/LOGIN ENVIRONMENT (IF IMAGE HAS GUI)
==========================================================================

11.1 Disable guest login (if applicable)
  - On modern Ubuntu with GDM, guest session is usually off by default.
  - If guest is present:
    - Configure GDM to disable automatic guest login or guest sessions:
      - gsettings or GDM config files (varies by version).
```

```
   - Ensure auto-login is disabled except possibly for competition user as directed by README.

 11.2 Configure login warning banners
   - Files:
     - /etc/issue (local console)
     - /etc/issue.net (remote SSH banner)
   - Insert approved legal warning text (short and generic, not competition-breaking).

 11.3 Screen lock and idle timeouts
   - Set system-wide idle lockout:
     - For GNOME:
       - gsettings set org.gnome.desktop.session idle-delay <seconds>
       - gsettings set org.gnome.desktop.screensaver lock-enabled true
     - Or use dconf/gsettings via script for each user if necessary.
   - For CP, this is usually low priority but safe.

 11.4 Disable automatic mounting of removable media (optional)
   - For GDM/GNOME:
     - Tweak settings to prevent auto-mounting of USB drives if consistent with scenario.
   - Usually more CIS-hardening than CP-scored; treat as optional.

================================================================================
12. ANTIVIRUS AND MALWARE SCANNING
================================================================================

 12.1 Install and configure ClamAV
   - Commands:
     - apt-get update
     - apt-get install -y clamav
     - freshclam (update virus DB)
     - clamscan -r -i /   (or target home directories, e.g., /home)
   - This may take a while; script can:
     - Run clamscan in background and log results to /root/clamav-scan.log.
   - Remove infected files ONLY if clearly malicious; otherwise log for manual review.

================================================================================
13. CHECKS FOR DUPLICATES AND ACCOUNT CONSISTENCY
================================================================================

 13.1 No duplicate UIDs, GIDs, usernames, or group names
   - Script steps:
     - Use awk and sort/uniq to detect duplicates in /etc/passwd and /etc/group.
     - For each duplicate:
       - Log issue and optionally prompt for manual fix.
```

  - This aligns with the idea that every account and group should be unique.

13.2 Root path integrity
  - Inspect root’s PATH environment:
    - Ensure no '.' (current directory) in PATH.
    - Ensure no world-writable directories in PATH.
    - Prefer /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin.
  - If modifications needed:
    - Adjust in /root/.profile, /etc/profile, or /etc/environment.

==

14. FINAL AUDIT / REPORT

==

14.1 Summary report for the team and scoring
  - At the end of script execution, print and save:
    - List of users and groups after changes.
    - Services still running and their status.
    - UFW status and rules.
    - Any suspicious files or directories left for manual review.
    - Any operations skipped because of README or high risk.

14.2 Dry-run mode (recommended)
  - Support a --dry-run flag:
    - Instead of making changes, only show what WOULD be changed.
    - Helpful for testing on CP practice images.

14.3 Exit codes
  - Exit non-zero if:
    - Critical steps failed (e.g., editing /etc/passwd or /etc/shadow had errors).
  - Otherwise, exit 0 and clearly state "HARDENING COMPLETED (check logs)."

==

NOTES FOR SCRIPT CREATION

- Always check the README FIRST; some services/files MUST remain as-is to earn points.
- Don’t blindly apply all CIS-style options in an auto script:
  - Use comments and flags (e.g., --cis-level1) for more aggressive steps like extra mount options,
    kernel module blacklisting, or GRUB hardening.
- Keep every destructive action reversible:
  - Use backups, logs, and minimal changes to maintain stability.
- Test your script thoroughly on training images (Ubuntu 22, Mint 21, etc.) before live CP rounds.