Summary

- Focus: Scanning and Vulnerability Analysis concepts.

Information Gathering

- Reconnaissance involves gathering public information about targets.
- Passive information gathering reduces detection risk.
- Open-Source Intelligence (OSINT) utilizes publicly available sources.
- OSINT techniques: Web Scraping, Google Dorking, social media profiling.
- Tools like host, nslookup, dig, whois, knockpy, netdiscover, traceroute, whatweb, theHarvester, sherlock, wfw00f are used.

Scanning and Vulnerability Analysis

- Scanning identifies system services and entry points using tools like nmap, zenmap, unicorn, nikto.
- Vulnerability analysis uncovers weaknesses using tools like nessus, searchsploit and OpenVAS, and Metasploit.

Tools

- Scanning and Vulnerability Analysis Tools:
  - OS and NW: nmap, nessus, openVAS, tripwire, wireshark, MSF
  - Web Applications: Burp Suite, nikto, OWASP ZAP
  - Mobile Applications: frida, drozer, MobSF, Burp Suite

Vulnerability Analysis Steps

- Scan for known vulnerabilities using databases like NVD.
- Assess vulnerability severity using CVSS/VPR.
- Submit a report with vulnerabilities, risk levels, and mitigation steps.

Environment Setup

- Example setup: Kali Linux (attacker) and Metasploitable 2 (target).

Metasploit Framework (MSF)

- MSF is an open-source penetration testing and exploitation platform.
- Supports all phases of penetration testing.
- Includes a command-line interface (msfconsole).
- Key concepts: vulnerability, exploit, and payload.

Msfconsole Interface

- msfconsole is a popular MSF interface providing access to options.

- `help` command lists available commands.

- Running `msfconsole` as `sudo` may be necessary.

- Focus: Scanning and Vulnerability Analysis concepts.

Metasploit Framework Anatomy

- Metasploit Framework files are located in `/usr/share/metasploit-framework/`.
- Interaction primarily occurs through seven modules within the `/usr/share/metasploit-framework/modules/` directory.
- Modules are Ruby scripts for specific tasks.

Module Types

- **Auxiliary**: Information gathering and vulnerability analysis (port scanners, sniffers, fuzzers). Examples: `syn.rb`, `tcp.rb`.
- **Exploits**: Exploiting vulnerabilities in OS, network services, applications for unauthorized access.
- **Payloads**: Code that runs remotely on a compromised system.
  - **Singles**: Self-contained, single-task payloads (e.g., command execution).
  - **Stagers**: Small payloads establishing a connection back to the attacker.
  - **Stages**: Larger payloads sent over the established connection.
- **Encoders**: Encoding payloads to evade detection (e.g., antivirus).
- **Nops**: Generating NOP sleds to modify payload signatures for evasion.
- **Evasion**: Bypassing security mechanisms like firewalls and IDS.
- **Post**: Post-exploitation activities (privilege escalation, data exfiltration, persistence).

Msfconsole Basic Commands

- Running `msfconsole` as `sudo` is recommended.

- Familiarization with commands is crucial.

- `help`: Lists available commands with descriptions. `help <command>` gives specific command help.

- `banner`: Displays an ASCII art banner with version and module counts.
- `exit/quit`: Exits `msfconsole`.

- `show nops`: Displays scripts, disclosure date, rank, check and description of each.

- `search <term>`: Searches for exploits, payloads, auxiliary modules. `search type:<module_type> <term>` filters by module type (e.g., `auxiliary`, `exploit`). `search cve:<CVE_ID>` searches by CVE ID.

- `searchsploit <term>`: Searches Exploit Database (EDB) for publicly available exploits.

- `info <module>`: Displays information about a specific module.

- `use <module>`: Changes context to a specific module, exposing module-specific commands. `back` returns to the main context.
- `show options`: Displays available/required settings for the current module.
- `show advanced`: Displays advanced options for the current module.
- `set <param> <value>`: Sets a parameter value.
- `unset <param>`: Removes a previously set parameter. `unset all` removes all assigned variables.
- `setg <param> <value>`: Sets a global parameter value for all modules.
- `run` or `exploit`: Executes the loaded and configured module.

Port Scanning with Metasploitable2

- `nmap -sV <ip of M2>`: Runs an nmap port scan from within `msfconsole`.
- Metasploit offers various internal port scanners within `auxiliary/scanner/portscan`.
- `search portscan`: Lists available port scanners.
- Example: Performing a SYN scan.
    - `use auxiliary/scanner/portscan/syn`
    - `show options`
- Focus: Scanning and Vulnerability Analysis concepts.

Port Scanning with Metasploitable2 (cont.)

- `set RHOSTS <IP of M2>`: Sets the target IP.
- `set THREADS 50`: Increases scan speed.
- `run`: Executes the SYN scan.
- The SYN scan identifies open ports on the target.
- Other scripts like `ack.rb` and `tcp.rb` can be used for comparison.

Version Scanning on Metasploitable2

- Determine service versions running on open ports.
- **SMB Version Scanning:**
    - `use auxiliary/scanner/smb/smb_version`
    - `set RHOSTS <IP of M2>`
    - `run`: Identifies the SMB service version.
- **FTP Version Scanning:**
    - `use auxiliary/scanner/ftp/ftp_version`

    - `set RHOSTS <IP of M2>`
    - `run`: Identifies the FTP service version (e.g., vsftpd 2.3.4).

- **HTTP Version Scanning:**
    - `use auxiliary/scanner/http/http_version`
    - `set RHOSTS <IP of M2>`
    - `run`: Identifies the web server version (e.g., Apache 2.2.8 with PHP 5.2.4).

- Scripts to find versions of SSH, SMB, MySQL, and Postgres are recommended for further practice.

Directory Scanning on Metasploitable2

- Uses auxiliary scanner modules to find directories, files, and shares.

- **HTTP Directory Scanning:**
    - `use auxiliary/scanner/http/dir_scanner`
    - `set RHOSTS <IP of M2>`
    - `run`: Discovers directories on the web server.

- Example: Accessing `http://<IP of M2>:80/phpMyAdmin`.

- Directory scanning on Tomcat server (port 8180) can reveal admin interfaces.
    - `use auxiliary/scanner/http/dir_scanner`
    - `set RHOSTS <IP of M2>`
    - `run`: Discovers directories like `/admin/`, `/webdav/`, `/tomcat-docs/`.

- Brute-forcing login credentials on discovered admin interfaces can be attempted.

Anonymous User Access in Network Services

- Checks for misconfigurations allowing anonymous access.

- **FTP Anonymous Access Check:**
    - `use auxiliary/scanner/ftp/anonymous`
    - `set RHOSTS <IP of M2>`
    - `run`: Determines if anonymous FTP login is enabled.

- If enabled, access with username `anonymous` and a blank password using `ftp <ip of M2>`.

- Focus: Scanning and Vulnerability Analysis concepts.

Brute-Force Login on Metasploitable2

- Apache Tomcat server runs on Metasploitable2 (M2) at port 8180.

- Admin panel accessible via `http://<IP of M2>:8180/admin` if credentials are known.

- `tomcat_mgr_login.rb` script performs brute-force attacks against Tomcat.

- Parameters:

  - `USERNAME` & `PASSWORD`: Single username/password.
  - `USER_FILE` & `PASS_FILE`: Files with usernames/passwords (one per line).
  - `USERPASS_FILE`: File with usernames/passwords (username space password).

- Example:

  - Use `auxiliary/scanner/http/tomcat_mgr_login`

  - Clear existing settings:

    ```
    set --clear username set --clear password set --clear
    user_file set --clear pass_file set --clear userpass_file
    ```

  - Set user and password files:

    ```
    set user_file /usr/share/Metasploit-framework/data/
    wordlists/tomcat_mgr_default_users.txt set pass_file /
    usr/share/Metasploit-framework/data/wordlists/
    tomcat_mgr_default_pass.txt
    ```

  - Set target and port:

    ```
    set RHOSTS <IP of M2> set RPORT 8180
    ```

  - `run`: Executes the brute-force attack.

  - Successful login credentials can be used to access Tomcat's admin panel.
  - Students should practice brute-force attacks using `ssh_login.rb`, `mysql_login.rb`, and `postgres_login.rb`.

Vulnerable Services on Metasploitable2

- List of vulnerable services running on Metasploitable2 with CVEs and attack vectors.

- TCP Port 21 - vsftpd 2.3.4 (FTP Server):

  - CVE: CVE-2011-2523
  - Attack Vector: Backdoor opens a reverse shell on port 6200/tcp upon login with username ending in `:)`.

  - TCP Port 22 - OpenSSH 4.7p1 (SSH Server):

  - CVE: No specific CVE.

  - Attack Vector: Susceptible to brute-force attacks due to weak configurations.

◦ TCP Port 23 - Telnet (Remote Login Service):

◦ CVE: No specific CVE.

◦ Attack Vector: Transmits data in plaintext, susceptible to network sniffing.

◦ TCP Port 25 - Postfix (SMTP Server):

◦ CVE: No specific CVE.

◦ Attack Vector: Misconfiguration can lead to open relay issues, exploited by spammers.

◦ TCP Port 53 - BIND 9.4.2 (DNS Server):

◦ CVE: CVE-2009-0025

◦ Attack Vector: Denial of service via DNSSEC validation issues.

◦ TCP Port 80 - Apache 2.2.8 (HTTP Server):

◦ CVE: CVE-2007-6750

◦ Attack Vector: Vulnerable to denial of service via partial HTTP requests.

◦ TCP Ports 139 & 445 - Samba 3.0.20 (SMB/CIFS):

◦ CVE: CVE-2007-2447

◦ Attack Vector: Flaw in "username map script" allows remote code execution.

◦ TCP Port 512, 513, 514 - Rexec, Rlogin, Rsh (Remote Execution Services):

◦ CVE: No specific CVEs.

◦ Attack Vector: Transmit data in plaintext, susceptible to interception.

◦ TCP Port 2049 - NFS (Network File System):

◦ CVE: No specific CVE.

◦ Attack Vector: Improper configuration allows remote attackers to mount NFS shares.

◦ TCP Port 2121 - ProFTPD 1.3.1 (FTP Server):

◦ CVE: CVE-2006-5815

◦ Attack Vector: Command injection flaw allows remote command execution.

◦ TCP Port 3306 - MySQL 5.0.51a (Database Server):

◦ CVE: No specific CVE.

◦ Attack Vector: Weak default configurations may allow root access without a password.

◦ TCP Port 5432 - PostgreSQL 8.3.0 (Database Server):

◦ CVE: No specific CVE.

◦ Attack Vector: Default or weak passwords allow unauthorized database access.

◦ TCP Port 5900 - VNC (Virtual Network Computing):

◦ CVE: No specific CVE.

◦ Attack Vector: Improperly secured VNC allows unauthorized remote desktop access.

◦ TCP Port 6667 - UnrealIRCd 3.2.8.1 (IRC Server):

◦ CVE: CVE-2010-2075

◦ Attack Vector: Backdoor allows remote command execution via crafted commands.