



A T M E
College of Engineering

GOLD

QS I-GAUGE
INDIAN COLLEGE RATINGS



MODULE 2

CHAPTER 3

PHYSICAL PENETRATION ATTACKS

Introduction To Physical Penetration Testing

Breaching Physical Barriers in Cybersecurity

Purpose & Advantages

- Gain internal access to networks & data
- Often easier than remote attacks

Challenges & Skills

- Varies by perimeter sophistication
- Requires social engineering skills
- Risk of detection

WHY PHYSICAL PENETRATION TESTING IS IMPORTANT

"Crunchy on the Outside, Soft on the Inside"

The "Crunchy Exterior" Reality

- Strong perimeter, weak internal controls
- Still common in many organizations

Crunchy on the Outside, Soft on the Inside"

Damage & Value of Information

- Assess real damage from local attackers
- Valuable private info for criminals

Client Awareness & Acceptance

- Often overlooked until explained
- Most accept once value is understood

Client Awareness & Acceptance

- Often overlooked until explained
- Most accept once value is understood

Client Awareness & Acceptance

- Often overlooked until explained
- Most accept once value is understood

NOTE: Physical testing reveals hidden risks. Essential for complete security assessment!

CONDUCTING A PHYSICAL PENETRATION TEST

Safe & Effective Execution

Timing: Business Hours

- Tests controls, procedures & personnel
- Avoid after-hours risks (armed response, arrest)

Client Coordination

- Designated contact for vouching
- Agreed time window
- Client acts unaware if encountered

Preparation & Execution

Lay groundwork → Plan → Execute ethically!

Entrances & Blending In

- Front vs Secondary Entrances
- Contractor/ Maintenance Ruse
- Observe Badges in Public

MENTAL PREPARATION FOR PHYSICAL PENETRATION

Staying Calm Under Pressure

Key Mental Aspects

- Stress Management
- Project Confidence
- Adopt "Insider" Mindset

Practical Preparation

- Role-play common questions with partner
- Quick thinking for unexpected scenarios

- **NOTE**

Confidence & Mindset = Success in blending in.
Practice makes perfect!

Common Ways into a Building

Understanding Physical Security Risks

- Physical Security Awareness
- Importance of Secure Building Access
- Preventing Unauthorized Entry

1.The Smokers' Door

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:**

Busy Traffic: May need to smoke a cigarette (can be faked). Be ready to explain your presence, who you work for, etc.

- **Accomplice**
- **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.



- Manned Checkpoints
- Multitenant Building Lobby Security
- Campus-Style or Single-Tenant Buildings
- Mantraps
- Locked Doors
- The Unmanned Foyer
- Biometric Locks

CHAPTER 4

INSIDER ATTACKS

Introduction to Insider Attacks

Attacking from the perspective of someone who *already has access* to the target's information systems (an insider).

Common Factor in Crimes: Unauthorized insider access is a common factor in:

Identity theft

Intellectual property theft

Stolen customer lists

Stock manipulation

Espionage

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:

- Learn details about overall security posture not discoverable from external-only tests.
- Even in large companies, insiders represent a smaller field of attackers than the public Internet.
- Potential for damage by insiders is demonstrably greater.
- Insiders typically have working knowledge of company security controls, processes, and data storage locations.

Conducting an Insider Attack

1. Tools and Preparation for Insider Attacks
2. Orientation (Initial Reconnaissance from Inside)
3. Gaining Local Administrator Privileges

Preferred Methods

- Using Offline NT Password and Registry Editor
- Recovering the Administrator Password (Cracking Hashes)

4. Disabling Antivirus (AV)
5. Raising Cain (Installation and Usage)

CHAPTER 5 METASPLOIT

Metasploit: The Big Picture

A free, downloadable framework for acquiring, developing, and launching exploits for computer software vulnerabilities.

Getting Metasploit

Using the Metasploit Console to Launch Exploits

Session Management

Example: Reverse Shell (windows/shell/reverse_tcp)

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.

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.

Automating and Scripting Metasploit

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.

2. *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.