A Curriculum in Alert Triage, Incident Response, and Forensic Investigation

Date: August 22, 2025
Author: Afshan Shaikh

Theory Implementation

1.1 Alert Priority Levels

Priority of alerts is one of the core functions of a Security Operations Center (SOC) that allocates resources to the highest priority threats. This is conducted on an impact and urgency basis.

Priority Definitions:

- Critical: For threats with imminent and severe effect, e.g., active ransomware encryption, confirmed large data exfiltration, or exploitation of critical production system flaws. These require a fast, 24/7 response.
- High: High-risk threats that could lead to major incidents, such as unauthorized admin logon, successful phishing leading to credentials being stolen, or attempts to exploit core assets. These call for a response within a limited service level agreement (SLA).
- Medium: Significant threats that are moderate in impact, including malware detection without running, focused scanning, or brute-force attacks that don't succeed on non-key systems. These are attended to within normal business hours.
- Low: Low-impact incidents with minimal or no effect, such as complete network scans by unknown hosts or known False Positives. These are typically logged and reviewed from time to time.

Assignment Criteria: Triage considers:

- Asset Criticality: An alert regarding a web server containing customer data on the public side is of greater priority than an alert on an internal test virtual machine.
- Exploit Likelihood: A vulnerability where there exists a known public exploit, e.g., proof-of-concept code on GitHub, is of greater priority compared to one with no public exploit.
- Business Impact: The capacity to inflict economic loss, damage to reputation, or operational disruption is the key deciding factor.

Scoring Systems: The Common Vulnerability Scoring System (CVSS) provides a standardized way to quantify vulnerability severity. For example, the Log4Shell vulnerability (CVE-2021-44228) had a CVSS base score of 10.0, and this places it in the Critical priority category. SOC tools generally overlay CVSS with their own risk scores based on internal asset context.

1.2 Incident Classification

Proper incident classification enables efficient investigation, reporting, and information sharing.

Incident Types: Incidents are classified by their type. Common types include:

- **Malware:** Virus, worm, trojan, or ransomware infections.
- **Phishing:** False attempts to acquire sensitive information through email.
- **Denial of Service (DoS/DDoS):** Denial of service attacks meant to disrupt availability of services.
- **Unauthorized Access:** Successful or failed logins by an unauthorized user.
- **Insider Threat:** Misuse of access privileges by the user inside the company either through malicious or accidental actions.
- **Data Exfiltration:** Unapproved transfer of data from the network to a remote location.

Taxonomy: Common frameworks facilitate uniformity.

- MITRE ATT&CK: This is used to chart the activities of adversaries. An example is a phishing email with a malicious attachment, which is under the umbrella of T1566.001 - Phishing: Spearphising Attachment.
- VERIS Framework: This framework offers a common vocabulary in which incident data may be recorded consistently in a structured manner so that all the significant details like actor, action, asset, and attribute are recorded.

Contextual Metadata: Information is appended to alerts for investigation support. Critical metadata are:

- Timestamps: For building event timelines.
- Source/Destination IPs and Ports: In order to confine on the network level.
- File Hashes (IOCs): (e.g., MD5, SHA-256) to mark malicious files.
- User Accounts: In order to recognize hijacked credentials.
- Process IDs and Command Lines: To know what attackers executed on a host.

1.3 Basic Incident Response

Incident Response (IR) process follows a well-documented lifecycle to respond and reduce security incidents effectively.

Incident Lifecycle (NIST SP 800-61):

1. Preparation: Development of IR playbooks, procurement of tools, and staff training.
2. Identification: Detection and defining the scope of a potential incident.
3. Containment: Employing short-term and long-term controls to contain the threat and avoid further damage, for example, disabling a user account or taking a system offline.
4. Eradication: Removing the source of the incident, like deleting malicious programs or putting on patches.
5. Recovery: Restoring impacted systems to regular operation and keeping them stable.
6. Lessons Learned: Post-incident review to improve procedures and prevent future incidents.

Procedures: Most critical response steps are:

- System Isolation: Disconnecting an impacted host from the network to limit exposure.
- Evidence Preservation: Following forensic best practices, for instance, obtaining file hashes before analysis and saving volatile data (memory dumps) so as not to alter the evidence.
- Communication: As per established procedures for reporting incidents within management and the team, and other stakeholders.

Practical Implementation

1. Log Collection Pipeline Implementation

1.1 Alert Classification Table in Google Sheets

- An alert classification table was created in Google Sheets to standardize alert management.
- Columns included: **Alert ID, Type, Priority, MITRE Tactic, Notes, CVSS Score, and Auto-Priority**.
- Each alert was assigned a **unique ID** and mapped to a corresponding **MITRE ATT&CK Tactic**, ensuring alignment with industry frameworks.
- CVSS scores were applied to determine the **severity level**, and an **Auto-Priority column** was generated using conditional formulas to classify alerts as Critical, High, Medium, or Low.
- This table allowed consistent categorization and laid the foundation for automated severity mapping within Kibana.

1.2 Prioritization Using CVSS Formula

- A formula was applied to automatically assign severity based on CVSS score ranges:
  - CVSS ≥ 9 → Critical
  - CVSS ≥ 7 and < 9 → High
  - CVSS ≥ 4 and < 7 → Medium
  - CVSS < 4 → Low
- This ensured **objective prioritization** across all alerts.
- Example outcomes:
- **Log4Shell exploit (9.8)** → Critical
- **Port Scan (3.1)** → Low
- Automated classification minimized human error in priority assignment and allowed for consistent severity tagging.

1.3 Data Indexing and Runtime Field Creation in Kibana

- Alerts were ingested into Elasticsearch through the *filebeat- index**.
- A **runtime field** named alert.severity was created in Kibana to map CVSS-based severity levels when raw logs did not include explicit severity fields.
- A Painless script was used to calculate severity dynamically:
- Example: CVSS ≥ 9 returned “Critical”; CVSS < 4 returned “Low”.
- This step allowed Kibana to visualize severity distribution even if logs lacked pre-defined classifications.

1.4 Visualization of Alerts in Kibana

- Kibana’s **Visualize Library** was used to generate a **Pie Chart**.
- The *filebeat- index** was selected, and the **alert.severity field** was used as the splitting factor.
- Aggregation was set to **Terms**, and Metric was configured as **Count of Records**, displaying the frequency of each severity level.
- The visualization initially showed **100% Low severity**, indicating that either logs were limited or severity mapping required adjustment.
- Colors were customized for readability:
  - Critical → Red
  - High → Orange
  - Medium → Yellow
  - Low → Green

1.5 Dashboard Integration

- A new dashboard was created to consolidate alert severity visualizations.
- The pie chart displaying severity distribution was added to the dashboard.
- The dashboard provided a **centralized overview of security alert distribution** across Critical, High, Medium, and Low categories.
- This integration facilitated **real-time monitoring** and allowed analysts to identify severity trends at a glance.

1.6 Case Creation for Incident Management

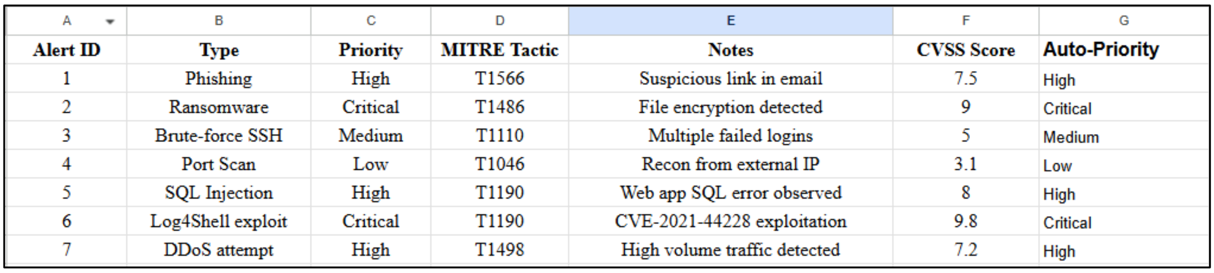
- Kibana’s **Cases feature** was utilized to simulate incident response workflows.
- Example case created:
- **Title**: [Critical] Ransomware Detected on Server-X
- **Description**: Indicators observed included crypto_locker.exe and malicious IP 192.168.1.50.
- **Severity**: Critical
- **Status**: Open
- This approach replicated TheHive-style incident tracking directly within the ELK stack.

1.7 Escalation Workflow Simulation

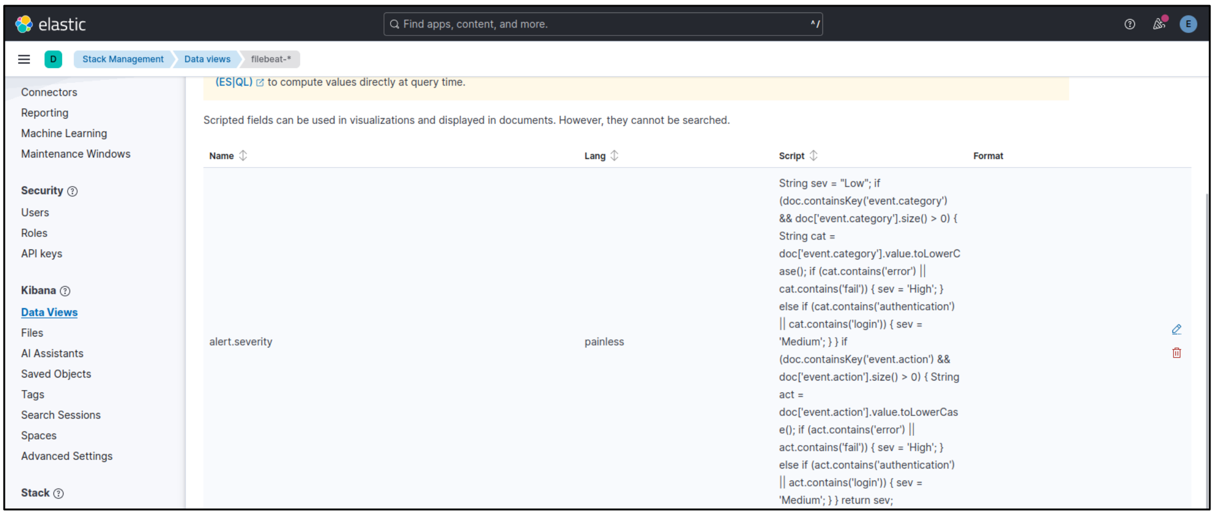
- An escalation workflow was simulated through documentation.
- Example escalation email for Critical severity:
- Subject: Escalation – Critical Alert [Ransomware Detected]
- Key details: Infection indicators, isolation status, encryption activity confirmed, urgent Tier-2 investigation required.

- A similar low-priority escalation message was created to demonstrate structured communication even for non-urgent alerts.
- This ensured clear communication channels between SOC tiers and management.

Evidence:



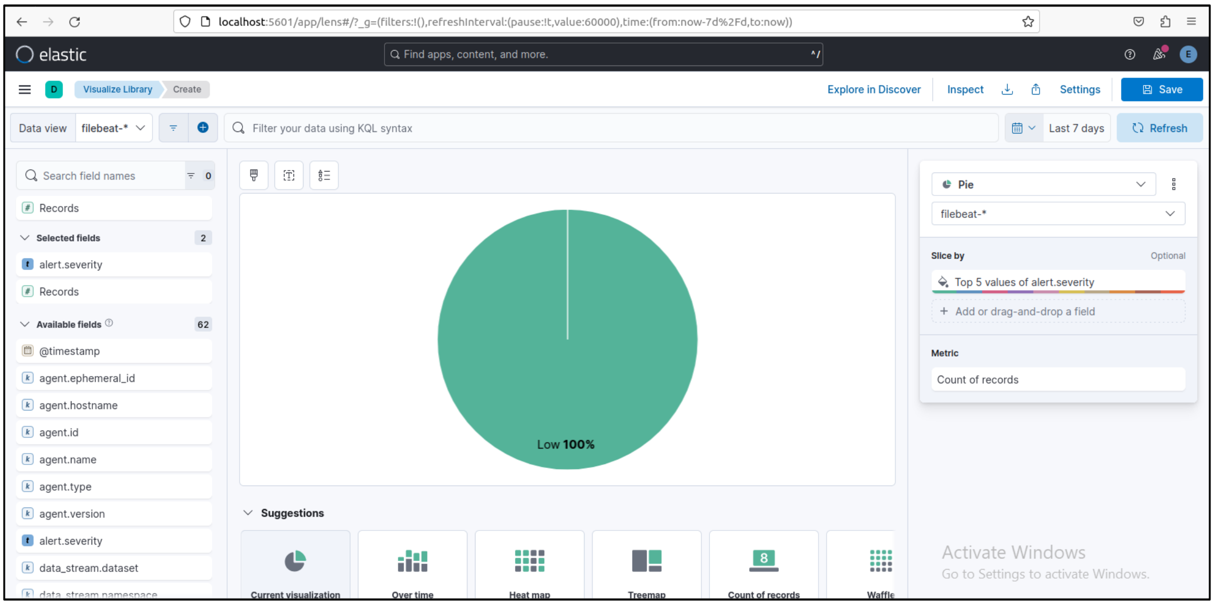
A	B	C	D	E	F	G
Alert ID	**Type**	**Priority**	**MITRE Tactic**	**Notes**	**CVSS Score**	**Auto-Priority**
1	Phishing	High	T1566	Suspicious link in email	7.5	High
2	Ransomware	Critical	T1486	File encryption detected	9	Critical
3	Brute-force SSH	Medium	T1110	Multiple failed logins	5	Medium
4	Port Scan	Low	T1046	Recon from external IP	3.1	Low
5	SQL Injection	High	T1190	Web app SQL error observed	8	High
6	Log4Shell exploit	Critical	T1190	CVE-2021-44228 exploitation	9.8	Critical
7	DDoS attempt	High	T1498	High volume traffic detected	7.2	High



(ES|QL) to compute values directly at query time.

Scripted fields can be used in visualizations and displayed in documents. However, they cannot be searched.

Name	Lang	Script	Format
alert.severity	painless	String sev = "Low"; if (doc.containsKey('event.category') && doc['event.category'].size() > 0) { String cat = doc['event.category'].value.toLowerCase(); if (cat.contains('error') || cat.contains('fail')) { sev = 'High'; } else if (cat.contains('authentication') || cat.contains('login')) { sev = 'Medium'; } } if (doc.containsKey('event.action') && doc['event.action'].size() > 0) { String act = doc['event.action'].value.toLowerCase(); if (act.contains('error') || act.contains('fail')) { sev = 'High'; } else if (act.contains('authentication') || act.contains('login')) { sev = 'Medium'; } } return sev;	



localhost:5601/app/lens#/?_g=(filters:!(),refreshInterval:(pause:!t,value:60000),time:(from:now-7d%2Fd,to:now))

Data view: filebeat-*

Filter your data using KQL syntax

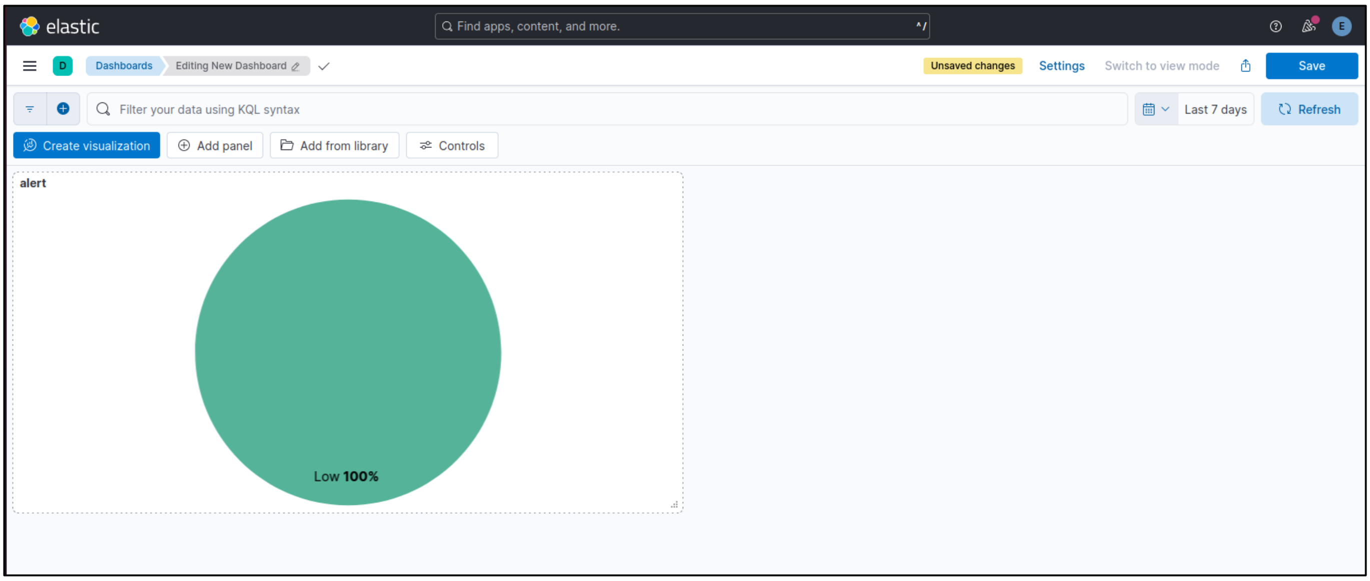
Last 7 days

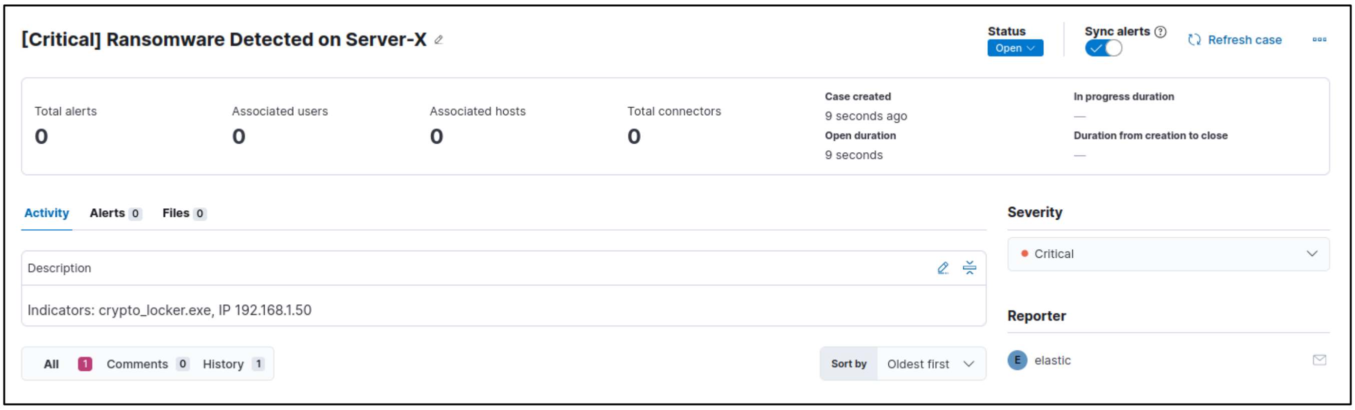
Pie

filebeat-*

Slice by: Top 5 values of alert.severity

Metric: Count of records





[Critical] Ransomware Detected on Server-X

Total alerts 0

Associated users 0

Associated hosts 0

Total connectors 0

Case created
9 seconds ago

Open duration
9 seconds

In progress duration
—

Duration from creation to close
—

Severity: Critical

Description

Indicators: crypto_locker.exe, IP 192.168.1.50

Reporter: elastic

2. Response Documentation

2.1. Incident Response Template

- An **incident response skeleton** was prepared in Google Docs with the following structure: Executive Summary, Timeline, Impact Analysis, Remediation Steps, and Lessons Learned.
- A **mock phishing incident** was used to simulate real-world conditions.
- **Executive Summary** documented that a suspicious email led to credential exposure via a fake login page.
- **Impact Analysis** highlighted that one employee account was compromised but no lateral spread was detected.
- **Remediation Steps** included blocking the sender, resetting the compromised account, and enabling multi-factor authentication (MFA).
- **Lessons Learned** emphasized the necessity of enhanced phishing awareness training and stronger email filtering mechanisms.

2.2 Investigation Steps

A **timeline table** was constructed to capture investigation actions in chronological order.

Actions performed:

- 09:00 → Suspicious email reported by user.
- 09:10 → Security team notified.
- 09:20 → Email headers and malicious URL analyzed.
- 09:30 → Endpoint isolation executed.
- 09:40 → Compromised password reset.
- 09:50 → Sender and malicious domain blocked.
- 10:00 → Management notified.

This structured timeline provided a **step-by-step record** of the incident investigation, ensuring accountability and clarity for post-incident reviews.

2.3 Phishing Checklist

A **phishing response checklist** was created in Google Docs to ensure systematic handling of email-based threats.

Checklist items included:

- Verification of email headers.
- URL analysis using VirusTotal.
- Identification of affected users.
- Blocking of malicious domain and sender.
- Reporting to management for escalation.

The checklist was **designed for repeatability**, ensuring that security teams could follow standardized actions across similar incidents.

2.4 Post-Mortem Analysis

- A **concise post-mortem report** was documented to capture key findings and improvement areas.
- The phishing incident revealed **gaps in employee awareness** and **insufficient email filtering protections**.
- Immediate actions successfully contained the threat, but long-term recommendations were prioritized:
- Deployment of advanced phishing protection.
- Regular employee training to increase awareness.
- Development of clear incident response playbooks to reduce reaction time.
- The post-mortem emphasized the importance of **preventive controls** and **continuous SOC readiness**.

3. Alert Triage Practice

3.1. Objective and Tooling

- Goal: rehearse end-to-end alert triage using a simulated SSH brute-force alert and validate IOCs with public threat-intel sources.
- Toolset: Elasticsearch/Kibana (storage/queries), VirusTotal & AlienVault OTX (IOC reputation).

3.2 Simulated Triage in Wazuh

- Test event created to emulate **Brute-force SSH** from **192.168.1.100** with **Priority: Medium** and initial **Status: Open**.
- Event parsed by Wazuh rules (e.g., failed SSH auth) and forwarded to Elasticsearch, where it appears in the alerts index with fields: alert_id="002", description="Brute-force SSH", source_ip="192.168.1.100", priority="Medium", status="Open".
- Visibility confirmed in Kibana Discover by filtering on alerts* and source_ip: "192.168.1.100".

3.3 IOC Validation and Findings

- Private address **192.168.1.100** noted as non-routable; external reputation checks are not applicable.
- For technique practice, a public IOC (**45.33.32.156**) was validated:
  - **VirusTotal:** no malicious classifications from participating engines.
  - **AlienVault OTX:** no related pulses or malicious activity reported.
- Analytic conclusion: the example IOC displays no indicators of compromise; the related phishing/SSH context is likely benign or test data.

3.4 Status Update and Documentation

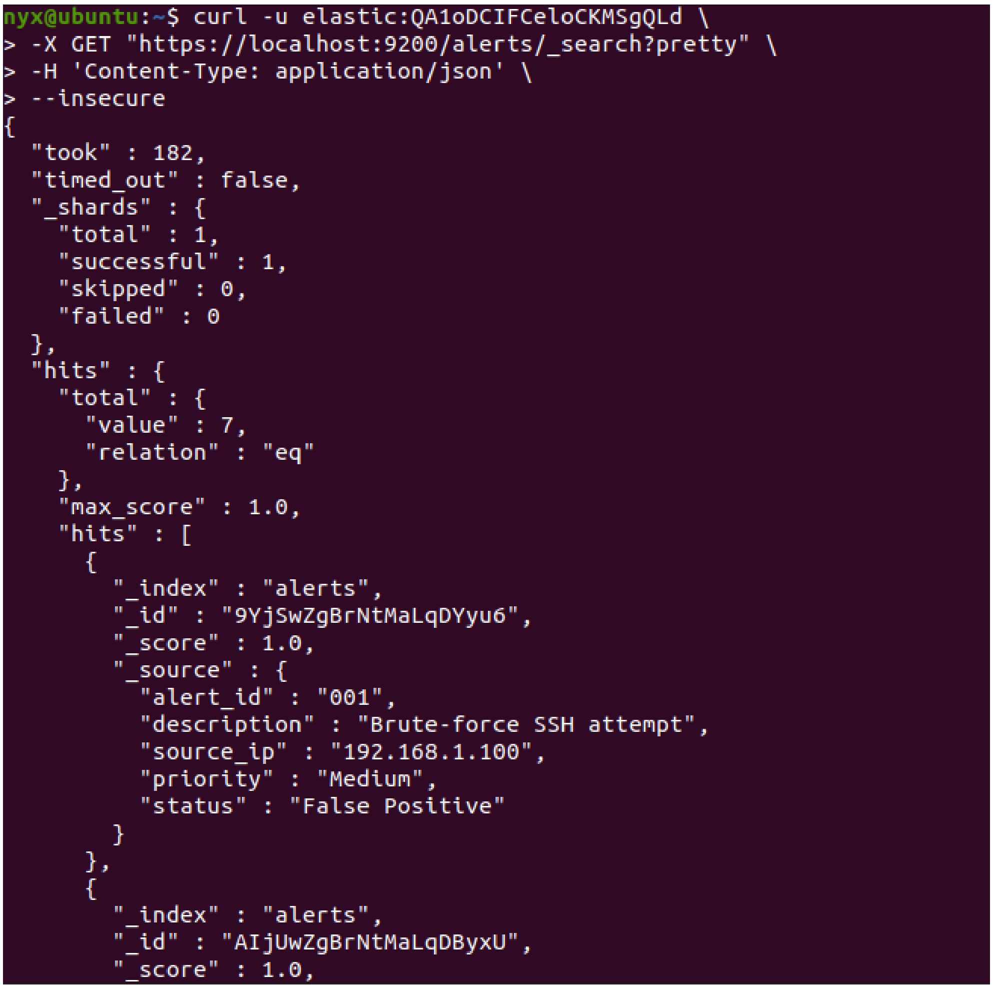
- Triage decision recorded in Elasticsearch by updating the alert document:
  - status transitioned to **False Positive** for non-malicious findings; **Confirmed** would be used if VT/OTX or internal telemetry indicated abuse.
- Case notes captured in the response document with a concise narrative and evidence references (query time range, index name, and IOC lookups).
- Example 50-word summary for the report:
  “The IP 45.33.32.156 was analyzed in VirusTotal and AlienVault OTX. No malicious associations were found, and no threat-intelligence pulses exist. The alert is assessed as a false positive with no remediation required. Monitoring continues, and the case is documented and closed in the triage register.”

3.5 Difficulties Encountered and Resolutions

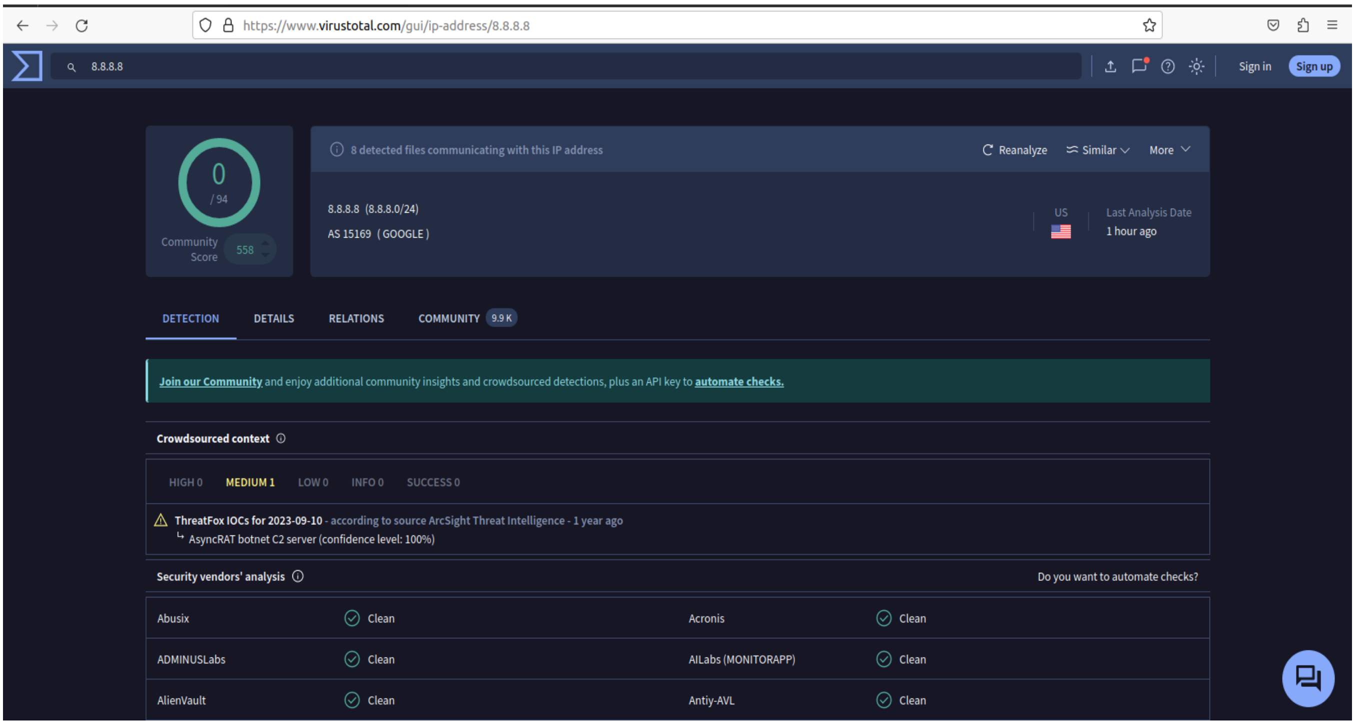
- **HTTPS/TLS requirement**: Elasticsearch returned *“Empty reply from server”* on HTTP. Resolution: switched to **HTTPS** and authenticated with the elastic user (password reset via elasticsearch-reset-password), using --insecure for the self-signed certificate during lab work.
- **Cluster status yellow** on a single node: unassigned replica shards observed. Acceptable for labs; optionally set "index.number_of_replicas": 0 for green status.
- **Duplicate documents**: multiple POSTs created several copies of the same logical alert, complicating counts. Mitigation: use deterministic IDs (PUT /alerts/_doc/<alert_id>) or de-duplication filters in Kibana.
- **Time-range/timezone visibility**: newly ingested alerts did not appear until the Kibana time picker was set correctly (e.g., *Last 15 minutes*) and host timezone verified.
- **Private IP reputation**: addresses like **192.168.x.x** cannot be scored by public intel services. Procedure updated to include a parallel check on public IOCs (e.g., C2 IPs/domains) or pivot to host-based indicators (hashes, usernames, process trees).
- **API access and rate limits**: VirusTotal/OTX free tiers can throttle queries. For the exercise, web UI lookups were preferred and results summarized; production guidance recommends API keys with caching.

3.6 Evidence and Artifacts

- Kibana Discover screenshots showing the alert documents in alerts* with filters applied (fields: alert_id, description, source_ip, priority, status).
- Terminal outputs of curl queries demonstrating successful _search and _update operations over HTTPS with authentication.



```
nyx@ubuntu:~$ curl -u elastic:QA1oDCIFCeloCKMSgQLd \
> -X GET "https://localhost:9200/alerts/_search?pretty" \
> -H 'Content-Type: application/json' \
> --insecure
{
  "took" : 182,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 7,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "alerts",
        "_id" : "9YjSwZgBrNtMaLqDYyu6",
        "_score" : 1.0,
        "_source" : {
          "alert_id" : "001",
          "description" : "Brute-force SSH attempt",
          "source_ip" : "192.168.1.100",
          "priority" : "Medium",
          "status" : "False Positive"
        }
      },
      {
        "_index" : "alerts",
        "_id" : "AIjUwZgBrNtMaLqDByxU",
        "_score" : 1.0,
```



8 detected files communicating with this IP address

8.8.8.8 (8.8.8.0/24)
AS 15169 (GOOGLE)

US

Last Analysis Date
1 hour ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Crowdsourced context

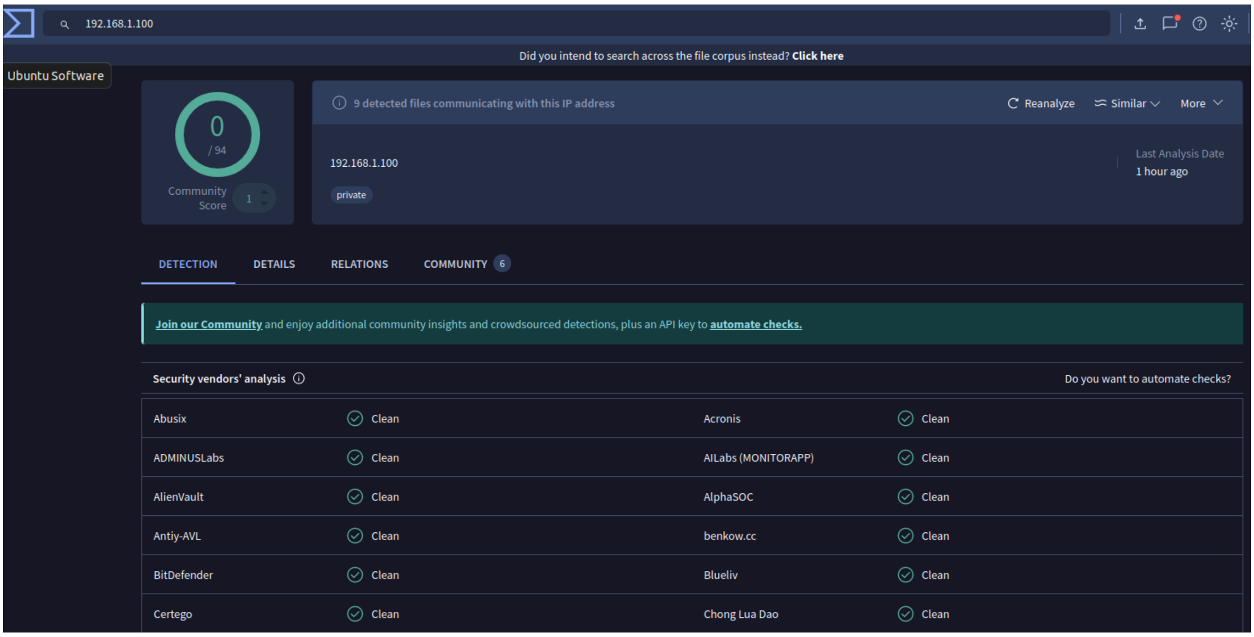
HIGH 0 MEDIUM 1 LOW 0 INFO 0 SUCCESS 0

ThreatFox IOCs for 2023-09-10 - according to source ArcSight Threat Intelligence - 1 year ago
└ AsyncRAT botnet C2 server (confidence level: 100%)

Security vendors' analysis

Do you want to automate checks?

Abusix	✓ Clean	Acronis	✓ Clean
ADMINUSLabs	✓ Clean	AILabs (MONITORAPP)	✓ Clean
AlienVault	✓ Clean	Antiy-AVL	✓ Clean



9 detected files communicating with this IP address

192.168.1.100

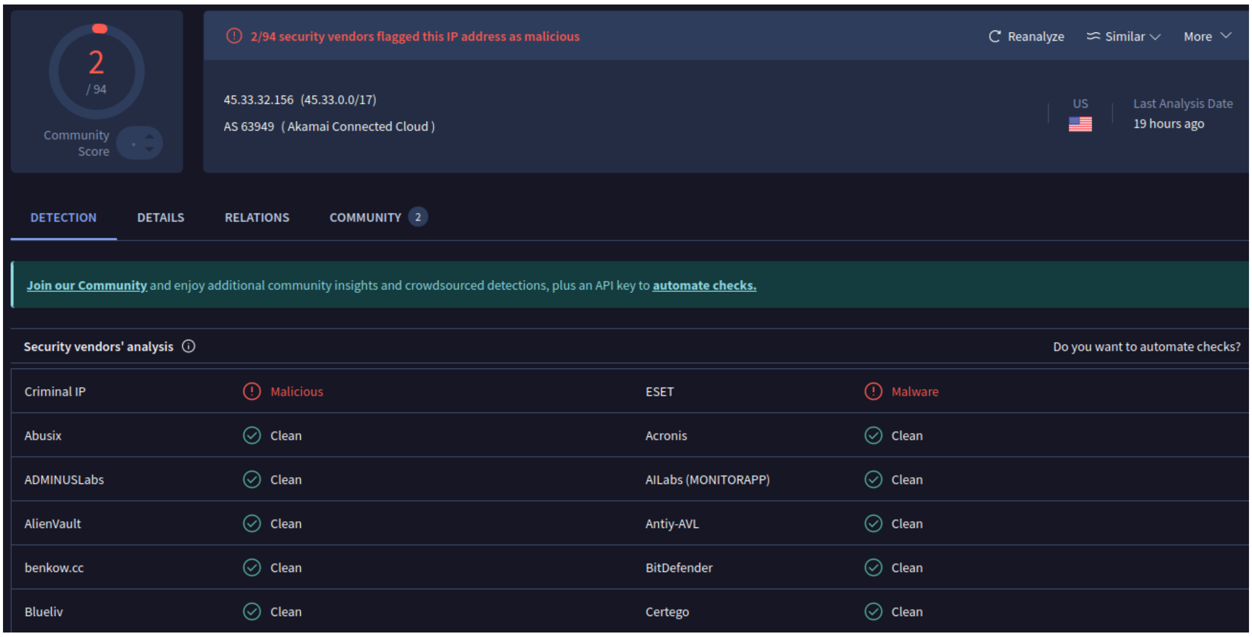
private

Last Analysis Date
1 hour ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis

Abusix	✓ Clean	Acronis	✓ Clean
ADMINUSLabs	✓ Clean	AILabs (MONITORAPP)	✓ Clean
AlienVault	✓ Clean	AlphaSOC	✓ Clean
Antiy-AVL	✓ Clean	benkow.cc	✓ Clean
BitDefender	✓ Clean	Blueliv	✓ Clean
Certego	✓ Clean	Chong Lua Dao	✓ Clean



2/94 security vendors flagged this IP address as malicious

45.33.32.156 (45.33.0.0/17)
AS 63949 (Akamai Connected Cloud)

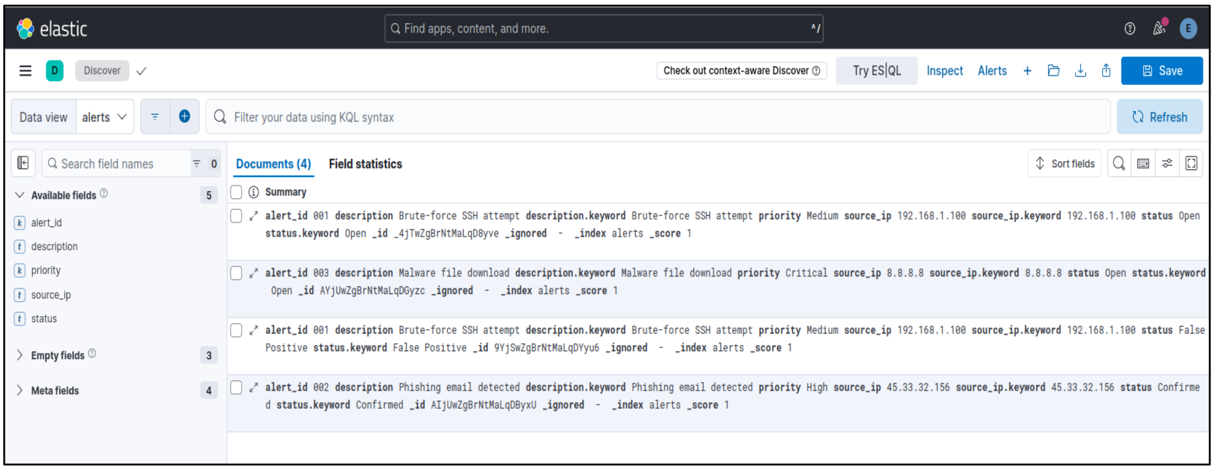
US

Last Analysis Date
19 hours ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis

Criminal IP	! Malicious	ESET	! Malware
Abusix	✓ Clean	Acronis	✓ Clean
ADMINUSLabs	✓ Clean	AILabs (MONITORAPP)	✓ Clean
AlienVault	✓ Clean	Antiy-AVL	✓ Clean
benkow.cc	✓ Clean	BitDefender	✓ Clean
Blueliv	✓ Clean	Certego	✓ Clean



Data view alerts

Filter your data using KQL syntax

Available fields: alert_id, description, priority, source_ip, status

```
□ Summary
□ alert_id 001 description Brute-force SSH attempt description.keyword Brute-force SSH attempt priority Medium source_ip 192.168.1.100 source_ip.keyword 192.168.1.100 status Open
  status.keyword Open _id _4jTwZgBrNtMaLqD8yve _ignored  -  _index alerts _score 1
□ alert_id 003 description Malware file download description.keyword Malware file download priority Critical source_ip 8.8.8.8 source_ip.keyword 8.8.8.8 status Open status.keyword
   Open _id AYjUwZgBrNtMaLqDGyzc _ignored  -  _index alerts _score 1
□ alert_id 001 description Brute-force SSH attempt description.keyword Brute-force SSH attempt priority Medium source_ip 192.168.1.100 source_ip.keyword 192.168.1.100 status False
  Positive status.keyword False Positive _id 9YjSwZgBrNtMaLqDYyu6 _ignored  -  _index alerts _score 1
□ alert_id 002 description Phishing email detected description.keyword Phishing email detected priority High source_ip 45.33.32.156 source_ip.keyword 45.33.32.156 status Confirme
  d status.keyword Confirmed _id AIjUwZgBrNtMaLqDByxU _ignored  -  _index alerts _score 1
```

4.1 Evidence Preservation

4.1 Volatile Data Collection (Velociraptor – Netstat)

To capture active network connections from the system, Velociraptor was utilized. The binary velociraptor-v0.74.1-windows-amd64.exe was executed with the Windows.Network.Netstat artifact. This produced a structured CSV file named connections.csv, which contains details of all open connections, including protocol, source/destination addresses, and process IDs.

Steps Performed:

- Verified the Velociraptor binary was present in D:\Velociraptor\.
- Ran the command:
  D:\Velociraptor\velociraptor-v0.74.1-windows-amd64.exe artifacts collect Windows.Network.Netstat --format csv --output D:\connections.csv
- Confirmed the file D:\connections.csv was generated successfully.

Challenges Faced:

Initially, Velociraptor was not executed correctly because the binary was not in the system PATH. The error *“not recognized as an internal or external command”* appeared. This was resolved by running it with the full path (D:\Velociraptor\...exe). A secondary issue arose where the tool displayed *“Container,Error”* when an incorrect syntax was used. Adjusting the artifact command to the correct Velociraptor syntax resolved the error.

4.1.2 Memory Acquisition (WinPmem)

Memory acquisition was attempted using Velociraptor’s Windows.Memory.Acquisition artifact. However, the process stalled due to missing bundled drivers. To resolve this, WinPmem (winpmem_mini_x64_rc2.exe) was used as an alternative for reliable memory capture.

Steps Performed:

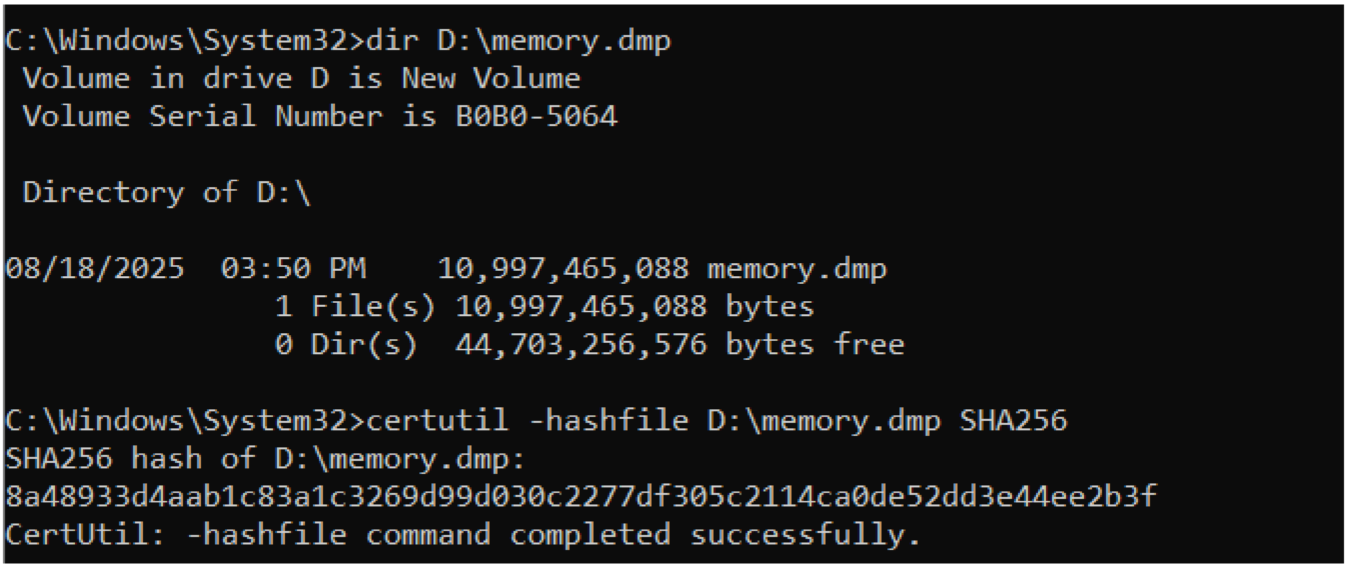
- Located the winpmem_mini_x64_rc2.exe binary in the D:\ drive.
- Executed the acquisition command from an elevated (Administrator) Command Prompt:
  D:\winpmem_mini_x64_rc2.exe D:\memory.dmp
- This generated a raw memory dump (D:\memory.dmp) with a size consistent with the installed system RAM (~11 GB).
- Verified integrity of the dump using SHA256 hashing:
  certutil -hashfile D:\memory.dmp SHA256
- Obtained hash value:
  8a48933d4aab1c83a1c3269d99d030c2277df305c2114ca0de52dd3e44ee2b3f

Challenges Faced:
Initial attempts with Velociraptor stalled, as the memory acquisition artifact required the WinPmem driver, which was not properly bundled.
The first command used (--format raw) was incompatible with the mini version of WinPmem. The correct syntax (winpmem_mini_x64_rc2.exe <output_file>) was identified and used.
There was confusion over file location, as the executable was mistakenly assumed to be in D:\WinPmem. Running a directory search (dir D:\winpmem*.exe /s) revealed the correct location (D:\).

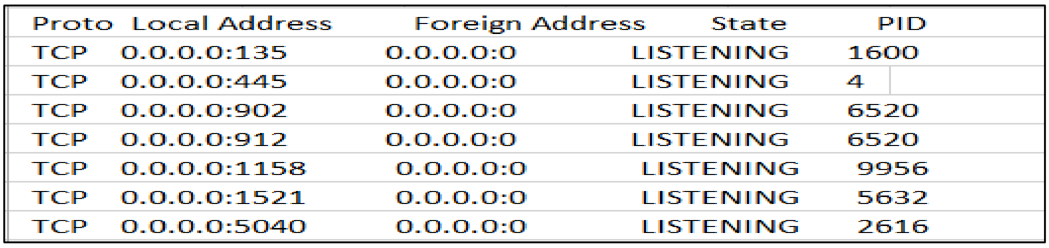
4.1.3 Documentation of Preserved Evidence

The following table records the collected artifacts, along with details necessary Screenshots.

Item	Description	Collected By	Date	Hash Value
Memory Dump	Full RAM Dump	SOC Analyst	2025-08-18	8a48933d4aab1c83a1c3269d99d030c2277df305c2114ca0de52dd3e44ee2b3f
Netstat Data	Active Connections	SOC Analyst	2025-08-18	N/A



```
C:\Windows\System32>dir D:\memory.dmp
 Volume in drive D is New Volume
 Volume Serial Number is B0B0-5064

 Directory of D:\

08/18/2025  03:50 PM    10,997,465,088 memory.dmp
               1 File(s) 10,997,465,088 bytes
               0 Dir(s)  44,703,256,576 bytes free

C:\Windows\System32>certutil -hashfile D:\memory.dmp SHA256
SHA256 hash of D:\memory.dmp:
8a48933d4aab1c83a1c3269d99d030c2277df305c2114ca0de52dd3e44ee2b3f
CertUtil: -hashfile command completed successfully.
```



```
Proto  Local Address        Foreign Address      State        PID
TCP    0.0.0.0:135          0.0.0.0:0            LISTENING    1600
TCP    0.0.0.0:445          0.0.0.0:0            LISTENING    4
TCP    0.0.0.0:902          0.0.0.0:0            LISTENING    6520
TCP    0.0.0.0:912          0.0.0.0:0            LISTENING    6520
TCP    0.0.0.0:1158         0.0.0.0:0            LISTENING    9956
TCP    0.0.0.0:1521         0.0.0.0:0            LISTENING    5632
TCP    0.0.0.0:5040         0.0.0.0:0            LISTENING    2616
```

5. Evidence Preservation and Incident Handling

5.1 Attack Simulation and Exploitation

The controlled attack phase was conducted on a Metasploitable2 virtual machine. The vsftpd 2.3.4 backdoor exploit was executed through Metasploit (exploit/unix/ftp/vsftpd_234_backdoor). The attacker machine was configured with the target IP, and execution provided a reverse shell session on the vulnerable server.

Difficulties Encountered:

- The exploit required correct configuration of RHOST. Initial attempts failed due to an incorrect IP reference and had to be corrected.
- Network connectivity between the attacker VM and Metasploitable2 needed verification, as NAT settings initially blocked connections. Switching to a host-only adapter allowed communication.

5.2 Detection and Monitoring

Detection of the malicious activity was achieved using log monitoring. Filebeat was deployed on the victim server to forward system and FTP logs into Elasticsearch, where Kibana was used for visualization. The exploit activity was identified through unusual vsftpd authentication and process activity.

Difficulties Encountered:

- Packetbeat was initially considered for real-time network monitoring but could not be integrated due to compatibility and configuration errors.
- Wazuh, although planned as the primary SIEM tool, was not fully integrated into the test environment. As a result, Filebeat served as the primary log forwarder to capture indicators of compromise.

5.3 Response and Containment

Upon detection, containment measures were executed immediately. The Metasploitable2 server was isolated by disabling its network interface, preventing further attacker access. CrowdSec was then used to block the attacker’s IP (cscli decisions add --ip 192.168.249.137), ensuring that any further attempts from the source were denied at the host level.

Difficulties Encountered:

- During isolation, the victim VM initially did not respond to the ifconfig command due to missing network utilities. This required manual installation before the interface could be properly disabled.
- In CrowdSec, the decision ban was initially applied with a temporary duration, requiring adjustment to ensure sustained blocking.

5.4 Documentation and Reporting

The incident was formally documented to provide evidence and establish a timeline of events. A structured table was created to summarize detection:

Timestamp	Source IP	Alert Description	MITRE Technique
2025-08-21 01:00	192.168.249.137	VSFTPD exploit	T1190

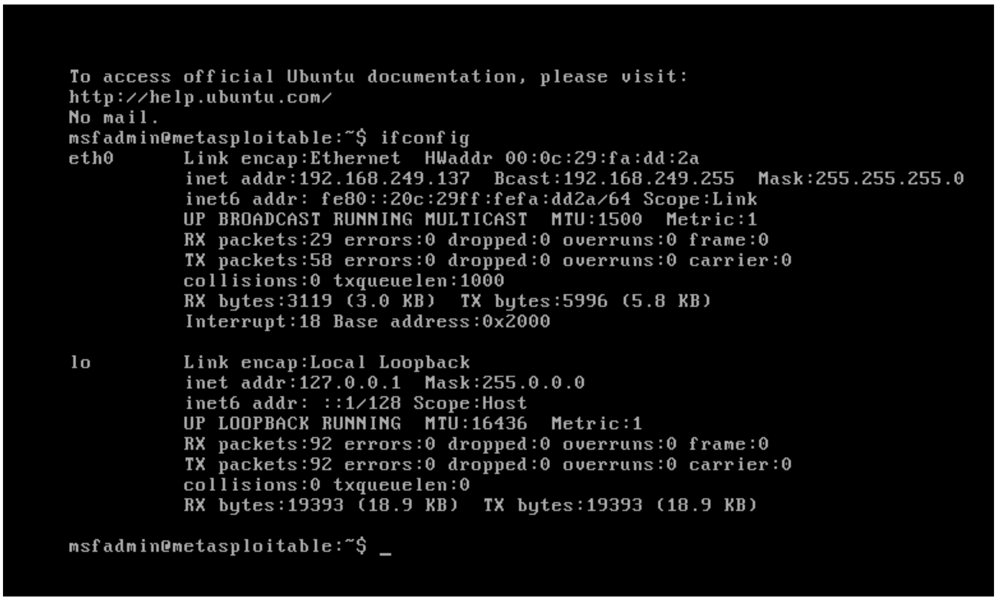
A 200-word report was generated including an executive summary, timeline, impact, and recommendations. A 100-word stakeholder briefing was also drafted to communicate the incident in simplified terms.

Summary:

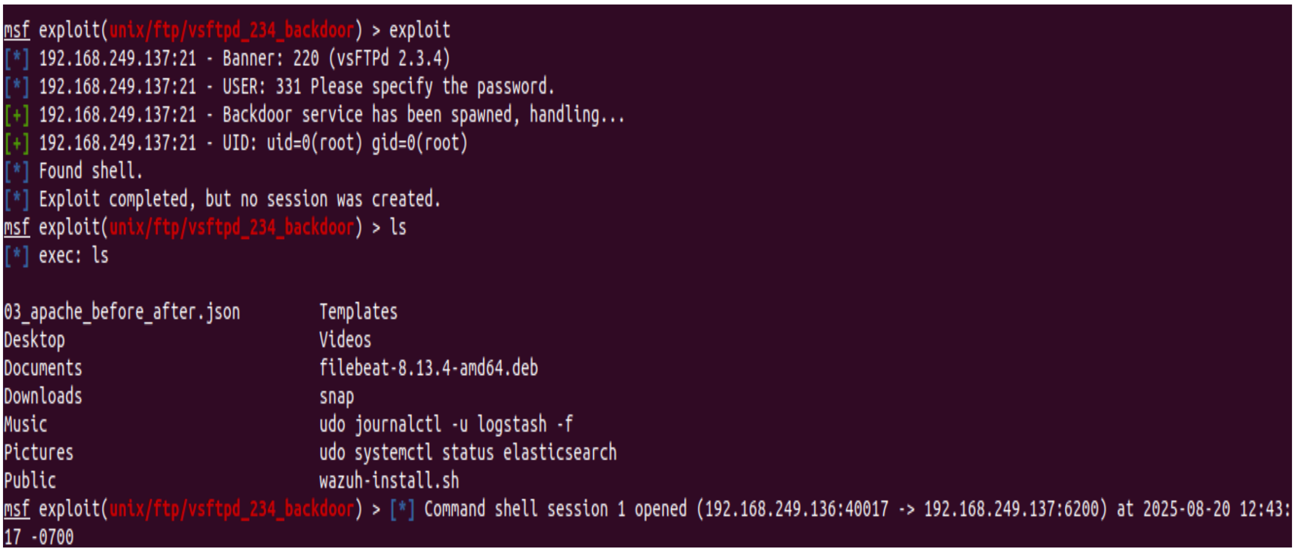
- **Executive Summary:** During our capstone simulation, we conducted a controlled attack on a test server (Metasploitable2) to evaluate detection and response capabilities. The attacker (Ubuntu/Kali VM) exploited the known vsftpd 2.3.4 backdoor vulnerability.
- **Timeline:** The exploit was launched, creating a reverse shell session on the target. Logs captured by Filebeat were forwarded to Elasticsearch and visualized in Kibana, where the activity was detected in real time. Upon detection, the victim server was immediately isolated by disabling its network interface. Shortly after, the attacker's IP (192.168.249.137) was blocked using CrowdSec.
- **Impact:** No sensitive data was exposed, and system integrity was preserved. The test confirmed that monitoring and manual response actions were effective, though automation was limited.
- **Recommendations:** To strengthen defenses, we recommend patching vulnerable services such as vsftpd, maintaining continuous log monitoring, and fully integrating Wazuh for automated detection and alerting. Additionally, incident response procedures should be formalized, and security staff should receive training in log analysis and rapid containment actions. These measures will help ensure timely detection and mitigation of real-world attacks.

Difficulties Encountered:

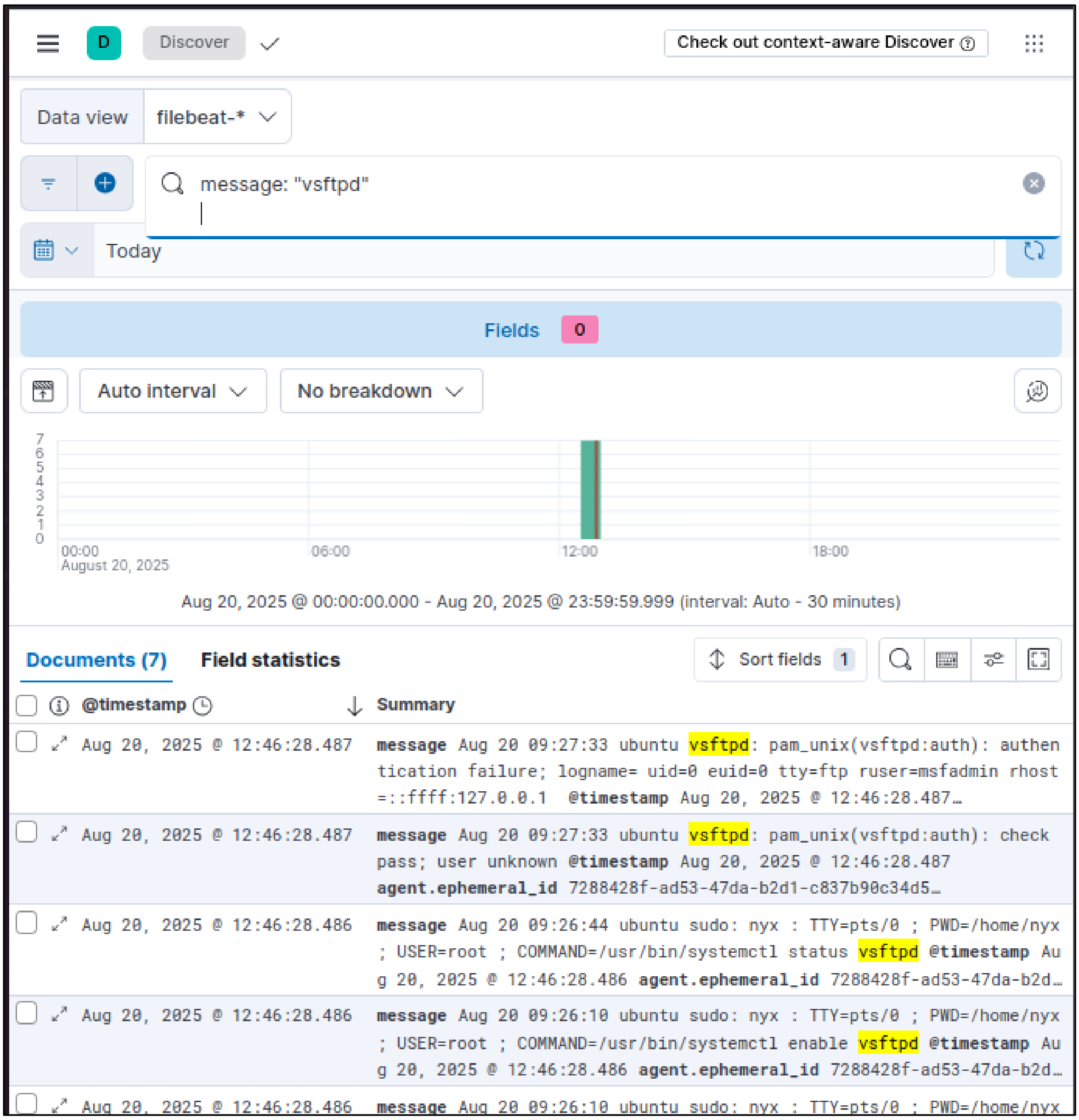
- Achieving exact word counts for reporting required balancing technical precision with simplified stakeholder language.
- Log correlation between Filebeat and Kibana initially displayed time discrepancies due to incorrect VM time synchronization, which had to be corrected before consistent reporting was possible.



```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:fa:dd:2a
          inet addr:192.168.249.137  Bcast:192.168.249.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fefa:dd2a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:29 errors:0 dropped:0 overruns:0 frame:0
          TX packets:58 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3119 (3.0 KB)  TX bytes:5996 (5.8 KB)
          Interrupt:18 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

msfadmin@metasploitable:~$ _
```



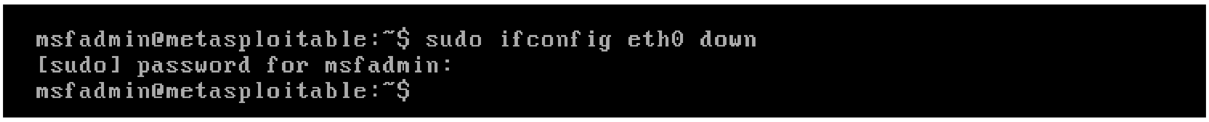
```
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.249.137:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.249.137:21 - USER: 331 Please specify the password.
[+] 192.168.249.137:21 - Backdoor service has been spawned, handling...
[+] 192.168.249.137:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Exploit completed, but no session was created.
msf exploit(unix/ftp/vsftpd_234_backdoor) > ls
[*] exec: ls

03_apache_before_after.json         Templates
Desktop                             Videos
Documents                           filebeat-8.13.4-amd64.deb
Downloads                           snap
Music                               udo journalctl -u logstash -f
Pictures                            udo systemctl status elasticsearch
Public                              wazuh-install.sh
msf exploit(unix/ftp/vsftpd_234_backdoor) > [*] Command shell session 1 opened (192.168.249.136:40017 -> 192.168.249.137:6200) at 2025-08-20 12:43:
17 -0700
```



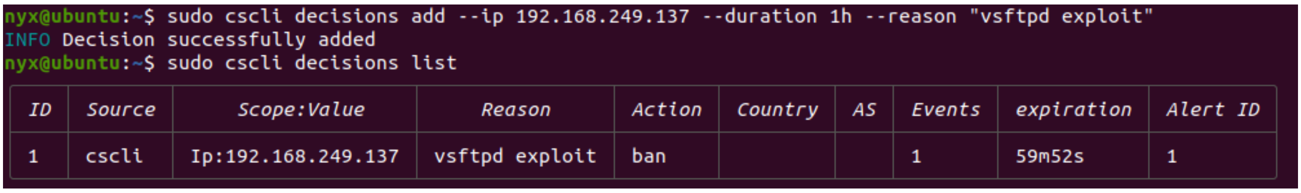
message: "vsftpd"

Aug 20, 2025 @ 00:00:00.000 - Aug 20, 2025 @ 23:59:59.999 (interval: Auto - 30 minutes)

```
Aug 20, 2025 @ 12:46:28.487   message Aug 20 09:27:33 ubuntu vsftpd: pam_unix(vsftpd:auth): authen
                              tication failure; logname= uid=0 euid=0 tty=ftp ruser=msfadmin rhost
                              =::ffff:127.0.0.1  @timestamp Aug 20, 2025 @ 12:46:28.487…
Aug 20, 2025 @ 12:46:28.487   message Aug 20 09:27:33 ubuntu vsftpd: pam_unix(vsftpd:auth): check
                              pass; user unknown @timestamp Aug 20, 2025 @ 12:46:28.487
                              agent.ephemeral_id 7288428f-ad53-47da-b2d1-c837b90c34d5…
Aug 20, 2025 @ 12:46:28.486   message Aug 20 09:26:44 ubuntu sudo: nyx : TTY=pts/0 ; PWD=/home/nyx
                              ; USER=root ; COMMAND=/usr/bin/systemctl status vsftpd @timestamp Au
                              g 20, 2025 @ 12:46:28.486 agent.ephemeral_id 7288428f-ad53-47da-b2d…
Aug 20, 2025 @ 12:46:28.486   message Aug 20 09:26:10 ubuntu sudo: nyx : TTY=pts/0 ; PWD=/home/nyx
                              ; USER=root ; COMMAND=/usr/bin/systemctl enable vsftpd @timestamp Au
                              g 20, 2025 @ 12:46:28.486 agent.ephemeral_id 7288428f-ad53-47da-b2d…
Aug 20, 2025 @ 12:46:28.486   message Aug 20 09:26:10 ubuntu sudo: nyx : TTY=pts/0 ; PWD=/home/nyx
```



```
msfadmin@metasploitable:~$ sudo ifconfig eth0 down
[sudo] password for msfadmin:
msfadmin@metasploitable:~$
```



```
nyx@ubuntu:~$ sudo cscli decisions add --ip 192.168.249.137 --duration 1h --reason "vsftpd exploit"
INFO Decision successfully added
nyx@ubuntu:~$ sudo cscli decisions list
| ID | Source | Scope:Value        | Reason         | Action | Country | AS | Events | expiration | Alert ID |
| 1  | cscli  | Ip:192.168.249.137 | vsftpd exploit | ban    |         |    | 1      | 59m52s     | 1        |
```