Proactive Security

Zeus Trojan

Department: Cybersecurity

Presented by:

Walaa Ahmed (2206192)

Youstina Malak (2206199)

Seif Ahmed (2206215)

Ahmed Adel (2206194)

Introduction:

This report demonstrates the process of detecting the Zeus malware using various tools, including Suricata, Splunk, Volatility, and YARA. Each step is elaborated to explain its purpose, execution, and expected alerts.

Step 1: Setting Up the Environment

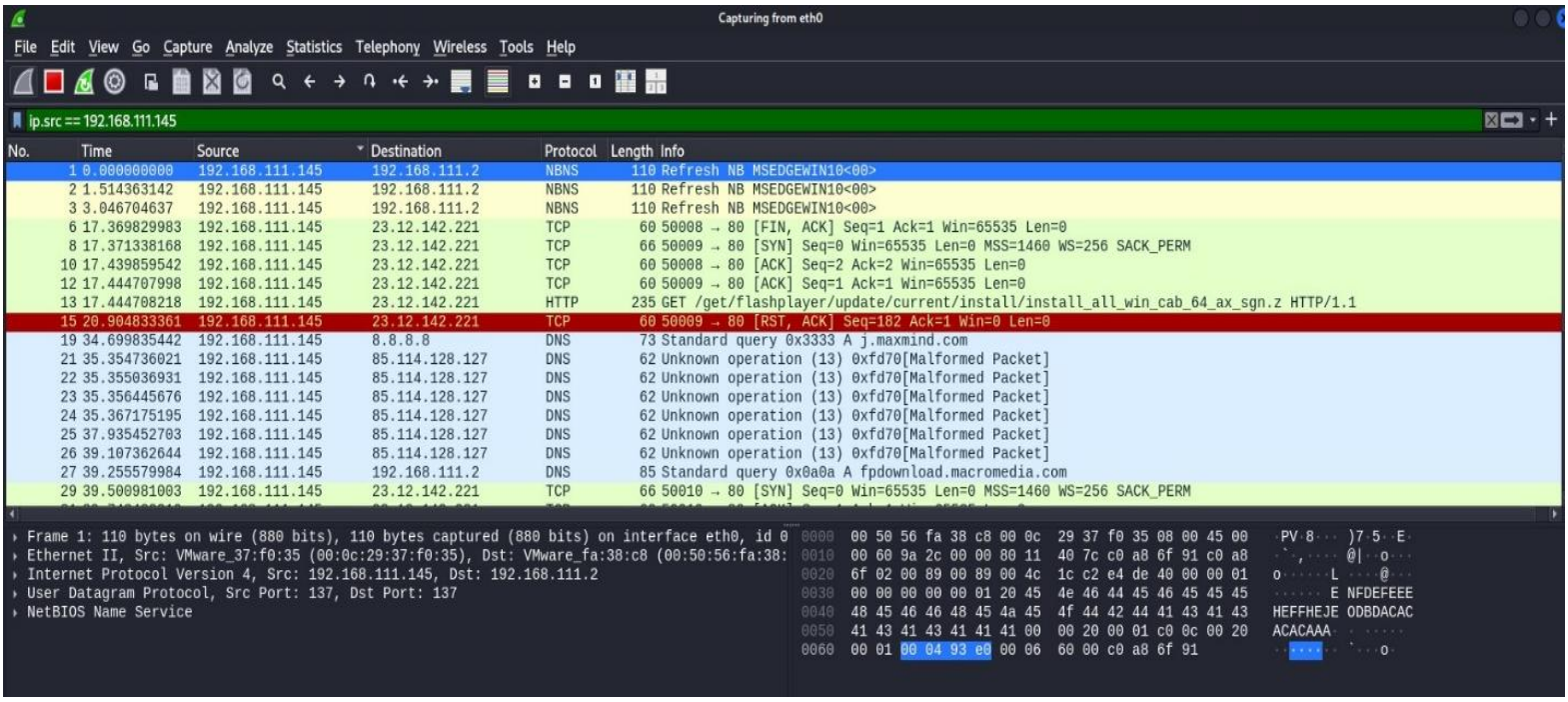
1. **Kali Linux Machine**:
    - Install **Suricata** and **Wireshark** on the Kali Linux machine to analyze network traffic and detect malicious activities.
2. **Windows 10 Machine**:
    - Install the Zeus malware's zip file
    - Disable **Windows Defender**, **antivirus software**, and the **firewall** to prevent the automatic removal of the malicious file.
3. **Network Configuration**:
    - Both machines must be on the same **NAT network** (192.168.111.0), ensuring they can communicate without external interference.

Purpose: This setup creates a secure, controlled environment to observe and analyze malware behavior without risking external contamination.

Step 2: Analyzing Traffic Using Wireshark

1. **IP Address Identification**:
    - Retrieve the IP address of the Windows machine to focus network
    - analysis on its traffic using ipconfig in CMD.

2. **Traffic Capture**:
    - Start live traffic capture on the infected VM's by selecting the appropriate network interface.
    - Apply a filter using the Windows machine's IP address to isolate its traffic.



3. **Analyzing Captured Traffic**:

Observations:

1. **Destination IP**: 85.114.128.127 (This IP has been flagged as malicious in threat intelligence databases, commonly linked to C2 activities or malware distribution)

2. **Protocol**:

- DNS: The infected machine likely resolves the domain associated with this IP.
- TCP: A persistent TCP connection is observed, indicating active communication between the infected machine and the C2 server.

3. **HTTP:** Request with GET method and /get/flashplayer/update/current/install_all_win_cab_64_ax_sgn.z url.
    1. This HTTP request appears to download a malicious file, disguised as a legitimate Flash Player update wich a common tactic used by Zeus malware to deliver updates or additional payloads to the infected machine.

4. **Indicators of Compromise (IOCs):** Malicious IP(85.114.128.127) Malicious Domain(Resolved by DNS traffic) URI Pattern (/get/flashplayer/update/current/install_all_win_cab_64_ax_sgn.z)

Step 3: Malware Detection Using Suricata

1. **Custom Rule Creation**:
    - Create a custom Suricata rule to detect the suspicious tcp and Dns requests or other malicious traffic patterns observed in Wireshark:

    - Save the rule in /etc/suricata/rules/zeus2.

Explanation of Rules:

1. **HTTP Rule**:
    - Triggers an alert when the specified GET request is observed by Matching the specific URI string.
2. **DNS Rule**:
    - Flags DNS queries to known malicious domains by Inspecting DNS queries for malicious indicators that Matches against a specific domain string

Purpose: These rules provide real-time alerts, enabling analysts to identify and act on Zeus-related activity.

2. **Suricata Configuration**:
    - Edit the suricata.yaml configuration file to:

- Define the home network as the Windows VM's IP address.

```
vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.111.145]"
    #HOME_NET: "[192.168.111.145]"
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
    #EXTERNAL_NET: "any"

    HTTP_SERVERS: "$HOME_NET"
    SMTP_SERVERS: "$HOME_NET"
    SQL_SERVERS: "$HOME_NET"
    DNS_SERVERS: "$HOME_NET"
    TELNET_SERVERS: "$HOME_NET"
    AIM_SERVERS: "$EXTERNAL_NET"
    DC_SERVERS: "$HOME_NET"
    DNP3_SERVER: "$HOME_NET"
    DNP3_CLIENT: "$HOME_NET"
    MODBUS_CLIENT: "$HOME_NET"
```

- Set the monitored interface to eth0.

```
# Linux high speed capture support
af-packet:
  - interface: eth0
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 106
    # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per hash.
    # This is only supported for Linux kernel > 3.1
    # possible value are:
    #  * cluster_flow: all packets of a given flow are sent to the same socket
    #  * cluster_cpu: all packets treated in kernel by a CPU are sent to the same socket
    #  * cluster_qm: all packets linked by network card to a RSS queue are sent to the same
    #  socket. Requires at least Linux 3.14.
    #  * cluster_ebpf: eBPF file load balancing. See doc/userguide/capture-hardware/ebpf-xdp.rst for
    #  more info.
    # Recommended modes are cluster_flow on most boxes and cluster_cpu or cluster_qm on system
    # with capture card using RSS (requires cpu affinity tuning and system IRQ tuning)
```

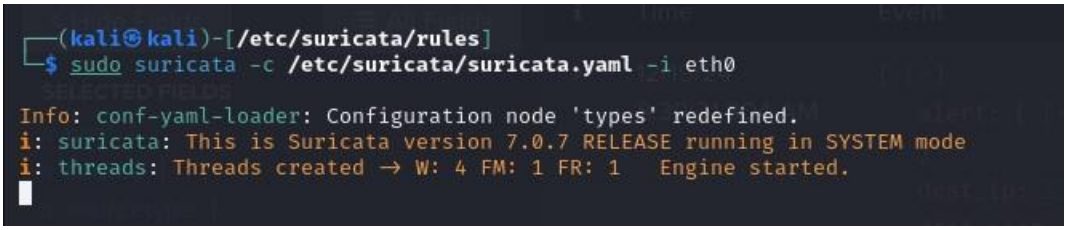
- Link the custom rule file in the rules-file section.

```
default-rule-path: /home/kali/Downloads/zeus

rule-files:
#   - /var/lib/suricata/rules/suricata.rules
    - /etc/suricata/rules/zeus2
```

- Ensure there are no syntax errors in the configuration.

```
┌──(kali㉿kali)-[~]
└─$ sudo suricata -T -c /etc/suricata/suricata.yaml

Info: conf-yaml-loader: Configuration node 'types' redefined.
i: suricata: This is Suricata version 7.0.7 RELEASE running in SYSTEM mode
i: suricata: Configuration provided was successfully loaded. Exiting.
```

3. Run Suricata on the specified interface

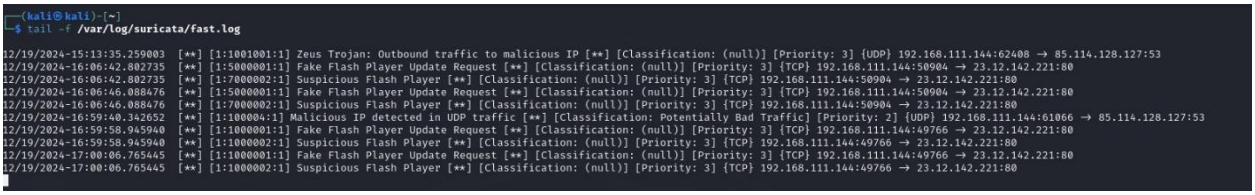


4. **Running the Malware**:

   - Extract and execute the Zeus zip file on the Windows VM. Suricata should now detect the malware's activity.

5. **Log Analysis**:

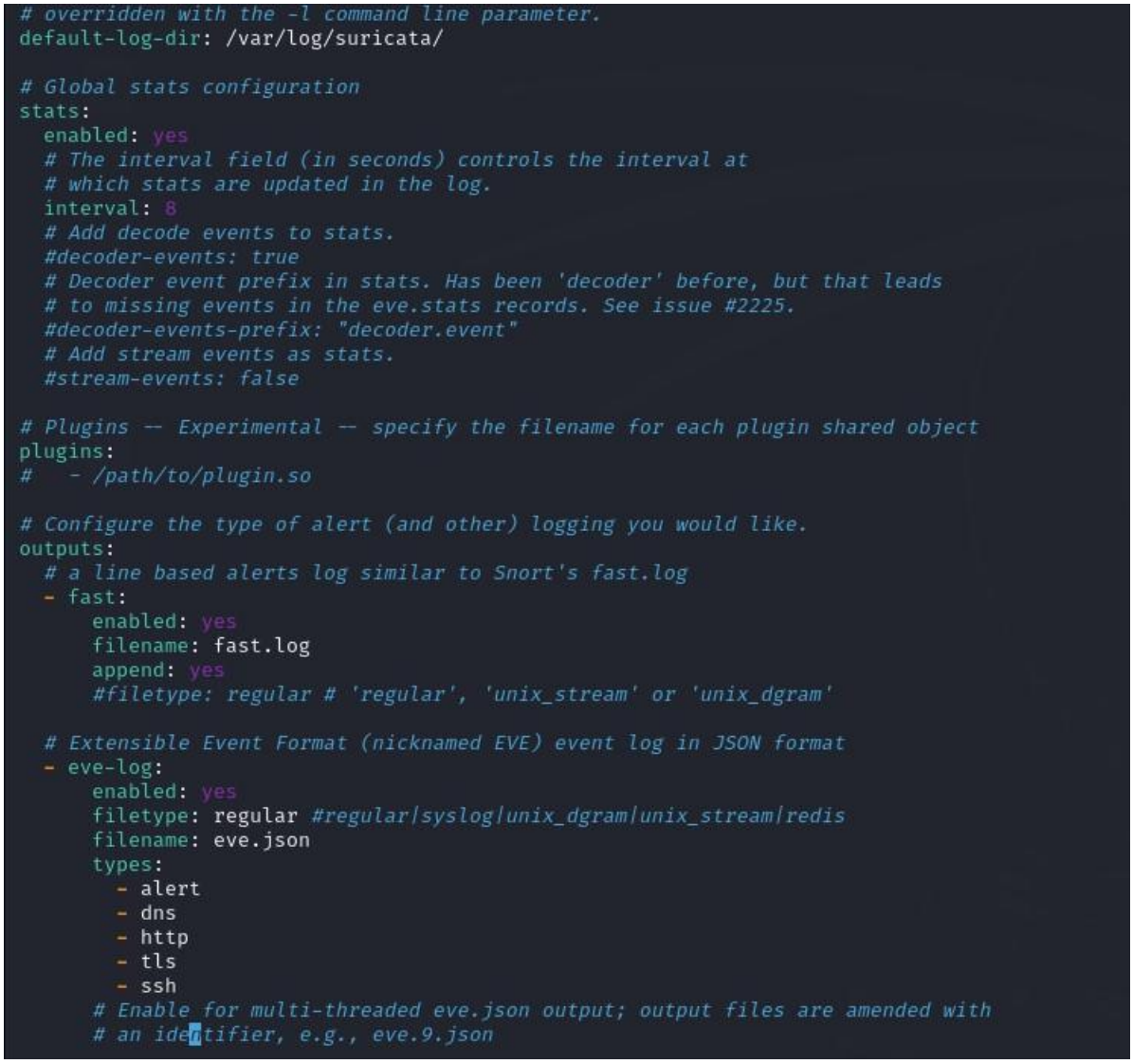
   - Check the fast.log file for alerts generated by Suricata:



Step 4: Forwarding logs :

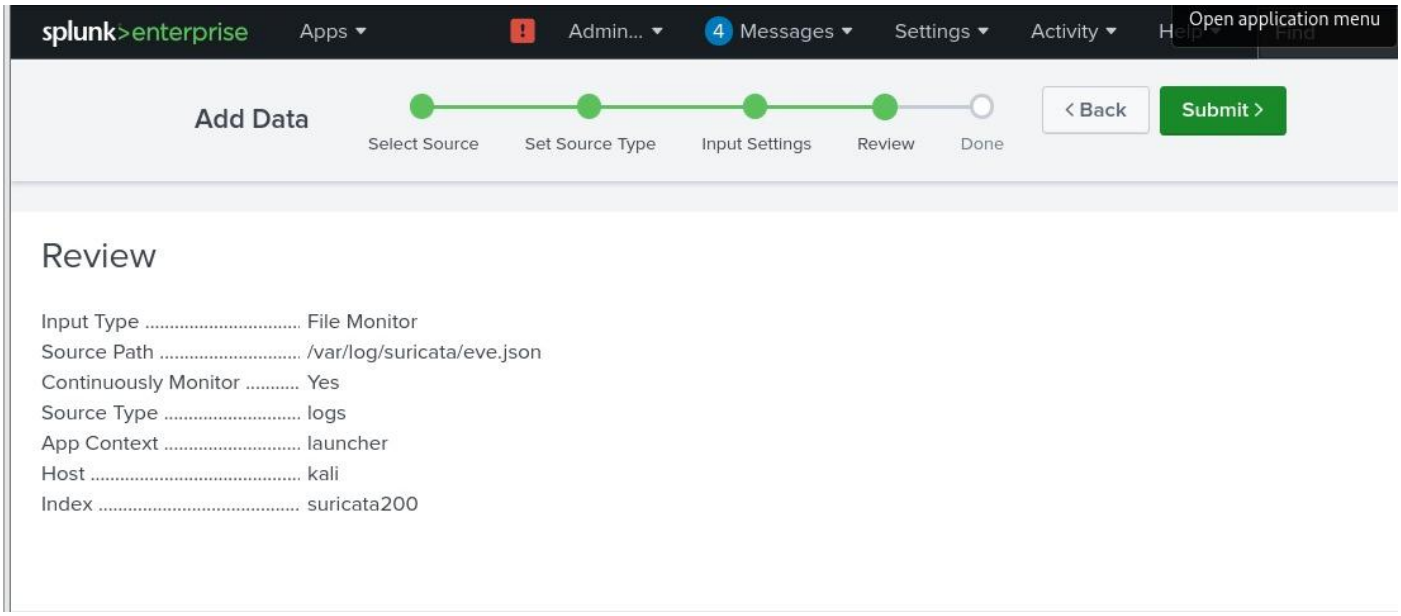
A) Configure suricata.yaml

- The suricata.yaml configuration file was modified to enable log forwarding to both eve.json and fast.log. The eve.json file provides detailed JSON-formatted logs, while fast.log delivers concise alert summaries



```
# overridden with the -l command line parameter.
default-log-dir: /var/log/suricata/

# Global stats configuration
stats:
  enabled: yes
  # The interval field (in seconds) controls the interval at
  # which stats are updated in the log.
  interval: 8
  # Add decode events to stats.
  #decoder-events: true
  # Decoder event prefix in stats. Has been 'decoder' before, but that leads
  # to missing events in the eve.stats records. See issue #2225.
  #decoder-events-prefix: "decoder.event"
  # Add stream events as stats.
  #stream-events: false

# Plugins -- Experimental -- specify the filename for each plugin shared object
plugins:
#  - /path/to/plugin.so

# Configure the type of alert (and other) logging you would like.
outputs:
  # a line based alerts log similar to Snort's fast.log
  - fast:
      enabled: yes
      filename: fast.log
      append: yes
      #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'

  # Extensible Event Format (nicknamed EVE) event log in JSON format
  - eve-log:
      enabled: yes
      filetype: regular #regular|syslog|unix_dgram|unix_stream|redis
      filename: eve.json
      types:
        - alert
        - dns
        - http
        - tls
        - ssh
      # Enable for multi-threaded eve.json output; output files are amended with
      # an identifier, e.g., eve.9.json
```

- These logs are subsequently forwarded to the Splunk instance for centralized monitoring and analysis. This setup ensures comprehensive visibility into network and security events.



B) Setting Up the Splunk Universal Forwarder

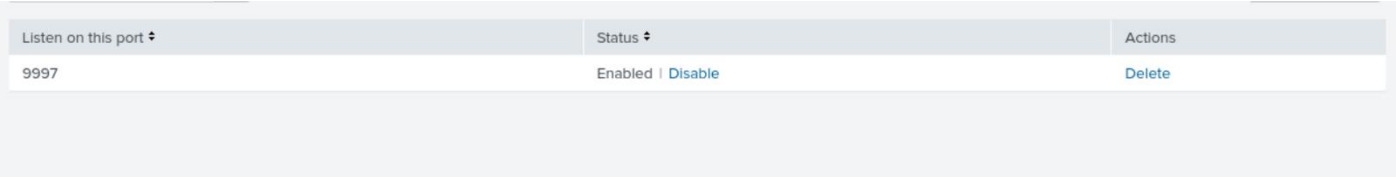
- The outputs.conf file was updated to direct logs to the Splunk monitoring server on the Kali machine with IP
- The inputs.conf file was configured to collect Windows Event Logs, specifically for the **Application**, **Security**, and **System** event sources:

```
[WinEventLog://Application]

disabled = 0

[WinEventLog://Security]

disabled = 0

[WinEventLog://System]

disabled = 0
```

- The configuration files are located at:
  C:\Program Files\SplunkUniversalForwarder\etc\system\local

- The Splunk instance on Kali was configured to listen for incoming data on port 9997.

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow 9997/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)

┌──(kali㉿kali)-[~]
└─$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
9997/tcp                   ALLOW IN    Anywhere
9997/tcp (v6)              ALLOW IN    Anywhere (v6)
```

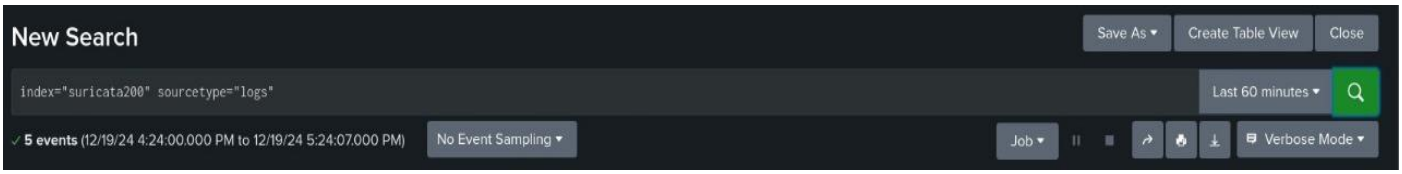


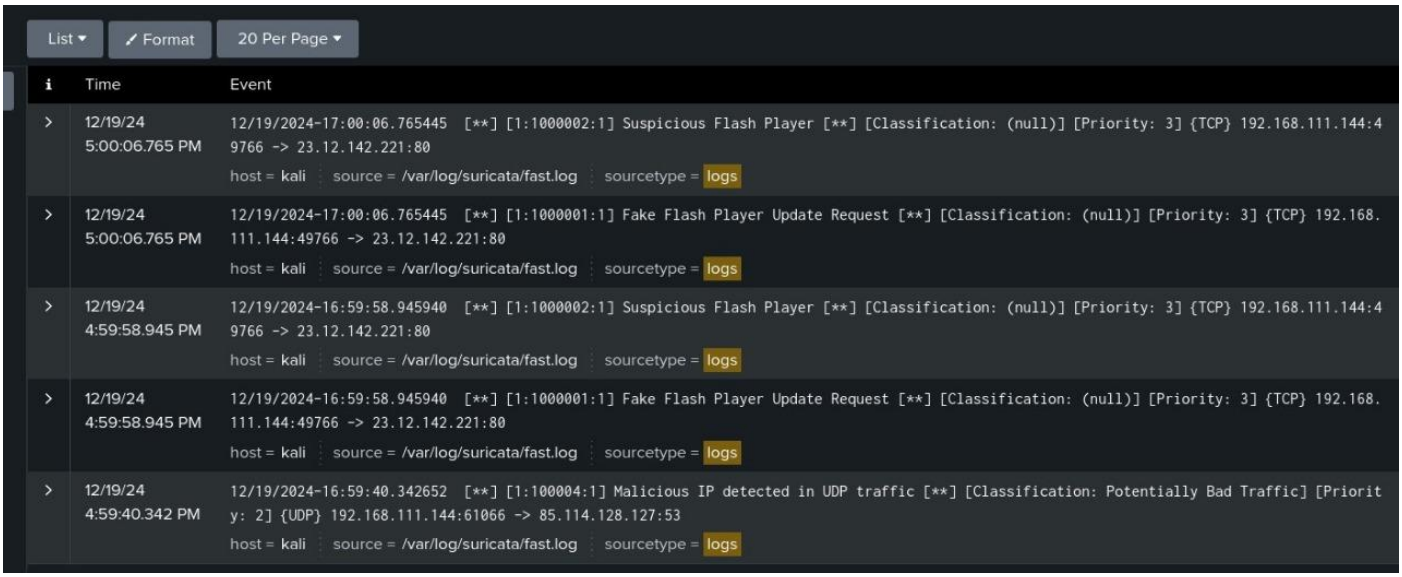
- The logs from the Windows machine successfully appeared in Splunk with sourcetype identifiers such as sourcetype="WinEventLog:System", confirming proper data ingestion.

Step 5: Monitoring and Observing Logs in Splunk:

A) Observing Suricata Alerts

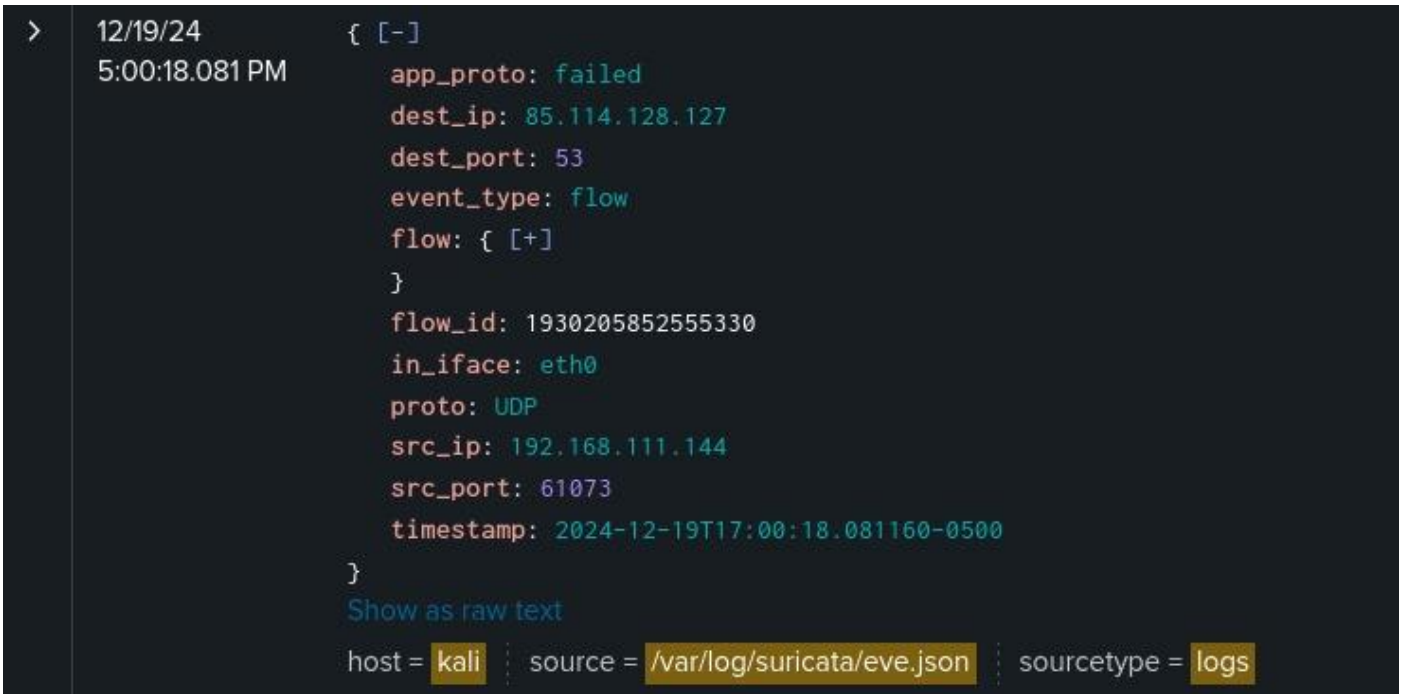
- Analysis of Suricata alerts revealed two suspicious IP addresses originating from the victim machine. These alerts were extracted using custom Splunk searches and visualizations.

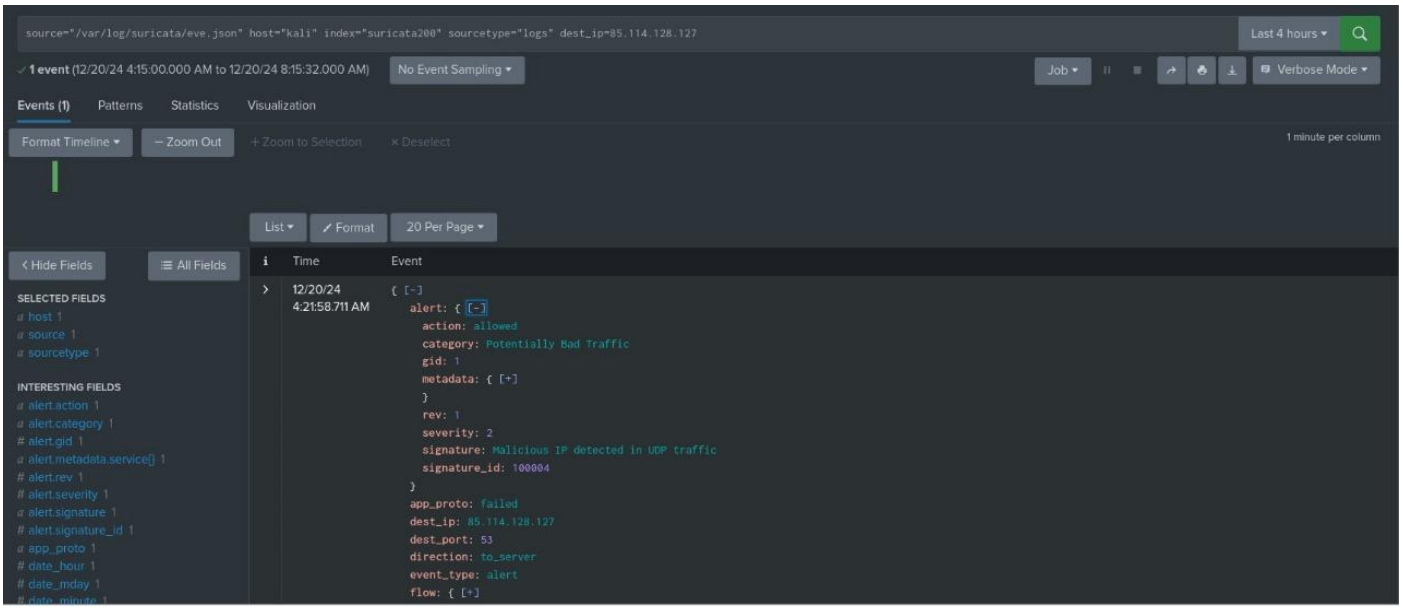


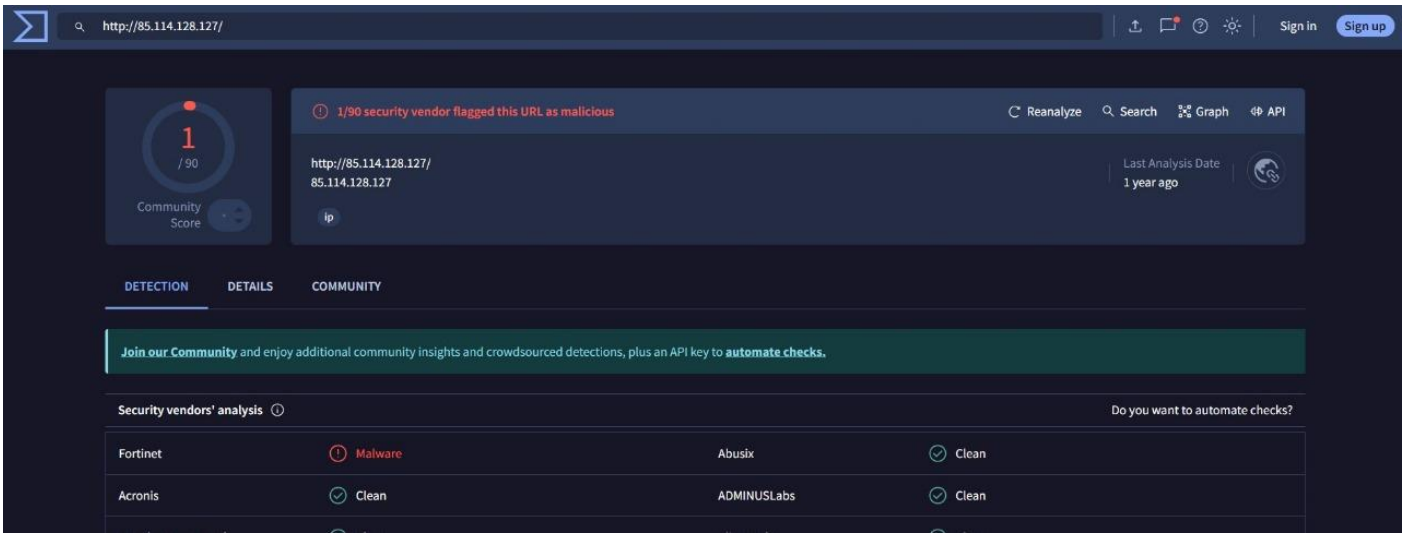


B) Investigating Suspicious IP Addresses

- The two identified IPs were queried in VirusTotal, revealing that one of them (85.114.128.127) is associated with DNS traffic on port 53. This activity strongly indicates that the Zeus Trojan on the victim machine is communicating with a malicious DNS server, indicative of **command-and-control (C2)** communicat

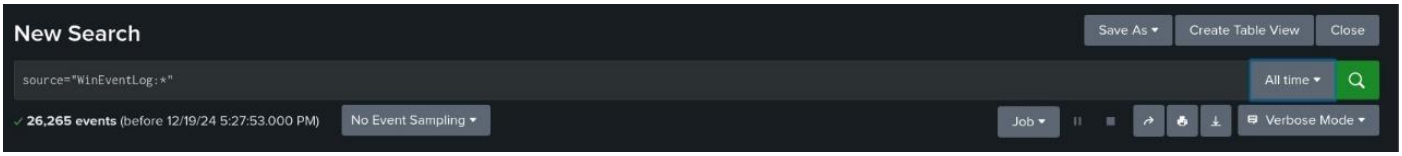

12/19/24
5:00:18.081 PM
{ [-]
app_proto: failed
dest_ip: 85.114.128.127
dest_port: 53
event_type: flow
flow: { [+]
}
flow_id: 1930205852555330
in_iface: eth0
proto: UDP
src_ip: 192.168.111.144
src_port: 61073
timestamp: 2024-12-19T17:00:18.081160-0500
}
Show as raw text
host = kali
source = /var/log/suricata/eve.json
sourcetype = logs

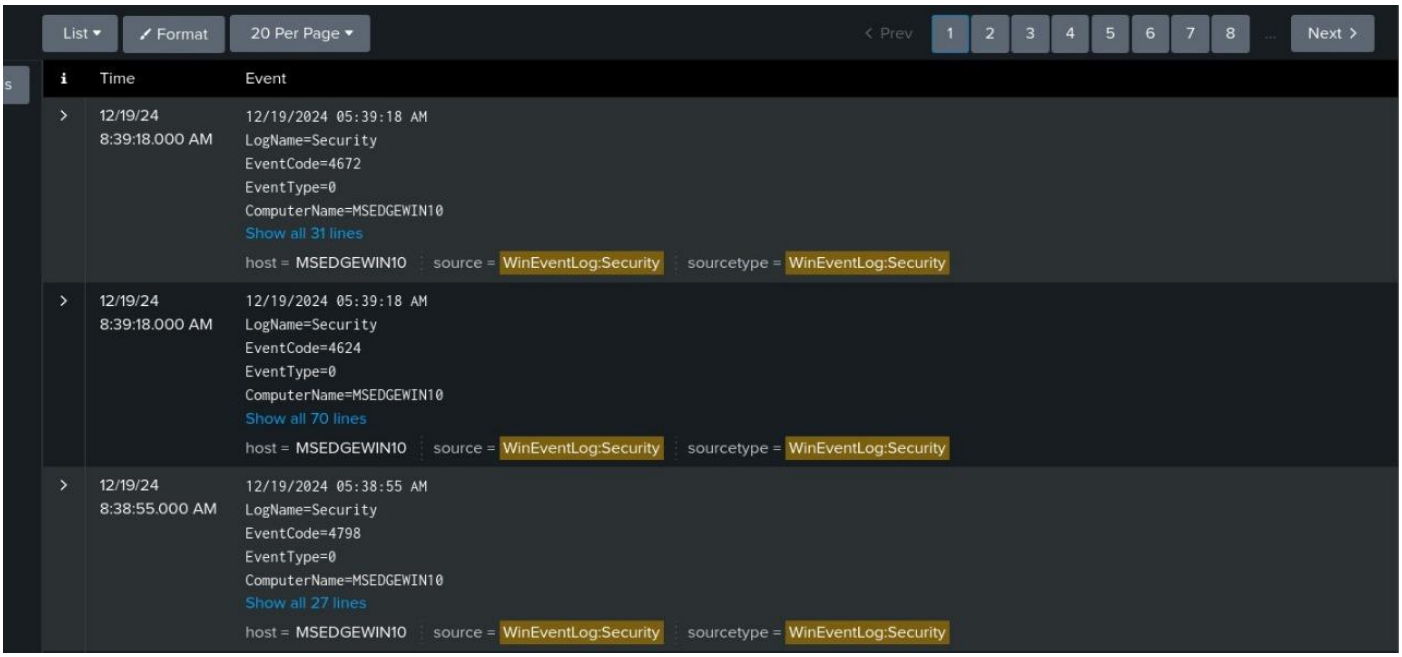

source="/var/log/suricata/eve.json" host="kali" index="suricata200" sourcetype="logs" dest_ip=85.114.128.127
Last 4 hours
1 event (12/20/24 4:15:00.000 AM to 12/20/24 8:15:32.000 AM)
No Event Sampling
Job
Verbose Mode
Events (1)
Patterns
Statistics
Visualization
Format Timeline
Zoom Out
Zoom to Selection
Deselect
1 minute per column
List
Format
20 Per Page
Hide Fields
All Fields
Time
Event
12/20/24
4:21:58.711 AM
{ [-]
alert: { [-]
action: allowed
category: Potentially Bad Traffic
gid: 1
metadata: { [+]
}
rev: 1
severity: 2
signature: Malicious IP detected in UDP traffic
signature_id: 100004
}
app_proto: failed
dest_ip: 85.114.128.127
dest_port: 53
direction: to_server
event_type: alert
flow: { [+]
SELECTED FIELDS
host 1
source 1
sourcetype 1
INTERESTING FIELDS
alert.action 1
alert.category 1
alert.gid 1
alert.metadata.service{} 1
alert.rev 1
alert.severity 1
alert.signature 1
alert.signature_id 1
app_proto 1
date_hour 1
date_mday 1


http://85.114.128.127/
Sign in
Sign up
1
/ 90
Community Score
1/90 security vendor flagged this URL as malicious
Reanalyze
Search
Graph
API
http://85.114.128.127/
85.114.128.127
ip
Last Analysis Date
1 year ago
DETECTION
DETAILS
COMMUNITY
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.
Security vendors' analysis
Do you want to automate checks?
Fortinet
Malware
Abusix
Clean
Acronis
Clean
ADMINUSLabs
Clean

C) Monitoring Windows Event Logs

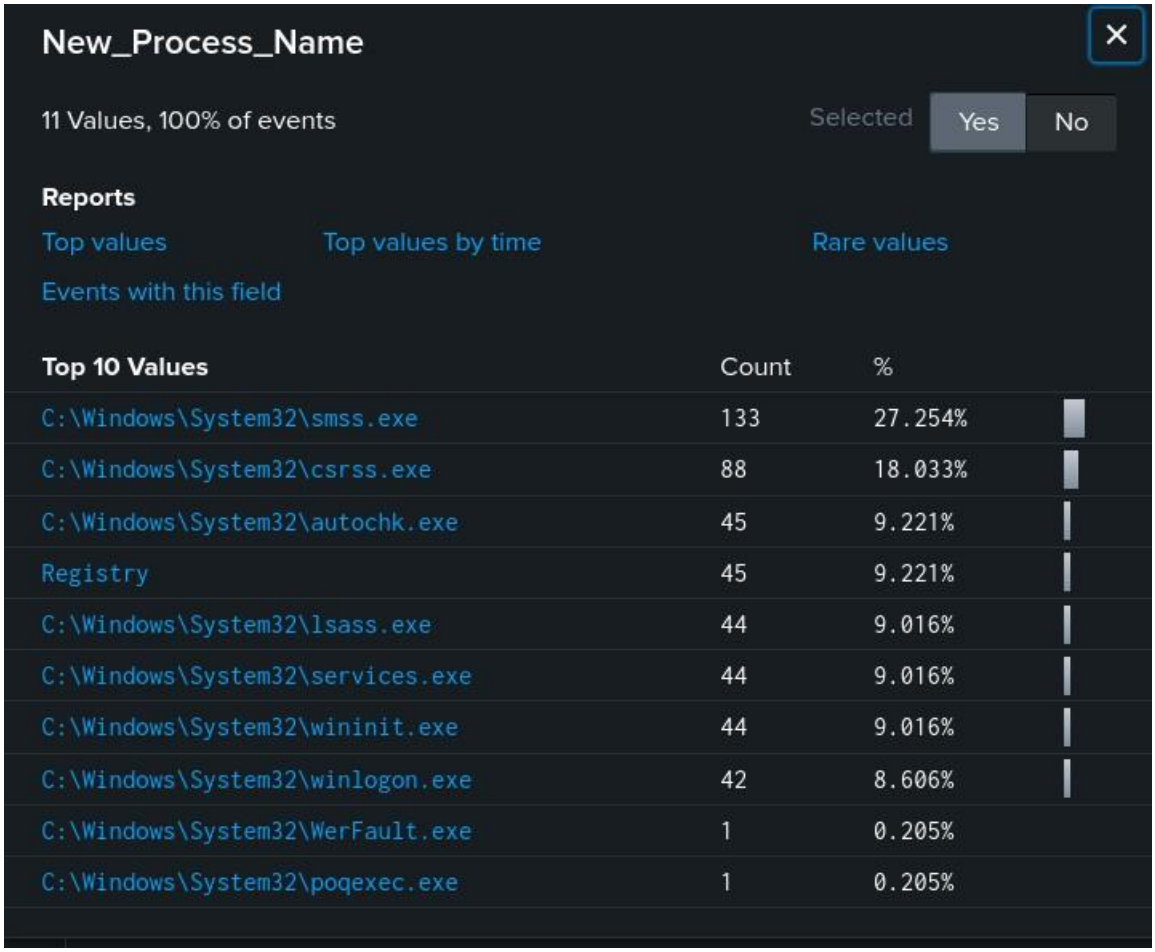
- Comprehensive searches of Windows Event Logs were performed to detect anomalies. These logs provided critical insights into system events, security incidents, and application behavior.





D) Identifying New Processes:

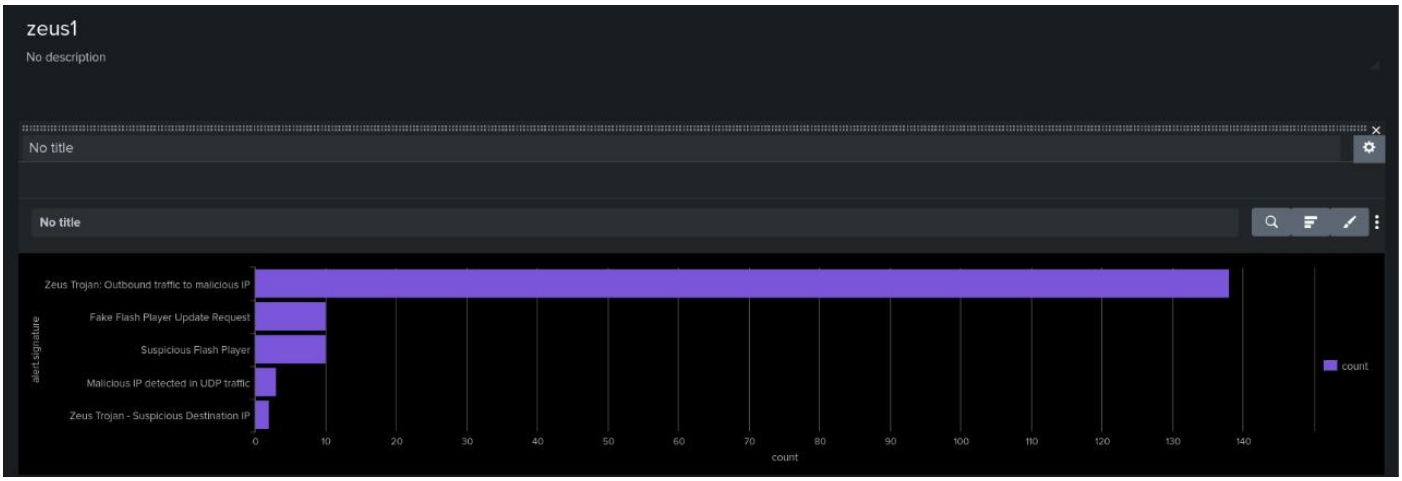
- Specific focus was given to identifying new processes spawned by the Zeus Trojan. These were detected by searching for Windows Event Log entries associated with process creation events (EventCode=4688).



E) Creating Zeus Dashboard :

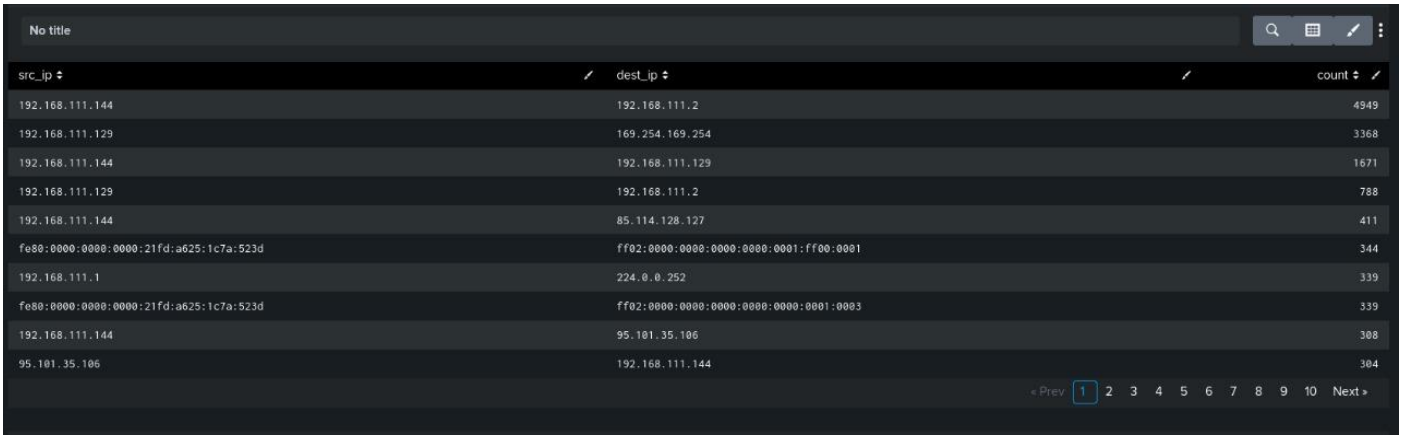
A) Panel 1 : Displays the frequency of detected alerts, grouped by signature .

Query: index=suricata200 sourcetype="logs" | stats count by alert.signature | sort – count



B) Panel 2: Highlights IPs generating or receiving the most traffic.

Query: index=suricata200 sourcetype="logs" | stats count by src_ip, dest_ip | sort – count

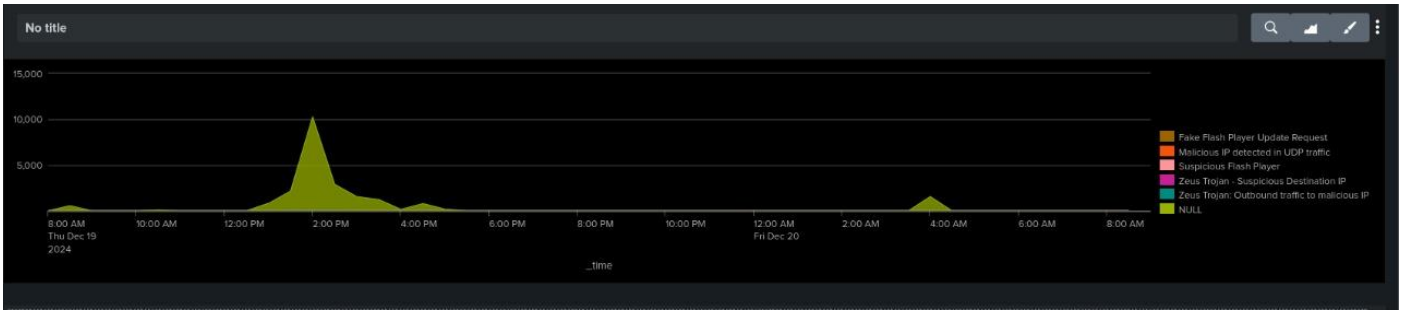


C) Panel 3: Tracks Zeus-related alerts over time.

Query: index=suricata200 sourcetype="logs" alert.signature="Zeus" | timechart count by alert.signature

D) Panel 4: Shows the trend of alerts over time.

Query: index=suricata200 sourcetype="logs" | timechart count by alert.signature

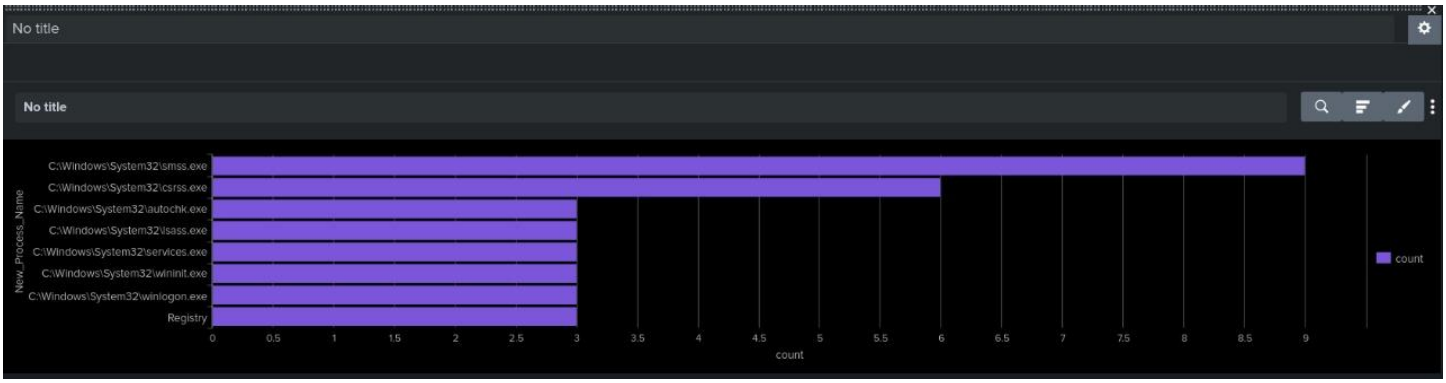


E) Panel 5 : Displays the frequency of detected processes, grouped by new_process .

Query: sourcetype="WinEventLog:Security" EventCode=4688

| stats count by New_Process_Name

| sort-count



Step 6: Analyzing the Memory Dump Using Volatility:

We will analyze a memory dump from a potentially compromised system using the Volatility framework. The memory image, **zeus2x4.vmem**, is analyzed to identify malicious processes and their characteristics.

The analysis will include the identification of running processes, network connections, and the extraction of executable files from memory.

1. **Image Information:**

First, we gather information about the memory image to determine the appropriate profile to use for analysis.

```
┌──(seif㉿final)-[~/volatility]
└─$ python2 vol.py -f ../Downloads/zeus2x4.vmem imageinfo
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apihooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.evtlogs (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.tcaudit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpregistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apihooks_kernel (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.registry.amcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.check_syscall_shadow (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.ssdt (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.registry.registryapi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.apihooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envars (ImportError: No module named Crypto.Hash)
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/seif/Downloads/zeus2x4.vmem)
                      PAE type : No PAE
                           DTB : 0x39000L
                          KDBG : 0x8054cde0L
          Number of Processors : 1
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2010-09-09 19:56:54 UTC+0000
     Image local date and time : 2010-09-09 15:56:54 -0400
```

The output suggests two possible profiles: **WinXPSP2x86** and **WinXPSP3x86**. We will use **WinXPSP2x86** for our subsequent analysis.

2. Process Listing:

Next, we list all running processes in the memory image to identify any suspicious activity.

python2 vol.py -f ../Downloads/zeus2x4.vmem --profile WinXPSP2x86 pslist

Offset(V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start
0x823c8a00	System	4	0	57	671	------	0	
0x82292da0	smss.exe	596	4	3	19	------	0	2010-09-02 12:25:18 UTC+0000
0x821f2978	csrss.exe	668	596	14	471	0	0	2010-09-02 12:25:21 UTC+0000
0x822c09f8	winlogon.exe	692	596	21	588	0	0	2010-09-02 12:25:22 UTC+0000
0x821a5da0	services.exe	744	692	15	279	0	0	2010-09-02 12:25:22 UTC+0000
0x822c8798	lsass.exe	756	692	24	437	0	0	2010-09-02 12:25:22 UTC+0000
0x82150b90	svchost.exe	912	744	20	202	0	0	2010-09-02 12:25:22 UTC+0000
0x822c8bf8	svchost.exe	992	744	10	277	0	0	2010-09-02 12:25:22 UTC+0000
0x82151da0	svchost.exe	1084	744	58	1327	0	0	2010-09-02 12:25:22 UTC+0000
0x821521b0	svchost.exe	1140	744	6	81	0	0	2010-09-02 12:25:22 UTC+0000
0x8214f488	svchost.exe	1192	744	13	175	0	0	2010-09-02 12:25:23 UTC+0000
0x8221e278	iscsiexe.exe	1436	744	6	78	0	0	2010-09-02 12:25:24 UTC+0000
0x82095500	spoolsv.exe	1616	744	13	140	0	0	2010-09-02 12:25:24 UTC+0000
0x821b2020	explorer.exe	1752	1720	22	520	0	0	2010-09-02 12:25:25 UTC+0000
0x822b96c0	SharedIntApp.ex	1900	1752	3	75	0	0	2010-09-02 12:25:25 UTC+0000
0x820ee580	prl_cc.exe	1908	1752	14	133	0	0	2010-09-02 12:25:25 UTC+0000
0x8212ada0	jusched.exe	1936	1752	1	43	0	0	2010-09-02 12:25:26 UTC+0000
0x82129370	svchost.exe	364	744	4	88	0	0	2010-09-02 12:25:33 UTC+0000
0x82089558	jqs.exe	472	744	5	146	0	0	2010-09-02 12:25:33 UTC+0000
0x8208abf0	sqlservr.exe	488	744	25	306	0	0	2010-09-02 12:25:33 UTC+0000
0x82077da0	coherence.exe	572	744	4	51	0	0	2010-09-02 12:25:36 UTC+0000
0x82189530	prl_tools_servi	436	744	3	78	0	0	2010-09-02 12:25:36 UTC+0000
0x82086798	prl_tools.exe	632	436	9	107	0	0	2010-09-02 12:25:36 UTC+0000
0x821aa7e8	sqlwriter.exe	660	744	4	84	0	0	2010-09-02 12:25:36 UTC+0000
0x8213dda0	wscntfy.exe	2180	1084	3	48	0	0	2010-09-02 12:25:41 UTC+0000
0x81e8a368	alg.exe	2588	744	6	107	0	0	2010-09-02 12:25:44 UTC+0000
0x8205dda0	wuauclt.exe	940	1084	4	126	0	0	2010-09-02 12:26:40 UTC+0000
0x82001ad0	ImmunityDebugge	2972	1752	2	87	0	0	2010-09-08 19:14:36 UTC+0000
0x8207bda0	nifek_locked.ex	2204	2972	2	38	0	0	2010-09-08 19:14:36 UTC+0000
0x82282380	ImmunityDebugge	1932	1752	2	86	0	0	2010-09-08 19:23:02 UTC+0000
0x8223c020	vaelh.exe	952	1932	2	40	0	0	2010-09-08 19:23:02 UTC+0000
0x81ffb6d8	ImmunityDebugge	3788	1752	2	103	0	0	2010-09-08 22:39:40 UTC+0000
0x8219e5c8	anaxu.exe	3508	3788	2	54	0	0	2010-09-08 22:39:40 UTC+0000
0x81eab2f8	wuauclt.exe	3984	1084	8	325	0	0	2010-09-09 19:52:45 UTC+0000
0x82066478	ImmunityDebugge	2404	1752	2	85	0	0	2010-09-09 19:56:19 UTC+0000
0x81f4bb28	b98679df6defbb3	3772	2404	1	46	0	0	2010-09-09 19:56:19 UTC+0000
0x81e87da0	ihah.exe	3276	3772	1	45	0	0	2010-09-09 19:56:32 UTC+0000
0x82311648	rundll32.exe	3768	1084	1	53	0	0	2010-09-09 19:56:33 UTC+0000

including **ImmunityDebugge**, **explorer.exe**, and others. Notably, we see processes with unusual names or those that are not typically found on a clean system, such as **ihah.exe** and **b98679df6defbb3**.

3. **Process Tree:**

   To understand the parent-child relationships between processes, we can visualize the process tree.

```
┌──(seif㉿final)-[~/volatility]
└─$ python2 vol.py -f ../Downloads/zeus2x4.vmem --profile WinXPSP2x86 pstree
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apihooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.evtlogs (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.tcaudit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpregistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apihooks_kernel (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.registry.amcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.check_syscall_shadow (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.ssdt (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.registry.registryapi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.apihooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envars (ImportError: No module named Crypto.Hash)
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x821b2020:explorer.exe                             1752   1720     22    520 2010-09-02 12:25:25 UTC+0000
. 0x82282380:ImmunityDebugge                         1932   1752      2     86 2010-09-08 19:23:02 UTC+0000
.. 0x8223c020:vaelh.exe                               952   1932      2     40 2010-09-08 19:23:02 UTC+0000
. 0x8212ada0:jusched.exe                             1936   1752      1     43 2010-09-02 12:25:26 UTC+0000
. 0x82001ad0:ImmunityDebugge                         2972   1752      2     87 2010-09-08 19:14:36 UTC+0000
.. 0x8207bda0:nifek_locked.ex                        2204   2972      2     38 2010-09-08 19:14:36 UTC+0000
. 0x81ffb6d8:ImmunityDebugge                         3788   1752      2    103 2010-09-08 22:39:40 UTC+0000
.. 0x8219e5c8:anaxu.exe                              3508   3788      2     54 2010-09-08 22:39:40 UTC+0000
. 0x820ee580:prl_cc.exe                              1908   1752     14    133 2010-09-02 12:25:25 UTC+0000
. 0x82066478:ImmunityDebugge                         2404   1752      2     85 2010-09-09 19:56:19 UTC+0000
.. 0x81f4bb28:b98679df6defbb3                        3772   2404      1     46 2010-09-09 19:56:19 UTC+0000
... 0x81e87da0:ihah.exe                              3276   3772      1     45 2010-09-09 19:56:32 UTC+0000
. 0x822b96c0:SharedIntApp.ex                         1900   1752      3     75 2010-09-02 12:25:25 UTC+0000
```

The process tree reveals that ImmunityDebugge is a parent process to several other processes, including ihah.exe. This could indicate
that ImmunityDebugge is being used as a debugger for malicious activities.

4. Network Connection:

Next, we check for any active network connections that may indicate communication with a command and control (C2) server.

```
┌──(seif㉿final)-[~/volatility]
└─$ python2 vol.py -f ../Downloads/zeus2x4.vmem --profile WinXPSP2x86 connscan
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apihooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.evtlogs (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.tcaudit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpregistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apihooks_kernel (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.registry.amcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.check_syscall_shadow (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.ssdt (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.registry.registryapi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.apihooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envars (ImportError: No module named Crypto.Hash)
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------- ---
0x020f5410 10.211.55.5:1427         65.54.81.89:80            1084
0x02125008 10.211.55.5:1423         207.46.21.123:80          1084
0x022ace08 10.211.55.5:1432         193.43.134.14:80          1752
```

```
┌──(seif㉿final)-[~/volatility]
└─$ python2 vol.py -f ../Downloads/zeus2x4.vmem --profile WinXPSP2x86 connections
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apihooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.evtlogs (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.tcaudit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpregistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apihooks_kernel (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.registry.amcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.check_syscall_shadow (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.ssdt (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.registry.registryapi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.apihooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envars (ImportError: No module named Crypto.Hash)
Offset(V)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------- ---
0x822ace08 10.211.55.5:1432         193.43.134.14:80          1752
```

The output shows several connections, including ip 193.43.134.14 with PID 1752 which is the PPID for the ImmunityDebugge .

5. Dumping Malicious Processes:

To analyze the contents of the suspicious processes, we can dump their executable files from memory.

```
┌──(seif㉿final)-[~/volatility]
└─$ python2 vol.py -f ../Downloads/zeus2×4.vmem --profile WinXPSP2×86 procdump -p 2404 -D procdump
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apihooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.evtlogs (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.tcaudit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpregistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apihooks_kernel (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.registry.amcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.check_syscall_shadow (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.ssdt (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.registry.registryapi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.apihooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envars (ImportError: No module named Crypto.Hash)
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0×82066478 0×00400000 ImmunityDebugge      OK: executable.2404.exe
```

This command dumps the executable of the process with PID 2404 (identified as ImmunityDebugge). We repeat this for other suspicious processes:

- **b98679df6defbb3**

```
┌──(seif㉿final)-[~/volatility]
└─$ python2 vol.py -f ../Downloads/zeus2×4.vmem --profile WinXPSP2×86 procdump -p 3772 -D procdump
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apihooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.evtlogs (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.tcaudit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpregistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apihooks_kernel (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.registry.amcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.check_syscall_shadow (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.ssdt (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.registry.registryapi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.apihooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envars (ImportError: No module named Crypto.Hash)
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0×81f4bb28 0×00400000 b98679df6defbb3      OK: executable.3772.exe
```

- **ihah.exe**

```
┌──(seif㉿final)-[~/volatility]
└─$ python2 vol.py -f ../Downloads/zeus2x4.vmem --profile WinXPSP2x86 procdump -p 3276 -D procdump
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apihooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.evtlogs (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.tcaudit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpregistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apihooks_kernel (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.registry.amcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.check_syscall_shadow (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.ssdt (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.registry.registryapi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.apihooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envars (ImportError: No module named Crypto.Hash)
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0x81e87da0 0x00400000 ihah.exe             OK: executable.3276.exe
```

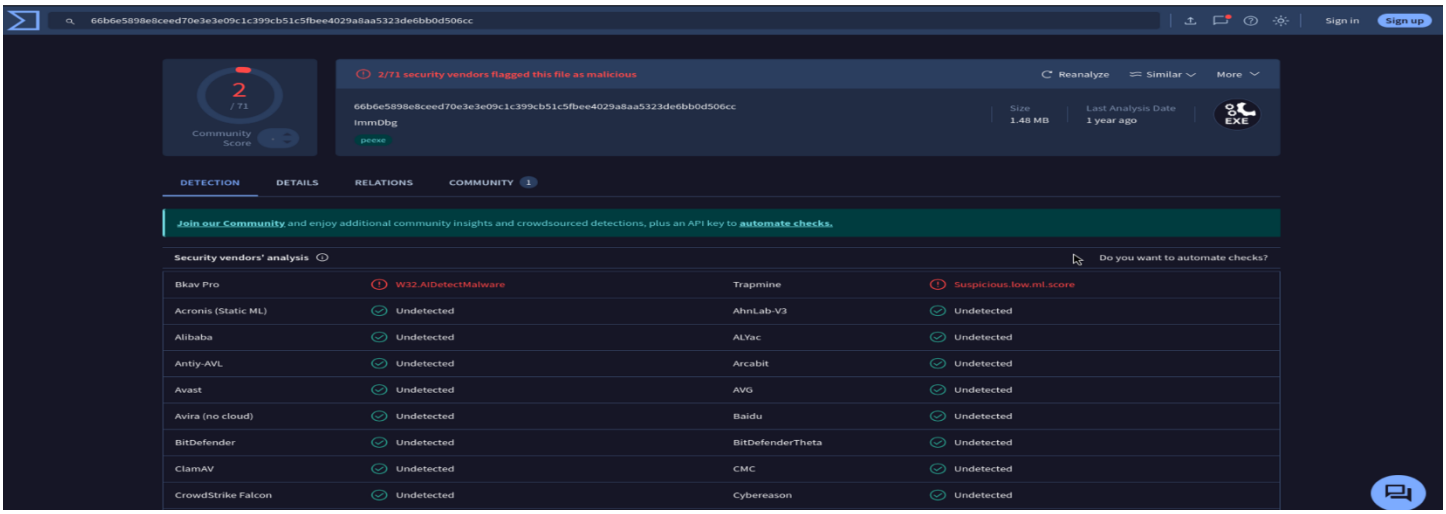
6. Analyzing Dumped Executables

After dumping the executables, we can calculate their SHA256 hashes to verify their integrity and check against known malware databases.

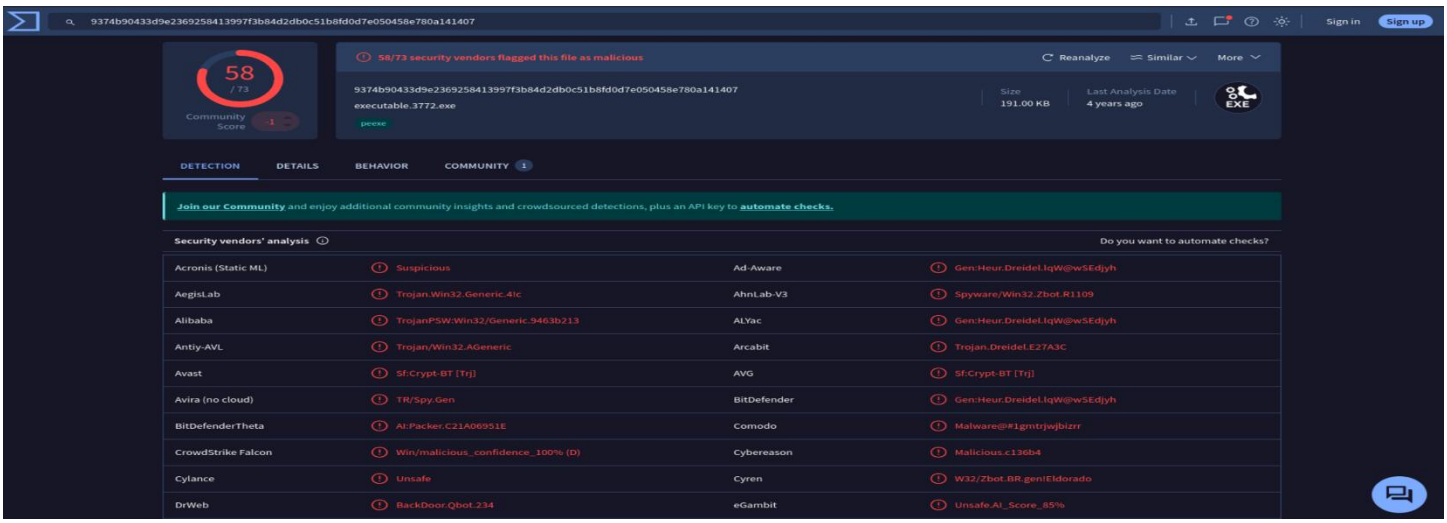
```
┌──(seif㉿final)-[~/volatility]
└─$ sha256sum procdump/executable.2404.exe
66b6e5898e8ceed70e3e3e09c1c399cb51c5fbee4029a8aa5323de6bb0d506cc  procdump/executable.2404.exe

┌──(seif㉿final)-[~/volatility]
└─$ sha256sum procdump/executable.3772.exe
9374b90433d9e2369258413997f3b84d2db0c51b8fd0d7e050458e780a141407  procdump/executable.3772.exe

┌──(seif㉿final)-[~/volatility]
└─$ sha256sum procdump/executable.3276.exe
c4b88f8e160f9eb145bb9e12e5122fa539a83b772e93929efb8846c8e1171eed  procdump/executable.3276.exe
```

The hashes can be used to search for known malware signatures on VirusTotal.

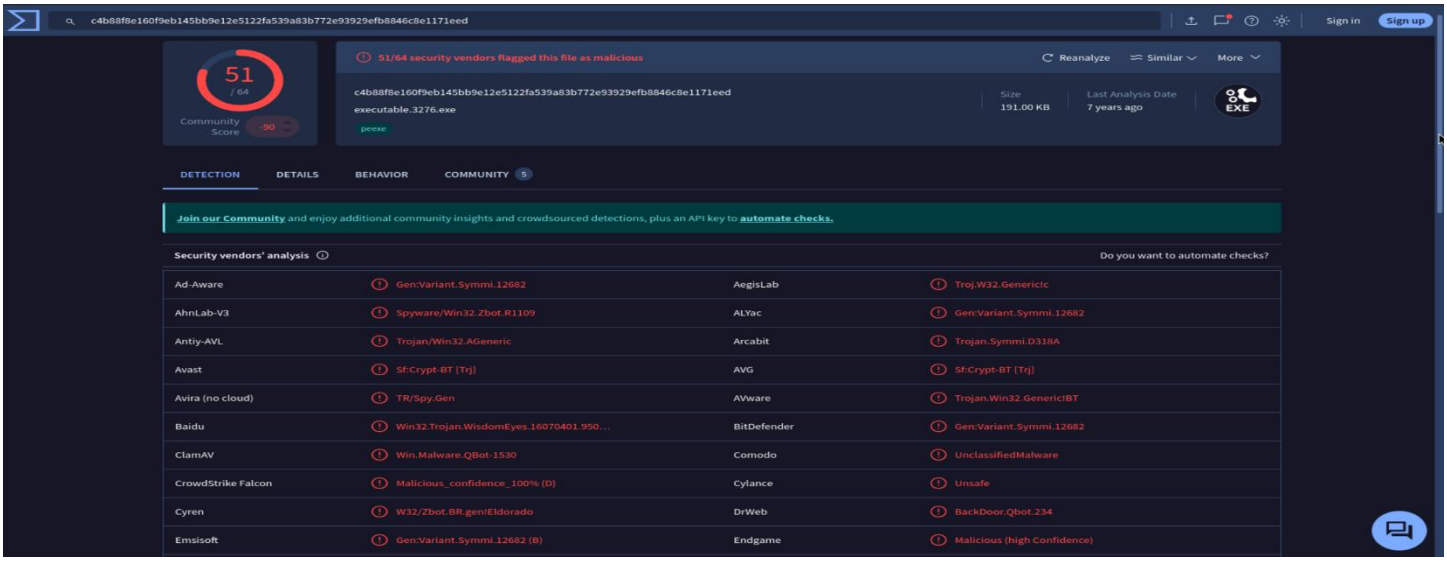
- **ImmunityDebugge**



- **b98679df6defbb3**



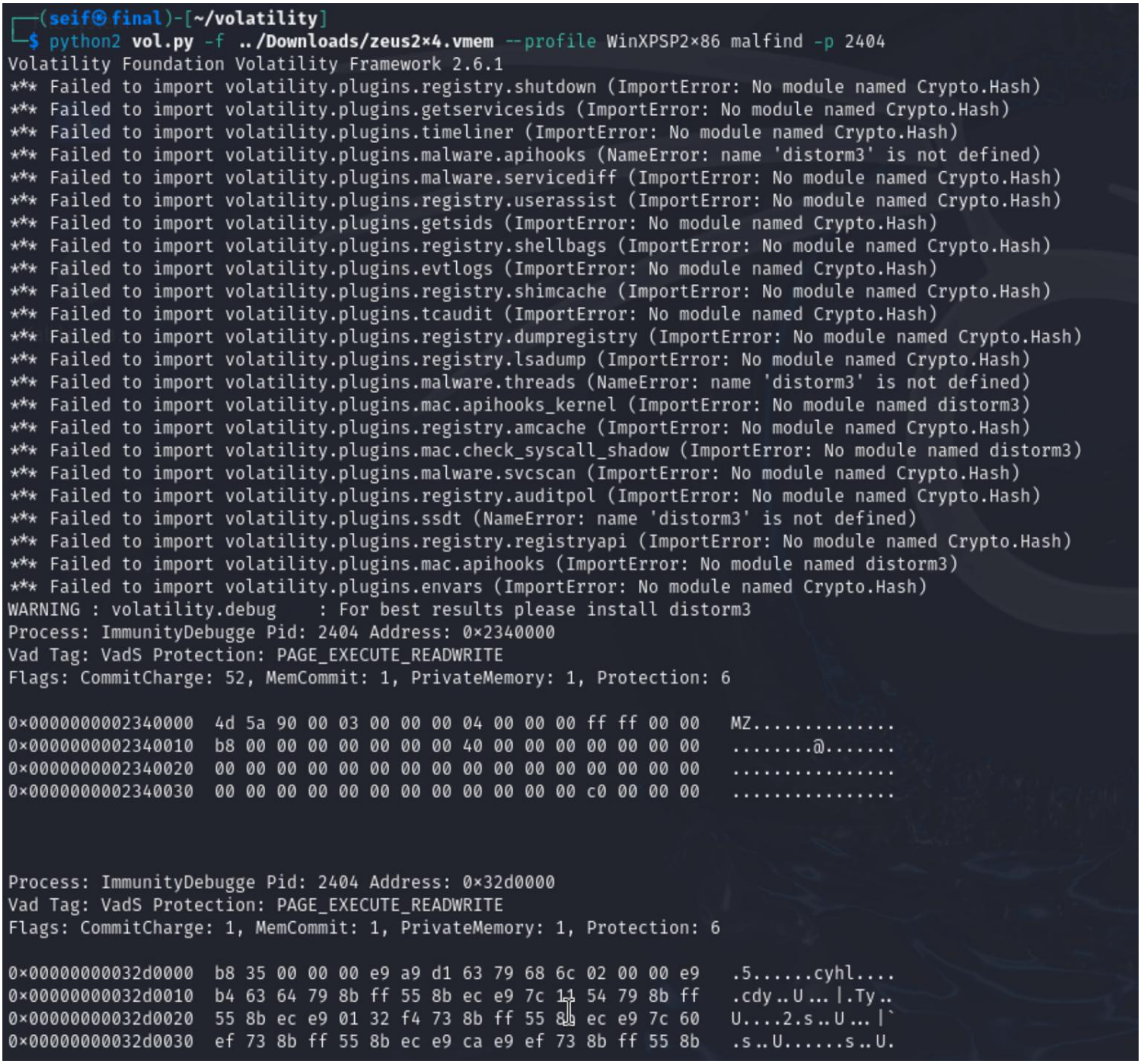
- **ihah.exe**



7. Memory Analysis for Malicious Code:

To further analyze the memory for injected code or other malicious artifacts, we can use the malfind plugin.

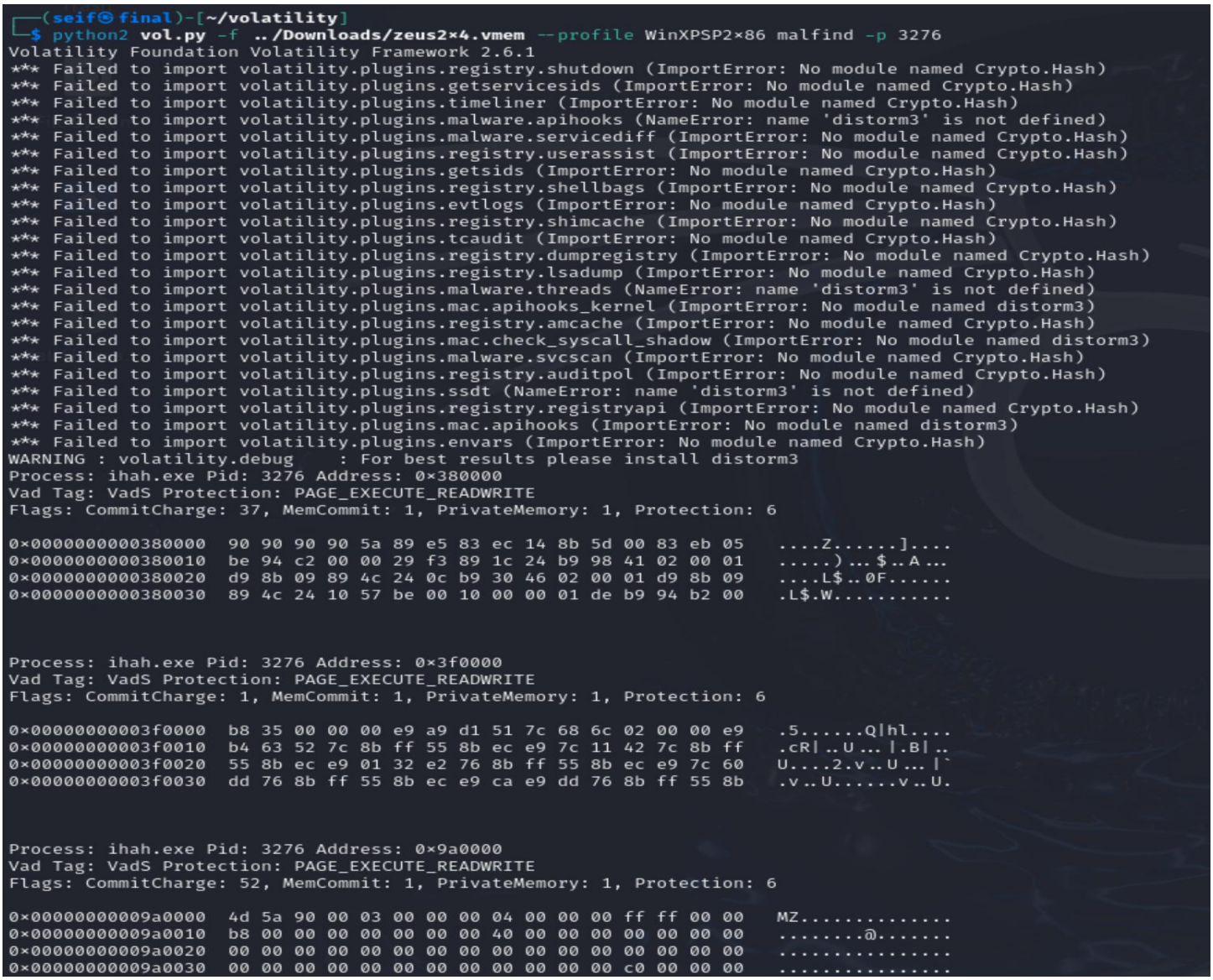
- **ImmunityDebugge**



- **b98679df6defbb3**

```
┌──(seif㉿final)-[~/volatility]
└─$ python2 vol.py -f ../Downloads/zeus2x4.vmem --profile WinXPSP2x86 malfind -p 3772
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apihooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.evtlogs (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.tcaudit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpregistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apihooks_kernel (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.registry.amcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.check_syscall_shadow (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.ssdt (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.registry.registryapi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.apihooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envars (ImportError: No module named Crypto.Hash)
WARNING : volatility.debug    : For best results please install distorm3
Process: b98679df6defbb3 Pid: 3772 Address: 0x380000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 36, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x0000000000380000  90 90 90 90 5a 89 e5 83 ec 14 8b 5d 00 83 eb 05   ....Z......]....
0x0000000000380010  be 0b 88 01 00 29 f3 89 1c 24 b9 84 44 02 00 01   .....)...$..D...
0x0000000000380020  d9 8b 09 89 4c 24 0c b9 74 43 02 00 01 d9 8b 09   ....L$..tC......
0x0000000000380030  89 4c 24 10 57 be 00 10 00 00 01 de b9 0b 78 01   .L$.W.........x.



Process: b98679df6defbb3 Pid: 3772 Address: 0x9a0000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 52, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x00000000009a0000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x00000000009a0010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x00000000009a0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00000000009a0030  00 00 00 00 00 00 00 00 00 00 00 00 c0 00 00 00   ................



Process: b98679df6defbb3 Pid: 3772 Address: 0xa30000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x0000000000a30000  b8 35 00 00 00 e9 a9 d1 ed 7b 68 6c 02 00 00 e9   .5.......{hl....
0x0000000000a30010  b4 63 ee 7b 8b ff 55 8b ec e9 7c 11 de 7b 8b ff   .c.{..U...|..{..
0x0000000000a30020  55 8b ec e9 01 32 7e 76 8b ff 55 8b ec e9 7c 60   U....2~v..U...|`
0x0000000000a30030  79 76 8b ff 55 8b ec e9 ca e9 79 76 8b ff 55 8b   yv..U.....yv..U.
```

- **ihah.exe**



The output shows suspicious memory regions associated with the process, including executable code that may not be part of the original executable.

- We can see that this process has MZ header and protection of PAGE_EXECUTE_READWRITE, which means that this memory region is marked as executable, and it can also be both read from and written to.
- This is unusual and suspicious because it allows an attacker to execute code from that memory region and also modify its content dynamically.
- In legit scenarios, memory regions won't be executable and writable at the same time.

Step 7: Detect Zeus with YARA Signatures:

First Create a Yara Rule file:

```
rule ZeusBanking_Detection {
        meta:
                author="Seif Ahmed"
                description="A detection rule against ZuesBankingVersion_26Nov2013"
        strings:
                $file_name="invoice_2318362983713_823931342io.pdf.exe" ascii
                // Suspected name of functions and DLL functionalities.
                $function_name_KERNEL32_CreateFileA="CellrotoCrudUntohighCols" ascii
                // PE Magic Byte.
                $PE_magic_byte = "MZ"
                // Hex String Function name.
                $hex_string = {43 61 6D 65 56 61 6C 65 57 61 75 6C 65 72}
        condition:
                $PE_magic_byte at 0 and $file_name
                and $function_name_KERNEL32_CreateFileA
                or $hex_string
}
```

Strings Section:

This section defines strings or patterns the rule looks for:

- $file_name:

Detects a suspicious file name: "invoice_2318362983713_823931342io.pdf.exe".

- The ascii modifier ensures it's interpreted as plain text.
- The filename appears like a phishing or malicious document disguised as a PDF but is actually an executable (.exe).

- $function_name:

- Matches a specific function name: "CellrotoCrudUntohighCols".
- This is likely a unique name associated with the Zeus Banking malware, possibly part of its internal code.

- $PE_magic_byte:
  - Matches the magic bytes { 4D 5A }, which represent "MZ" in hexadecimal.
  - These bytes indicate the start of a Portable Executable (PE) file, common in Windows executables.
  - The at 0 condition ensures this pattern is at the very beginning of the file, as expected in a valid PE file.
- $hex_string:
  - Matches a specific binary pattern in hexadecimal: { 43 61 6D 65 56 61 6C 65 57 61 75 6C 65 72 }.
  - This corresponds to the ASCII string "CameValeWauler".
  - This might be a unique or uncommon string embedded in the malware for identification or functionality.

Condition Section:

The condition section specifies the logic for triggering the rule. This is the decision-making part of the rule.

- $PE_magic_byte at 0:
  - Ensures the file starts with MZ, confirming it is a Windows executable.
- $file_name and $function_name: Both must be present for a match.
- or $hex_string: The hex string can trigger the rule on its own, without the file name or function name.

The overall logic:

- If the file is a valid PE file (MZ at the start), And if:

A known malicious file name ($file_name) is present and a specific function name ($function_name) is found, OR a unique binary pattern ($hex_string) is present, Then the rule matches.

This setup ensures the rule is specific to the malware while still flexible enough to catch variations.

Second Detect the Malware with Yara:

```
┌──(seif㉿final)-[~]
└─$ yara rules.yara ./Downloads/test/invoice_2318362983713_823931342io.pdf.exe -s -w -p 32
ZeusBanking_Detection ./Downloads/test/invoice_2318362983713_823931342io.pdf.exe
0x3176c:$function_name: CellrotoCrudUntohighCols
0x0:$PE_magic_byte: 4D 5A
0x31716:$hex_string: 43 61 6D 65 56 61 6C 65 57 61 75 6C 65 72
```

Additional Step Create a Yara Rule with YarGen:

First create a Yara Rule to the ZeusBankingVersion_26Nov2013:

```
/*
   YARA Rule Set
   Author: yarGen Rule Generator
   Date: 2024-12-20
   Identifier: test
   Reference: https://github.com/Neo23x0/yarGen
*/

/* Rule Set ----------------------------------------------------------------- */

rule ZeusBankingVersion_26Nov2013 {
   meta:
      description = "test - file ZeusBankingVersion_26Nov2013.zip"
      author = "yarGen Rule Generator"
      reference = "https://github.com/Neo23x0/yarGen"
      date = "2024-12-20"
      hash1 = "4644b5fb10fb84c0d47bec4b5a48d5e60165e8ae2130fca5c055633aaad73162"
   strings:
      $s1 = "invoice_2318362983713_823931342io.pdf.exe" fullword ascii
      $s2 = "invoice_2318362983713_823931342io.pdf.exePK" fullword ascii
      $s4 = "@THBep!" fullword ascii
      $s5 = "OWVjY6S" fullword ascii
      $s6 = "oBPD@='" fullword ascii
      $s7 = ".tjB,`" fullword ascii
      $s8 = "IlBU,\"" fullword ascii
      $s9 = "asRt#U!" fullword ascii
      $s10 = "iYqK/e^" fullword ascii
      $s11 = "Kl)'tNrIIA_wAn" fullword ascii
      $s12 = "lXxgM0O" fullword ascii
      $s13 = "vtaMp?p" fullword ascii
      $s14 = "PMzrbuX" fullword ascii
      $s15 = "InYv?8" fullword ascii
      $s16 = "\\/vVuW" fullword ascii
      $s17 = "UofQk3" fullword ascii
      $s18 = "2G]zoh" fullword ascii
      $s19 = "G[IfNy#" fullword ascii
      $s20 = "!+d+1_" fullword ascii
   condition:
      uint16(0) == 0x4b50 and filesize < 500KB and
      8 of them
}
```

Strings Section:

This section defines patterns (strings) the rule searches for in a file. Each $sN variable represents a specific pattern:

- Strings like $s1 and $s2 are filenames or parts of filenames (invoice_2318362983713_823931342io.pdf.exe).
- Other strings, such as $s3 to $s20, are ASCII sequences potentially indicating unique or suspicious content (e.g., malicious code or markers of the Zeus malware).

Condition Section:

The condition determines whether the rule triggers. The conditions for this rule are:

- uint16(0) == 0x4b50: Checks if the file's first two bytes are 0x4b50, which indicates a ZIP file.
- filesize < 500KB: Ensures the file size is less than 500 KB.
- 8 of them: At least 8 of the defined strings must match for the rule to trigger.

This rule is tailored to detect ZIP files with specific malicious content associated with the Zeus malware.

<u>Detect it with Yara:</u>

Second create a Yara Rule to the invoice_2318362983713_823931342io_pdf:

```
rule invoice_2318362983713_823931342io_pdf {
   meta:
      description = "test - file invoice_2318362983713_823931342io.pdf.exe"
      author = "yarGen Rule Generator"
      reference = "https://github.com/Neo23x0/yarGen"
      date = "2024-12-20"
      hash1 = "69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169"
   strings:
      $s1 = "ejDmZKid5htD0UB[gZTHJVrLlTaTBBsBS18pEuDJBuMks{0H0zRNleRt2kh8S:QPqP/2v2JFYWjpubc,vQKhJvYCDZsyJKTWY,B6xyzRzzHY6Ezu44u6U6LOL[dhqVMn" ascii
      $s2 = "corect.com" fullword ascii
      $s3 = "USER32.GetShellWindow" fullword ascii
      $s4 = "KERNEL32.GetThreadPriority" fullword ascii
      $s5 = "KERNEL32.SystemTimeToFileTime" fullword ascii
      $s6 = "USER32.GetKeyNameTextA" fullword ascii
      $s7 = "KERNEL32.CreateIoCompletionPort" fullword ascii
      $s8 = "KERNEL32.GetShortPathNameA" fullword ascii
      $s9 = "KERNEL32.GetWindowsDirectoryA" fullword ascii
      $s10 = "KERNEL32.GetStartupInfoW" fullword ascii
      $s11 = "kqKDSPX2HCYOP/CYRnffTI[QZT{BN8Tafn,Jg2Ko[0X+i1oOknPp4ubEZniy2Q:OfQpxex4frsHQLes46ehHemEMxU9LPw{6VUKMC06pOw6cLW395ZdQdqxqDI6UQu7W" ascii
      $s12 = "9tc34LSgjT7ksJmvD1NxsNewhlynXj97U7O2OIsjnaNv0Vglp5FzexmnW7uVORnovysoxu0sKAIn0NYuxRcwu81fYFOEugVLBVJ+3jUAl/w2{hHZhK9leprOkc:ehsEO" ascii
      $s13 = "IKe397ub8CXtoFKc4rpl7t{DViecb2T7YM1yKaiMRmyCfs8Q:m[+PtURL3Myem6ZTR6kTSYjeph4xg1wlgrno+H0p81Wmn78yBOY76uEWgJRfJUWBsYj9UhYSyka,41W" ascii
      $s14 = "Dumpcotsavo" fullword ascii
      $s15 = "        <requestedExecutionLevel level='asInvoker' uiAccess=\"false\"/>" fullword ascii
      $s16 = "USER32.GetUpdateRgn" fullword ascii
      $s17 = "gi4HzEwf0b9TQHjtEoOXk3TgcahTZe3sCGwEOg5iVBZz3WW7wkiNIMrnH0ZuSagxOTBaU93fuzD4BD7yiAU9MT6yUdT+fdoMjVpOOlOGZZVdXPV7cfpzMrUnxewB5eYr" ascii
      $s18 = "USER32.GetMonitorInfoW" fullword ascii
      $s19 = "jqiXzmPzIU8V590Xs8,5xbUM7YgXcpsjiizfRlhaQhH/pYXxG8LJqjhVskFt34KOlaJG9KCGjT,brQrWn/xuwTW3xm,CyP60F936QWqfEhEgN1gM830gOtrTb6hbP7ir" ascii
      $s20 = "fvifsB4KEyDcEPd9ma,mZmhSNAXYsZEbZZclOdUQCS6p8uEip/hwoawNRzsRy6G5JkiyRhp/pLoGOKTt68dv6HMz:ofAI7VI7o8lZxQpqKq51M3U,Nsk0Fy1rZVIdPKI" ascii
   condition:
      uint16(0) == 0x5a4d and filesize < 700KB and
      8 of them
}
```

Strings Section:

- $s1: A long ASCII sequence, likely representing encoded or obfuscated data.
- $s2: References corect.com, which may be associated with malicious infrastructure.
- $s3 to $s10: Function names from Windows APIs, suggesting this file interacts with the operating system, potentially for malicious purposes.
- $s15: A line from a manifest file indicating a requested execution level, showing the program's privileges.
- $s14: "Dumpcotsavo" is likely a specific string marker.
- $s11, $s12, $s13, $s17, $s19, $s20: Long ASCII sequences indicating additional suspicious patterns.

Condition Section:

- uint16(0) == 0x5a4d: Checks if the file starts with 0x5a4d, indicating a Windows executable (PE file).
- filesize < 700KB: Ensures the file size is less than 700 KB.
- 8 of them: At least 8 of the defined strings must match for the rule to trigger.

This rule is focused on detecting executable files related to a suspicious invoice filename with patterns of malicious behavior.

Detect it with Yara: