Theoretical Knowledge

Core Concepts:

Priority Definitions

- **Understand severity levels** such as **Critical, High, Medium, Low** — typically defined by:
  - **Impact** (e.g., data breach, service disruption)
  - **Urgency** (e.g., active exploitation)
  - **Example mapping**:
    - **Critical**: ransomware encryption, immediate business shutdown
    - **High**: unauthorized administrative access
    - **Medium/Low** follow similar logic based on business and technical evaluation

---

2. Assignment Criteria — Alert Prioritization

When ranking alerts, consider:

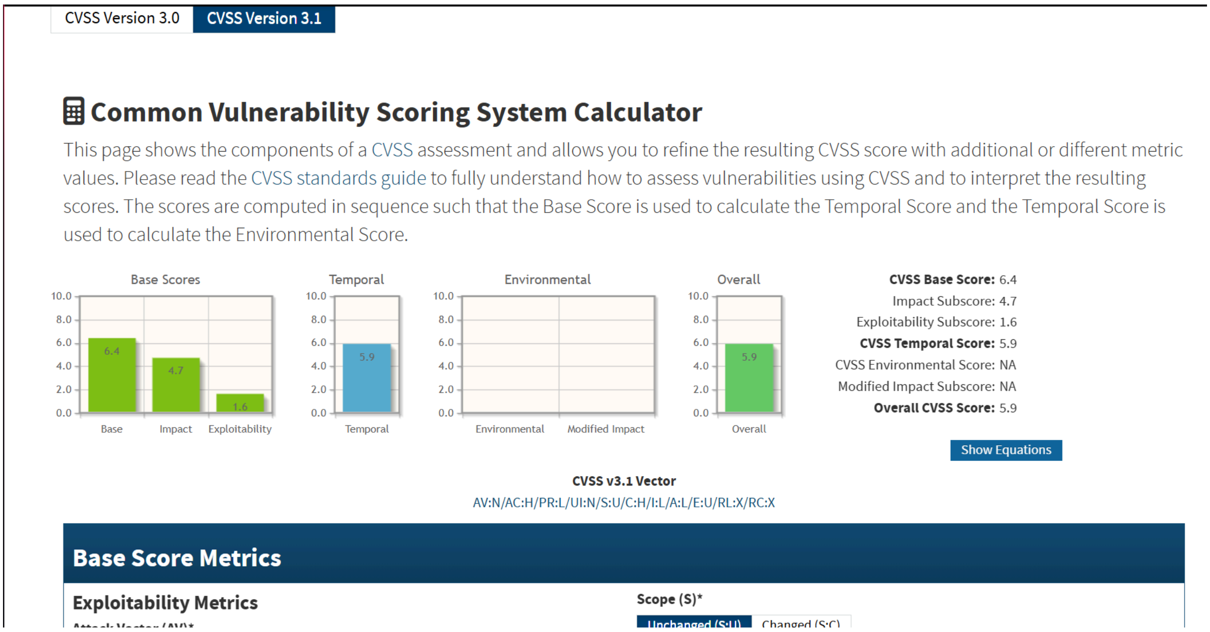
- **Asset Criticality**: production vs. test systems
- **Exploit Likelihood**: known CVE, public proof-of-concept
- **Business Impact**: financial loss, compliance exposure
- **Example**: A CVSS score of 9.8 (e.g., Log4Shell, CVE-2021-44228) often maps to a **Critical** priority level

---

3. Scoring Systems and Risk Quantification

SOC teams don't just rely on gut feelings—they use **standard scoring systems** to reduce bias.

- **CVSS (Common Vulnerability Scoring System)**:
    - **Base Metrics** measure intrinsic severity (constant traits)
    - **Temporal Metrics** reflect time-sensitive factors (e.g., exploit maturity, patch availability)
    - **Environmental Metrics** consider your environment's specifics and business context
    - In **CVSS v4.0**, there are four groups—Base, Threat (similar to Temporal), Environmental, and Supplemental [FIRSTWikipedia].
    - The **National Vulnerability Database (NVD)** provides Base scores and a CVSS calculator—but not full Temporal or Environmental assessments [NVD].



SOC Risk Scoring (SIEM-based)

- Tools like Splunk or Elastic assign **risk points** based on:
    - The type of event (e.g., malware detection = +50).
    - The asset affected (e.g., production server = +30).
    - The user involved (e.g., admin account = +20).
- When risk exceeds a threshold (say 80+), the alert is **Critical**.

- **Severity categories** (common for v3.x and v4.0):
    - **Low**: 0.1–3.9
    - **Medium**: 4.0–6.9
    - **High**: 7.0–8.9
    - **Critical**: 9.0–10.0 NVDCobaltWikipedia
- **Temporal metrics** example: availability of proof-of-concept or patch can lower or raise overall score Information Security Stack ExchangeWikipedia.
- **Environmental example**: a banking web server vulnerability may be scored higher if critical to service and widely deployed NIST.

---

4. NIST Incident Severity Classification & Prioritization Workflows

- **NIST SP 800-61 Revision 3 (released April 2025)** is now the **latest**, replacing the archived Rev 2 NIST Publications+1.
- It outlines **incident prioritization**, **severity estimation**, **response initiation**, and **recovery processes** NIST Publications.
- NIST emphasizes using **functional impact**, **information impact**, and **recoverability** to prioritize — particularly for third-party or supply-chain incidents Mitratech.
- The older **SP 800-61 Revision 2** (from 2012, now withdrawn) still provides foundational guidance on incident analysis and response—but should be referenced for historical context only NIST Computer Security Resource CenterNIST Publications.
- **Practical frameworks** based on these NIST standards—for example, incident response playbooks and templates—outline the four lifecycle phases: Preparation; Detection & Analysis; Containment, Eradication & Recovery; and Post-Incident Activity ConcertiumNIST Computer Security Resource CenterNIST Publications.

5. Key Learnings from Study

Through my study of alert prioritization frameworks and scoring systems, I gained several important insights:

1. **CVSS Scoring**

    ○ I learned how **CVSS (Common Vulnerability Scoring System)** provides a structured way to measure the severity of vulnerabilities.

    ○ The **Base metrics** give a static severity assessment (e.g., impact on confidentiality, integrity, availability).

    ○ **Temporal metrics** reflect real-world exploitability factors like whether a proof-of-concept exploit is publicly available.

    ○ **Environmental metrics** allow organizations to adjust severity based on their specific context, such as the criticality of the affected asset or compensating controls.

    ○ I also saw how **severity bands** are categorized: Low (0.1–3.9), Medium (4.0–6.9), High (7.0–8.9), and Critical (9.0–10.0). This mapping helps translate CVSS scores into alert priority levels.

2. **Incident Prioritization Workflows (NIST Guidance)**

    ○ From the **NIST incident handling guide (SP 800-61)**, I learned that prioritization doesn't rely on technical severity alone — it also considers **functional impact**, **information impact**, and **recoverability**.

    ○ For example, an attack with limited technical severity could still be treated as high-priority if it affects regulated data or damages business reputation.

    ○ NIST also emphasizes a lifecycle approach: **Preparation; Detection & Analysis; Containment, Eradication & Recovery; and Post-Incident Activity**. Prioritization is embedded in the **Detection & Analysis** phase, guiding how quickly the SOC responds.

3. **Practical Application**

    ○ A case study like **Log4Shell (CVE-2021-44228)** helped me connect theory to practice. Its CVSS score of 9.8 clearly placed it in the **Critical** category, but what made it urgent was not only the score, but also its widespread exploitability and business impact (threats to production systems worldwide).

    ○ This reinforced that **scores are a starting point**, but SOC analysts must also weigh exploit availability, asset exposure, and organizational context before finalizing alert priority.

4. **Takeaways for SOC Operations**

- Alert prioritization is essentially about balancing **quantitative scoring** (like CVSS) with **qualitative judgment** (like business impact).
- By combining these, analysts can reduce noise, focus on the most pressing threats, and align response efforts with organizational risk tolerance.

2. Incident Classification

Core Concepts:

Through my study of incident classification frameworks and case studies, I developed a strong understanding of how incidents are systematically categorized and enriched to support investigations.

1. **Incident Categories**
    - I learned that incidents are commonly classified by type, such as **malware infections, phishing attacks, denial-of-service (DoS/DDoS), insider threats, and data exfiltration**.
    - Each category carries distinct response considerations. For example, **insider threats** may involve unauthorized data exports by employees or contractors, requiring both technical and HR/legal response processes.
    - Recognizing these categories allows analysts to quickly align alerts with appropriate playbooks and escalation paths.
2. **Taxonomies and Frameworks**
    - I studied several frameworks that provide **standardized language and structure** for incident classification:
        - **MITRE ATT&CK**: maps incidents to specific tactics and techniques (e.g., *T1566 – Phishing* under the Initial Access tactic).
        - **ENISA Incident Taxonomy**: defines high-level categories for consistent reporting across organizations in Europe.
        - **VERIS (Vocabulary for Event Recording and Incident Sharing)**: focuses on structured incident reporting, especially for sharing data across organizations.
    - By applying these taxonomies, incidents can be labeled consistently, which helps with **trend analysis, reporting, and threat intelligence sharing**.
3. **Contextual Metadata Enrichment**

- Beyond categorization, I learned the importance of enriching incidents with **contextual metadata** such as affected assets, system roles (e.g., production vs. test), timestamps, attacker IPs, and Indicators of Compromise (IOCs) like file hashes or malicious domains.

- This enrichment transforms a raw alert into a **complete incident record**, making it easier for responders to analyze patterns and determine scope.

In addition to understanding the theory of incident categories, I explored several frameworks and real-world examples to build practical skills in classification and enrichment:

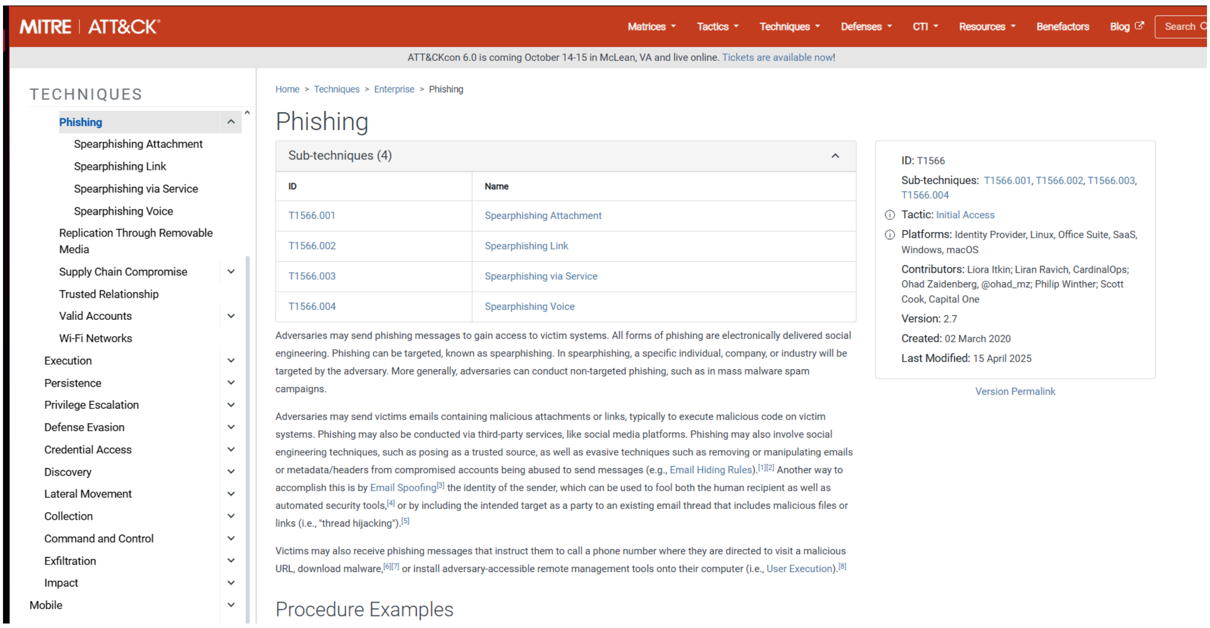
1. **MITRE ATT&CK Framework**

    - I studied how **MITRE ATT&CK** breaks down adversary behavior into **tactics (the "why")** and **techniques (the "how")**.

    - For example, phishing (T1566) falls under the **Initial Access** tactic, but ATT&CK further breaks it down into **spearphishing attachment**, **spearphishing link**, and **spearphishing via service**.

    - Mapping incidents to ATT&CK helped me move beyond just saying "phishing happened" to clearly identifying the attacker's method and intent.

    - I also learned how ATT&CK **matrices and detection mappings** can be used in a SOC to standardize how incidents are documented and investigated.



2. **ENISA Incident Taxonomy**

    - From studying ENISA's work, I learned that their taxonomy provides **broad, high-level categories** like Malicious Code, Unauthorized Access, Information

Leakage, and Availability.

  - What stood out was ENISA's emphasis on **uniform reporting across organizations and countries**. This showed me the importance of classification not just for internal investigations but also for compliance and cross-border information sharing.

3. **VERIS Framework (Vocabulary for Event Recording and Incident Sharing)**

   - I studied VERIS as a structured way to document incidents for **metrics and analysis**.

   - Unlike ATT&CK, which focuses on attacker behavior, VERIS emphasizes **who, what, why, and how** of an incident.

   - For example, in a phishing case, VERIS captures not just that phishing occurred, but also details such as **actor type (external), action (social engineering), asset (email server, endpoint), and impact (credentials compromised)**.

   - I realized that VERIS is especially powerful for **data-driven analysis**, like the Verizon DBIR (Data Breach Investigations Report), which is built on VERIS.

4. **Case Studies and Practical Exercises**

   - I reviewed phishing campaign case studies (such as those in the SANS Reading Room). These showed how analysts apply classification and metadata enrichment in real investigations.

   - For instance, one phishing campaign was mapped as:

     - **Category**: Phishing (ENISA)

     - **Technique**: T1566.001 – Spearphishing Attachment (MITRE ATT&CK)

     - **Metadata**: Malicious macro-enabled document (MD5 hash provided), delivery timestamp, source IP ranges, and affected user accounts

     - **VERIS Recording**: Actor = External, Action = Social, Asset = User email account, Impact = Confidentiality breach (credential loss)

   - This exercise made it clear how these frameworks complement each other: **ATT&CK** for behavior, **ENISA** for broad reporting, **VERIS** for structured incident sharing.

3. Basic Incident Response

Core Concepts:

While studying incident response, I focused on both the theoretical lifecycle and the practical skills required to manage real-world incidents. The main insights I gained are:

1. Incident Lifecycle

- I learned that the **incident response lifecycle** follows six key phases, as defined by NIST and reinforced by SANS:

    1. **Preparation** – building readiness through playbooks, policies, logging strategies, and tools. For example, having a phishing playbook ensures the SOC responds consistently every time.

    2. **Identification** – triaging alerts to confirm whether an actual incident is occurring. This involves reviewing logs, IOCs, and threat intelligence to distinguish between false positives and real threats.

    3. **Containment** – limiting the spread of an incident. I studied examples like isolating compromised endpoints from the network or disabling a malicious user account.

    4. **Eradication** – removing the root cause of the attack, such as deleting malware, closing exploited vulnerabilities, or resetting compromised credentials.

    5. **Recovery** – restoring systems to production, often in stages. For example, bringing a patched web server back online gradually while monitoring for reinfection.

    6. **Lessons Learned** – conducting a **post-mortem** to analyze what worked, what failed, and how processes or defenses can be improved.

2. Core Procedures

- I studied the **practical procedures** that SOC teams apply during incidents:

    - **System isolation**: Using EDR tools or firewall rules to quarantine compromised hosts.

    - **Evidence preservation**: Capturing volatile data (e.g., memory dumps, running processes), creating forensic images, and hashing files to maintain chain of custody.

    - **Communication protocols**: Following escalation paths, informing stakeholders in non-technical language, and ensuring no sensitive information leaks outside the approved channels.

  - **SOAR (Security Orchestration, Automation, and Response)**: I explored how tools like **Splunk Phantom** or **Cortex XSOAR** automate repetitive tasks such as blocking IPs or generating incident tickets, freeing analysts to focus on higher-level analysis.

As I studied incident response, I approached it from three perspectives: **frameworks (NIST), best practices (SANS), and hands-on simulation (Let's Defend and SOAR tools).**

1. NIST SP 800-61 Incident Handling Guide

- I studied the NIST guide in depth and found it invaluable for understanding the **structured lifecycle of incident response**.
- It emphasized that the most important phase is **Preparation**, since the effectiveness of response depends on readiness (playbooks, trained staff, logging, detection systems).
- I learned that NIST categorizes incidents based on **functional impact**, **information impact**, and **recoverability**, which influences how the lifecycle phases are executed.
- A key insight I gained is that **incident response is iterative** — for example, containment strategies may be short-term (quarantining a system) or long-term (deploying permanent firewall rules), and these decisions are made dynamically.
- Studying NIST also showed me how important **documentation and evidence tracking** are for both compliance and future lessons learned.

2. SANS Incident Handler's Handbook

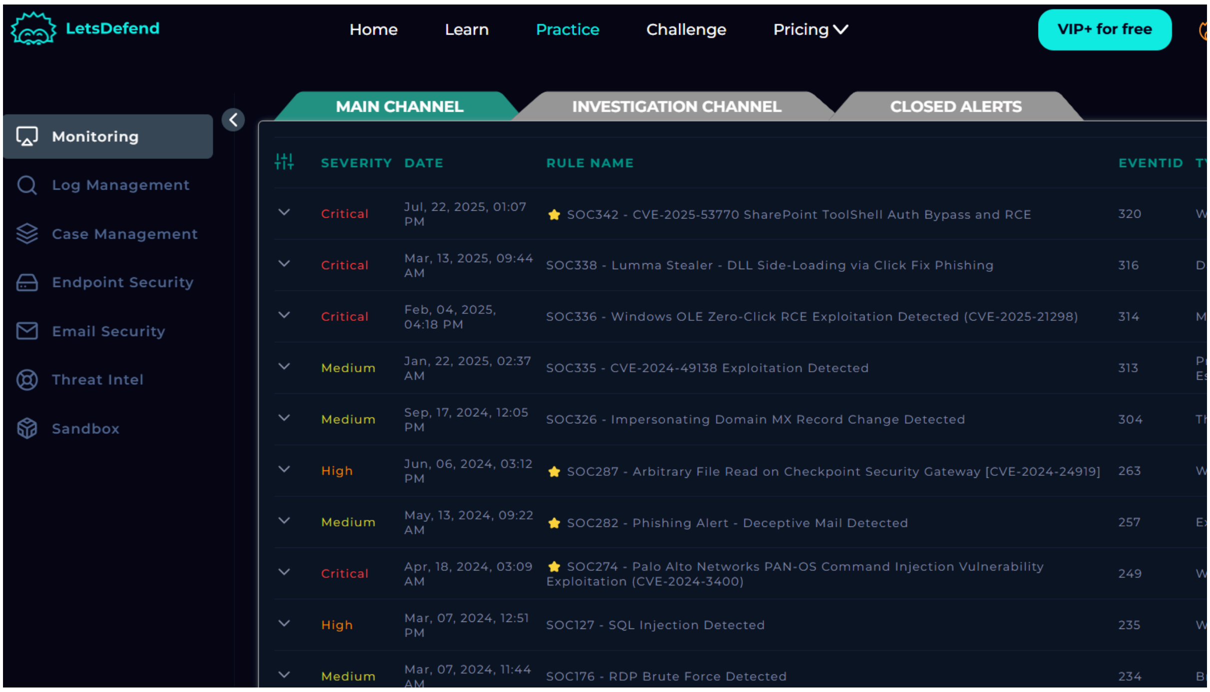
- The SANS handbook translated NIST's lifecycle into **practical, analyst-friendly checklists and templates**.
- I learned how to perform actions such as:
  - Writing an **incident diary** to capture evidence and decisions in real time.
  - Following a **containment checklist** (e.g., disconnecting a host, revoking credentials, or disabling services).
  - Conducting **post-incident reviews** with both technical teams and business stakeholders.
- The handbook also emphasized **communication protocols**, which was something I hadn't initially considered as deeply technical, but it's actually critical to response success. For example, during containment, SOC analysts should notify business

owners before shutting down a production system to avoid unnecessary downtime.

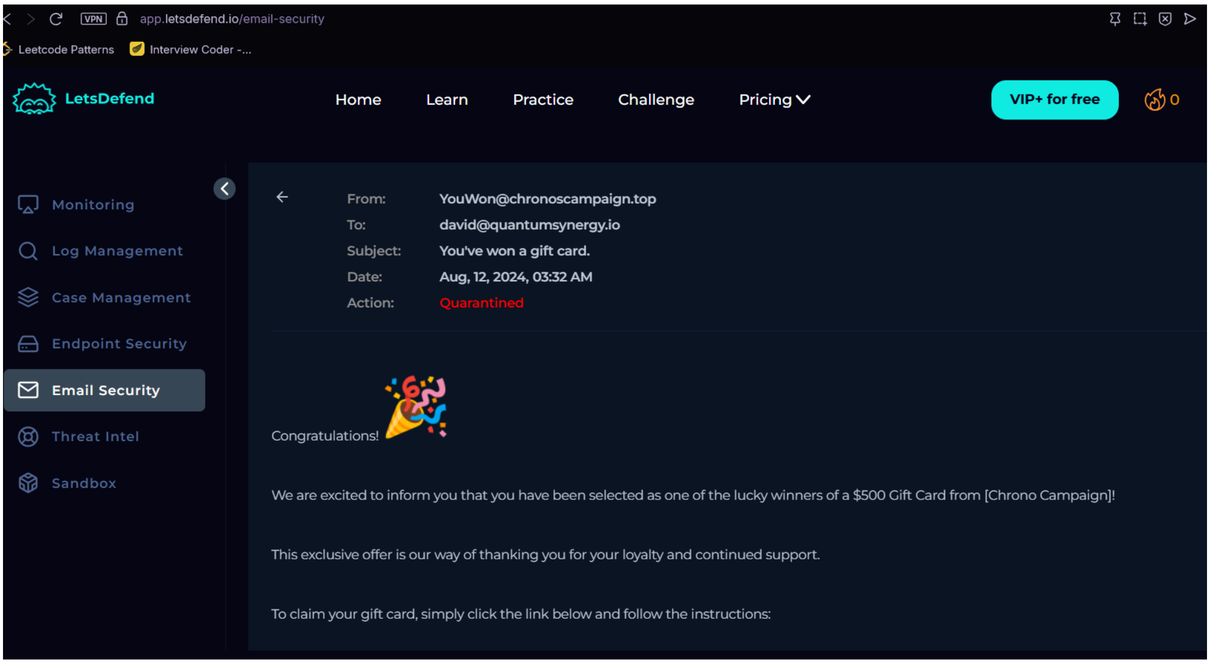
- Overall, the SANS perspective helped me see how to bridge the gap between **theory and real SOC operations**.

3. Hands-On Practice (Let's Defend and SOAR Concepts)

- I applied what I studied through **Let's Defend labs**, which provided **simulated SOC scenarios**. These exercises gave me confidence in moving through the lifecycle phases, especially **Identification** (triaging alerts and recognizing true positives) and **Containment** (isolating compromised hosts).



- In one phishing simulation, I practiced gathering indicators from malicious emails, analyzing them, and correlating them with MITRE ATT&CK techniques — reinforcing how incident classification ties into response.



- I also explored **SOAR (Security Orchestration, Automation, and Response)** tools like Splunk Phantom. I learned how automation can:
    - Automatically enrich alerts with threat intelligence.
    - Trigger containment actions like blocking IPs or disabling accounts.
    - Generate standardized reports.
- These simulations highlighted that **automation speeds up repetitive tasks**, but human judgment is still required for high-impact decisions, especially during eradication and recovery

Practical Application
1. Alert Management Practice
Activities:

Tools: Google Sheets, Wazuh, TheHive.

Tasks: Create an alert classification system, prioritize alerts, document response procedures, create incident tickets, and practice escalation.

Enhanced Tasks:

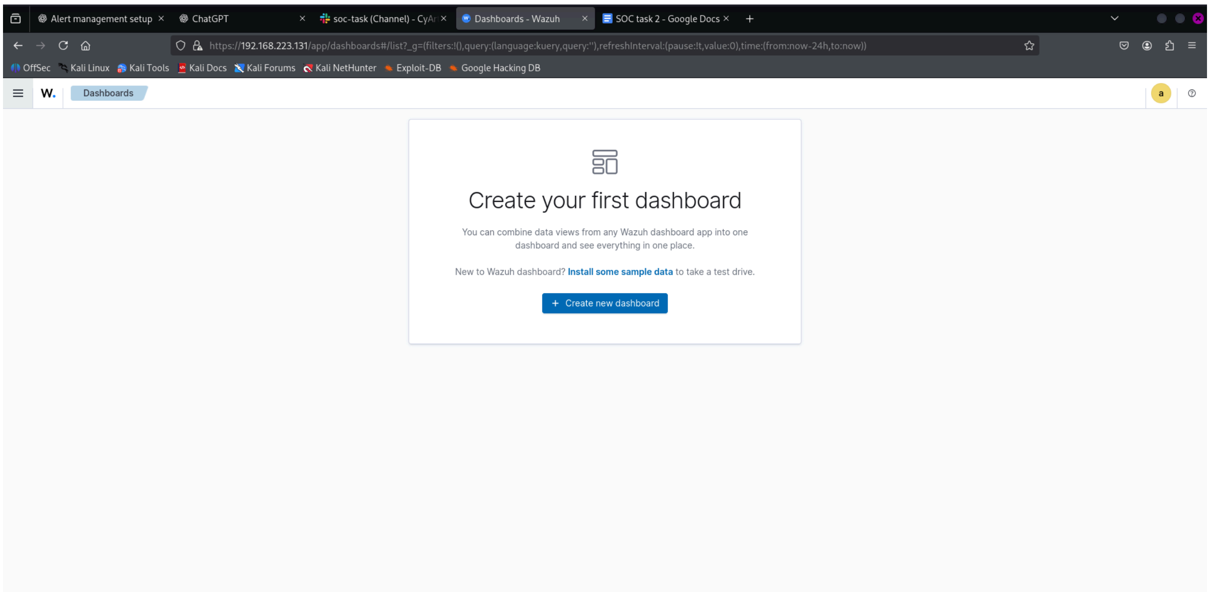
Alert Classification System: Create a Google Sheets table to map alerts to MITRE ATT&CK techniques:

Alert ID	Type	Priority	MITRE Tactic
001	Phishing	High	T1566

Test with a mock alert (e.g., "Phishing Email: Suspicious Link").

Prioritize Alerts: Simulate alerts (e.g., “Critical: Log4Shell Exploit Detected” vs. “Low: Port Scan”) and score using CVSS in Google Sheets. Example: Log4Shell CVSS 9.8 = Critical.

Dashboard Creation: In Wazuh, create a dashboard to visualize alert priorities (e.g., pie chart for Critical vs. High alerts).

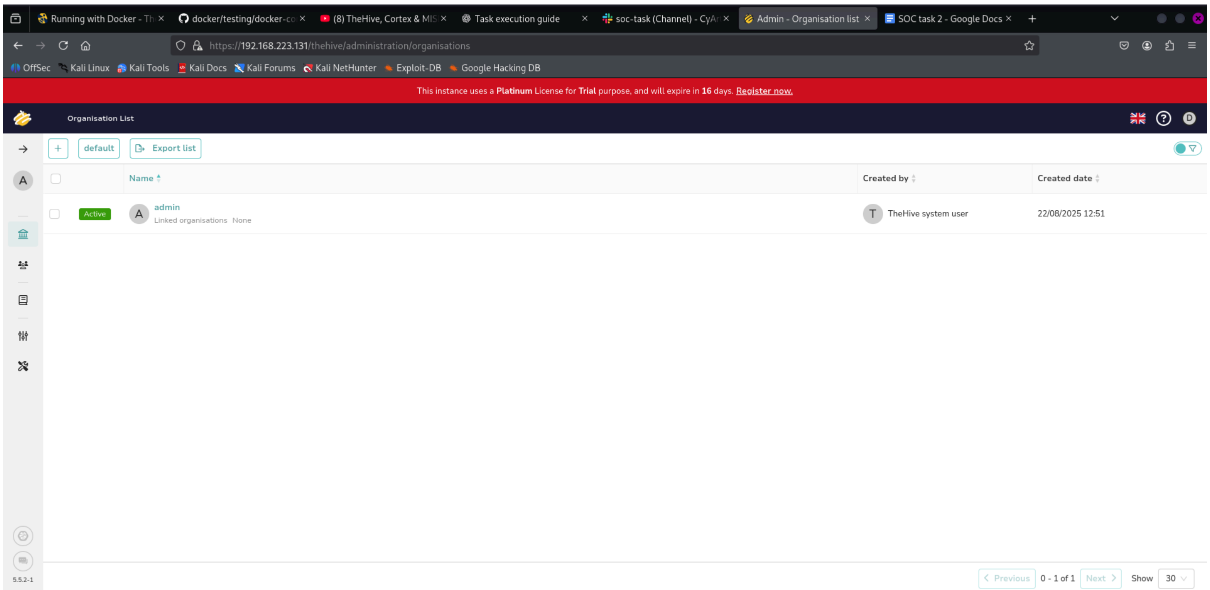


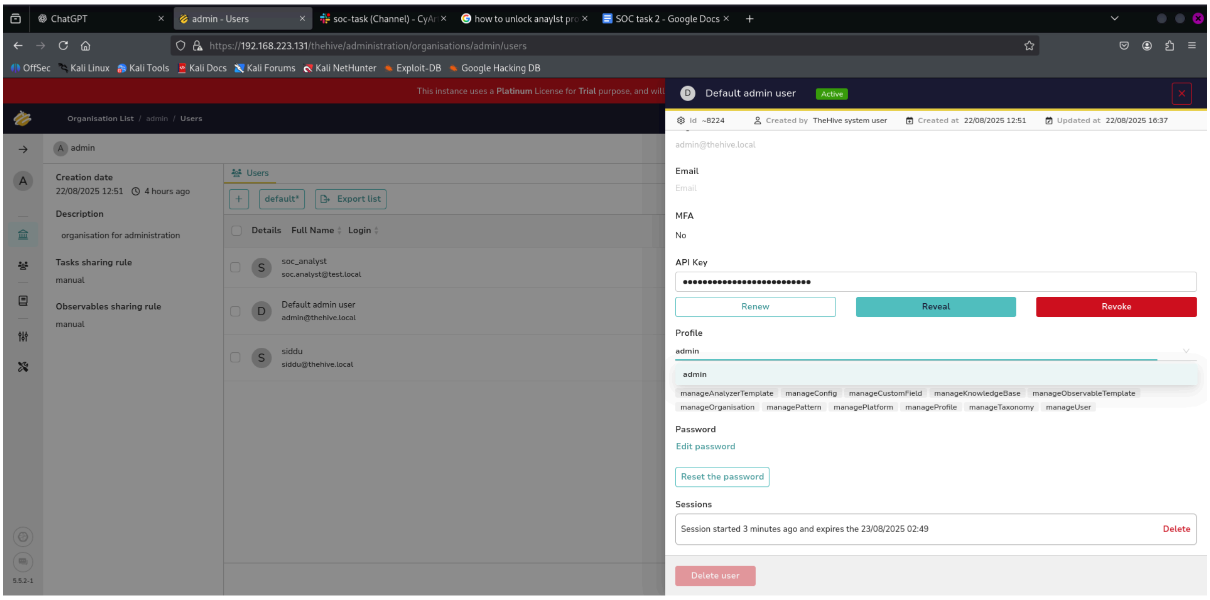
Incident Ticket: Draft a ticket in TheHive with fields:

Title: [Critical] Ransomware Detected on Server-X

Description: Indicators: [File: crypto_locker.exe], [IP: 192.168.1.50]

Priority: Critical





I cant create a new case in this version anymore and i cant create a analyst user it's only letting me create a admin acc which doesnt have create case option

Assignee: SOC Analyst

Escalation Role-Play: Draft a 100-word email to escalate a Critical alert to Tier 2, summarizing the incident and IOCs.

Dear Tier-2 Team,

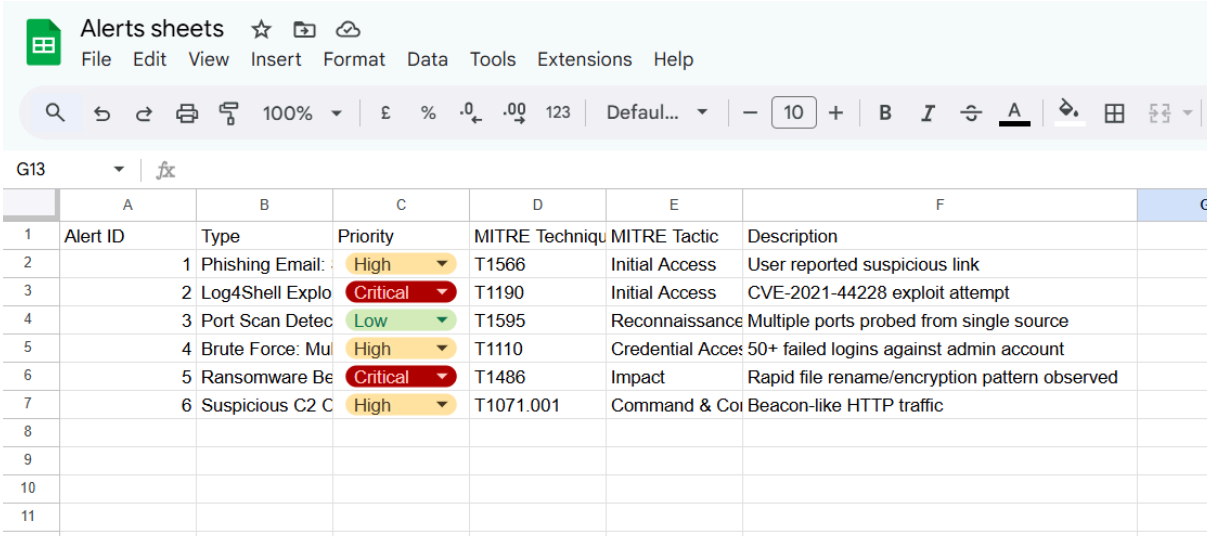
We have detected a critical ransomware incident on Server-X. Wazuh flagged malicious file activity (crypto_locker.exe) originating from IP 192.168.1.50. Indicators suggest active encryption attempts. Current impact: user files are being locked.

Immediate containment is recommended. Please investigate lateral movement, block malicious IPs, and isolate the affected host. Related MITRE ATT&CK mapping: T1486 (Data Encrypted for Impact).

I have logged a case in TheHive (#INC-2025-01). Screenshots and logs are attached.

Regards,
K.Sai Sidhartha
SOC Tier-1 Analyst

Test data:



Alert ID	Type	Priority	MITRE Techniqu	MITRE Tactic	Description
1	Phishing Email:	High	T1566	Initial Access	User reported suspicious link
2	Log4Shell Explo	Critical	T1190	Initial Access	CVE-2021-44228 exploit attempt
3	Port Scan Detec	Low	T1595	Reconnaissance	Multiple ports probed from single source
4	Brute Force: Mul	High	T1110	Credential Acce:	50+ failed logins against admin account
5	Ransomware Be	Critical	T1486	Impact	Rapid file rename/encryption pattern observed
6	Suspicious C2 C	High	T1071.001	Command & Coı	Beacon-like HTTP traffic

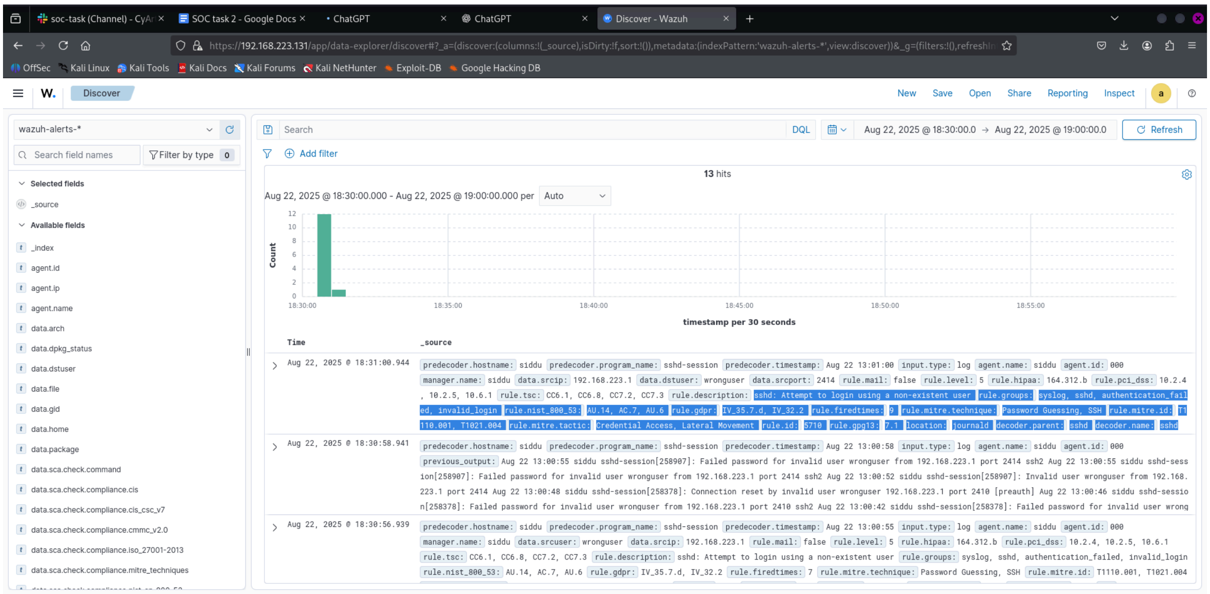
3. Alert Triage Practice
Activities:

- **Tools:** Wazuh, VirusTotal, AlienVault OTX.
- **Tasks:** Simulate triage with sample alerts and validate false positives.

Enhanced Tasks:

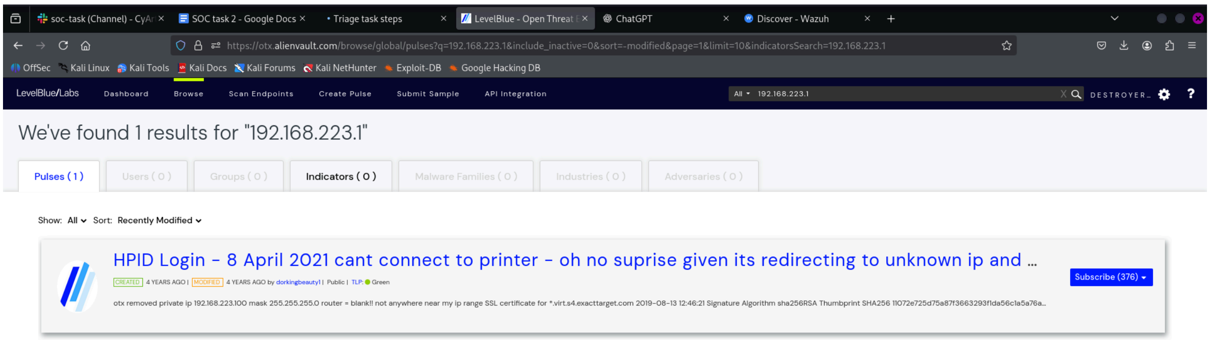
- **Triage Simulation:** Analyze a mock alert (e.g., “Brute-force SSH Attempts”) in Wazuh. Document:

Alert ID	Description	Source IP	Priority	Status
002	Brute-force SSH	192.168.1.100	Medium	Open



Alert ID	Description	Source IP	Priority	Status
5710	sshd: Attempt to login using a non-existent user	192.168.223.1	Medium	Open

- **Threat Intelligence Validation:** Cross-reference the alert's IP or file hash with AlienVault OTX to validate IOCs. Summarize findings in 50 words.



https://otx.alienvault.com/pulse/606f00888300861b977f0f15

Since it cant detect private ip i got nothing

Step 3: Document a 50-word Threat Intel Summary

You can write something like:

The IP 192.168.223.1 is an internal host attempting SSH logins with a non-existent user. AlienVault OTX has no reports of malicious activity for this IP. This likely indicates either misconfiguration or internal testing. Continuous monitoring is recommended, and repeated failed attempts should be blocked if persistent.

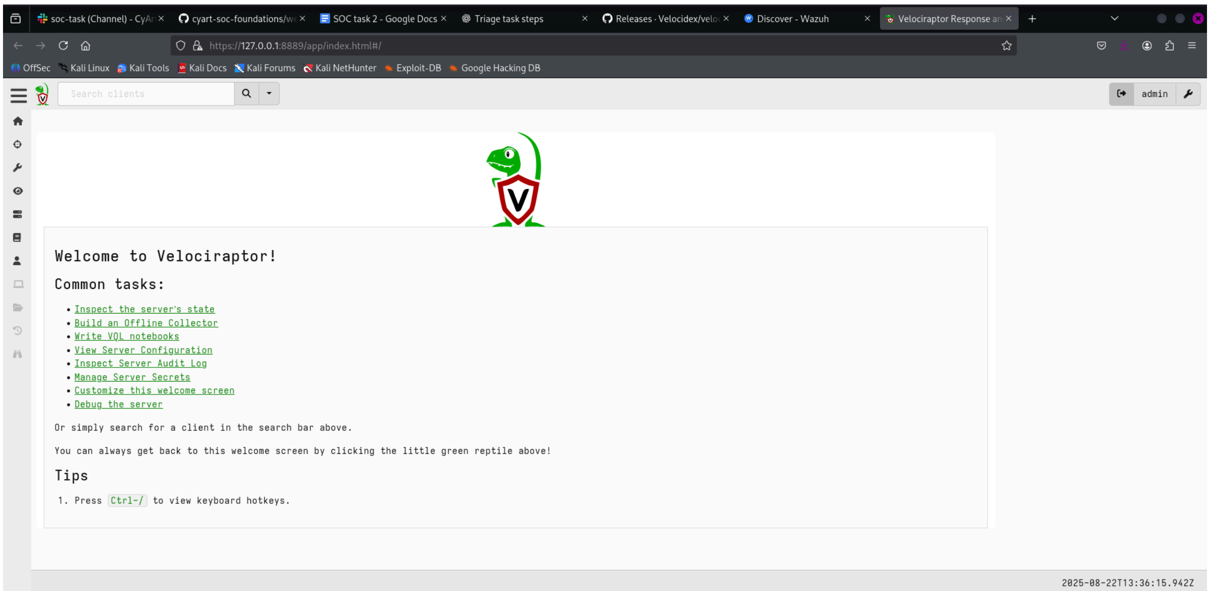
4. Evidence Preservation

Activities:

- **Tools:** Velociraptor, FTK Imager.
- **Tasks:** Practice evidence preservation and chain-of-custody documentation.

Enhanced Tasks:

- **Volatile Data Collection:** Use Velociraptor to collect network connections (SELECT * FROM netstat) from a Windows VM. Save to CSV.

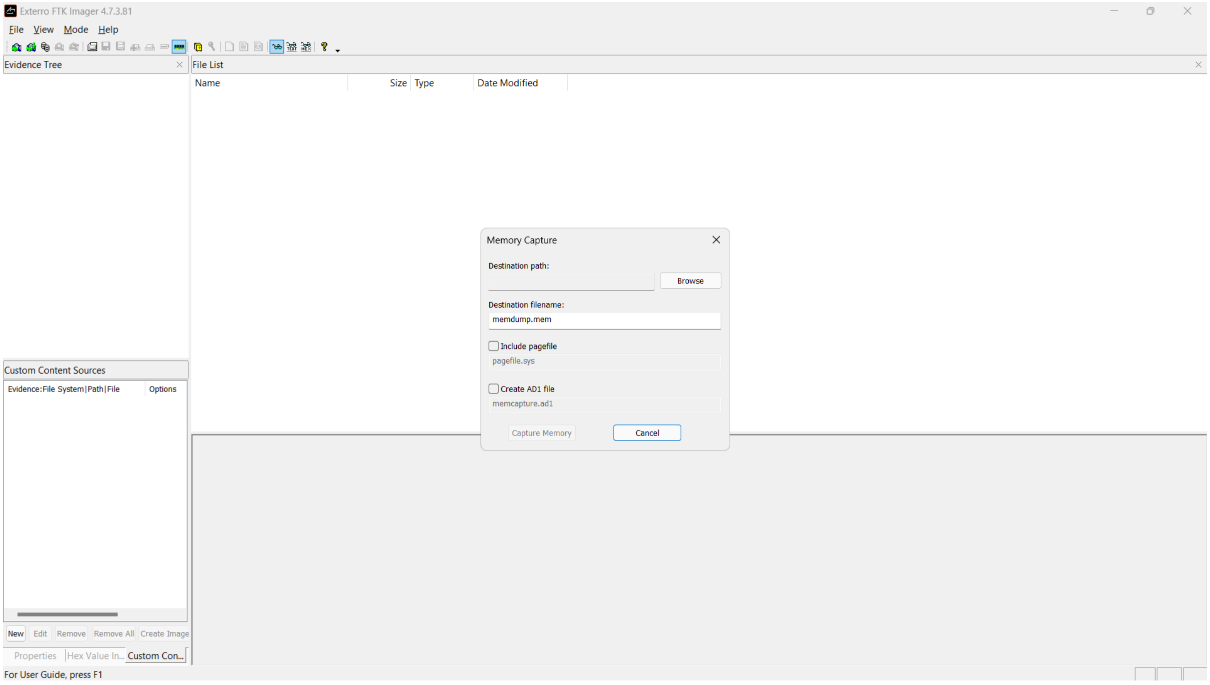


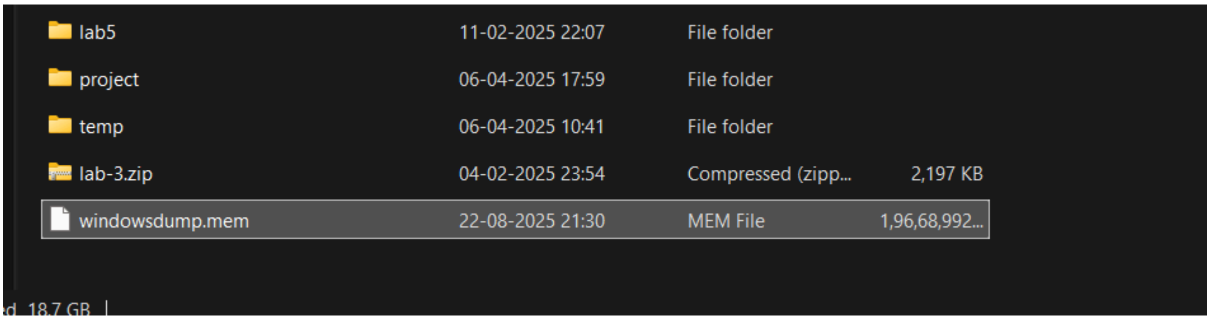

```
Active Connections

  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
```

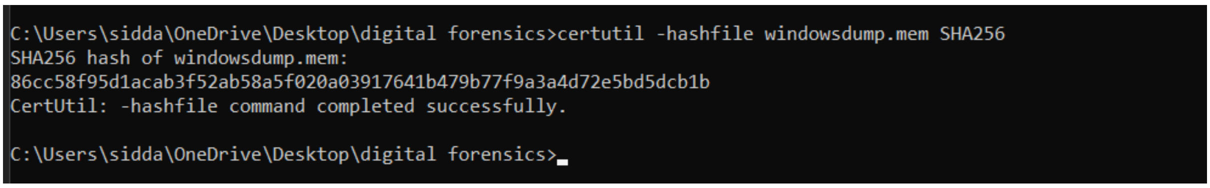
```
  TCP    0.0.0.0:5040           0.0.0.0:0              LISTENING
2616
  TCP    0.0.0.0:5520           0.0.0.0:0              LISTENING
9956
  TCP    0.0.0.0:5560           0.0.0.0:0              LISTENING
6196
  TCP    0.0.0.0:5580           0.0.0.0:0              LISTENING
6196
  TCP    0.0.0.0:7680           0.0.0.0:0              LISTENING
16132
  TCP    0.0.0.0:27036          0.0.0.0:0              LISTENING
4596
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING
1296
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING
1124
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING
2200
  TCP    0.0.0.0:49667          0.0.0.0:0              LISTENING
3364
  TCP    0.0.0.0:49668          0.0.0.0:0              LISTENING
4512
  TCP    0.0.0.0:49675          0.0.0.0:0              LISTENING
1268
  TCP    127.0.0.1:27017        0.0.0.0:0              LISTENING
5448
  TCP    127.0.0.1:27060        0.0.0.0:0              LISTENING
4596
  TCP    127.0.0.1:27275        0.0.0.0:0              LISTENING
4560
  TCP    127.0.0.1:49670        0.0.0.0:0              LISTENING
5632
  TCP    127.0.0.1:49984        0.0.0.0:0              LISTENING
4596
  TCP    127.0.0.1:49984        127.0.0.1:49991        ESTABLISHED
4596
  TCP    127.0.0.1:49986        0.0.0.0:0              LISTENING
4596
  TCP    127.0.0.1:49986        127.0.0.1:49990        ESTABLISHED
4596
  TCP    127.0.0.1:49990        127.0.0.1:49986        ESTABLISHED
2400
  TCP    127.0.0.1:49991        127.0.0.1:49984        ESTABLISHED
2400
  TCP    192.168.16.139:139     0.0.0.0:0              LISTENING       4
  TCP    192.168.16.139:2493    64.233.170.188:5228    ESTABLISHED
```

```
13900
  TCP    192.168.16.139:2498    4.213.25.241:443       ESTABLISHED
3432
  TCP    192.168.16.139:2518    163.70.143.60:5222     ESTABLISHED
13900
  TCP    192.168.16.139:2521    172.64.148.235:443     ESTABLISHED
13900
  TCP    192.168.16.139:2527    155.133.224.23:443     ESTABLISHED
4596
  TCP    192.168.16.139:2531    35.230.116.55:7500     ESTABLISHED
4560
  TCP    192.168.16.139:2694    145.190.36.0:443       ESTABLISHED
4872
  TCP    192.168.16.139:19677   34.98.110.65:443       LAST_ACK
4560
  TCP    192.168.16.139:35716   4.224.116.131:443      ESTABLISHED
11768
  TCP    192.168.16.139:35719   172.64.155.209:443     ESTABLISHED
13900
  TCP    192.168.16.139:35722   172.64.155.209:443     ESTABLISHED
13900
  TCP    192.168.16.139:35730   185.199.109.133:443    ESTABLISHED
13900
  TCP    192.168.16.139:35736   103.17.159.178:443     ESTABLISHED
4560
  TCP    192.168.16.139:35737   34.98.110.65:443       ESTABLISHED
4560
  TCP    192.168.16.139:35738   35.186.243.246:443     TIME_WAIT       0
  TCP    192.168.16.139:38428   34.98.110.65:443       CLOSE_WAIT
4560
  TCP    192.168.16.139:38441   140.82.113.26:443      ESTABLISHED
13900
  TCP    192.168.16.139:38461   34.107.243.93:443      CLOSE_WAIT
5656
```

- **Evidence Collection:** Collect a memory dump (SELECT * FROM Artifact.Windows.Memory.Acquisition) and hash it using sha256sum. Document:







Item	Description	Collected By	Date	Hash Value
Memory Dump	Server-X Dump	SOC Analyst	2025-08-18	86cc58f95d1acab3f52ab58a5f020a03917641b479b77f9a3a4d72e5bd5dcb1b

5. Capstone Project: Full Alert-to-Response Cycle

Activities:

- **Tools:** Metasploit, Wazuh, CrowdSec, Google Docs.
- **Tasks:** Simulate an attack, detect, triage, respond, and document.

Enhanced Tasks:

Attack Simulation: Exploit a Metasploitable2 vulnerability with Metasploit (e.g., vsftpd backdoor: use exploit/unix/ftp/vsftpd_234_backdoor). Follow Metasploit Unleashed.

- **Detection and Triage:** Configure Wazuh to alert on the attack. Document:

Timestamp	Source IP	Alert Description	MITRE Technique
2025-08-18 11:00:00	192.168.1.100	VSFTPD exploit	T1190

- **Response:** Isolate the VM and block the attacker's IP with CrowdSec. Verify with a ping test.
- **Reporting:** Write a 200-word report in Google Docs using a SANS template, including Executive Summary, Timeline, and Recommendations.
- **Stakeholder Briefing:** Draft a 100-word briefing for a non-technical manager, summarizing the incident and actions taken.

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:5a:bd:3f brd ff:ff:ff:ff:ff:ff
    inet 192.168.223.133/24 brd 192.168.223.255 scope global eth0
    inet6 fe80::20c:29ff:fe5a:bd3f/64 scope link
       valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ ^[[2~
```

```
┌──(siddu0034㉿siddu)-[~]
└─$ sudo cscli decisions add --ip 192.168.223.133 --duration 1h --reason "vsftpd exploit"

INFO[23-08-2025 21:01:43] Decision successfully added

┌──(siddu0034㉿siddu)-[~]
└─$ sudo cscli decisions list
```

ID	Source	Scope:Value	Reason	Action	Country	AS	Events	expiration	Alert ID
15001	cscli	Ip:192.168.223.133	vsftpd exploit	ban			1	59m53.233471582s	2

```
┌──(siddu0034㉿siddu)-[~]
└─$
```

ıg 20, 2025 @ 23:59:59.999 (interval: Auto - 30 minutes)

Sort fields 1

Aug 20 09:27:33 ubuntu vsftpd: pam_unix(vsftpd:auth): authen
ı failure; logname= uid=0 euid=0 tty=ftp ruser=msfadmin rhost
:127.0.0.1 @timestamp Aug 20, 2025 @ 12:46:28.487…