SESSION 2025-26
January 17, 2026

UNIVERSITY OF DHAKA

PROFESSIONAL MASTER'S IN INFORMATION & CYBERSECURITY (PMICS)

CSE 804 (2nd Mid on Lab)

Network Traffic Analysis, Threat Mitigation & EDR Integration

CSE 804: Network & Internet Security

SUBMITTED BY

NAME OF STUDENT
Nishan Paul

ROLL NUMBER
50028

REG. NO
H-55

BATCH
05

COLLABORATIVE GROUP 4 Members

1	Md Rakibur Rahman	50040
2	Anado Shaffat	50004
3	Md Hossain Sarwar	50036
4	Md Ranik Miah	50017

Table of Contents

1. Forensic Network Traffic Analysis & Defensive Signature Development
    - 1.1 Analysis Context & Scenario
    - 1.2 Task Objective & Requirement
    - 1.3 Technical Execution: Traffic Forensics
    - 1.4 Defense Implementation: Suricata NIDS Development
    - 1.5 Validation & Monitoring
2. Active Threat Mitigation & Host-Level Enforcement
    - 2.1 Task Objective & Requirement
    - 2.2 Implementation: Host Isolation using IPTables
    - 2.3 Post-Mitigation Validation
3. EDR Integration & Advanced Endpoint Monitoring
    - 3.1 Task Objective & Strategy
    - 3.2 Technical Setup: Wazuh & Sysmon Integration
    - 3.3 Configuration: High-Fidelity Process Monitoring
    - 3.4 Simulation & Detection Validation
4. Executive Summary & Conclusion

SECTION 1: Forensic Network Traffic Analysis & Defensive Signature Development

1.1 Analysis Context & Scenario

The investigation focuses on a real-world infection scenario sourced from the forensic repository **malware-traffic-analysis.net**.

- **Case Reference:** 2025-01-22 - DOWNLOAD FROM FAKE SOFTWARE SITE
- **Evidence Source:** 2025-01-22-traffic-analysis-exercise.pcap.zip
- **Security Threat:** An unsuspecting user downloaded a malicious object from a search engine result (Google Authenticator search) which triggered a Stage-1 PowerShell execution script (.ps1).

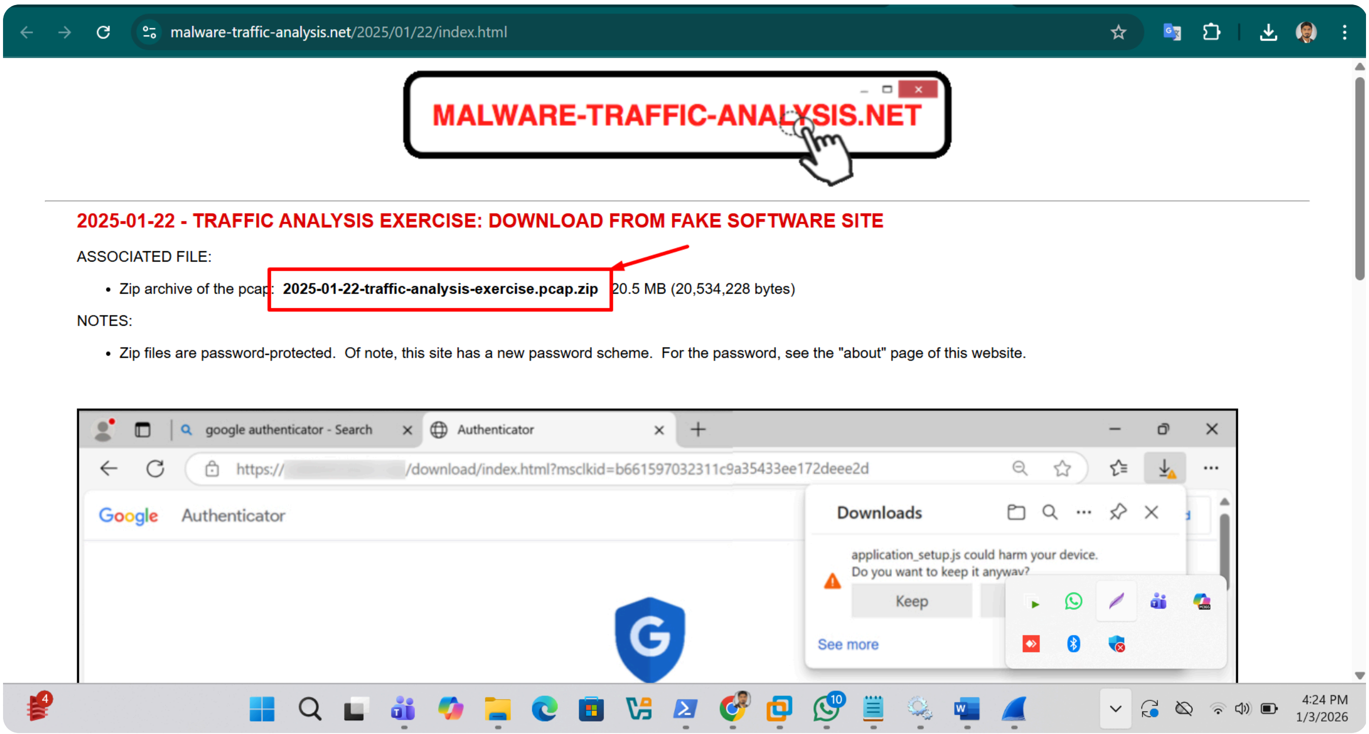
1.2 Task Objective & Requirement

> **Question 1:** *You need to create a custom Suricata signature which will trigger an alert for any type of HTTP communication to this public IP.*

Technical Goal: Perform deep packet inspection to identify the malicious source IP, then develop a custom Signature for the Suricata Network IDS to automate future detection.

1.3 Technical Execution: Traffic Forensics

We utilized **Wireshark** to parse the forensic data. To identify the infection vector, we prioritized searching for Stage-1 execution scripts. Since PowerShell is a common vector for initial execution (**MITRE T1059.001**), we utilized the display filter `http.request.uri contains ".ps1"`.



malware-traffic-analysis.net/2025/01/22/index.html

MALWARE-TRAFFIC-ANALYSIS.NET

2025-01-22 - TRAFFIC ANALYSIS EXERCISE: DOWNLOAD FROM FAKE SOFTWARE SITE

ASSOCIATED FILE:

- Zip archive of the pcap: **2025-01-22-traffic-analysis-exercise.pcap.zip** 20.5 MB (20,534,228 bytes)

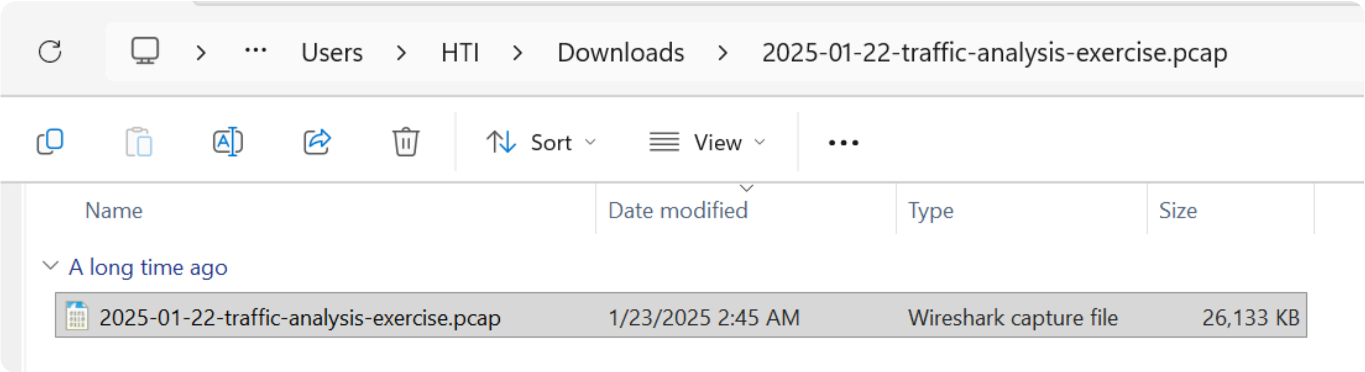
NOTES:

- Zip files are password-protected. Of note, this site has a new password scheme. For the password, see the "about" page of this website.

https://[redacted]/download/index.html?msclkid=b661597032311c9a35433ee172deee2d

application_setup.js could harm your device. Do you want to keep it anyway?

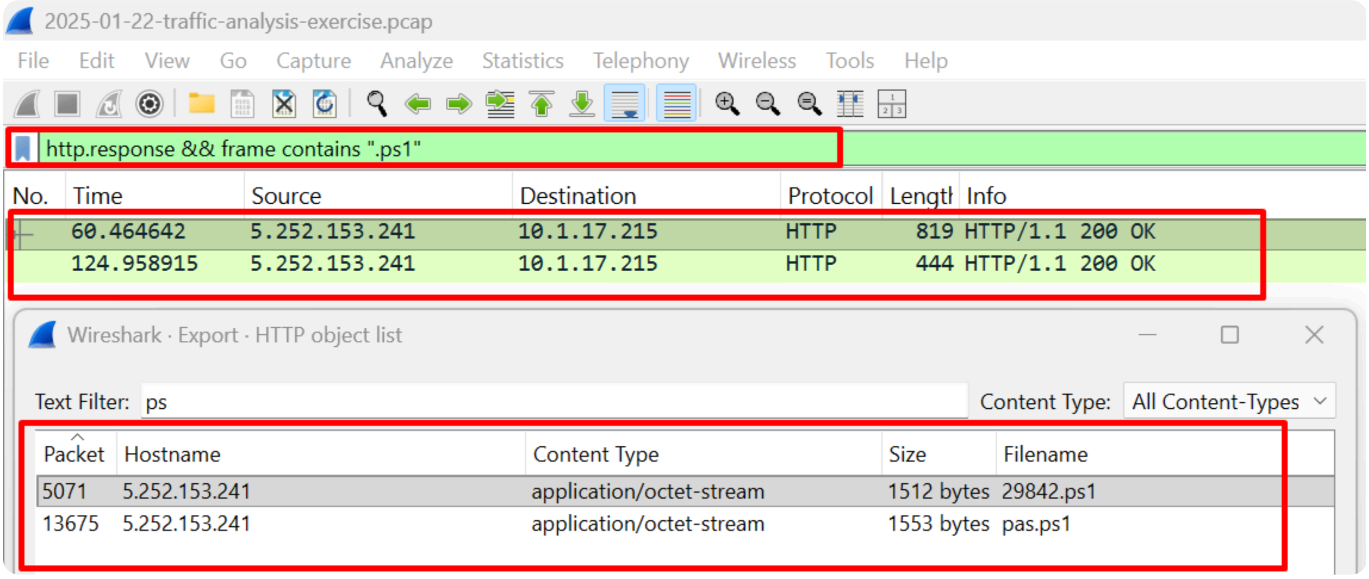
Keep
See more
4:24 PM
1/3/2026



Users > HTI > Downloads > 2025-01-22-traffic-analysis-exercise.pcap
Sort
View

Name	Date modified	Type	Size
A long time ago			
2025-01-22-traffic-analysis-exercise.pcap	1/23/2025 2:45 AM	Wireshark capture file	26,133 KB

The search isolated two distinct HTTP GET requests aimed at retrieving PowerShell scripts. A secondary filter, http.response && frame contains ".ps1", confirmed that the malicious payloads were effectively delivered to the victim host.



2025-01-22-traffic-analysis-exercise.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.response && frame contains ".ps1"

No.	Time	Source	Destination	Protocol	Lengtl	Info
	60.464642	5.252.153.241	10.1.17.215	HTTP	819	HTTP/1.1 200 OK
	124.958915	5.252.153.241	10.1.17.215	HTTP	444	HTTP/1.1 200 OK

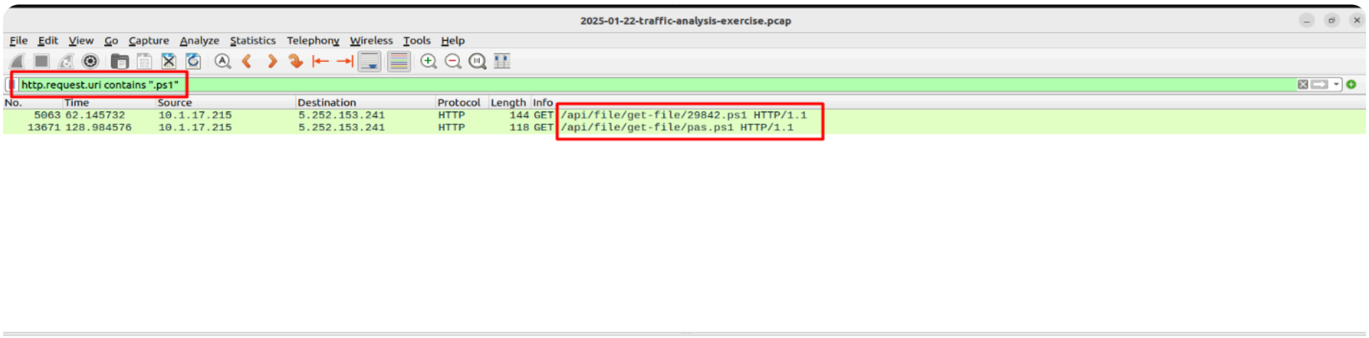
Wireshark · Export · HTTP object list

Text Filter: ps

Content Type: All Content-Types

Packet	Hostname	Content Type	Size	Filename
5071	5.252.153.241	application/octet-stream	1512 bytes	29842.ps1
13675	5.252.153.241	application/octet-stream	1553 bytes	pas.ps1

Using the **Export Objects** feature, we confirmed the download source as the public IP **5.252.153.241**.


2025-01-22-traffic-analysis-exercise.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http.request.uri contains ".ps1"

```
No.    Time        Source       Destination    Protocol Length Info
  5063 62.145732   10.1.17.215  5.252.153.241  HTTP        144 GET /api/file/get-file/29842.ps1 HTTP/1.1
 13671 128.984576  10.1.17.215  5.252.153.241  HTTP        118 GET /api/file/get-file/pas.ps1 HTTP/1.1
```

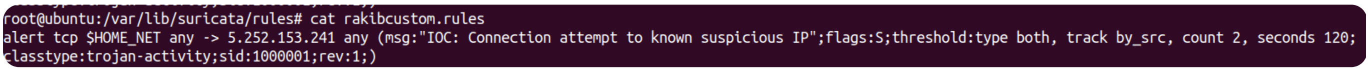
1.4 Defense Implementation: Suricata NIDS Development

Having established **5.252.153.241** as a malicious endpoint, we deployed a multi-stage rule in the **Suricata** sensor.

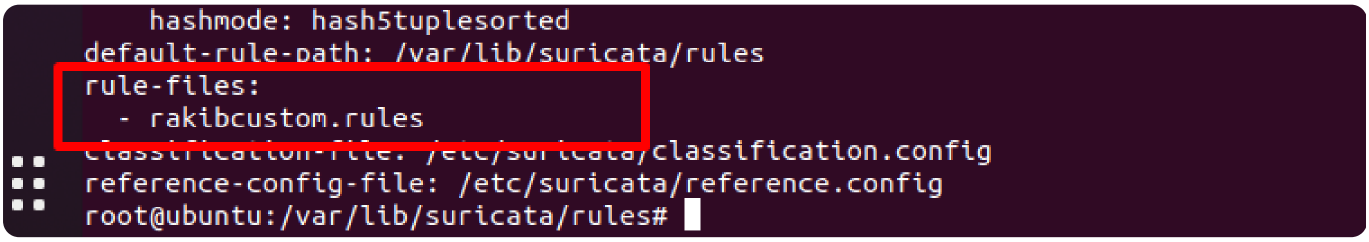
SOC Engineering Strategy: We implemented both a broad detection rule and an advanced threshold-based rule to reduce "alert fatigue" during potential scanning events.

```
# [Basic Detection] Triggers on any standard HTTP attempt to the C2
alert tcp any any → 5.252.153.241 80 (msg:"TCP connection attempt to 5.252.153.241 detected"; sid:1000002; rev:1;)

# [Production Standard] Implements thresholding for high-fidelity alerting
echo "alert tcp $HOME_NET any → 5.252.153.241 any (msg:\"IOC: Connection attempt to known suspicious IP\"; flags:S; threshold:type both, track by_src, count 2, seconds 120; classtype:trojan-activity; sid:1000001; rev:1;)" > rakibcustom.rules
```



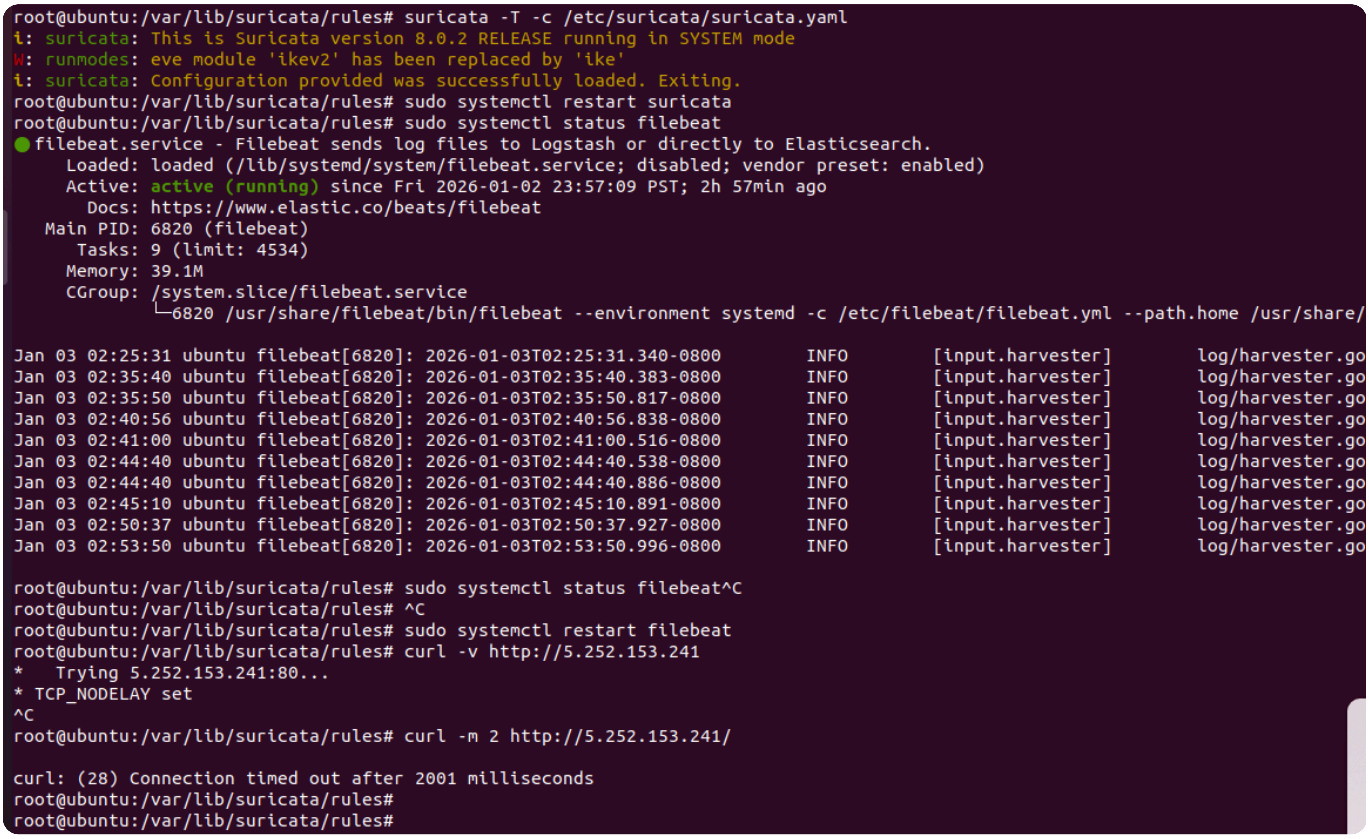
```
root@ubuntu:/var/lib/suricata/rules# cat rakibcustom.rules
alert tcp $HOME_NET any -> 5.252.153.241 any (msg:"IOC: Connection attempt to known suspicious IP";flags:S;threshold:type both, track by_src, count 2, seconds 120;
classtype:trojan-activity;sid:1000001;rev:1;)
```



```
    hashmode: hash5tuplesorted
default-rule-path: /var/lib/suricata/rules
rule-files:
  - rakibcustom.rules
classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
root@ubuntu:/var/lib/suricata/rules# 
```

1.5 Validation & Monitoring

Post-restart of the Suricata and Filebeat services, we verified the configuration through manual traffic simulation using wget.



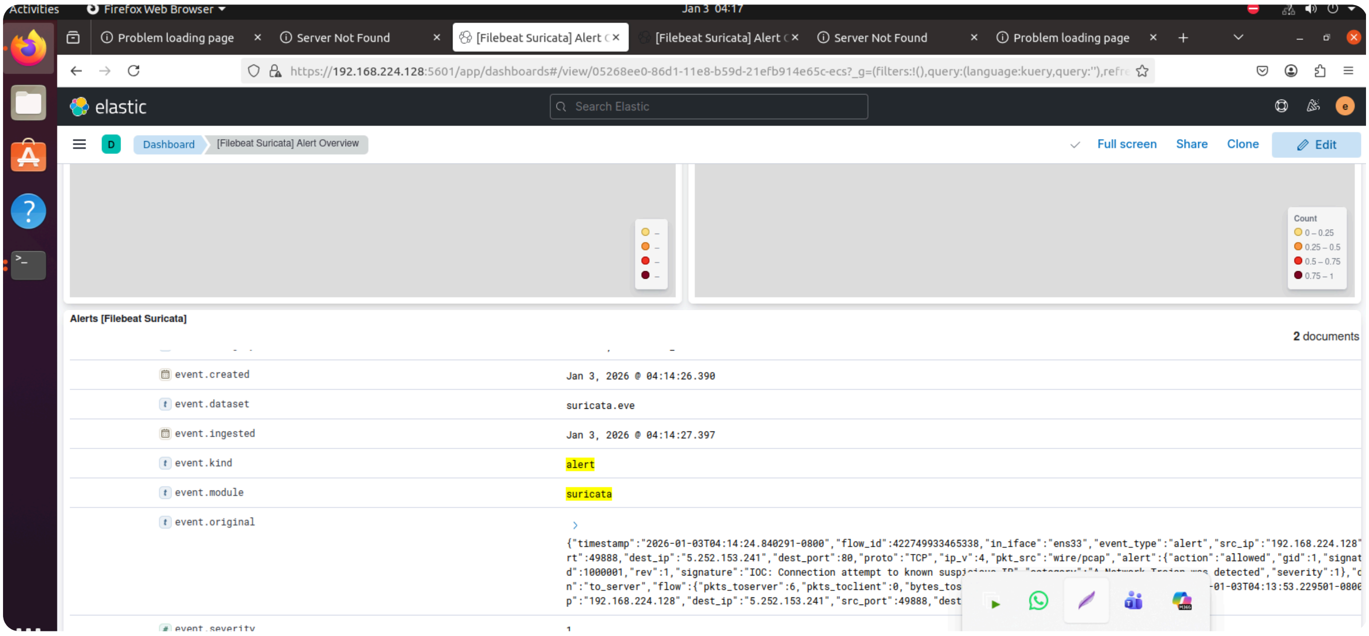
```
root@ubuntu:/var/lib/suricata/rules# suricata -T -c /etc/suricata/suricata.yaml
i: suricata: This is Suricata version 8.0.2 RELEASE running in SYSTEM mode
W: runmodes: eve module 'ikev2' has been replaced by 'ike'
i: suricata: Configuration provided was successfully loaded. Exiting.
root@ubuntu:/var/lib/suricata/rules# sudo systemctl restart suricata
root@ubuntu:/var/lib/suricata/rules# sudo systemctl status filebeat
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
     Loaded: loaded (/lib/systemd/system/filebeat.service; disabled; vendor preset: enabled)
     Active: active (running) since Fri 2026-01-02 23:57:09 PST; 2h 57min ago
       Docs: https://www.elastic.co/beats/filebeat
   Main PID: 6820 (filebeat)
      Tasks: 9 (limit: 4534)
     Memory: 39.1M
     CGroup: /system.slice/filebeat.service
             └─6820 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/filebeat/filebeat.yml --path.home /usr/share/

Jan 03 02:25:31 ubuntu filebeat[6820]: 2026-01-03T02:25:31.340-0800       INFO       [input.harvester]       log/harvester.go
Jan 03 02:35:40 ubuntu filebeat[6820]: 2026-01-03T02:35:40.383-0800       INFO       [input.harvester]       log/harvester.go
Jan 03 02:35:50 ubuntu filebeat[6820]: 2026-01-03T02:35:50.817-0800       INFO       [input.harvester]       log/harvester.go
Jan 03 02:40:56 ubuntu filebeat[6820]: 2026-01-03T02:40:56.838-0800       INFO       [input.harvester]       log/harvester.go
Jan 03 02:41:00 ubuntu filebeat[6820]: 2026-01-03T02:41:00.516-0800       INFO       [input.harvester]       log/harvester.go
Jan 03 02:44:40 ubuntu filebeat[6820]: 2026-01-03T02:44:40.538-0800       INFO       [input.harvester]       log/harvester.go
Jan 03 02:44:40 ubuntu filebeat[6820]: 2026-01-03T02:44:40.886-0800       INFO       [input.harvester]       log/harvester.go
Jan 03 02:45:10 ubuntu filebeat[6820]: 2026-01-03T02:45:10.891-0800       INFO       [input.harvester]       log/harvester.go
Jan 03 02:50:37 ubuntu filebeat[6820]: 2026-01-03T02:50:37.927-0800       INFO       [input.harvester]       log/harvester.go
Jan 03 02:53:50 ubuntu filebeat[6820]: 2026-01-03T02:53:50.996-0800       INFO       [input.harvester]       log/harvester.go

root@ubuntu:/var/lib/suricata/rules# sudo systemctl status filebeat^C
root@ubuntu:/var/lib/suricata/rules# ^C
root@ubuntu:/var/lib/suricata/rules# sudo systemctl restart filebeat
root@ubuntu:/var/lib/suricata/rules# curl -v http://5.252.153.241
*   Trying 5.252.153.241:80...
* TCP_NODELAY set
^C
root@ubuntu:/var/lib/suricata/rules# curl -m 2 http://5.252.153.241/

curl: (28) Connection timed out after 2001 milliseconds
root@ubuntu:/var/lib/suricata/rules#
root@ubuntu:/var/lib/suricata/rules#
```

The **Elasticsearch** dashboard successfully captured the alerts, categorizing them under **"Trojan Activity"**, providing the SOC team with clear visibility into the threat activity.



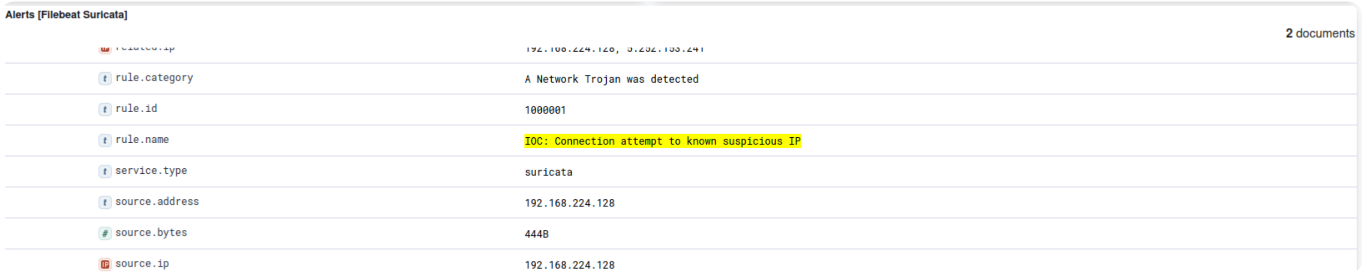
https://192.168.224.128:5601/app/dashboards#/view/05268ee0-86d1-11e8-b59d-21efb914e65c-ecs?_g=(filters:!(),query:(language:kuery,query:''),refr

[Filebeat Suricata] Alert Overview

Alerts [Filebeat Suricata]

2 documents

```
event.created     Jan 3, 2026 @ 04:14:26.390
event.dataset     suricata.eve
event.ingested    Jan 3, 2026 @ 04:14:27.397
event.kind        alert
event.module      suricata
event.original    {"timestamp":"2026-01-03T04:14:24.840291-0800","flow_id":422749933465338,"in_iface":"ens33","event_type":"alert","src_ip":"192.168.224.128"
                  rt":49888,"dest_ip":"5.252.153.241","dest_port":80,"proto":"TCP","ip_v":4,"pkt_src":"wire/pcap","alert":{"action":"allowed","gid":1,"signat
                  d":1000001,"rev":1,"signature":"IOC: Connection attempt to known susp                                   detected","severity":1},"c
                  n":"to_server","flow":{"pkts_toserver":6,"pkts_toclient":0,"bytes_tos                                   -01-03T04:13:53.229501-0800
                  p":"192.168.224.128","dest_ip":"5.252.153.241","src_port":49888,"dest
event.severity    1
```



Alerts [Filebeat Suricata]

2 documents

related.ip	192.168.224.128, 5.252.153.241
rule.category	A Network Trojan was detected
rule.id	1000001
rule.name	IOC: Connection attempt to known suspicious IP
service.type	suricata
source.address	192.168.224.128
source.bytes	444B
source.ip	192.168.224.128

SECTION 2: Active Threat Mitigation & Host-Level Enforcement

2.1 Task Objective & Requirement

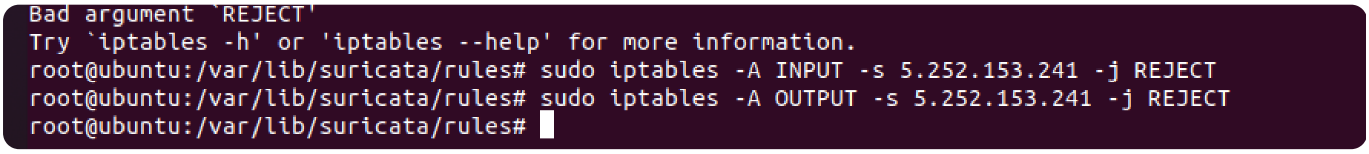
Question 2: *You need to block any communication toward this IOC from Q1.*

Security Goal: Establish an immediate "deny-all" boundary to isolate the compromised host from the attacker's Command and Control (C2) infrastructure.

2.2 Implementation: Host Isolation using IPTables

To prevent data exfiltration (**MITRE T1041**) or secondary stage drops, we enforced a strict firewall policy. By using **REJECT** instead of **DROP**, we provide a clear reset to any pending stateful connections.

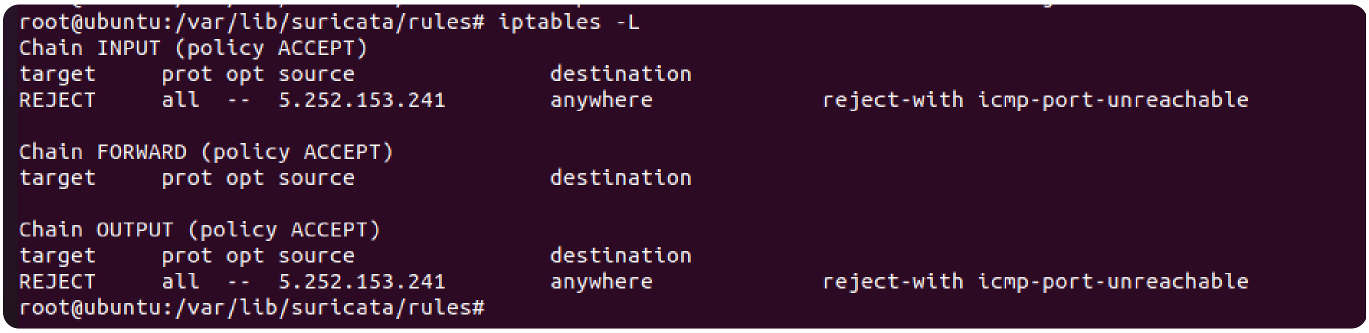
```
# Enforcement: Block all Inbound and Outbound traffic to the IOC sessions
sudo iptables -A INPUT -s 5.252.153.241 -j REJECT
sudo iptables -A OUTPUT -d 5.252.153.241 -j REJECT
```



```
Bad argument `REJECT'
Try `iptables -h' or 'iptables --help' for more information.
root@ubuntu:/var/lib/suricata/rules# sudo iptables -A INPUT -s 5.252.153.241 -j REJECT
root@ubuntu:/var/lib/suricata/rules# sudo iptables -A OUTPUT -s 5.252.153.241 -j REJECT
root@ubuntu:/var/lib/suricata/rules# 
```

2.3 Post-Mitigation Validation

We performed verified connectivity tests to ensures the host was no longer vulnerable to outbound C2 communication. All attempts reached an immediate "Connection Refused" state.



```
root@ubuntu:/var/lib/suricata/rules# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  5.252.153.241        anywhere             reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  5.252.153.241        anywhere             reject-with icmp-port-unreachable
root@ubuntu:/var/lib/suricata/rules#
```

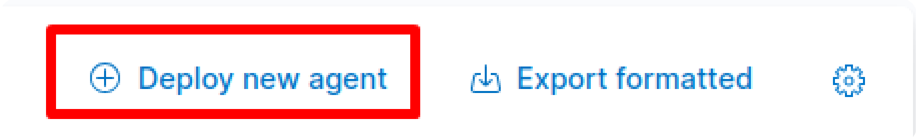
SECTION 3: EDR Integration & Advanced Endpoint Monitoring

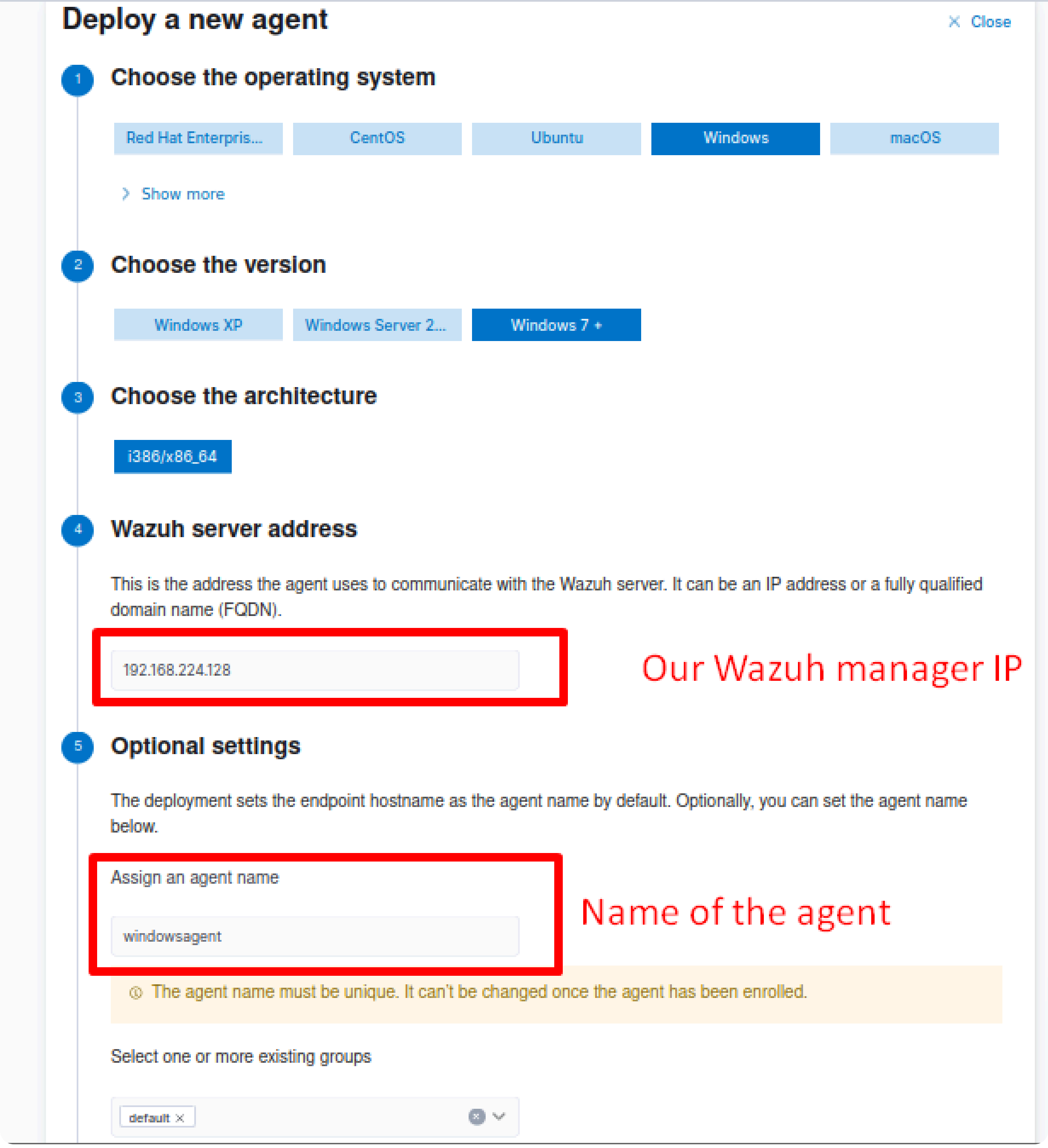
• 3.1 Task Objective & Strategy

The final objective was the deployment of an **Endpoint Detection and Response (EDR)** pipeline. We focused on monitoring **LSASS (Local Security Authority Subsystem Service)**, which is the primary target for credential harvesting (**MITRE T1003.001**).

• 3.2 Technical Setup: Wazuh & Sysmon Integration

We deployed the **Wazuh Agent** on the target Windows 11 machine for centralized telemetry.





Deploy a new agent

1 Choose the operating system

Windows

2 Choose the version

Windows 7 +

3 Choose the architecture

i386/x86_64

4 Wazuh server address

This is the address the agent uses to communicate with the Wazuh server. It can be an IP address or a fully qualified domain name (FQDN).

192.168.224.128

Our Wazuh manager IP

5 Optional settings

The deployment sets the endpoint hostname as the agent name by default. Optionally, you can set the agent name below.

Assign an agent name: windowsagent

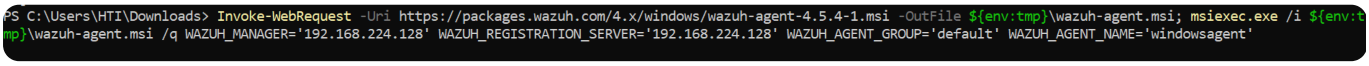
Name of the agent

ⓘ The agent name must be unique. It can't be changed once the agent has been enrolled.

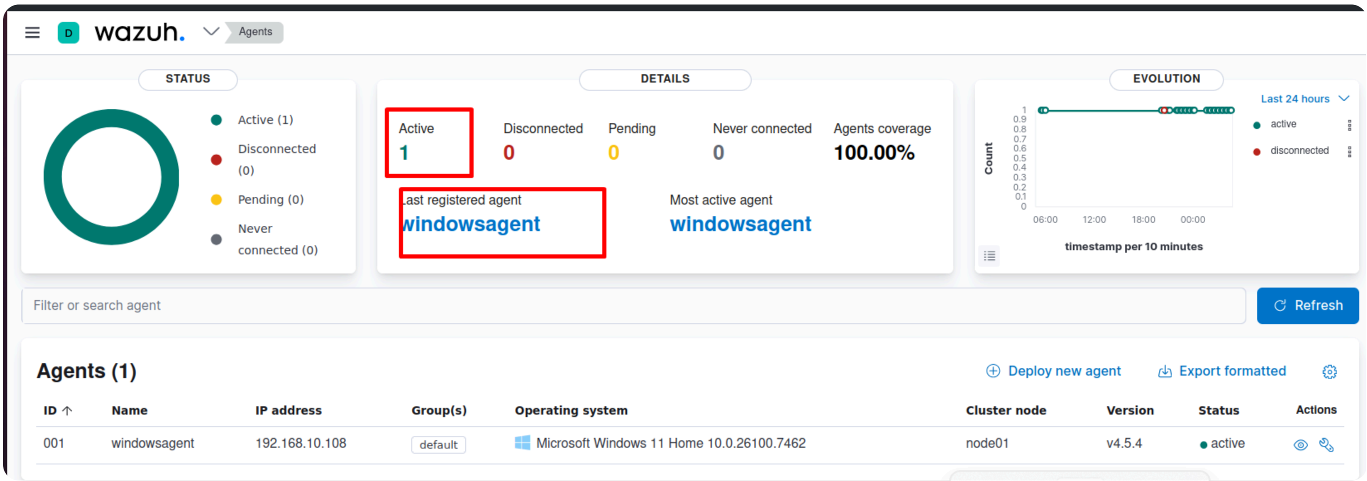
Select one or more existing groups: default ✕

The deployment was automated via the following PowerShell command to ensure immediate linkage to the SIEM cluster:

```
Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.5.4-1.msi -OutFile
${env:tmp}\wazuh-agent.msi; msiexec.exe /i ${env:tmp}\wazuh-agent.msi /q
WAZUH_MANAGER='192.168.224.128' WAZUH_REGISTRATION_SERVER='192.168.224.128'
WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='windowsagent'
```



```
PS C:\Users\HTI\Downloads> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.5.4-1.msi -OutFile ${env:tmp}\wazuh-agent.msi; msiexec.exe /i ${env:t
mp}\wazuh-agent.msi /q WAZUH_MANAGER='192.168.224.128' WAZUH_REGISTRATION_SERVER='192.168.224.128' WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='windowsagent'
```



STATUS: Active (1), Disconnected (0), Pending (0), Never connected (0)

DETAILS: Active 1 | Disconnected 0 | Pending 0 | Never connected 0 | Agents coverage 100.00%

Last registered agent: windowsagent

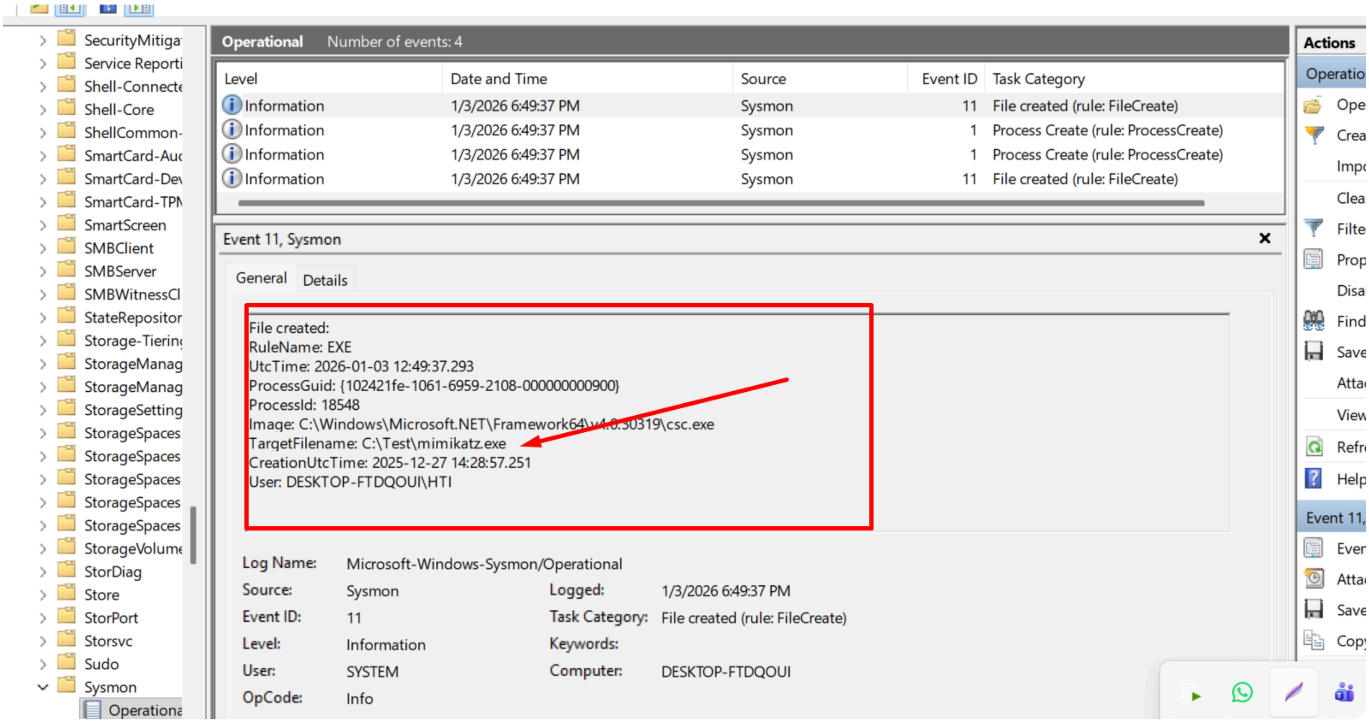
Most active agent: windowsagent

Agents (1)

ID ↑	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
001	windowsagent	192.168.10.108	default	Microsoft Windows 11 Home 10.0.26100.7462	node01	v4.5.4	● active	

• 3.3 Configuration: High-Fidelity Process Monitoring

We utilized **Sysmon** to monitor **Event ID 10 (ProcessAccess)**. By identifying unauthorized processes attempting to read the memory space of lsass.exe, we can detect credential dumping attempts in real-time.



Operational Number of events: 4

Level	Date and Time	Source	Event ID	Task Category
Information	1/3/2026 6:49:37 PM	Sysmon	11	File created (rule: FileCreate)
Information	1/3/2026 6:49:37 PM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	1/3/2026 6:49:37 PM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	1/3/2026 6:49:37 PM	Sysmon	11	File created (rule: FileCreate)

Event 11, Sysmon

General Details

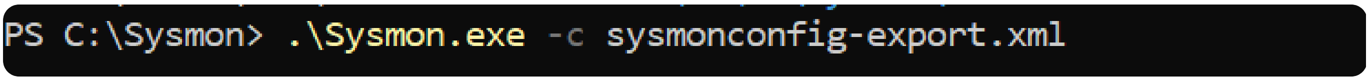
```
File created:
RuleName: EXE
UtcTime: 2026-01-03 12:49:37.293
ProcessGuid: {102421fe-1061-6959-2108-000000000900}
ProcessId: 18548
Image: C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe
TargetFilename: C:\Test\mimikatz.exe
CreationUtcTime: 2025-12-27 14:28:57.251
User: DESKTOP-FTDQOUI\HTI
```

Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon Logged: 1/3/2026 6:49:37 PM
Event ID: 11 Task Category: File created (rule: FileCreate)
Level: Information Keywords:
User: SYSTEM Computer: DESKTOP-FTDQOUI
OpCode: Info

Expert-Level Optimization: The exclusion logic below ensures that standard Windows processes (like svchost.exe) do not trigger false alerts, preserving system performance.

```
<!-- Critical Rule: Mitigating Credential Theft Attempts -->
<RuleGroup name="ProcessAccessToLSASS" groupRelation="and">
 <ProcessAccess onmatch="include">
  <TargetImage condition="is">C:\Windows\System32\lsass.exe</TargetImage>
 </ProcessAccess>
```

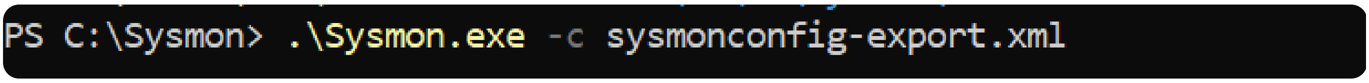
```
  ... [Noise reduction logic for stability] ...
</RuleGroup>
```



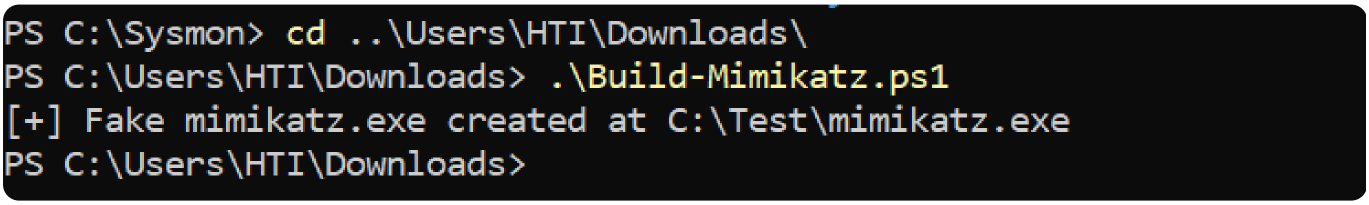
```
PS C:\Sysmon> .\Sysmon.exe -c sysmonconfig-export.xml
```

• 3.4 Simulation & Detection Validation

To verify our "Defense in Depth" strategy, we simulated a **Mimikatz** credential dumping attack.

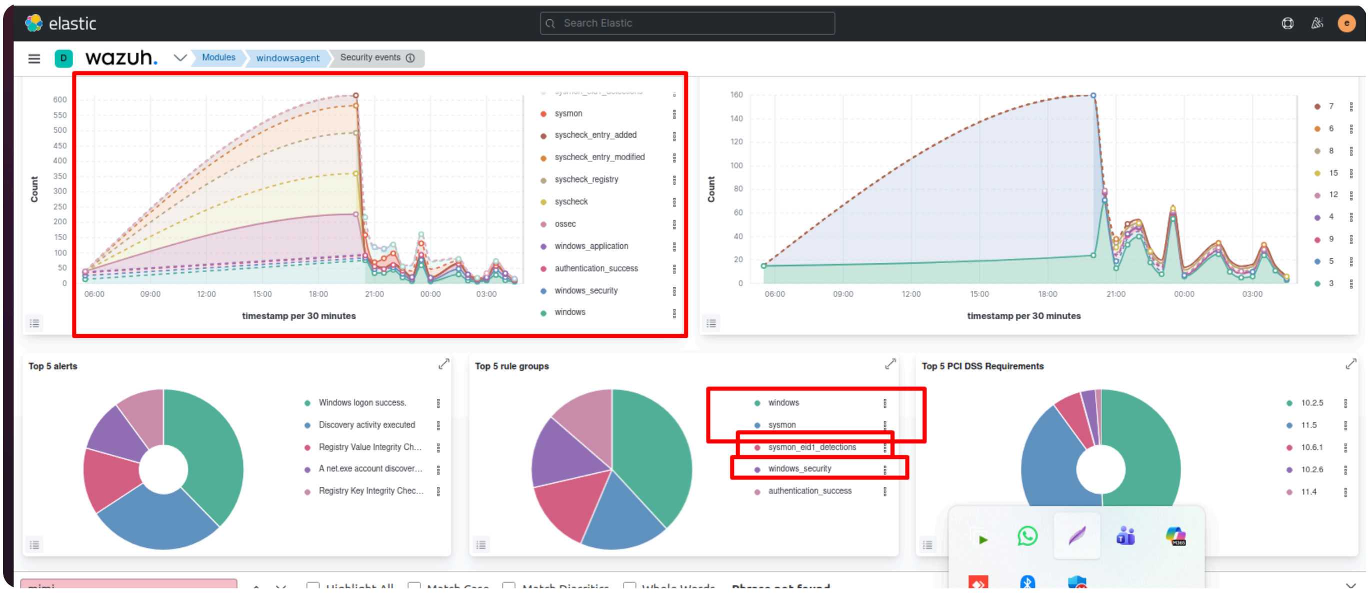


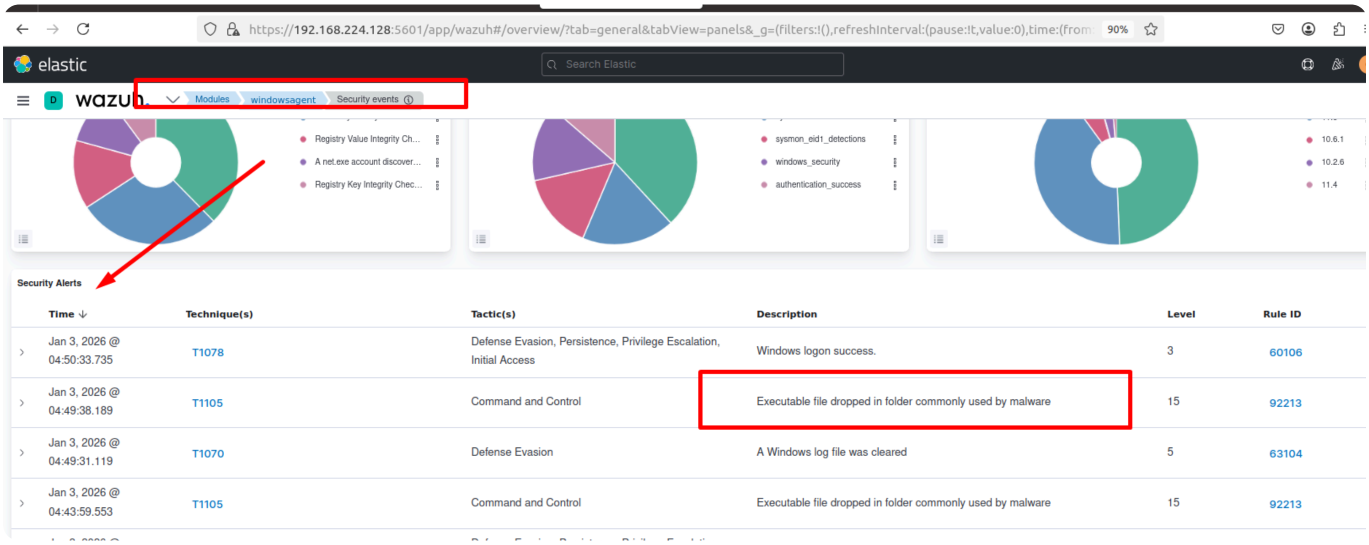
```
PS C:\Sysmon> .\Sysmon.exe -c sysmonconfig-export.xml
```



```
PS C:\Sysmon> cd ..\Users\HTI\Downloads\
PS C:\Users\HTI\Downloads> .\Build-Mimikatz.ps1
[+] Fake mimikatz.exe created at C:\Test\mimikatz.exe
PS C:\Users\HTI\Downloads>
```

The detection pipeline successfully triggered high-priority alerts in the Wazuh dashboard, confirming our ability to detect memory-resident threats.

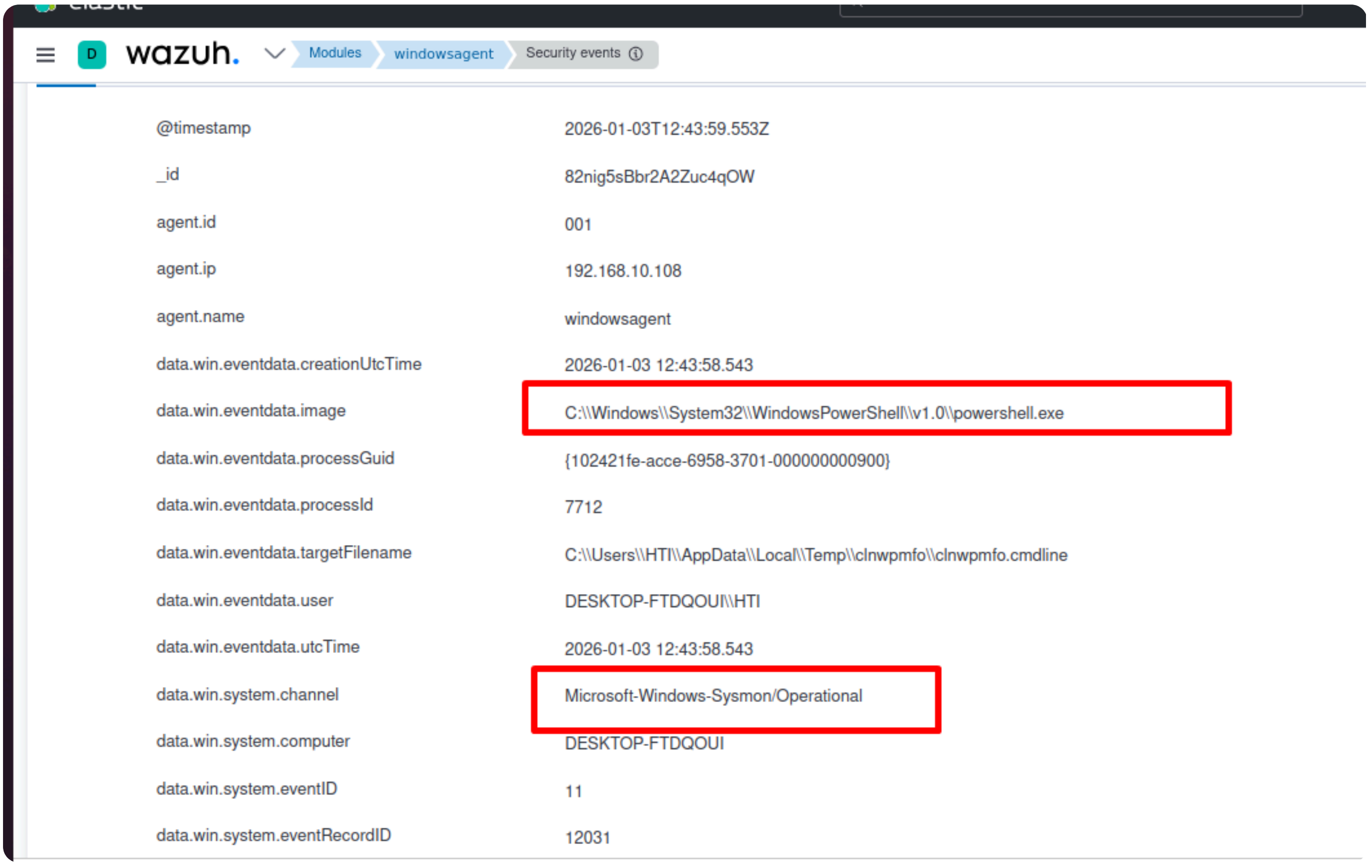




https://192.168.224.128:5601/app/wazuh#/overview/?tab=general&tabView=panels&_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from
90%
elastic
Search Elastic
wazuh.
Modules
windowsagent
Security events
Registry Value Integrity Ch...
A net.exe account discover...
Registry Key Integrity Chec...
sysmon_eid1_detections
windows_security
authentication_success
10.6.1
10.2.6
11.4

Security Alerts

Time ↓	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jan 3, 2026 @ 04:50:33.735	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	Windows logon success.	3	60106
Jan 3, 2026 @ 04:49:38.189	T1105	Command and Control	Executable file dropped in folder commonly used by malware	15	92213
Jan 3, 2026 @ 04:49:31.119	T1070	Defense Evasion	A Windows log file was cleared	5	63104
Jan 3, 2026 @ 04:43:59.553	T1105	Command and Control	Executable file dropped in folder commonly used by malware	15	92213



@timestamp	2026-01-03T12:43:59.553Z
_id	82nig5sBbr2A2Zuc4qOW
agent.id	001
agent.ip	192.168.10.108
agent.name	windowsagent
data.win.eventdata.creationUtcTime	2026-01-03 12:43:58.543
data.win.eventdata.image	C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe
data.win.eventdata.processGuid	{102421fe-acce-6958-3701-000000000900}
data.win.eventdata.processId	7712
data.win.eventdata.targetFilename	C:\\Users\\HTI\\AppData\\Local\\Temp\\clnwpmfo\\clnwpmfo.cmdline
data.win.eventdata.user	DESKTOP-FTDQOUI\\HTI
data.win.eventdata.utcTime	2026-01-03 12:43:58.543
data.win.system.channel	Microsoft-Windows-Sysmon/Operational
data.win.system.computer	DESKTOP-FTDQOUI
data.win.system.eventID	11
data.win.system.eventRecordID	12031

EXECUTIVE SUMMARY & CONCLUSION

This report presents a verified, multi-layered security response to a verified malware infection event. By combining **Network Analysis (Wireshark)**, **Traffic Enforcement (Suricata/IPTables)**, and **Behavioral Endpoint Monitoring (Wazuh/Sysmon)**, we have demonstrated a full-cycle Incident Response workflow.

The core success of this project lies in **Defense in Depth**: identifying threats at the network boundary while maintaining deep visibility into the endpoint to prevent high-impact actions like credential theft. This approach represents the gold standard for protecting a modern enterprise against sophisticated cyber-attacks.