STUDENT HANDOUT · WEEK 11

Hardening, Assets, and Vulnerability Management

SY0-701 objectives 4.1, 4.2, 4.3. Notes, exercises, review questions, and the lab.

Name	
Date	

THIS WEEK YOU WILL

1. Describe how a secure baseline is established, deployed, and maintained.
2. Choose hardening steps for workstations, servers, network gear, cloud, mobile, ICS, embedded, and IoT.
3. Recommend wireless and mobile settings, including WPA3, RADIUS, MDM, and a deployment model.
4. Explain application security techniques such as input validation, secure cookies, and code signing.
5. Trace an asset from procurement to certified disposal and name the security task at each stage.
6. Compare vulnerability identification methods, from scanning to bug bounties and threat feeds.
7. Prioritize vulnerabilities with CVSS and context, then pick a response and validate the fix.

KEY IDEAS

A baseline is the known good starting point

1. Establish	Start from a CIS Benchmark or vendor guide, then adapt it
2. Test	Pilot on a few machines so the baseline breaks nothing
3. Deploy	Push it with GPO, MDM, images, or configuration scripts
4. Maintain	Rescan for drift, update with new versions and threats

Every device class hardens a little differently

Target	Key hardening moves
Workstations and servers	Remove unused software, patch, EDR, host firewall
Switches and routers	SSH not Telnet, change defaults, update firmware
Mobile devices	MDM, screen lock, encryption, app store controls
Cloud infrastructure	Least privilege IAM, no public buckets, logging on
ICS and SCADA	Segment from IT, restrict remote access, monitor
Embedded, RTOS, IoT	Change defaults, update firmware, isolate network

Plan the radio, then lock down the connection

Site survey	Walk the site to measure signal, noise, and neighbors (Before and after install)
Heat map	Color map of signal strength over the floor plan (Finds dead zones and leaks)
WPA3	SAE replaces the WPA2 pre-shared key exchange (WPA3-Personal and Enterprise)
AAA and RADIUS	802.1X sends each user's login to a RADIUS server (Per-user accounts and logs)
EAP methods	EAP-TLS, PEAP, and EAP-TTLS carry the credentials (EAP-TLS: certificates both sides)

Who owns the phone decides how much you control

Model	Who owns it	Security tradeoff
BYOD	The employee	Cheap, least control; use containerization
COPE	The company, personal use allowed	Full control, with privacy rules for staff
CYOD	The company, user picks from a list	Control with a small, supported model list
MDM (all models)	Managed by IT	Enforces lock, encryption, wipe, app rules
Connection methods	Cellular, Wi-Fi, Bluetooth	Disable unused radios, VPN on open Wi-Fi

Build security into the application

Input validation	Check every input server side against an allow list

Secure cookies	Set Secure, HttpOnly, and SameSite on session cookies
Static code analysis	Scan source code for flaws before it runs
Code signing	Signature proves the author and that code is unchanged
Sandboxing	Run code in isolation, away from the host system
Monitoring	Log app errors and abuse, and alert on odd patterns

Track every asset from purchase to disposal

1. Acquire	Buy from vetted vendors; security reviews the purchase
2. Assign	Record an owner and a data classification
3. Monitor	Keep the inventory current with discovery scans
4. Dispose	Sanitize or destroy, get a certificate, honor retention

Many ways to find a weakness before attackers

Method	What it finds
Vulnerability scan	Known flaws and missing patches across hosts
Static and dynamic analysis	Code flaws at rest (SAST) and running (DAST)
Package monitoring	Vulnerable third-party libraries in your apps
Threat feeds	OSINT, vendor, ISAC, and dark web intelligence
Penetration testing	Exploitable paths a skilled tester can chain
Disclosure and bug bounty	Flaws reported by outside researchers

Confirm, prioritize, fix, and prove it is fixed

1. Confirm	Rule out false positives; watch for false negatives
2. Prioritize	CVSS plus exposure, environment, and impact
3. Respond	Patch, segment, compensate, insure, or except
4. Validate	Rescan, audit, and verify the fix holds
5. Report	Show owners the trend, open items, and exceptions

EXERCISE 1: EXERCISE 1: HARDEN LAKESIDE FAMILY CLINIC

8 MINUTES · PAIRS

For each problem, name the device class, the first hardening step, and the baseline setting that keeps it fixed.

#	Prompt	Your answer
1	Front desk PCs all log in with one shared local admin account	
2	Staff Wi-Fi uses WPA2-Personal; the password is on a sticky note	
3	Doctors read email on personal phones with no lock screen	
4	The core switch is managed over Telnet with the default password	
5	The vaccine fridge sensor is an IoT device on the main network	
6	The patient portal login form passes input straight to SQL	
7	The file server runs FTP and a print spooler it does not need	
8	Old laptops sit in a closet, waiting to be thrown away	

EXERCISE 2: EXERCISE 2: RANK THE CLINIC'S FINDINGS

7 MINUTES · SMALL GROUPS

Rank the six findings from first to last fix. For each, pick a response and say how you will validate it.

#	Prompt	Your answer
1	A: VPN appliance, CVSS 9.8, internet facing, listed in CISA KEV	
2	B: Lab test server, CVSS 9.1, no network path to patient data	
3	C: EHR web server, CVSS 6.5, public exploit, holds patient records	
4	D: Printer, CVSS 7.5, but the vendor backported the fix last month	
5	E: Intranet server still allows TLS 1.0, CVSS 5.3	
6	F: Old X-ray PC, CVSS 8.8, vendor will not support a patch	

REVIEW QUESTIONS

1. What are the three stages in the life of a secure baseline?

2. Why should a site survey come before installing wireless access points?

3. How do COPE and CYOD differ?

4. What is the difference between sanitization and destruction?

5. A scan flags a critical CVE, but the server has the vendor's backported fix. What is this, and what next?

KEY TAKEAWAYS

- **Baselines beat one-off fixes.** Write the standard once, deploy it everywhere, and scan for drift.
- **Fit hardening to the device.** A switch, a phone, and a PLC each need their own hardening steps.
- **Know every asset you own.** You cannot patch, track, or wipe a device that is not in the inventory.
- **Score, then add context.** CVSS starts the ranking; exposure and impact finish it. Then rescan.

LAB 11: HARDEN A LAB HOST AGAINST A CIS BENCHMARK AND RESCAN

60 MINUTES

Use the Ubuntu Linux VM from Lab 1 and the free CIS Benchmark PDF for your Ubuntu version from cisecurity.org. Scan only your own lab VMs, never a school or work network.

Steps

1. Take a snapshot of the Ubuntu VM named pre-hardening, so you can roll back any change that breaks it.
2. Install Lynis with sudo apt install lynis and run sudo lynis audit system. Record the hardening index and the top five warnings or suggestions.
3. From your second VM, run nmap -sV LAB_IP against the Ubuntu VM (or a Greenbone scan as in Lab 5). Save the open ports and findings.
4. Choose eight Level 1 recommendations from the CIS Benchmark that apply to your VM, such as SSH settings, the host firewall, unused services, password quality, and automatic updates. Note each one's section number.
5. Apply them. Examples: set PermitRootLogin no in /etc/ssh/sshd_config and run sudo systemctl restart ssh; run sudo ufw default deny incoming, sudo ufw allow 22/tcp, and sudo ufw enable; stop a service you do not need with sudo systemctl disable --now cups.
6. Install and turn on unattended-upgrades and auditd, then reboot and confirm you can still log in and use the VM.
7. Rerun Lynis and the same Nmap or Greenbone scan. Compare the hardening index, open ports, and findings before and after.
8. For any recommendation you chose not to apply, write a one-line exception: the reason, the risk owner, a compensating control, and a review date.

Hand in

A two-page report with before and after screenshots of Lynis and the scan, a table of your eight recommendations with CIS section numbers and status, and your exception list.

Questions to answer in your report

1. Which change improved the hardening index or scan results the most, and why?
2. Did any hardening step break something? How would a pilot group have caught it before deployment?
3. How would you deploy this baseline to every clinic machine of this type and detect drift next month?
4. Which finding remained after hardening, and which response fits it: patch, compensating control, or exception?