Week 1 – Required Reading & Notes

Topics Included:

- MITRE ATT&CK (TTP Notes)
- OWASP Top 10
- Reconnaissance Automation

Prepared by: *Nirmiti Dhawade*

Internship: *The Cyber Ledger – Red Team*

Week: *1*

Date: *2025*

SECTION 1 –

MITRE ATT&CK NOTES

Detailed notes on two TTPs from the MITRE ATT&CK framework.

MITRE TTP 1: T1595 – Active Scanning (Reconnaissance)

Overview

Active Scanning is a technique used by attackers BEFORE exploitation. The goal is to gather as much technical information about a target as possible.
Unlike passive reconnaissance (OSINT, public sources), active scanning involves **direct interaction** with a system.

It typically includes:

- Port scanning
- Service discovery
- Vulnerability probing
- Technology fingerprinting
- SSL/TLS fingerprinting
- DNS scanning

Attackers use tools like:

- **Nmap**
- **Masscan**
- **ZMap**
- **WhatWeb**
- **httpx**
- **Nuclei**
- **Shodan / Censys automation**

Why attackers use Active Scanning

1. **Map all live systems** inside a target organization.
2. **Identify open ports/services** that could be vulnerable.
3. **Fingerprint technologies** (Apache, nginx, PHP, Windows version).
4. **Search for outdated versions** with known CVEs.
5. **Prepare the attack path** in advance.

Attackers may scan:

- Cloud assets
- Exposed VPN portals
- Email servers
- Web applications
- Misconfigured endpoints

These scans may be subtle — slow, spread out across IP ranges — or very aggressive (e.g., 10M packets/sec with Masscan).

Real-World Example

The **Palo Alto GlobalProtect 2.3M scan wave** you researched is a perfect example of **T1595 Active Scanning**:

- Attackers targeted /global-protect/login.esp
- Scanned millions of VPN portals
- Used coordinated automation through ASNs
- Searched for brute-force opportunities

Defensive Controls

Organizations defend against this by:

- Enforcing rate limiting
- Blocking suspicious IP ranges
- Using WAF rules
- Enabling Cloudflare/Zero-Trust access
- Monitoring scan patterns in SIEM
- Avoiding direct exposure of sensitive login portals

MITRE TTP 2: T1566 – Phishing (Initial Access)

Overview

Phishing is one of the most common and successful methods for attackers to gain initial access.
It targets the human layer — tricking victims into:

- Clicking malicious links
- Opening infected attachments
- Entering credentials into fake login pages
- Installing malware

Types of Phishing

1. **Bulk phishing**
    - Mass emails
    - Fake bank alerts
    - Simple social engineering
2. **Spear-phishing**
    - Targeted at one individual (CEO, engineer, finance team)
    - Customized messages
3. **Whaling**
    - Targets top executives, CFO, CEO
4. **Clone phishing**
    - Replaces a legitimate email with a malicious one
5. **Credential harvesting**
    - Fake Microsoft/Google login pages

Common Payloads

- PDF/Word malware
- Malicious macros
- “Invoice” attachments
- Fake MFA emails
- OAuth permission abuse
- QR code phishing

Why attackers use T1566

- Highest success rate
- Low cost and easy automation
- Direct path to internal systems
- Access to VPN, mailbox, corporate apps

Real-World Relevance

Most ransomware groups (LockBit, BlackCat, Black Basta) start with phishing.
Once credentials are stolen, they log in via:

- Office 365
- VPN
- Remote Desktop
- Okta

Defensive Controls

- Mandatory MFA
- Email filtering
- Attachment sandboxing
- Phishing-awareness training
- Zero-trust identity
- Domain impersonation protection

SECTION 2 –

OWASP Top 10 – Notes & Explanation

A structured breakdown of the OWASP Top 10 vulnerabilities with examples.

A01 – Broken Access Control

Definition:

Access control decides *who is allowed to do what*. Broken access control happens when attackers can access data or functions they **shouldn't**, such as viewing others' accounts, modifying data, or accessing admin pages.

Common Causes:

- Missing or weak authorization checks
- URL manipulation (e.g., /admin, /users?id=2)
- API endpoints returning extra data
- Privilege escalation flaws

Impact:

- Unauthorized access
- Data theft or modification
- Account takeover
- System compromise

Mitigations:

- Enforce server-side access checks on every request
- Implement role-based access control (RBAC)
- Use deny-by-default security
- Prevent ID tampering by using UUIDs instead of numeric IDs
- Test endpoints for privilege escalation

\

A02 – Cryptographic Failures (Sensitive Data Exposure)

Definition:

Weak or missing encryption that exposes sensitive information like passwords, credit card numbers, tokens, or personal data.

Common Causes:

- Storing passwords in plain text
- Using HTTP instead of HTTPS
- Weak hashing (MD5, SHA1)
- Exposed encryption keys

Impact:

- Credential leaks
- Identity theft
- Database compromise

Mitigations:

- Use TLS (HTTPS) everywhere
- Hash passwords using bcrypt, scrypt, or Argon2
- Encrypt sensitive data at rest
- Rotate encryption keys regularly

A03 – Injection (SQLi, OS Command Injection, etc.)

Definition:

User input is executed as a command or query (SQL, OS, LDAP), letting attackers inject malicious code.

Common Causes:

- Directly using user input in queries
- No validation or sanitation
- Legacy string concatenation

Impact:

- Database dump
- RCE (Remote Code Execution)
- Full system compromise

Mitigations:

- Use prepared statements / parameterized queries
- Validate and sanitize input
- Apply allow-list filtering
- Use ORM frameworks

A04 – Insecure Design

Definition:

Security missing at architecture level — bad design choices that create opportunities for future attacks.

Examples:

- No rate limiting
- No threat model
- Weak authentication flows

Impact:

- Large vulnerabilities later
- Widespread exploitation

Mitigations:

- Do threat modeling
- Use secure design patterns

- Apply defense-in-depth
- Enforce rate limiting & API throttling

A05 – Security Misconfiguration

Definition:

Improperly configured servers, databases, APIs, or cloud settings.

Examples:

- Default passwords
- Directory listing enabled
- Missing patches
- Open S3 buckets

Impact:

- Data leaks
- Unauthorized access
- Server takeover

Mitigations:

- Disable unnecessary features
- Patch regularly
- Harden server configs
- Automate configuration scanning

A06 – Vulnerable & Outdated Components

Definition:

Using outdated libraries, frameworks, or software with known CVEs.

Examples:

- Old versions of Apache, Nginx
- Vulnerable JavaScript libraries
- Unmaintained dependencies

Impact:

- RCE
- Data breaches
- Supply chain attacks

Mitigations:

- Maintain SBOM (Software Bill of Materials)
- Update dependencies regularly
- Use tools like Dependabot, Snyk
- Remove unused components

A07 – Identification & Authentication Failures

Definition:

Weak login mechanisms that allow attackers to bypass authentication.

Examples:

- Weak passwords
- Missing MFA
- Session ID stolen or predictable
- Poor session expiration

Impact:

- Account takeover
- Privilege abuse

Mitigations:

- Enforce strong password policy
- Use MFA (OTP, authenticator apps)
- Regenerate session IDs after login
- Use secure cookies & short session timeouts

A08 – Software & Data Integrity Failures

Definition:

When systems don't verify integrity of updates, dependencies, or critical data.

Examples:

- Relying on untrusted plugins
- No signature verification
- Supply chain attacks

Impact:

- Malware-injected updates
- System tampering

Mitigations:

- Use signed packages & code signing
- Implement CI/CD pipeline security
- Use checksum verification

A09 – Security Logging & Monitoring Failures

Definition:

Weak or missing logging, monitoring, and alerting, allowing attacks to go unnoticed.

Examples:

- No logs for failed logins
- Logs not analyzed
- Missing alert triggers

Impact:

- Delayed detection
- Extended breaches
- Larger damage

Mitigations:

- Log all critical events
- Store logs securely
- Use SIEM tools (Splunk, ELK)
- Enable alerting for brute-force, anomalies

A10 – Server-Side Request Forgery (SSRF)

Definition:

Attacker tricks the server into making internal or external requests on their behalf.

Examples:

http://target.com/?url=http://127.0.0.1/admin

Impact:

- Access to internal networks
- Fetch AWS metadata → full cloud takeover

Mitigations:

- Block internal IP ranges
- Enforce URL allow-list
- Disable redirects
- Use network segmentation

SECTION 3 –

Reconnaissance Automation –Notes

Comprehensive notes on automated recon workflows and tooling.

What Is Reconnaissance Automation?

Reconnaissance automation is the process of using specialized tools and scripts to automatically gather information about a target, such as networks, systems, or websites, in order to find vulnerabilities, misconfigurations, and other weaknesses. This automation streamlines reconnaissance by performing repetitive tasks quickly and consistently, allowing cybersecurity professionals and ethical hackers to save time and focus on more complex analysis.

Automation helps attackers (and ethical hackers) gather large-scale intelligence efficiently. Instead of manually scanning one domain, automation pipelines collect:

- Subdomains
- Ports
- Services
- Endpoints
- Technologies
- Vulnerabilities

1. Subdomain Enumeration

Tools that discover subdomains automatically.

• Amass

Enterprise-grade subdomain enumeration using OSINT + brute-force.
Fully automated with rich output.

• Subfinder

Fast passive subdomain finder powered by many OSINT sources**.**

• Assetfinder

Quick OSINT-based tool for finding associated domains/subdomains**.**

2. Live Host & HTTP Probing

Tools that check which domains are alive.

• Httpx

Checks live hosts, fetches status code, title, tech stack.

• Httprobe

Simple tool that filters live web hosts from large lists.

3. Port Scanning & Service Discovery

• Nmap

Most widely used scanner — ports, services, OS detection.

• Masscan

Internet-wide port scanning, extremely fast.

• Naabu

Fast TCP port scanner built by ProjectDiscovery**.**

4. Technology Fingerprinting

• WhatWeb

Identifies technologies used on websites (CMS, frameworks, servers).

• Wappalyzer (CLI)

Detects tech stacks, JavaScript libraries, analytics, etc.

5. Directory & Endpoint Discovery

• Gobuster

Brute-force directories and files on web servers.

• Feroxbuster

High-speed recursive directory enumeration.

6. Vulnerability Automation

• Nuclei

Template-based scanner that automates vulnerability detection.

• Nikto

Web server scanner — checks for misconfigurations and old versions.

7. OSINT for People & Company Data

• theHarvester

Gathers emails, subdomains, IPs from public sources.

• Sherlock

Finds usernames across social platforms.

• Hunter.io / Email OSINT tools

Discovers corporate email patterns.

8. DNS Enumeration

- **Dnsx**

DNS resolver with filtering capabilities.

- **Dig / Nslookup**

Classic DNS OSINT tools.

Key aspects and benefits

Speed and efficiency: Automated tools can scan large digital landscapes much faster than a human, covering thousands of IP addresses or domains in minutes.

Consistency: Automated tools follow predefined instructions, which reduces the risk of human error or oversights.

Scalability: Automation can handle large and complex networks with ease, making it suitable for organizations with extensive digital footprints.

Focus on high-value tasks: By automating the tedious and repetitive parts of information gathering, security professionals can spend more time on manual analysis and strategic thinking.

Frameworks and tools: Recon automation often involves creating or using frameworks that chain together various tools to perform a sequence of tasks, such as subdomain enumeration, port scanning, and vulnerability checks.

Examples of automated tasks

- Finding subdomains and checking if they are live
- Scanning for open ports and services
- Performing vulnerability checks for things like SQL injection or cross-site scripting (XSS)
- Gathering information from sources like the Wayback Machine
- Searching code repositories for sensitive information like API keys or passwords .