COMPTIA SECURITY+ SY0-701 · WEEK 5

Vulnerability Types Across the Enterprise



Objective 2.3 · 90-minute lesson

Presenter: Trapper Brown

What you will be able to do in 90 minutes

1. Explain memory injection, buffer overflow, race conditions, and malicious updates.
2. Distinguish SQL injection from cross-site scripting and name the main fix for each.
3. Explain why end-of-life, legacy, and firmware weaknesses persist and how to contain them.
4. Describe VM escape, resource reuse, and common cloud-specific vulnerabilities.
5. Identify supply chain, cryptographic, and misconfiguration weaknesses in a scenario.
6. Explain side loading, jailbreaking, and zero-day vulnerabilities and their controls.
7. Prioritize findings using CVE, CVSS, and real exposure.

AGENDA

4 min	From threats to weaknesses
11 min	Application vulnerabilities
10 min	Web-based vulnerabilities
8 min	Operating system and hardware
8 min	Virtualization and cloud-specific flaws
10 min	Exercise 1: name the vulnerability
7 min	Supply chain vulnerabilities
8 min	Cryptographic and misconfiguration flaws
7 min	Mobile devices and zero-days
5 min	Naming and ranking what you find
9 min	Exercise 2: Lakeside's weak spots
3 min	Wrap-up

Application flaws make code misbehave



Memory injection

Attacker code is written into a running process's memory.

DLL injection



Buffer overflow

Input larger than its buffer overwrites nearby memory.

Overwritten return address



Race condition

State changes between the time of check and time of use.

TOC/TOU



Malicious update

A trusted update channel delivers attacker code.

SolarWinds Orion, 2020

SQL injection and XSS trust user input

Flaw	What goes wrong	Main fix
SQL injection	User input runs as part of a database query	Parameterized queries
Stored XSS	Script saved on the site runs for every visitor	Output encoding, input validation
Reflected XSS	Script in a link is echoed back in the response	Output encoding, Content Security Policy
DOM-based XSS	The page's own JavaScript writes input unsafely	Safe DOM methods, no innerHTML with input

Old systems and firmware outlive their patches

Operating system

- Unpatched kernel and service flaws
- Privilege escalation bugs turn a user into an admin
- Unneeded services left listening
- Fixed by vendor patches on a regular cycle

Hardware

- Firmware flaws in BIOS, UEFI, and device controllers
- End-of-life: the vendor stops selling, then supporting
- Legacy: old gear still in use, often unpatchable
- Fixed by firmware updates, isolation, or replacement

Shared hardware creates shared weaknesses

Vulnerability	What happens	Defense
VM escape	Code in a guest breaks out to the hypervisor	Patch hypervisors, limit guest features
Resource reuse	Memory or disk from one VM is given to another	Clear memory and storage before reuse
Cloud misconfiguration	Storage or services exposed to the internet	Configuration scanning, least privilege
Weak cloud identity	Overprivileged keys, accounts without MFA	MFA, short-lived keys, access reviews
Insecure APIs	Management API lacks auth or rate limits	Authenticate, authorize, and log each call

Name the vulnerability



8

MINUTES · PAIRS

For each scenario, name the vulnerability type from objective 2.3 and one control that would reduce it.

1. A login form accepts ' OR 1=1 -- and logs the user in as admin
2. Two transfers sent at once both pass the balance check
3. A forum post runs a script in every reader's browser
4. A clinic X-ray PC runs an OS the vendor no longer patches
5. A new VM finds fragments of another tenant's data on its disk
6. A signed vendor update installs a backdoor at 300 customers
7. A staff phone installs an app from a website, not the store
8. Attackers exploit a flaw the vendor has never heard of

You inherit the weaknesses of your suppliers

Service provider

An MSP or vendor with remote access is breached first.

Kaseya VSA, 2021

Hardware provider

Counterfeit or tampered parts ship with flaws built in.

Gray-market network gear

Software provider

Vendor code or a library you depend on is flawed or poisoned.

Log4Shell in Log4j, 2021

Weak crypto and bad settings undo good tools

Cryptographic

- Deprecated algorithms: DES, RC4, MD5, SHA-1
- Short keys or old protocols such as SSL and TLS 1.0
- Keys stored next to the data they protect
- Flawed libraries, such as Heartbleed in OpenSSL

Misconfiguration

- Default usernames and passwords left in place
- Storage buckets or shares open to everyone
- Unneeded ports, services, and sample files
- Excess permissions and verbose error pages

Mobile devices and flaws no one has patched

Vulnerability	What it means	Control
Side loading	Installing apps from outside the official store	MDM policy that blocks unknown sources
Jailbreaking	Removing iOS restrictions to gain root access	MDM jailbreak detection, deny access
Rooting	The Android equivalent of jailbreaking	Compliance check before email access
Zero-day	A flaw exploited before a vendor patch exists	Defense in depth, behavior-based detection

Rank findings by severity and exposure

9.0 to 10
CVSS critical range

7.0 to 8.9
CVSS high range

CVE
One ID per public flaw

KEV
CISA list of exploited flaws

Score is not priority: a 7.5 on an internet-facing server can outrank a 9.8 on an isolated lab box.

Find the clinic's vulnerabilities



7

MINUTES · SMALL GROUPS

For each Lakeside item, name the vulnerability type, rate it high, medium, or low, and pick the first fix.

1. Front desk PCs still on Windows 10, past end of support
2. Patient portal search box that builds SQL from user text
3. Wi-Fi router with the admin password printed on its label
4. Backups encrypted with a DES key hardcoded in a script
5. Billing vendor with an always-on remote access tool
6. Physician's jailbroken phone that syncs clinic email
7. Ultrasound cart the maker says can never be patched

REVIEW

Five questions: write your answers first

1. A program checks a file's permissions, then opens it, and an attacker swaps the file in between. What is this?
2. What is the main fix for SQL injection?
3. Why is an end-of-life system a vulnerability even if it has no known flaws today?
4. What is resource reuse in a virtualized or cloud environment?
5. How does a vulnerability's priority differ from its CVSS severity?

TAKEAWAYS

Four things to take with you

01

Name the type, find the fix

SQLi points to code, a default password to settings, EOL to the budget.

02

Input is not code

Parameterized queries stop SQLi; output encoding stops XSS.

03

Contain what you cannot patch

Isolate EOL, legacy, and firmware-bound devices and watch them.

04

Severity is not priority

Add exposure, asset value, and KEV status to the CVSS score.

NEXT Lab 5: Discovery and vulnerability scanning with Nmap and Greenbone