MACCDC 2026 Quick Lockdown Playbook

A fast, low-drama hardening plan for the 11-VM pod (Linux, Windows, Palo Alto, Cisco FTD, VyOS).

Purpose: Rapidly reduce compromise risk while keeping scored services stable. This is written as a "do it in order" checklist you can follow under pressure.

Competition guardrails (do these wrong and you can self-own scoring):

- **Do not** move services between public IPs, and **do not** change internal addressing/VLANs unless an inject tells you to.
- **Do not** change system names or IP addresses unless an inject tells you to.
- Maintain ICMP (ping) on all competition devices except the Core port of the Palo Alto VM.
- Any firewall/IDS/IPS action that interferes with the scoring engine is your responsibility (and your score).
- VM scrubs/reverts are limited and penalized - assume every change must be reversible by you (config backups).

Wall poster: the obvious order

Step	Do this
1	Open Team Portal/NISE on a dedicated screen. Treat it as ground truth for what is scored and what is broken.
2	Rotate privileged credentials everywhere (local admin/root, device admin). Avoid changing service accounts until verified.
3	Lock down management access: only your workstations/jump hosts can reach SSH/RDP/GUI management.
4	Edge firewalls first: allow-list inbound to ONLY the scored ports on the assigned public IPs; deny the rest; keep ICMP per rules.
5	Turn on MAC safely: SELinux (permissive -> fix -> enforcing) on Fedora/Oracle; AppArmor enforce on Ubuntu.
6	Host hardening: patch, disable unused services, restrict remote login, tighten local firewalls (after edge rules are stable).
7	Central visibility: forward key logs to Splunk; enable audit logging; confirm time sync.
8	After every change: re-check NISE + run a local service test; roll back fast if scoring turns red.

Phase 1: Stabilize and control (first 15-30 minutes)

Goal: stop the obvious compromises and regain control without changing service behavior.

Checklist

- Confirm what is scored and current status in the Team Portal/NISE. Keep it open and refresh often.
- Change privileged passwords (local admin/root, device admin, platform admins). Admin/root accounts are typically not used for scoring; still, avoid touching unknown service accounts until confirmed.
- Restrict management paths: allow admin access only from your designated management workstations/jump hosts (SSH, RDP, web admin GUIs, Splunk admin).
- Back up configs BEFORE major changes (firewall exports, key config files, Windows policy exports). You do not have snapshots.

Fast edge firewall pattern (use on Palo Alto + Cisco FTD)

Start with inbound allow-listing on the external/public side. Permit only the ports that NISE shows as scored for each public IP, and deny everything else inbound. Keep ICMP allowed on competition devices except the Palo Alto core port. Avoid aggressive outbound blocking early - you may need patching and research.

Do NOT do these in the first hour

- Do not change public IP mappings for services, and do not change internal addressing/hostnames unless an inject says so.
- Do not deploy blanket IPS policies that might block the scoring engine checks (treat scoring traffic as production traffic).
- Do not 'fix' web content unless you are sure scoring is not comparing content (scoring may compare pages byte-for-byte).

Phase 2: MAC (SELinux / AppArmor) with minimal scoring risk

Goal: turn on Mandatory Access Control safely: permissive first, observe, then enforce. Do not guess.

Fedora 42 and Oracle Linux 9: SELinux safe enable ladder

Use targeted policy. If SELinux is disabled, enable in permissive mode, relabel, then fix contexts/booleans, then enforce.

```
# 1) Check current state
sestatus
getenforce

# 2) If disabled: enable targeted policy in permissive mode and relabel
sudo sed -i 's/^SELINUX=.*/SELINUX=permissive/' /etc/selinux/config
sudo sed -i 's/^SELINUXTYPE=.*/SELINUXTYPE=targeted/' /etc/selinux/config
sudo touch /.autorelabel
sudo reboot

# 3) After reboot (still permissive): inspect what would be blocked
sudo ausearch -m avc -ts recent
sudo journalctl -t setroubleshoot --since "10 min ago" 2>/dev/null

# 4) Fix the common safe issues
sudo restorecon -Rv /etc /var /opt

# 5) Flip to enforcing only after services test clean
sudo setenforce 1
sudo sed -i 's/^SELINUX=.*/SELINUX=enforcing/' /etc/selinux/config
```

SELinux: avoid the classic trap

Under time pressure, teams often generate broad custom allow rules that quietly gut SELinux. Prefer: correct labels (restorecon) and narrow booleans. Only write custom policy if you understand the AVCs and the daemon involved.

Ubuntu 24.04: AppArmor (quick wins)

```
# 1) Ensure AppArmor is running
sudo systemctl enable --now apparmor
sudo aa-status

# 2) Enforce only the profiles you actually need (examples)
sudo aa-enforce /etc/apparmor.d/usr.sbin.sshd 2>/dev/null
# For web/mail, enforce the matching profile files present on your host:
ls /etc/apparmor.d/ | egrep 'apache2|nginx|postfix|dovecot|named|vsftpd' || true
```

Windows (Server 2019/2022, Windows 11): fast, safe hardening

- Turn on and update Microsoft Defender. Enable cloud protection if permitted. Avoid third-party AV changes unless you know scoring impact.
- Enable Windows Firewall (Domain/Private). Start with inbound allow rules for scored services and explicit admin access from your jump host only.
- Reduce remote admin exposure: limit RDP to admins and jump host; disable legacy protocols you do not need.
- Audit before blocking: if you enable aggressive ASR/WDAC-style controls, start in Audit mode first to avoid breaking scored functionality.

Phase 3: Shrink attack surface and improve detection

Goal: make exploitation harder and response faster, while keeping scoring green.

Patch and baseline safely

- Patch OS and critical packages in controlled batches (one host at a time). Re-check NISE after each patch cycle.
- Disable or remove clearly unused services (but do not remove roles you suspect are scored).
- Harden SSH: disable password auth if you can (keys), or at least disable root login and restrict by IP; keep an out-of-band console plan.
- Harden Windows auth: reduce local admins, disable unused local accounts, tighten password policy, and monitor new accounts/groups.

Central logs (Splunk) - minimum viable signals

- Linux: auth logs (/var/log/auth.log or /var/log/secure), sudo logs, audit logs if enabled.
- Windows: Security log (logons, new users, group changes), System log, PowerShell operational logs if available.
- Firewalls: config changes, admin logins, deny logs for inbound public zones, and threat logs (if enabled).

Verification loop (do this after every major change)

- Refresh NISE service status: did anything turn red?
- Run a local functional check from an internal host: curl your web pages, nslookup your DNS, test mail/POP3/FTP as applicable.
- If a service breaks: roll back the last change first, then investigate (do not stack changes while blind).

Per-system quick actions (top 3 things each)

Use this as a delegation sheet: assign one person per box during the first hour.

Ubuntu Ecom (Ubuntu 24.04 server)

- Confirm web stack is stable; do not change site content unless you verify scoring expectations.
- Enforce AppArmor for sshd and your web daemon; restrict admin SSH to jump host IPs.
- Patch in small batches; keep a quick rollback plan (package history, config backups).

Fedora Webmail (Fedora 42)

- Enable SELinux targeted (permissive -> fix -> enforcing) and resolve AVCs safely (restorecon/booleans).
- Lock down mail surface: restrict admin access, keep only required mail ports exposed via edge firewall.
- Forward logs to Splunk; watch for new accounts and suspicious auth.

Splunk (Oracle Linux 9)

- Enable SELinux targeted with the safe ladder; label/restorecon for Splunk directories if needed.
- Change Splunk admin credentials; restrict web/UI access to internal management only.
- Set up basic dashboards/alerts: auth spikes, new users, firewall denies, suspicious PowerShell.

Ubuntu Workstation (Ubuntu 24.04 desktop)

- Treat as a jump host: patch it, enable full disk encryption if already present, and keep browsers/extensions minimal.
- Restrict outbound admin tools to what you need (SSH, browser to firewall UI); do not install random tools mid-event.
- Use it as the primary admin path for Palo Alto management; keep session logs/notes here.

Windows Server 2019 AD/DNS

- Confirm DNS and AD health first (do not break AD - other scoring may depend on it).
- Restrict RDP/WinRM to jump host; enable Windows Firewall inbound allow-listing.
- Enable auditing for logons, account management, and group policy changes; forward to Splunk.

Windows Server 2019 Web

- Confirm IIS sites and bindings; avoid content changes without validation.
- Firewall: allow only required ports; restrict admin access to jump host.
- Patch carefully; monitor IIS logs and new scheduled tasks/services.

Windows Server 2022 FTP

- Confirm FTP mode (passive range). Make sure edge firewall rules match the configured passive port range.
- Firewall: allow FTP control + passive range from outside only as needed; restrict admin access to jump host.
- Turn on FTP logging; watch for brute force and new file drops.

Windows 11 Workstation

- Treat as a jump host: patch, verify Defender is healthy, and keep it clean (no extra software).
- Use it as the primary admin path for Cisco FTD management; restrict browser access to internal management networks.
- Lock down local admin group membership; disable unused local accounts and enable firewall.

Palo Alto VM

- Restrict management UI/SSH to internal management only (jump host).
- Build inbound allow-list policy for scored services on the public IPs; keep ICMP per rules.
- Export config periodically so you can roll back quickly.

Cisco FTD VM

- Restrict management access to the Windows 11 jump workstation (as designed).
- Inbound allow-list to scored services; avoid IPS policies that break scoring checks.
- Export config periodically and track every rule change.

VyOS Router

- Restrict admin access to management networks only; disable unused services.
- Confirm routing/NAT is stable; do not change addressing unless injected.
- Back up config and record changes.

Appendix: scored services - how to avoid overblocking

Do not guess what ports are scored - read them from NISE. As a reminder, typical functional checks include web (HTTP/HTTPS), mail (SMTP/POP3), DNS, and sometimes FTP. Where scoring compares content (web), changing pages can cost points even if the service is up.

Service type	Typical ports	Common gotcha
Web	80/tcp, 443/tcp	Scoring may compare returned content; do not 'improve' pages casually.
SMTP	25/tcp (sometimes 587/tcp)	Ensure outbound delivery paths and local mailbox flow stay intact.
POP3	110/tcp (sometimes 995/tcp)	Auth may use AD usernames; do not break directory integration.
DNS	53/udp and 53/tcp	TCP/53 matters for large responses/zone transfers; don't block it blindly.
FTP	21/tcp + passive range	Passive port range must match server config and firewall rules.

Sources

2026 Mid-Atlantic Collegiate Cyber Defense Competition Qualifier - Team Packet (rules, scoring behaviors, and topology).