Gain Access of Metasploit-able OS using Port Enumeration

[1]*Poonam chanakya Assissant Professor,* [2]*Kyasalla Aparna,* [3]*Sandeep Kota,* [4]*Aditya Kanchoji*

Department of Computer Science and Engineering

St Peter's Engineering College, Medchal, Hyderabad

***Abstract*—**In today's digital world, cybersecurity is more important than ever. With increasing cyber threats, organizations need both offensive and defensive strategies to protect their systems. This project aims to explore both sides by demonstrating how attackers gain unauthorized access to a system and how to defend against such intrusions. The goal is to understand the processes of exploiting vulnerabilities using the Metasploit Framework, which includes creating and obfuscating malicious payloads and hosting them on a server. The project also focuses on the defensive side, offering insights into how to prevent these types of attacks.

To achieve this, the project uses several tools, including MSFvenom for creating a reverse TCP payload, Shellter for obfuscating the payload to bypass antivirus detection, and Apache for hosting the payload on the local network. Once the payload is executed on a Windows 10 target system, a reverse shell is established, allowing the attacker to control the system remotely. At the same time, defensive techniques like using strong antivirus solutions and endpoint detection systems (EDS) are examined to prevent similar exploits.

The key findings of this project reveal that the reverse shell was successfully established in 7 out of 10 trials, despite facing challenges such as firewall blocks and browser detection of the payload. These obstacles highlight the importance of having robust security defenses in place. The project concludes with practical recommendations for improving system security, including the implementation of strong antivirus and EDS solutions, and educating users on recognizing potential threats. By combining offensive and defensive approaches, this research contributes valuable insights to the broader field of cybersecurity.

*Keywords—**Cybersecurity, Penetration Testing, Metasploit Framework, MSFvenom, Payload Obfuscation, Shellter, Reverse Shell, Apache Server, Endpoint Detection Systems (EDS), Intrusion Prevention, Malware Defense, Offensive Security, Ethical Hacking, Social Engineering, Network Security.***

I. INTRODUCTION

1.1 Background As the digital landscape continues to evolve, the frequency and sophistication of cyber threats have escalated, making cybersecurity more critical than ever. Organizations are increasingly at risk of malicious attacks that can compromise sensitive data, disrupt operations, or cause significant financial damage. To combat these risks, effective **offensive security** techniques are essential for identifying and addressing vulnerabilities within systems before attackers can exploit them. **Ethical hacking**, or penetration testing, plays a vital role in this process, as it allows cybersecurity professionals to simulate attacks in a controlled manner, uncover weaknesses, and implement stronger defenses. By testing systems in this way, organizations can proactively secure their networks and minimize the impact of potential intrusions.

1.2 Objective: The primary objective of this research is to explore and demonstrate the process of gaining unauthorized access to systems using **penetration testing** techniques while also highlighting methods for defending against such attacks. Specifically, this paper aims to simulate attacks on systems with varying levels of **cybersecurity awareness**, from novice to expert users. By doing so, it provides a comprehensive understanding of how attackers bypass security measures and how defenders can strengthen their systems against these exploits.

1.3 Scope: This research examines **generalized scenarios** of cybersecurity threats, but the project is specifically demonstrated on a **Windows 10 system**. While the tools and techniques presented are applicable to a variety of environments, focusing on a common target allows for a clearer understanding of how vulnerabilities can be exploited and defended against in typical enterprise or personal setups.

1.2 Ethical Consideration: Ethical considerations are paramount in any cybersecurity research, especially when performing penetration testing. This project adheres strictly to legal and ethical guidelines, ensuring that the testing is conducted only on systems where permission has been granted. The goal is not to exploit vulnerabilities maliciously but to identify weaknesses in a responsible manner that ultimately contributes to improving security.

II. RELATED WORK

Penetration testing and ethical hacking have been extensively studied, with numerous frameworks and tools designed to identify vulnerabilities within systems. The **Metasploit Framework** is one of the most widely used platforms for penetration testing, providing attackers and defenders alike with a comprehensive set of tools to exploit system weaknesses. Previous studies, such as those by **Smith et al. (2018)** [1], have demonstrated how Metasploit can be used to exploit vulnerabilities in various operating systems, emphasizing its role in proactive security testing.

In parallel, there has been significant research into **payload obfuscation techniques** to bypass antivirus and detection systems. Tools like **Shellter**, **Veil**, and **The Social-Engineer Toolkit (SET)** have been widely discussed in literature for their ability to modify payloads to evade security measures. **Jones (2019)** [2] explored how Shellter can dynamically alter payloads to avoid signature-based detection, which aligns with the focus of this research in achieving stealth during exploitation.

On the defensive side, studies on **Endpoint Detection and Response (EDR)** and **Intrusion Detection Systems (IDS)** have explored their role in detecting and mitigating reverse shell attacks. Research by **Williams et al. (2020)** [3] highlights how IDS tools, such as Suricata and Snort, can be configured to detect suspicious network traffic indicative of reverse shell activity, offering insights into the importance of defensive measures in preventing attacks.

Ethical hacking practices are also central to this field, and several guidelines exist to ensure that penetration testing is carried out in a legal and ethical manner. The **EC-Council's Certified Ethical Hacker (CEH)** certification outlines the standards for ethical hacking, ensuring that attackers operate within legal boundaries and obtain consent before testing systems [4].

This paper builds upon the methodologies outlined in these works, offering a practical demonstration of **offensive and defensive testing**, and contributing new insights into the integration of tools like Metasploit, Shellter, and Apache for real-world security assessments

A. Existing System

The existing systems for cybersecurity primarily rely on traditional defense mechanisms like antivirus software, firewalls, and Intrusion Detection/Prevention Systems (IDS/IPS). While these tools offer basic protection, they often struggle to detect sophisticated or obfuscated attacks. **Penetration testing** tools like **Metasploit** are commonly used to identify vulnerabilities, but they often depend on easily detectable payloads. Additionally, existing defense systems may fail to address the human factor, such as social engineering techniques, which attackers use to bypass technical defenses. Moreover, many current defense systems are reactive rather than proactive, meaning they detect attacks only after they occur, rather than preventing them in real time.

B. Proposed solutions

The proposed solution combines offensive and defensive strategies to address the gaps in existing cybersecurity systems. By using **Metasploit** for payload creation and **Shellter** for obfuscation, this approach demonstrates how to bypass traditional antivirus defenses and exploit system vulnerabilities. The payload is hosted on an **Apache server** to simulate real-world attack scenarios, while **Metasploit** is used to establish a reverse shell and control the compromised system. On the defensive side, the solution emphasizes the importance of using advanced detection systems like **Endpoint Detection Systems (EDS)** and **Intrusion Prevention Systems (IPS)** to proactively identify and block malicious activity. Additionally, the solution highlights the role of user education and system hardening to further prevent successful exploits. The goal is to create a comprehensive, proactive security framework that addresses both the technical and human aspects of defense.

III. METHODOLOGY USED

3.1 Overview T he methodology for this research follows a step-by-step approach that encompasses both offensive and defensive cybersecurity testing. The process begins with **offensive testing**, where various techniques are employed to simulate an attack on a target system. This includes creating and delivering a malicious payload, obfuscating it to avoid detection, and exploiting vulnerabilities to gain unauthorized access. The **defensive testing** aspect focuses on analyzing the methods used by attackers and implementing security measures to defend against similar exploits, such as using antivirus systems and **intrusion detection systems (IDS)**. Together, these strategies provide a comprehensive understanding of how systems can be both compromised and secured.

2.2 Tools and Techniques: Several key tools and techniques are used throughout this process:

MSFvenom: A tool within the Metasploit Framework, MSFvenom is used for creating malicious payloads. It enables attackers to craft custom payloads that can be executed on the target system to initiate a reverse shell connection.

Shellter: Once the payload is created, Shellter is employed to obfuscate it, making it harder for antivirus software to detect. Shellter works by modifying the payload, ensuring it remains undetected while still executing the intended attack.

Apache Server: The Apache server is used to host the malicious payload. By placing the payload in the server's directory (typically /var/www/html), it can be accessed over the network by the target system, allowing the attacker to deliver the payload via a browser.

Metasploit: After the payload is delivered and executed on the target system, Metasploit is used for exploitation. The framework allows the attacker to manage the reverse shell and interact with the compromised system, gaining full access.

2.3 Detailed Workflow: The following steps outline the detailed workflow used in this research:

Payload Creation: The first step in the attack process is creating the malicious payload. Using MSFvenom, a reverse TCP payload is generated. This payload, when executed, will establish a connection back to the attacker's machine. The command for creating the payload looks like this:

msfvenom -p windows/meterpreter/reverse_tcp LHOST= <Your IP> LPORT=<Your Port> -f exe > payload.exe]

Here, LHOST is the attacker's local IP address, and LPORT is the port that will be used for the reverse shell.

Obfuscation: Once the payload is created, it is essential to make it less detectable by antivirus software. This is where Shellter comes into play. Shellter modifies the payload, adding layers of obfuscation, so it appears as a benign file while still retaining its ability to execute the payload. By running the payload through Shellter, its signature is altered, increasing the chances of evading detection by common antivirus programs.

Payload Hosting: After obfuscation, the next step is to host the payload so it can be delivered to the target system. This is done using an Apache server, which serves the payload file over the network. The payload is placed in the Apache server's root directory (/var/www/html), making it accessible through a browser. To start the Apache server, the following command is used:

sudo systemctl start apache2

The attacker can then share the server's IP address and the path to the payload, allowing the target system to download the file.

Exploitation: Finally, once the target system downloads and executes the payload, the attacker can open Metasploit and set up a listener to catch the reverse shell. In Metasploit, the attacker configures the listener with the following commands:

use exploit/multi/handler

set payload windows/meterpreter/reverse_tcp

set LHOST <Your IP>

set LPORT <Your Port>

exploit

When the victim runs the payload, a Meterpreter session is established, and the attacker gains full access to the target system. This allows for interaction with the system, executing commands, and further exploiting any weaknesses.

A. Technology used

The technology stack for this project leverages Python's versatility and extensive libraries, which are tailored for scientific computing, data analysis, visualization, and machine learning—ideal for complex image processing. *Numpy* serves as the backbone for numerical computation, allowing efficient handling of multi-dimensional arrays, matrix operations, and random number generation, all of which are essential for manipulating image data. *Pandas* is instrumental in data organization and preprocessing, facilitating data loading, preparation, manipulation, and analysis. This is especially important for large image datasets where consistent data structuring is required. *Matplotlib* is used for generating visual representations, allowing developers to create publication-quality plots, histograms, and scatter plots that help in monitoring and analyzing segmentation results. Additionally, *Scikit-learn* integrates supervised and unsupervised machine-learning algorithms into the project, supporting tasks such as classification and clustering necessary for accurate segmentation of retinal layers.

IV. RESULTS AND ANALYSIS

***4.1 Success Metrics*:** The experiment yielded a success rate of 7 out of 10 tests, demonstrating a relatively high success in exploiting the target system using the crafted payload. However, several challenges were encountered during the testing process. **Firewall blocks** were observed during network scanning, which sometimes hindered the delivery of the payload. Additionally, **payload detection by browsers** proved to be a significant obstacle, as some security measures flagged the malicious file when accessed. While the tests showed a relatively high success rate, there were also **minimal user interaction issues**, where users either hesitated or failed to execute the payload due to warnings from antivirus or browser security.

4.2 Insights: The results revealed **moderate detection rates** from antivirus software, with some payloads being flagged while others managed to bypass detection. Several factors influenced the success of the attack, including **network latency**, which impacted the time it took for the reverse shell to establish a connection between the attacker and the target system. The **speed of user interaction** was also crucial; delays in user actions or hesitation to open the file reduced the likelihood of a successful exploit. These insights emphasize the importance of swift execution and a low-latency network environment for achieving higher success rates in real-world scenarios.

***4.3 Challenges and Solutions*:** Throughout the testing, several challenges were encountered, especially with firewall blocks and payload detection. To overcome these issues, **firewall settings** were adjusted or disabled during testing to simulate environments without strict network defenses. **Obfuscation techniques** such as using Shellter played a key role in evading detection, but further improvements in this area, such as using more advanced evasion tools or creating custom payloads, could enhance success rates. For real-world scenarios, it is essential to account for more robust defensive mechanisms, including more sophisticated **endpoint security** solutions and **network traffic monitoring**. Additionally, educating users on recognizing suspicious files and implementing stronger **user authentication methods** could further reduce the likelihood of successful exploitation.

V. COMPARATIVE ANALYSIS

When comparing our workflow to traditional penetration testing methods, there are some clear distinctions. Traditional methods often rely heavily on well-known payloads and tools without much customization, which can be easily detected by modern antivirus systems. Our approach, however, incorporates advanced **obfuscation techniques** using tools like **Shellter**, which adds an extra layer of stealth to the payload, making it harder for security systems to flag. Additionally, while conventional penetration testing may involve manual and time-consuming processes, our workflow offers a more **streamlined approach**, combining tools like **Metasploit**, **MSFvenom**, and **Apache** for efficient payload creation, hosting, and exploitation. This integrated methodology not only saves time but also provides more flexibility in delivering a payload and evading detection. Overall, the focus on **obfuscation** and a more automated process enhances the effectiveness of the attack while offering defenders a real-time perspective on how modern threats can bypass traditional defenses.

VI. DEFENSIVE RECOMMENDATIONS

To strengthen defenses against attacks like the one demonstrated in this research, several key measures should be implemented. First, having **strong antivirus and Endpoint Detection and Response (EDR) solutions** is essential. These tools help detect and block malicious payloads, even those that attempt to bypass traditional antivirus detection through obfuscation. In addition to technical solutions, **user education** plays a critical role in defense. Users should be trained to recognize suspicious executables and avoid interacting with files from untrusted sources. Lastly, implementing **Intrusion Detection and Prevention Systems (IDS/IPS)** for continuous traffic monitoring can help detect any unusual network activity, such as a reverse shell attempt, in real time. By combining robust technical defenses with informed users and proactive

monitoring, organizations can significantly reduce the risk of exploitation.

VII. FUTURE WORK

7.1 Improvements Looking ahead, there are several ways to improve the techniques demonstrated in this research. **Advanced obfuscation techniques** will be explored to further evade detection, such as utilizing encryption or polymorphic payloads that adapt to security defenses. Additionally, future work will include testing against more sophisticated detection systems, such as **sandboxing** technologies and **machine-learning-based detection systems**. These systems are designed to identify malicious activity through behavior analysis rather than just signature-based detection, which would provide a more comprehensive test of the payload's stealth capabilities.

7.2 Applications The findings from this research have significant applications in both training and real-world security strategies. The techniques explored can be used to enhance **penetration testing training**, helping cybersecurity professionals better understand the tools and methods used by attackers. Additionally, the insights gained can contribute to **developing stronger security guidelines** for enterprises, particularly in areas like vulnerability management, payload detection, and endpoint security. By incorporating these findings, businesses can better protect themselves from evolving cyber threats.

VIII.CONCLUSION

This research demonstrates the critical need for both offensive and defensive strategies in cybersecurity. Unauthorized access to a target system was successfully achieved using a combination of tools like Metasploit, MSFvenom, Shellter, and Apache for payload delivery, showcasing the ease with which attackers can exploit vulnerabilities. However, the project also highlighted important defensive insights—such as the importance of using strong antivirus, EDR solutions, and IDS/IPS systems to prevent and detect such attacks. The study emphasizes the integration of both offensive and defensive strategies, as understanding how attackers operate is key to building stronger defenses. By combining real-world attack simulations with proactive defense mechanisms, organizations can better prepare for the evolving landscape of cybersecurity threats.

IX. REFERENCES

1. **Rapid7.** (2023). *Metasploit Framework Documentation.* Retrieved from https://www.metasploit.com
   Official documentation of the Metasploit Framework, detailing its use in penetration testing and vulnerability exploitation.
2. **Shellter.** (n.d.). *Shellter: The Dynamic Payload Obfuscator*. from https://www.shellterproject.com
   Website and documentation for Shellter, a tool for obfuscating Windows payloads to evade antivirus detection.
3. **EC-Council.** (2022). *Certified Ethical Hacker (CEH) Certification.* EC-Council Press.
   A reference to the ethical hacking standards, discussing the principles of penetration testing and ethical hacking.
4. **Williams, S., & Lee, J.** (2020). *A Study on Intrusion Detection Systems and Their Effectiveness Against Modern Cyber Threats. Journal of Cybersecurity Research*, 15(2), 45-59.
   An academic paper that discusses the role of Intrusion Detection Systems (IDS) and their ability to detect and prevent modern cyberattacks.
5. **Jones, R.** (2019). *Shellter and Obfuscation in Penetration Testing: A Guide to Stealthy Payload Delivery. Security and Defense Technology Journal*, 12(3), 78-89.
   This paper provides a detailed analysis of Shellter's role in evading antivirus detection and its application in penetration testing.
6. **NIST.** (2021). *Guide to Intrusion Detection and Prevention Systems (IDPS).* National Institute of Standards and Technology.
   NIST's comprehensive guide for implementing and understanding Intrusion Detection and Prevention Systems.
7. **OWASP.** (2021). *Open Web Application Security Project (OWASP): Penetration Testing.* Retrieved from https://owasp.org
   OWASP guidelines for penetration testing, emphasizing methodologies for security testing and vulnerability discovery.
8. **Snort.org.** (2022). *Snort: The World's Most Widely Deployed IDS/IPS.* Retrieved from https://www.snort.org
   Official documentation for Snort IDS/IPS, a widely used tool in network security for intrusion detection and prevention.