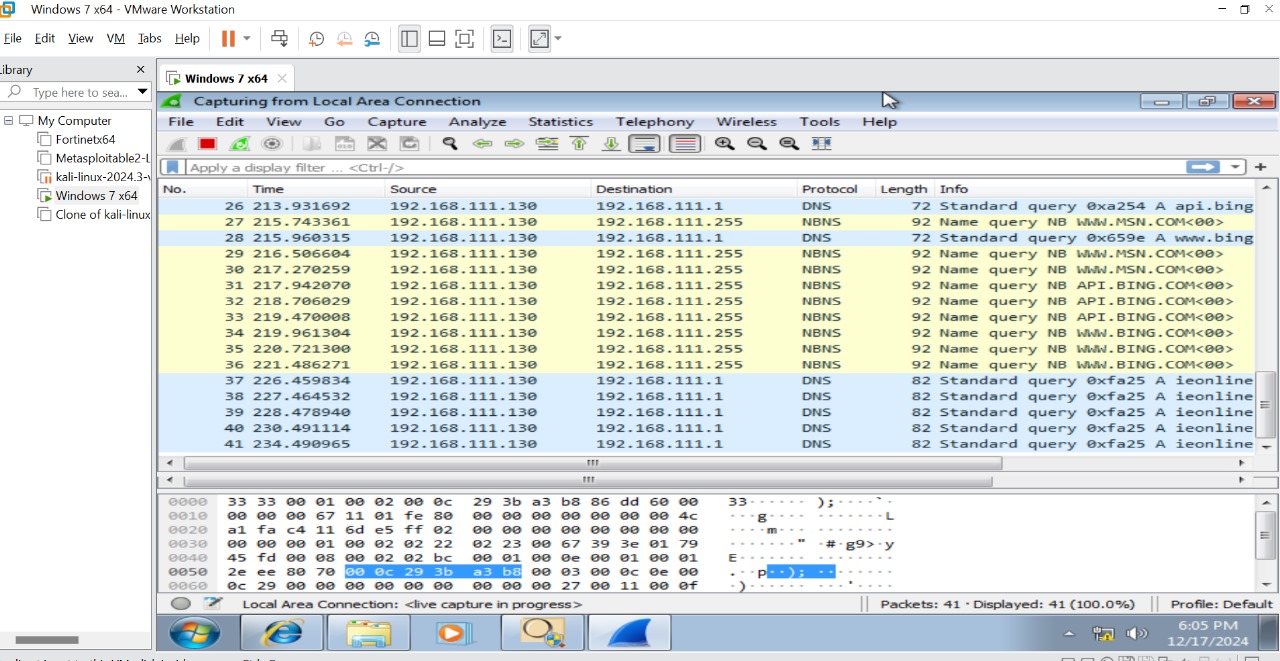
**Name:** Mamdouh Ahmed Amin 20221441410

**Name:** Nour Eldeen Ahmed Ibrahim 2022447712  
**Name:** Mohamed Elsayed Fawzy 20221374509  
**Namen:** Mahmoud Saeed Mansour 20221443364  
  
  
Firstly we ran the malware on a windows 7 virtual then we captured the network traffic using wireshark we already installed on the infected machine   
We took the pcap file of the traffic and transferred it using a USB driver to our Kali Linux workstation to investigate it using Suricata and Splunk (1st and 2nd Task)

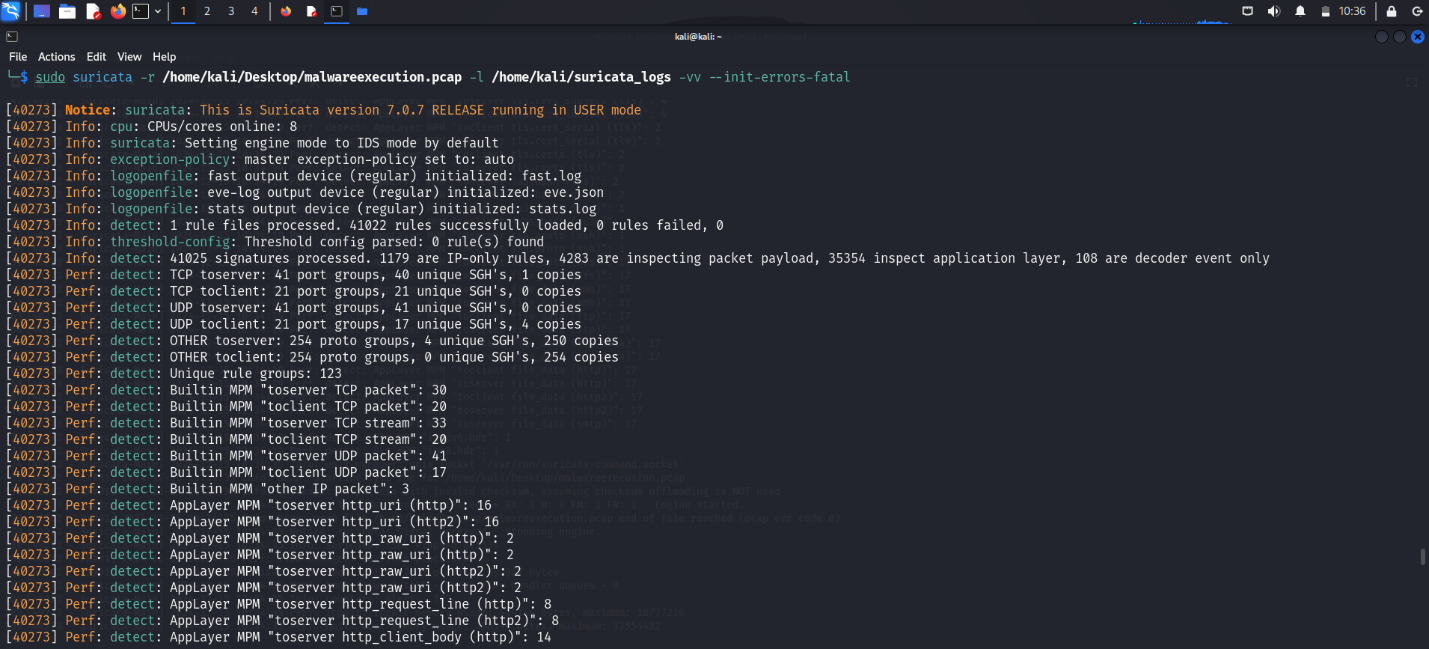
  
Running the malware and capturing the traffic:

**Suricata Downloading :**

A screenshot of a computer

Description automatically generated

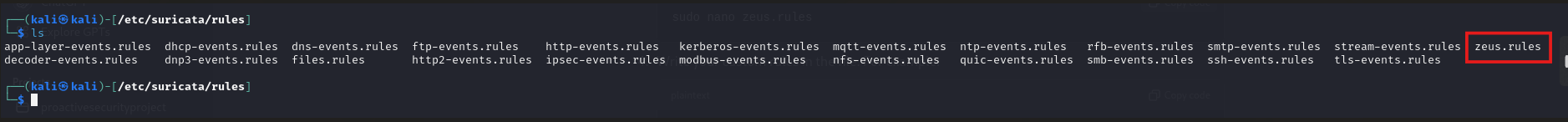
**Suricata Alerts detection:**

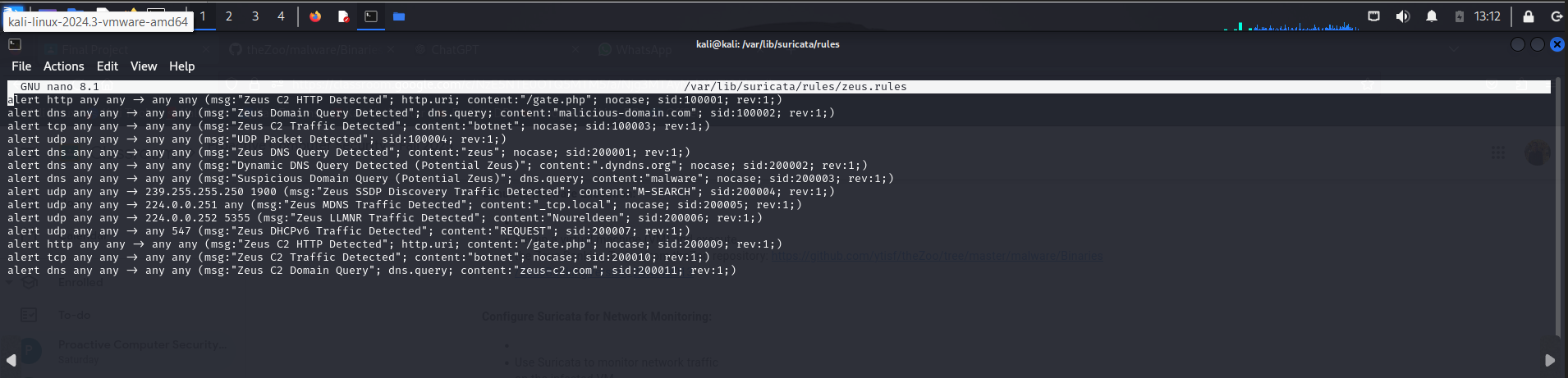


Here in this screen shot we passed the PCAP file to Suricata and it detected alerts

Using the default Suricata Rules

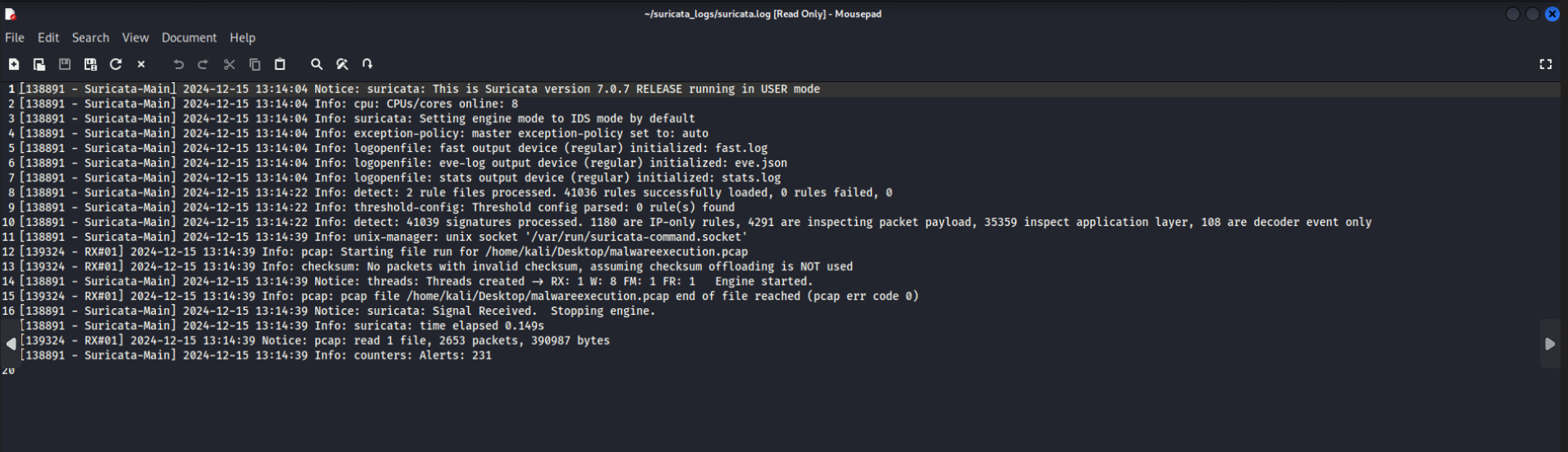
So we made custom rules to detect more and more alerts

**Firstly we made Rules files in the Suricata :**

**Then we added some Custom rules in it :**

**This rules mainly detect:**

* Zeus malware network communications, including its attempts to contact Command and Control (C2) servers.
* Botnet-related DNS queries.
* Abnormal or suspicious multicast and unicast traffic.
* General malware indicators in HTTP and DNS protocols.

After running with our custom rules we detected some new alerts:

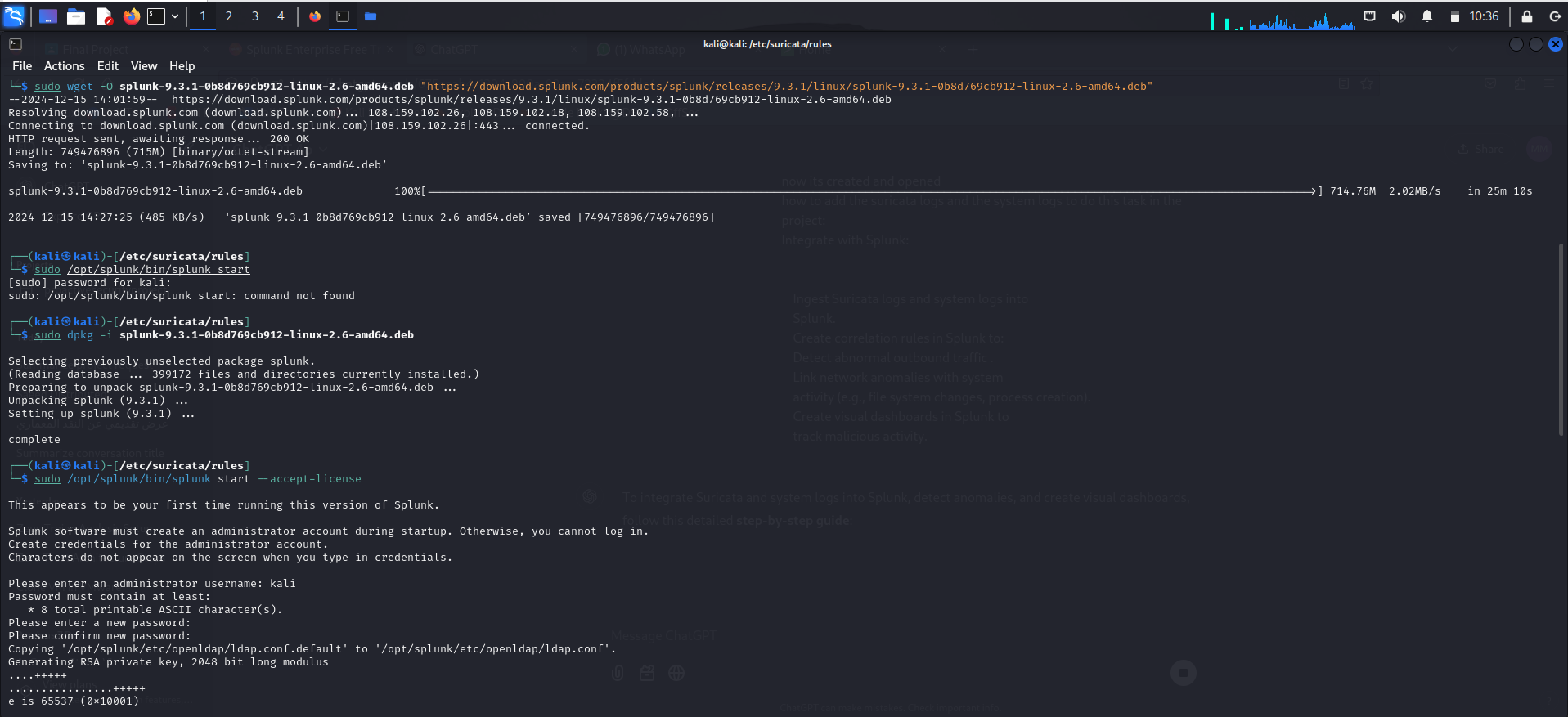
**Specifically 231 Alerts**

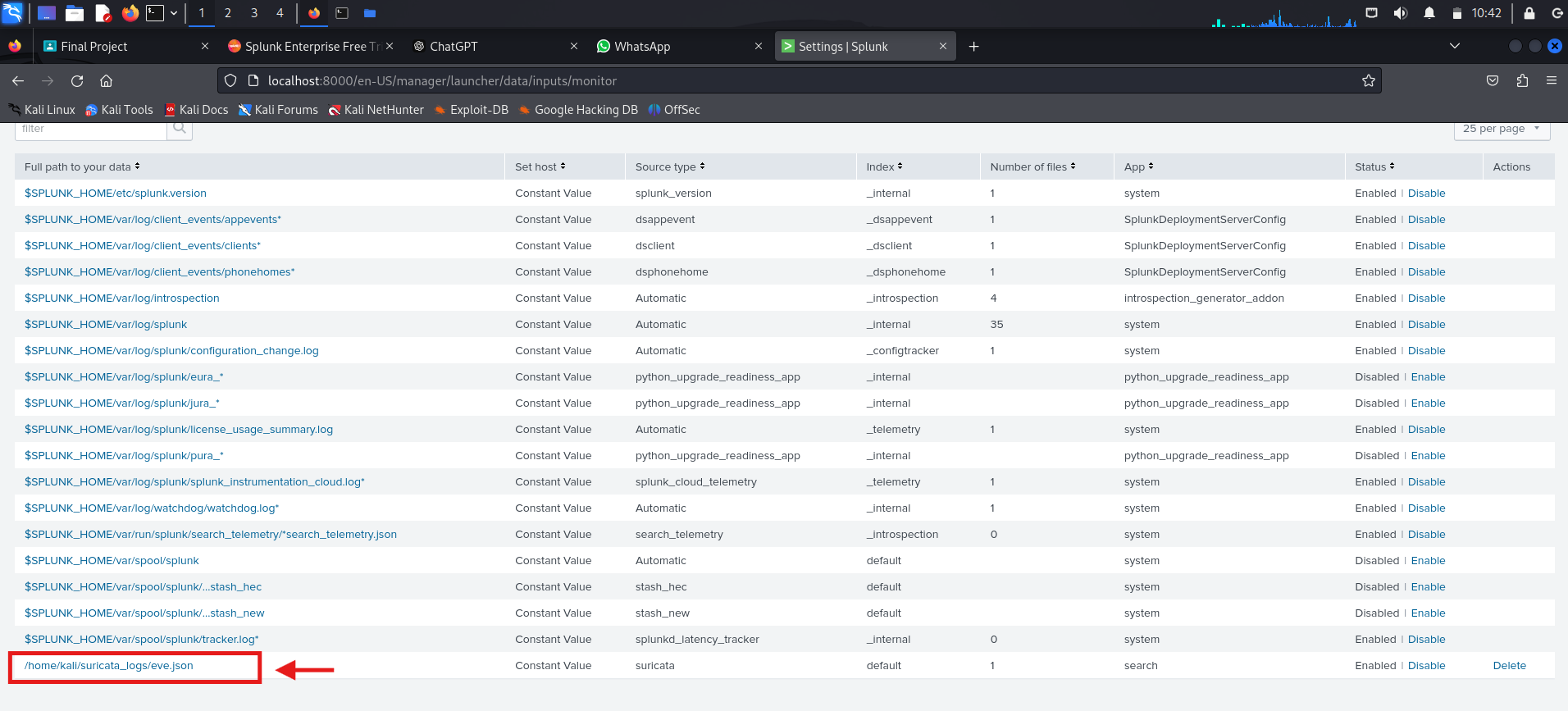
Once we finished Suricata detection it generated logs files that we gonna use into Splunk

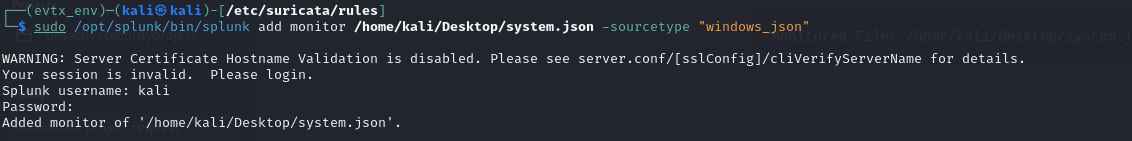
**A screenshot of a computer

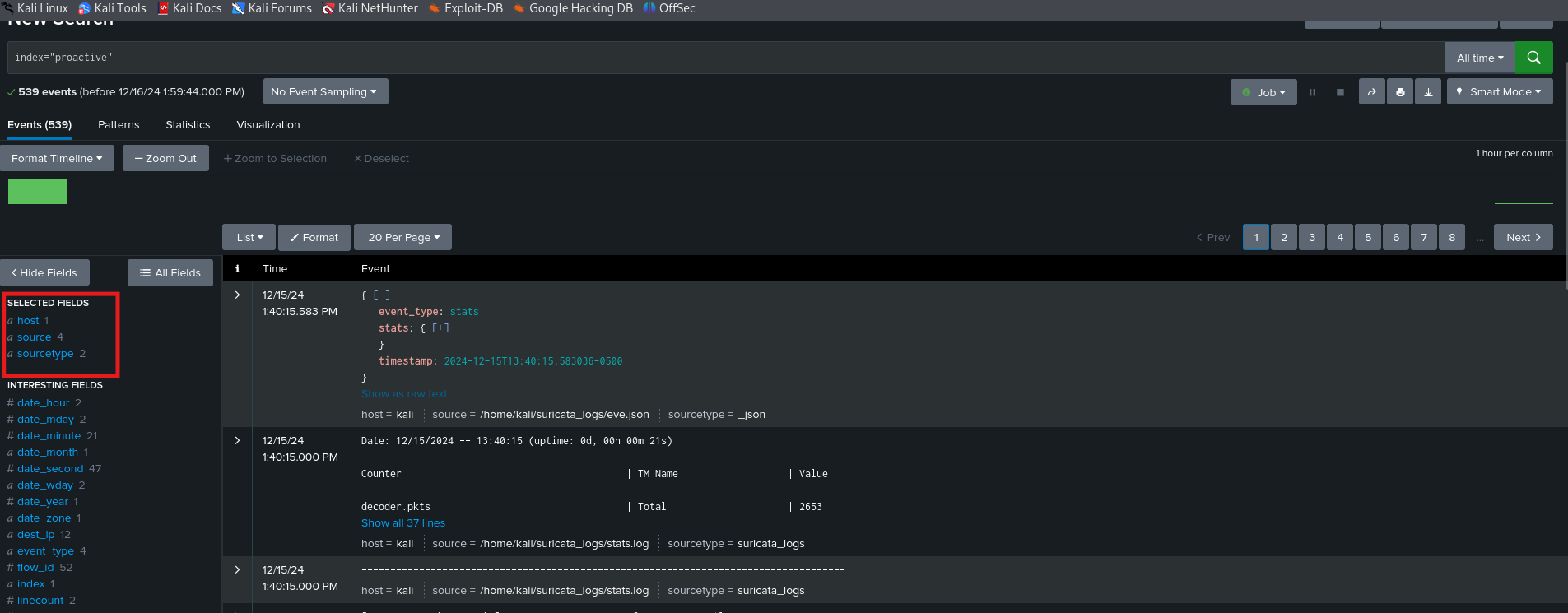
Description automatically generatedSuricata logs:**

Then we downloaded Splunk and ingested both Suricata logs and System logs   
We took windows System logs using Event viewer built in Application in Windows

Downloading Splunk:

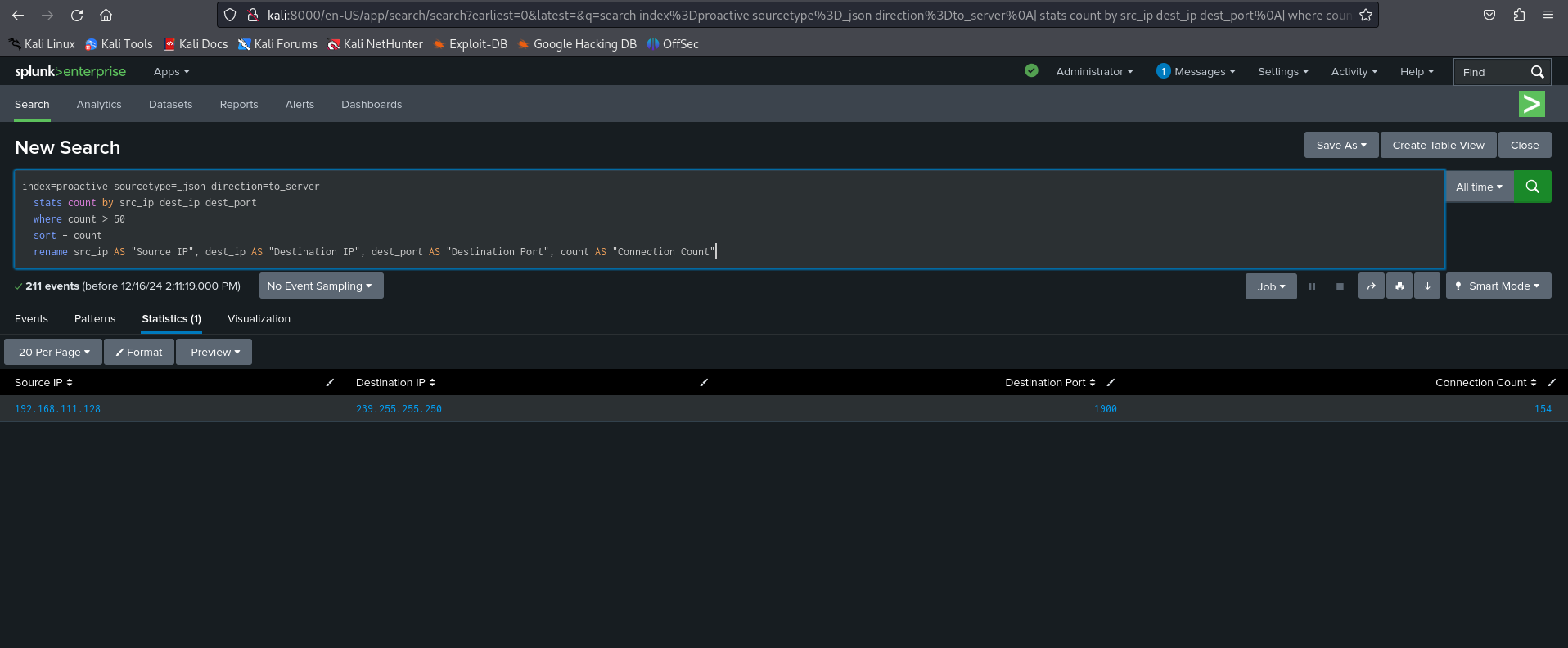
Adding Suricata logs in the Splunk:

Ingesting the System logs into Splunk

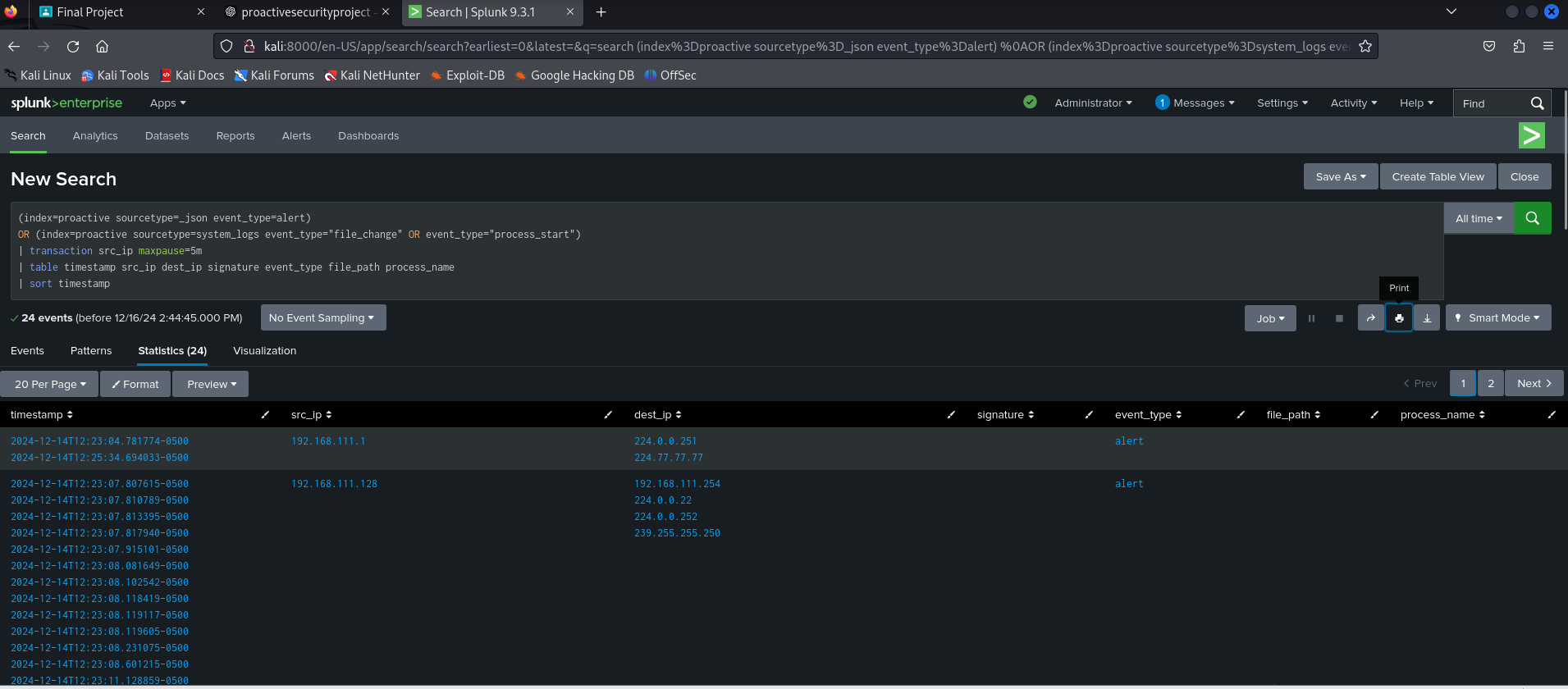
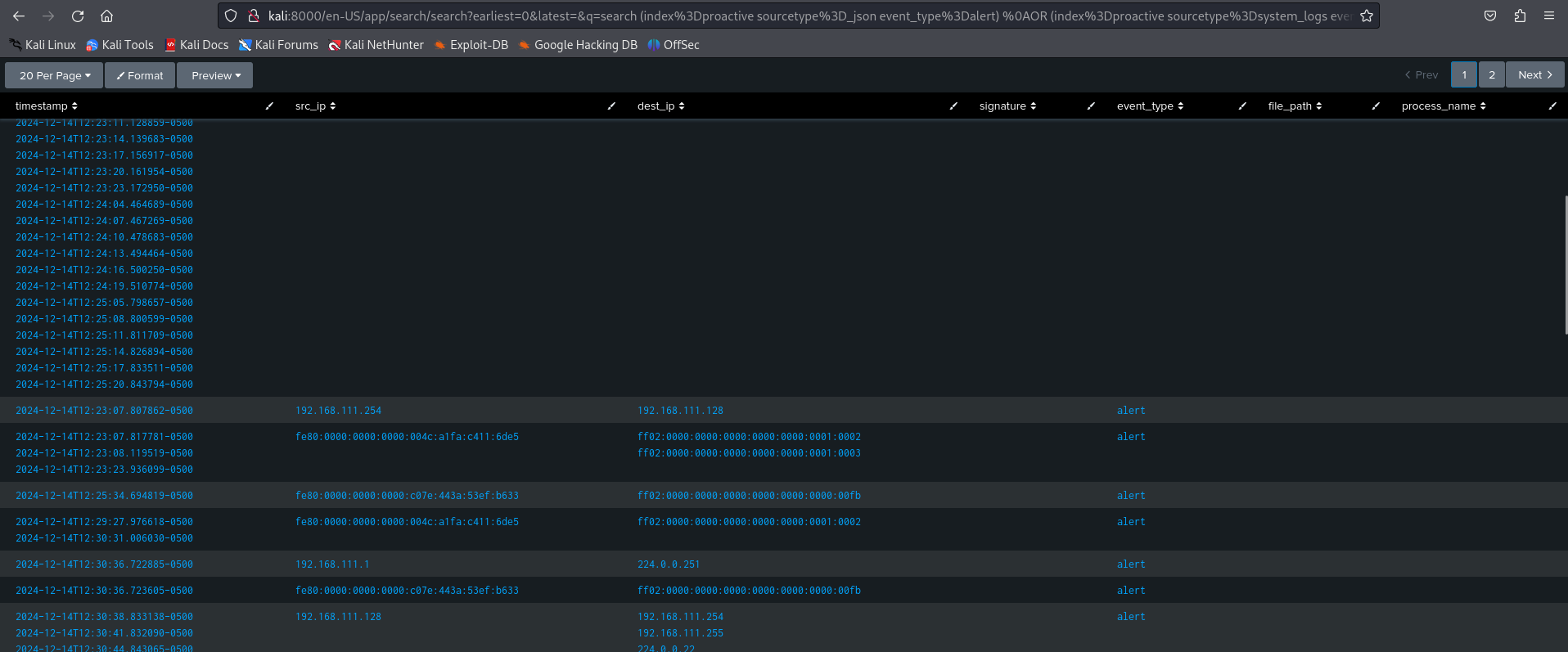
Ingested Suricata and System logs in the Splunk :

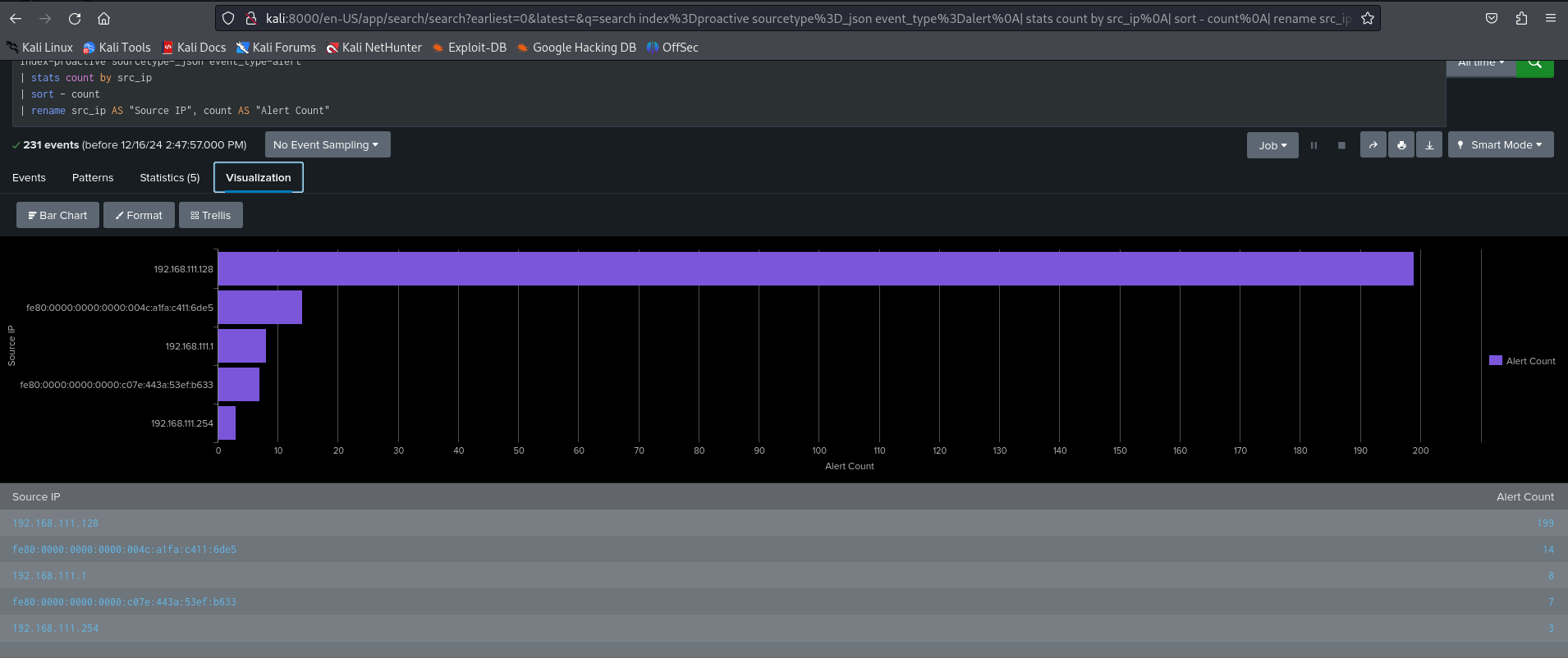
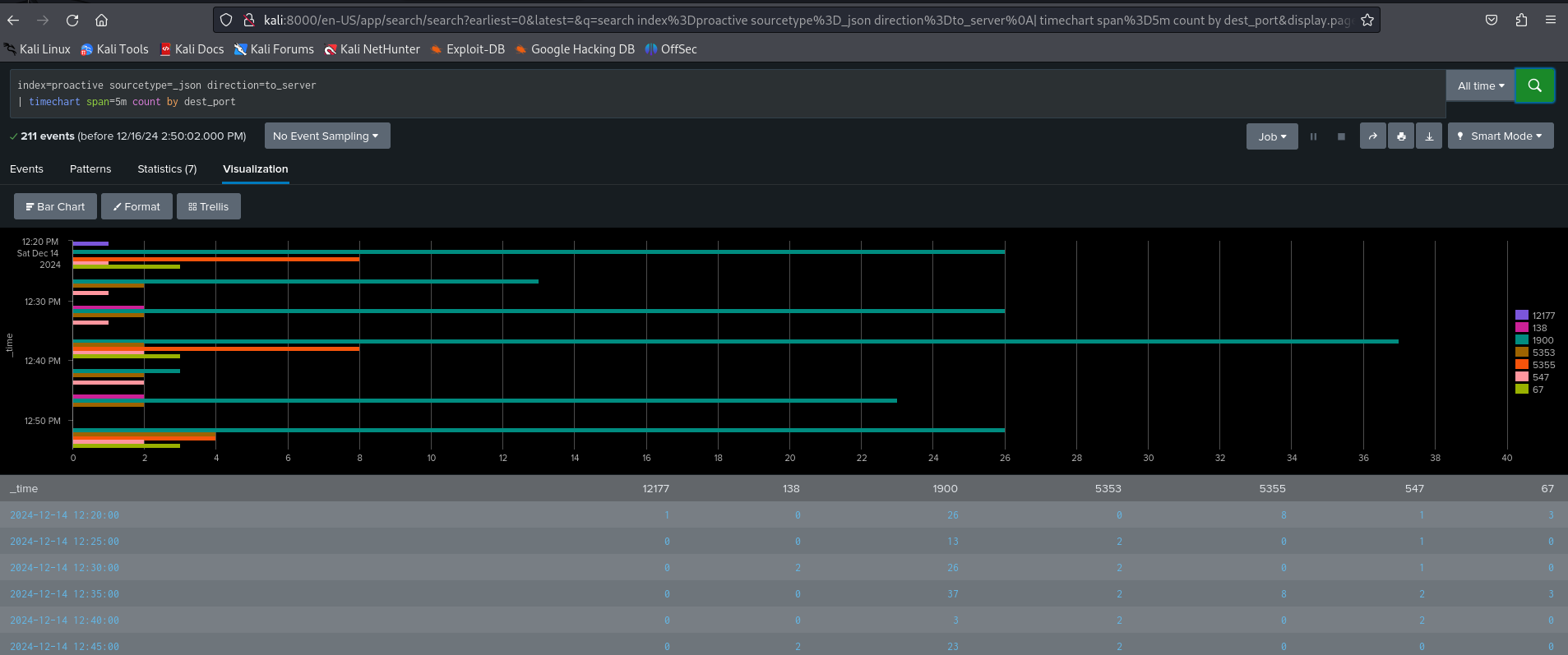
When we used the Splunk we detected abnormal Traffic activity from certain IPs

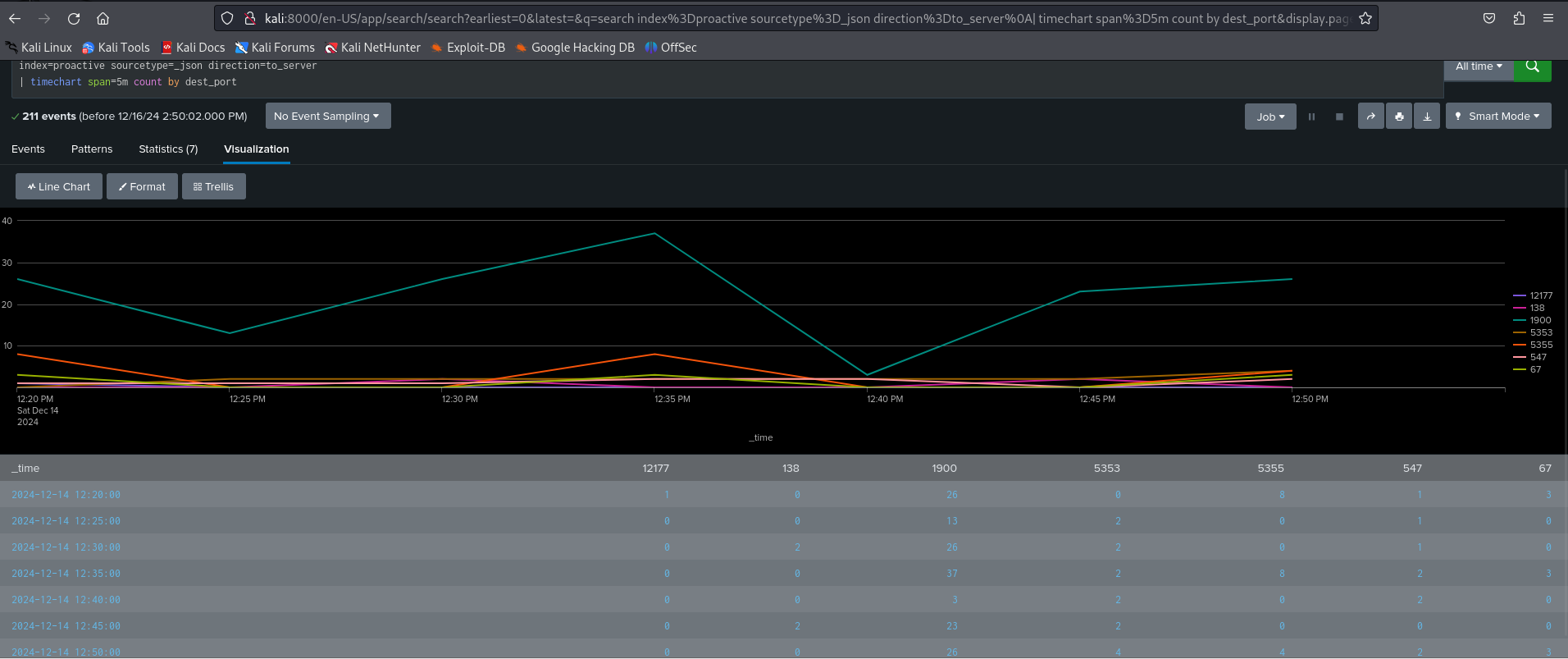
It was abnormal as the virtual machine were in host only network statues so it should not have all this network request from three or less IPs



There was one specific IP that sent 154 request in such a short time (3 minutes)

Then we correlated Suricata alert and the System logs using this search Query :  
  
And we detected a lot of Alerts and this was a sample of it

Here we made Dashboard to visualize the output to see a better investigation:



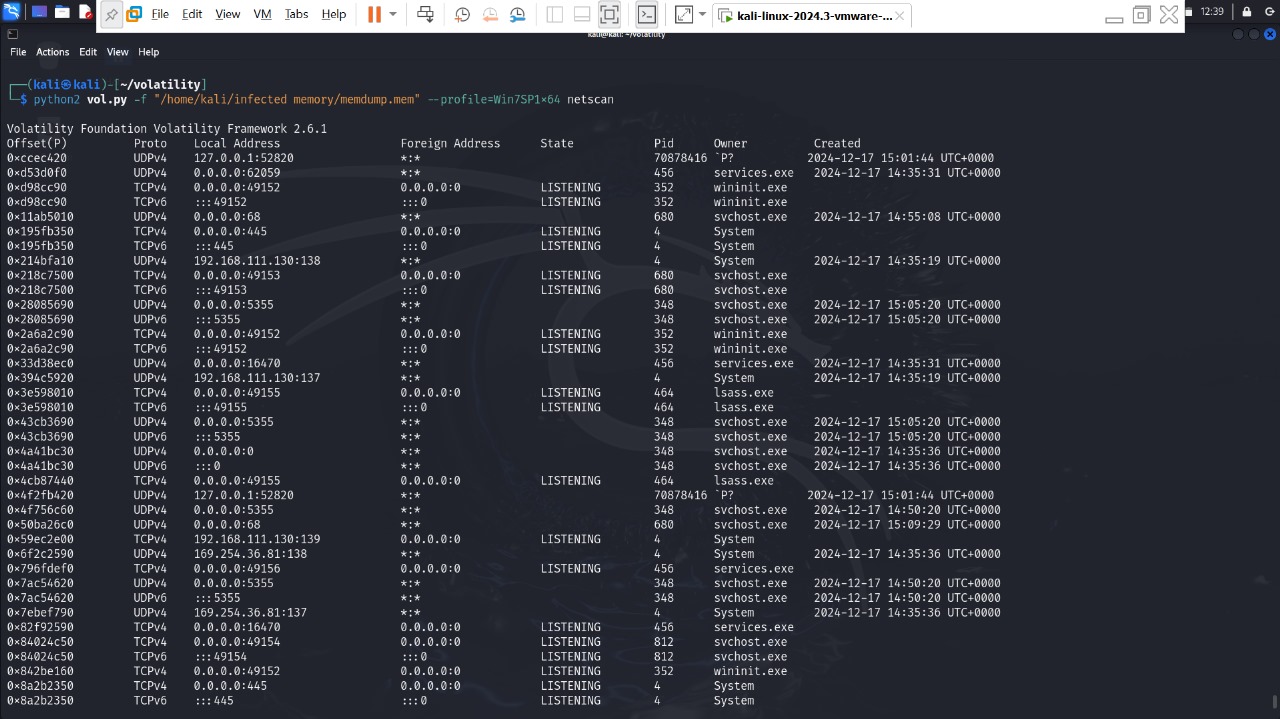
By this part we finished using Suricata and Splunk and moved into moving using Volatility and YARA tool (3rd and 4th Task)  
  
Firstly we captured the memory Dump on the infected windows 7 using FTK Imager and transferred it using a USB driver to our Kali Linux workstation to investigate it

Firstly, we showed the process list that was running after the execution of the maware using this commandA screenshot of a computer

Description automatically generated

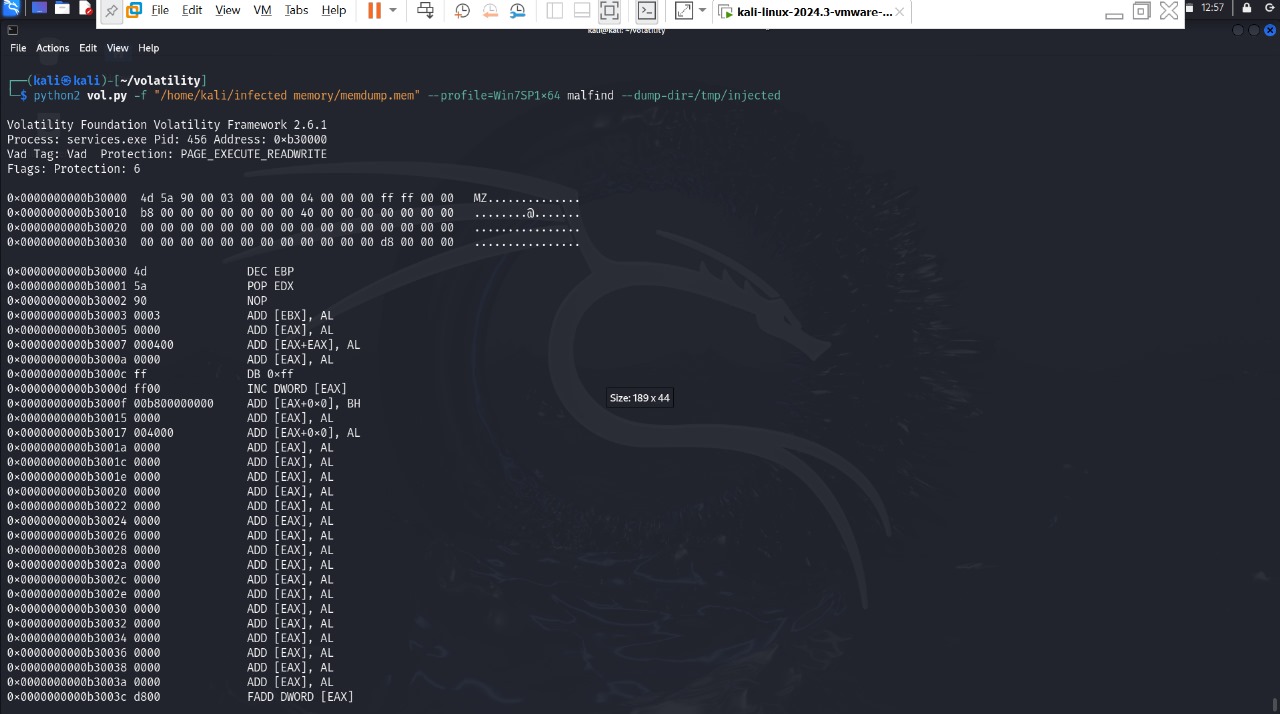
Then we used this command ” python2 vol.py -f "/home/kali/infected memory/memdump.mem" --profile=Win7SP1x64 netscan”

to identifies active network connections, including IP addresses, ports, and the associated processes. This can reveal Zeus-related network activity.

Dump process memory for analysis

Then we used this command ” python2 vol.py -f "/home/kali/infected memory/memdump.mem" --profile=Win7SP1x64 malfind --dump-dir=/tmp/injected”

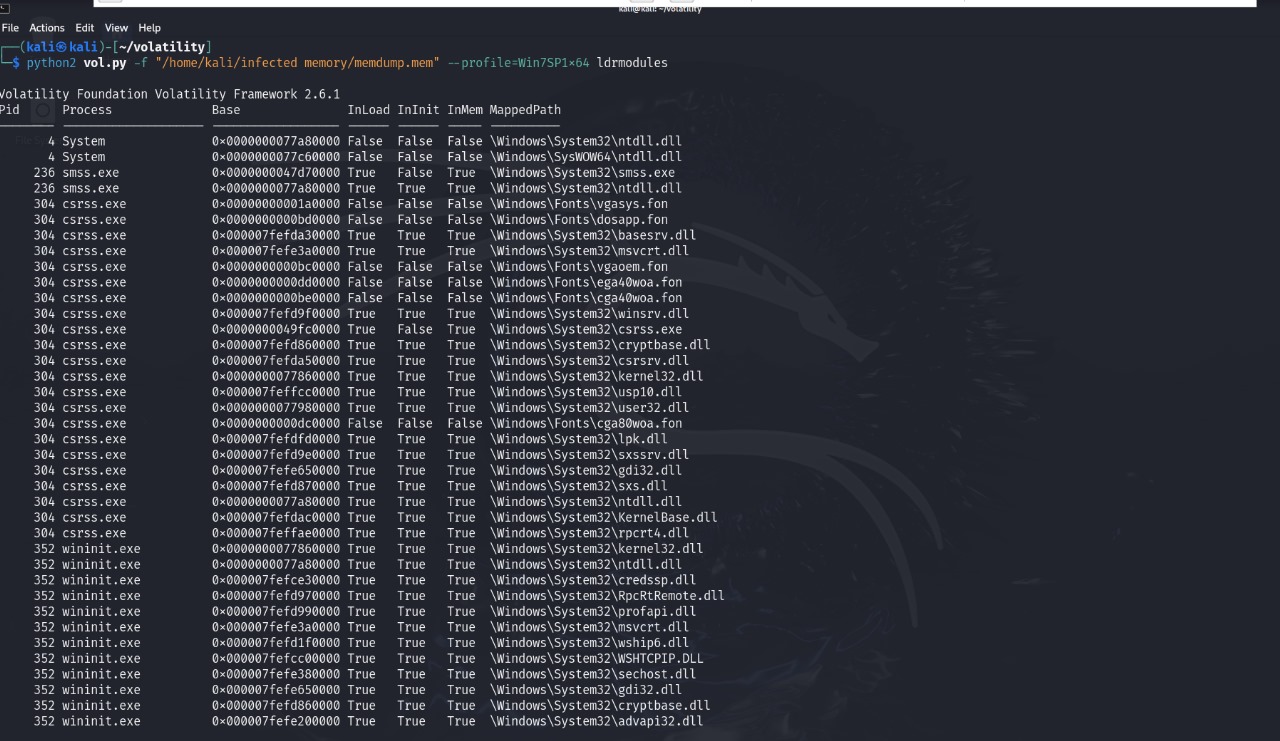
to identifies injected memory regions and dumps them for analysis. Malfind specifically targets injected code often used by malware like Zeus.

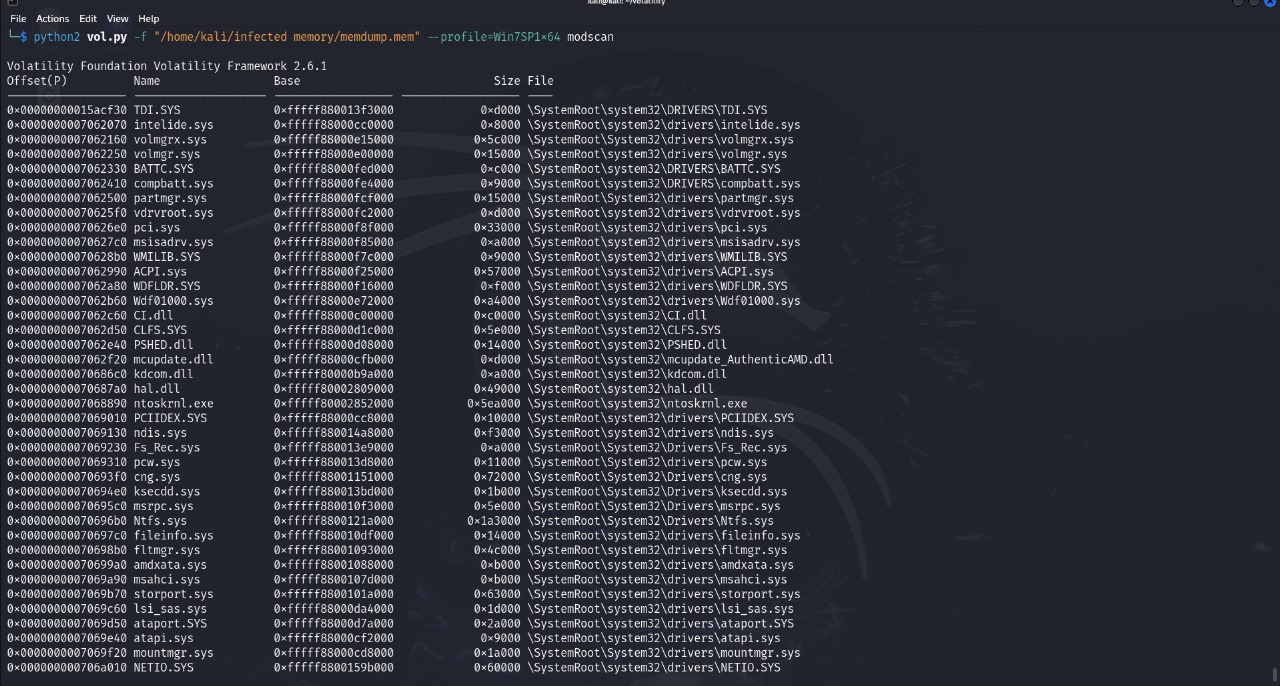
Analyze Zeus-related network connections

Then we used this command “ python2 vol.py -f "/home/kali/infected memory/memdump.mem" --profile=Win7SP1x64 procdump -p 456 --dump-dir=/tmp/dumps”

To dumps the memory of a specific process (PID 456) suspected to be related to Zeus for further analysis

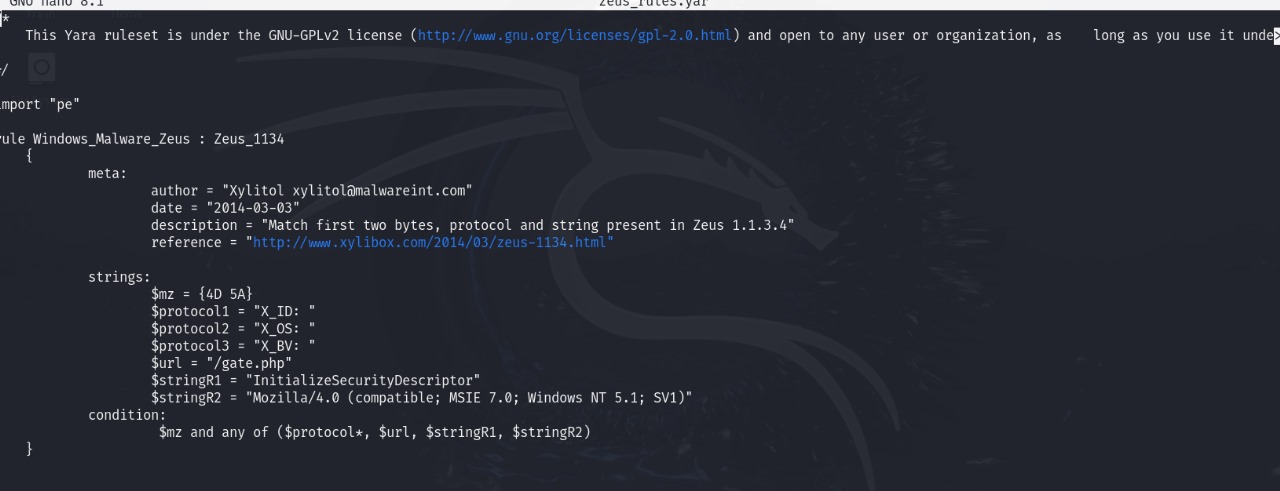
then the ldrmodules plugin from Volatility identifies DLL modules loaded into memory, highlighting discrepancies such as potential DLL injection or process hollowing. This is critical for detecting malware that manipulates legitimate processes.



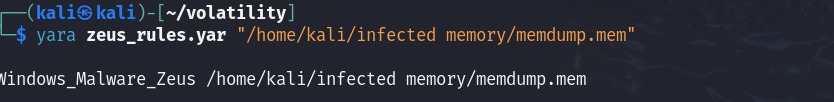
Then the modscan plugin detects kernel-mode drivers and modules in memory. It highlights potentially malicious drivers that could indicate rootkit activity or other kernel-level threats.

Now we went into Yara tool This YARA rule is customized to identify Zeus malware by looking for specific strings, headers, and protocols characteristic of Zeus. The "meta" section provides the rule's author and purpose, while the "strings" and "conditions" sections focus on matching Zeus-related markers in memory.

This rule is instrumental in detecting Zeus's binary signature in memory dumps, enabling forensic identification of compromised systems.



This shows a YARA scan using the Zeus rule on a memory dump file. The rule successfully identifies Zeus-related artifacts in memory, flagging the malware's presence.

Insights: This provides direct evidence of Zeus infection and can guide further remediation efforts.

Thank you for reading till this far