SOC Alert Management, Incident Response, and Threat Analysis

Harsh Vasoya
SOC task - 2

Summary

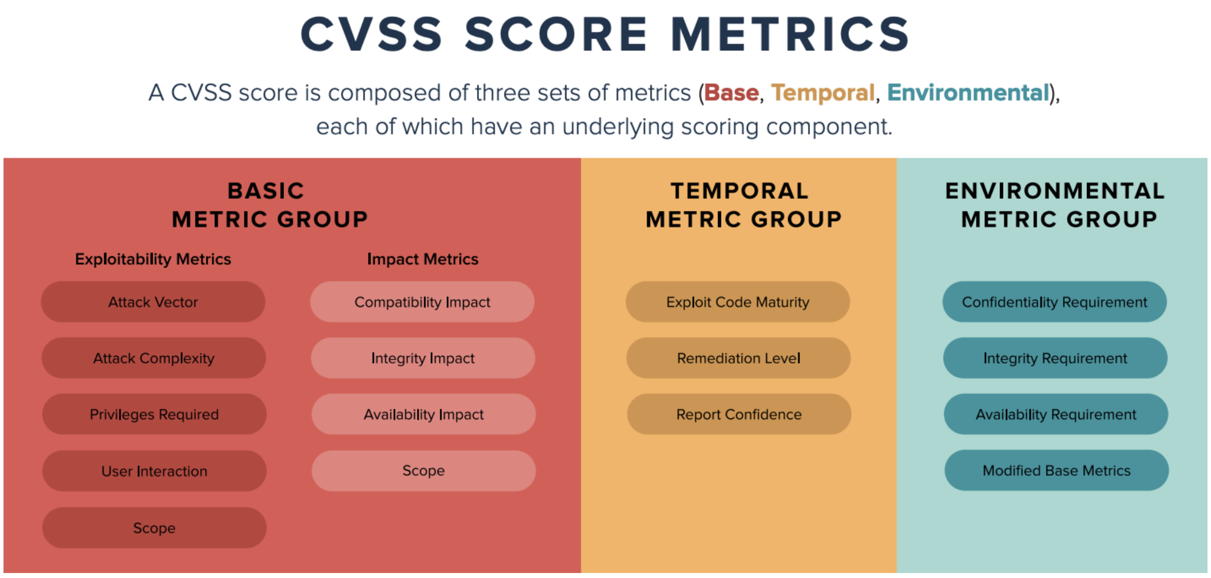
The Week 2 assignment for the cyart-soc-team is all about learning how a SOC analyst actually works, moving from reading books to actually using tools. First, you have to learn about alert priority levels so you know which problems to fix first, usually by looking at how bad the impact is and how fast you need to act. You will study things like the CVSS scoring system and NIST guidelines to understand why some alerts are critical while others are low priority. You also need to learn how to classify different types of attacks, like phishing or malware, using standard frameworks like MITRE ATT&CK. The theory part ends with learning the incident response lifecycle, which is basically the step-by-step plan for finding, stopping, and fixing a security breach.

For the practical part, you will get your hands dirty with tools like Wazuh and Google Sheets to organize alerts and map them to specific hacker techniques. You have to practice documenting everything, which means making response templates in Google Docs and writing down every action you take during an investigation. You will also try out triage by checking suspicious IP addresses on sites like VirusTotal to see if they are actually dangerous. There is even a section on forensics where you use a tool called Velociraptor to collect data from a computer and save it with a special hash value to prove the evidence hasn't been messed with.

The biggest part of the week is the capstone project where you do everything from start to finish. You will use Metasploit to launch a fake attack on a target, detect that attack using Wazuh, and then use CrowdSec to block the attacker.

Theoretical Knowledge

Alert Priority and Classification



In a SOC, not all alerts are equal, so we use a system to decide which ones need attention first. Alert priority is based on how much damage an attack could do and how fast it is spreading. We often use the CVSS (Common Vulnerability Scoring System) to give each alert a number from 0 to 10; for example, a score of 9.8 would be critical and needs to be fixed immediately, while a 4.0 might be medium and can wait a bit. Incident classification helps us label the type of attack we are seeing, like phishing, malware, or a DDoS attack. By using standard labels from the MITRE ATT&CK framework, every analyst on the team understands exactly what the hacker is trying to do, which makes the investigation much faster.

The Incident Response Lifecycle

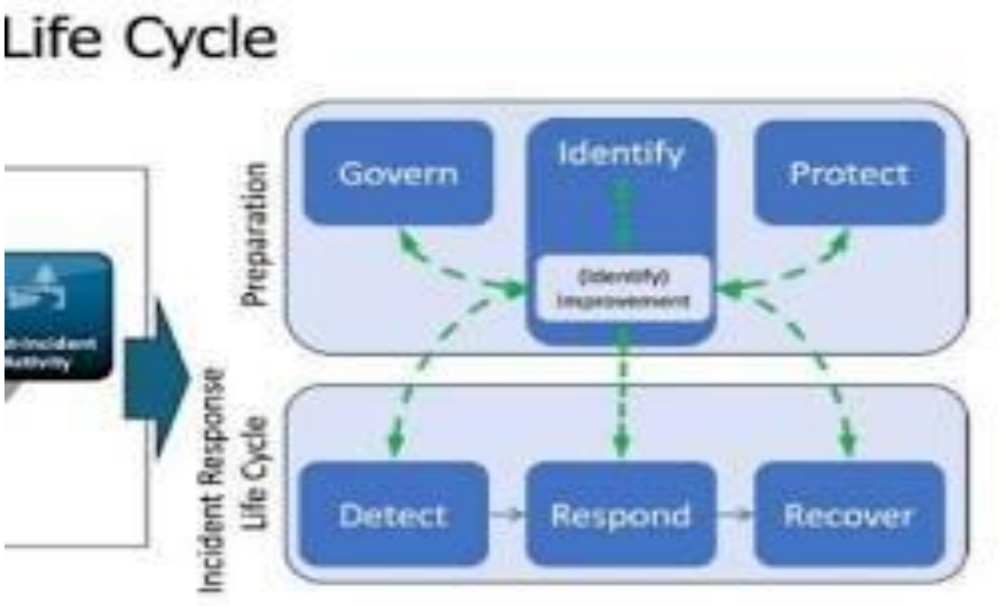
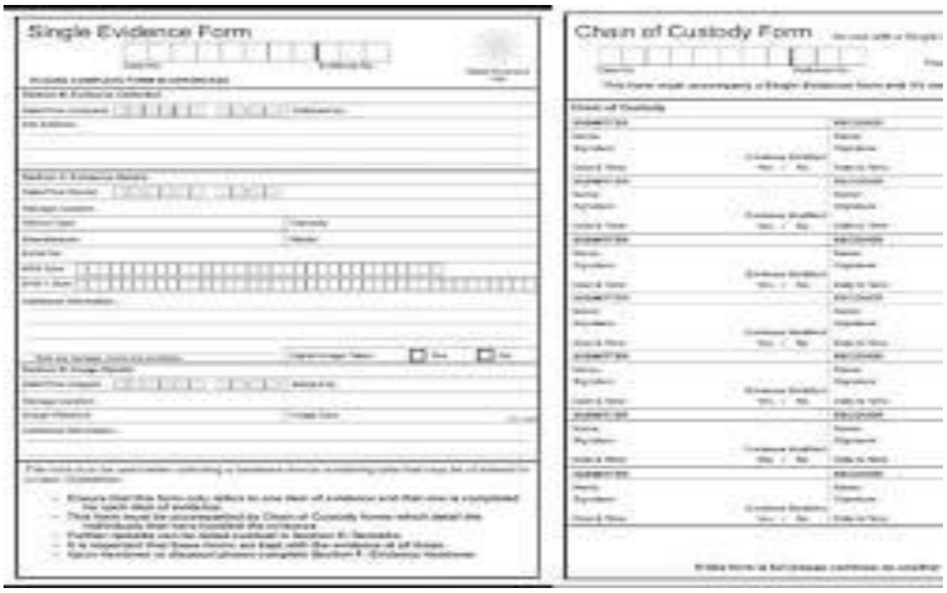


Fig. 2. Incident response life cycle model based on CSF 2.0 Functions

When a real threat is confirmed, we follow a specific game plan called the incident response lifecycle, which is usually based on the NIST SP 800-61 guide. It consists of six main steps to ensure nothing is missed during the chaos of an attack:

- Preparation: Setting up tools like Wazuh and creating playbooks before an attack happens.
- Identification: Triage and analysis to confirm if an alert is a true positive or just a false alarm.
- Containment: Stopping the threat from spreading, such as isolating an infected VM from the rest of the network.
- Eradication: Completely removing the cause of the incident, like deleting malware or closing a backdoored account.
- Recovery: Getting the systems back to normal and monitoring them to make sure the attacker doesn't return.
- Lessons Learned: A post-mortem meeting to discuss what went well and how to improve our defenses for next time.

Evidence Preservation and Forensics



Single Evidence Form

[illegible]

Chain of Custody Form

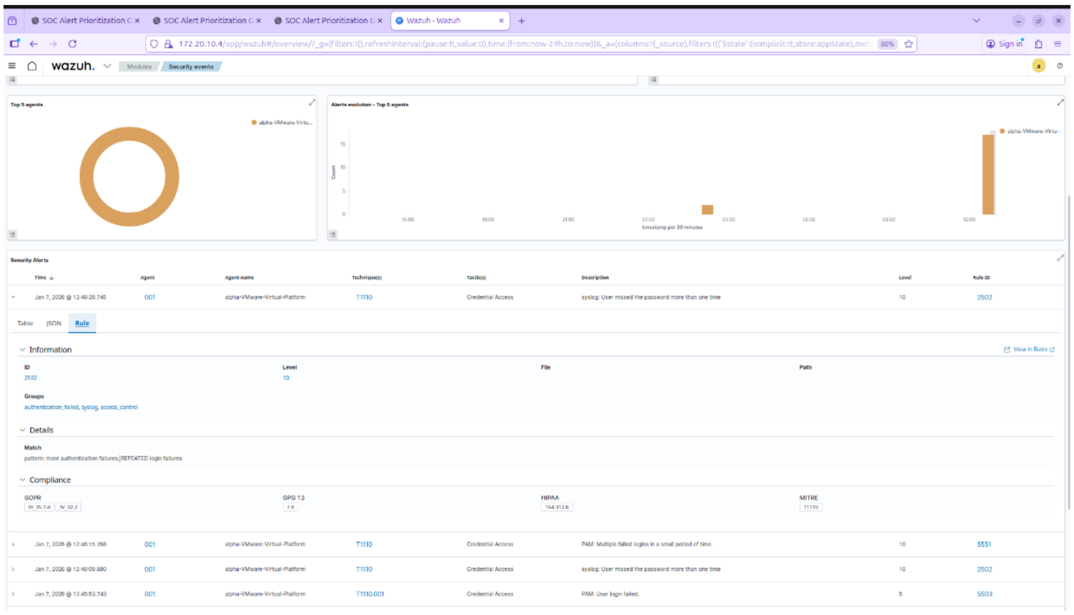
[illegible]

A key part of being an analyst is making sure that any data you collect can be used as proof later. This is called evidence preservation. When we find a compromised system, we use tools like Velociraptor or FTK Imager to grab volatile data, which is information that disappears when a computer is turned off, like running processes or network connections. To make sure the evidence is never changed or faked, we create a SHA256 hash, which is a unique digital fingerprint for the file. If even one tiny bit of the file changes, the hash will change, proving that the evidence has been tampered with.

SOC Lab Environment

The practical tasks for this week were done using a lab setup with two virtual machines to simulate a real security operations center. The first machine is the manager, which runs the Wazuh dashboard and is responsible for collecting all the logs and showing alerts when something suspicious happens. The second machine is the agent, This machine acts as the endpoint that is being watched.

This setup is important because it shows how a real SOC works, where many different computers all send their security data to one central monitoring system. By using these two machines, I was able to generate a real attack on the agent and then see exactly how the manager detected it and created an alert.



Security Alerts

	Time	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
˅	Jan 7, 2026 @ 12:46:29.745	001	alpha-VMware-Virtual-Platform	T1110	Credential Access	syslog: User missed the password more than one time	10	2502
›	Jan 7, 2026 @ 12:46:15.768	001	alpha-VMware-Virtual-Platform	T1110	Credential Access	PAM: Multiple failed logins in a small period of time.	10	5551
›	Jan 7, 2026 @ 12:46:09.880	001	alpha-VMware-Virtual-Platform	T1110	Credential Access	syslog: User missed the password more than one time	10	2502
›	Jan 7, 2026 @ 12:45:53.743	001	alpha-VMware-Virtual-Platform	T1110.001	Credential Access	PAM: User login failed.	5	5503

Table JSON **Rule**

Information — View in Rules

ID	Level	File	Path
2502	10		

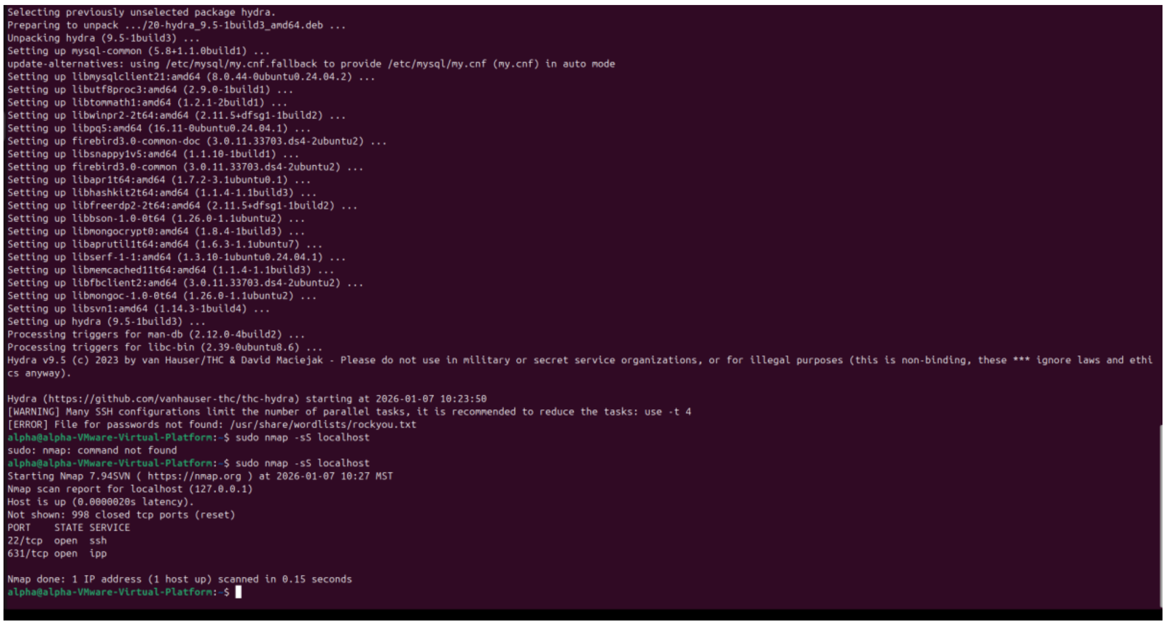
Groups
authentication_failed, syslog, access_control

Details

Match
pattern: more authentication failures;|REPEATED login failures

Compliance

GDPR	GPG 13	HIPAA	MITRE
IV_35.7.d IV_32.2	7.8	164.312.b	T1110

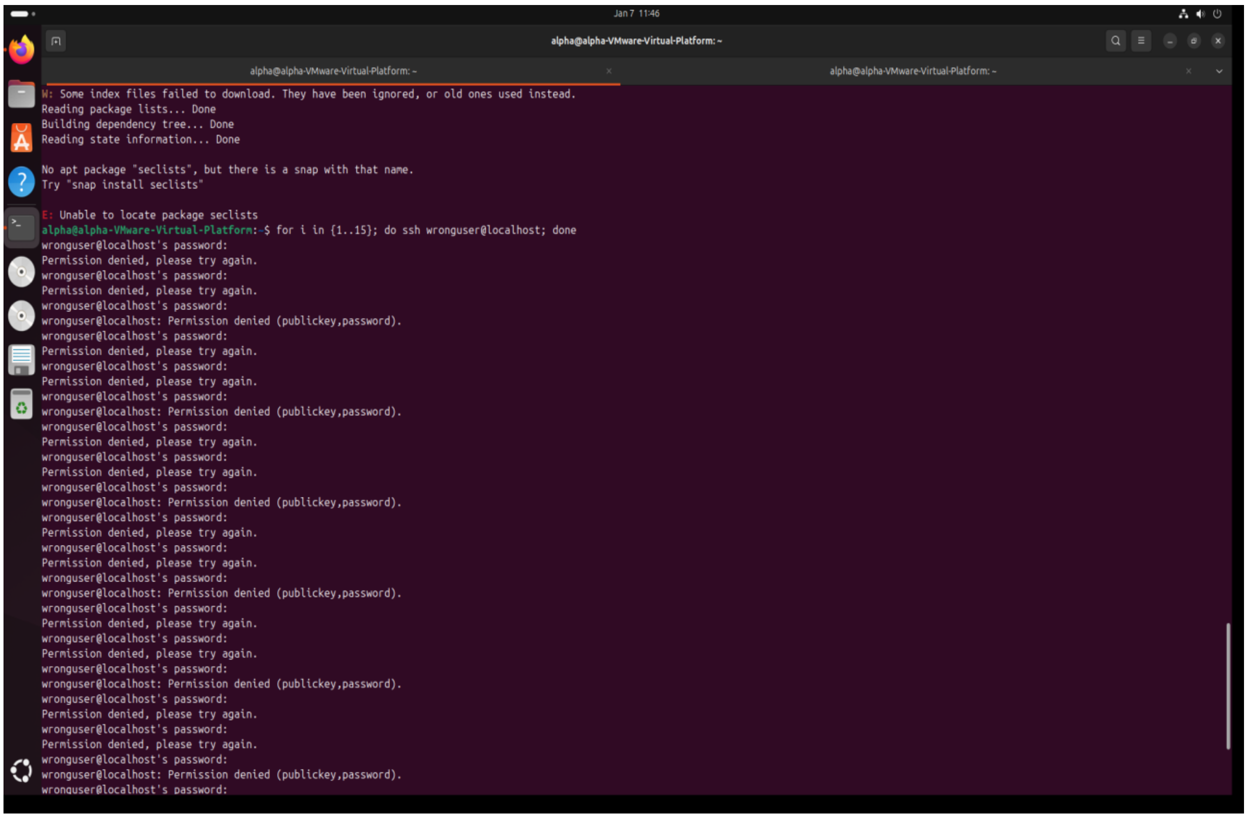


```
Selecting previously unselected package hydra.
Preparing to unpack .../20-hydra_9.5-1build3_amd64.deb ...
Unpacking hydra (9.5-1build3) ...
Setting up mysql-common (5.8+1.1.0build1) ...
update-alternatives: using /etc/mysql/my.cnf.fallback to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Setting up libmysqlclient21:amd64 (8.0.44-0ubuntu0.24.04.2) ...
Setting up libutf8proc3:amd64 (2.9.0-1build1) ...
Setting up libtommath1:amd64 (1.2.1-2build1) ...
Setting up libwinpr2-2t64:amd64 (2.11.5+dfsg1-1build2) ...
Setting up libpq5:amd64 (16.11-0ubuntu0.24.04.1) ...
Setting up firebird3.0-common-doc (3.0.11.33703.ds4-2ubuntu2) ...
Setting up libsnappy1v5:amd64 (1.1.10-1build1) ...
Setting up firebird3.0-common (3.0.11.33703.ds4-2ubuntu2) ...
Setting up libapr1t64:amd64 (1.7.2-3.1ubuntu0.1) ...
Setting up libhashkit2t64:amd64 (1.1.4-1.1build3) ...
Setting up libfreerdp2-2t64:amd64 (2.11.5+dfsg1-1build2) ...
Setting up libbson-1.0-0t64 (1.26.0-1.1ubuntu2) ...
Setting up libmongocrypt0:amd64 (1.8.4-1build3) ...
Setting up libaprutil1t64:amd64 (1.6.3-1.1ubuntu7) ...
Setting up libserf-1-1:amd64 (1.3.10-1ubuntu0.24.04.1) ...
Setting up libmemcached11t64:amd64 (1.1.4-1.1build3) ...
Setting up libfbclient2:amd64 (3.0.11.33703.ds4-2ubuntu2) ...
Setting up libmongoc-1.0-0t64 (1.26.0-1.1ubuntu2) ...
Setting up libsvn1:amd64 (1.14.3-1build4) ...
Setting up hydra (9.5-1build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethi
cs anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-07 10:23:50
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[ERROR] File for passwords not found: /usr/share/wordlists/rockyou.txt
alpha@alpha-VMware-Virtual-Platform:~$ sudo nmap -sS localhost
sudo: nmap: command not found
alpha@alpha-VMware-Virtual-Platform:~$ sudo nmap -sS localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2026-01-07 10:27 MST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000020s latency).
Not shown: 998 closed tcp ports (reset)
PORT    STATE SERVICE
22/tcp  open  ssh
631/tcp open  ipp

Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
alpha@alpha-VMware-Virtual-Platform:~$
```

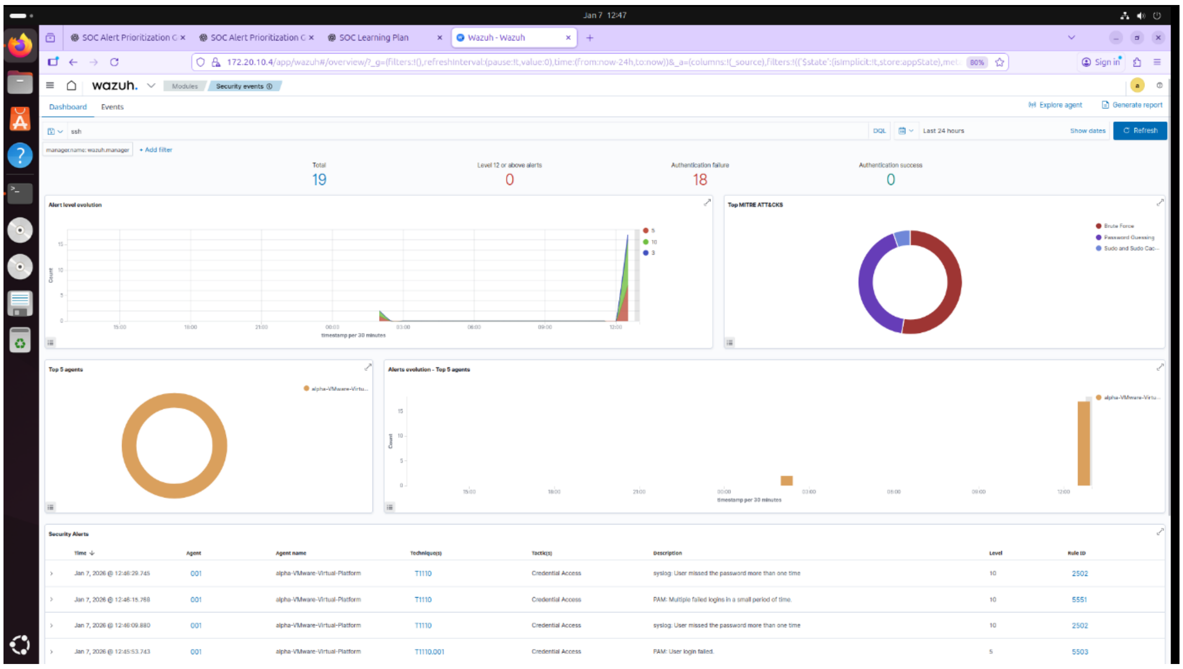
Alert Generation and Detection

The process starts when the agent machine generates logs based on activities happening on the system, such as a user trying to log in via SSH. These logs are sent to the Wazuh manager, which compares the raw data against a set of pre-defined security rules. For this task, the manager detected a series of failed login attempts that matched rule ID 2502, which is specifically for users missing their password more than once.

Once a rule is triggered, the system generates a formal alert in the dashboard. This alert includes important technical details like the rule level, which was set to 10 for high severity, and the MITRE ATT&CK ID T1110 to show it was a brute force attempt. This automation is what allows a SOC to detect attacks in real-time instead of manually searching through thousands of log files.


Jan 7 11:46
alpha@alpha-VMware-Virtual-Platform: ~
alpha@alpha-VMware-Virtual-Platform: ~
alpha@alpha-VMware-Virtual-Platform: ~

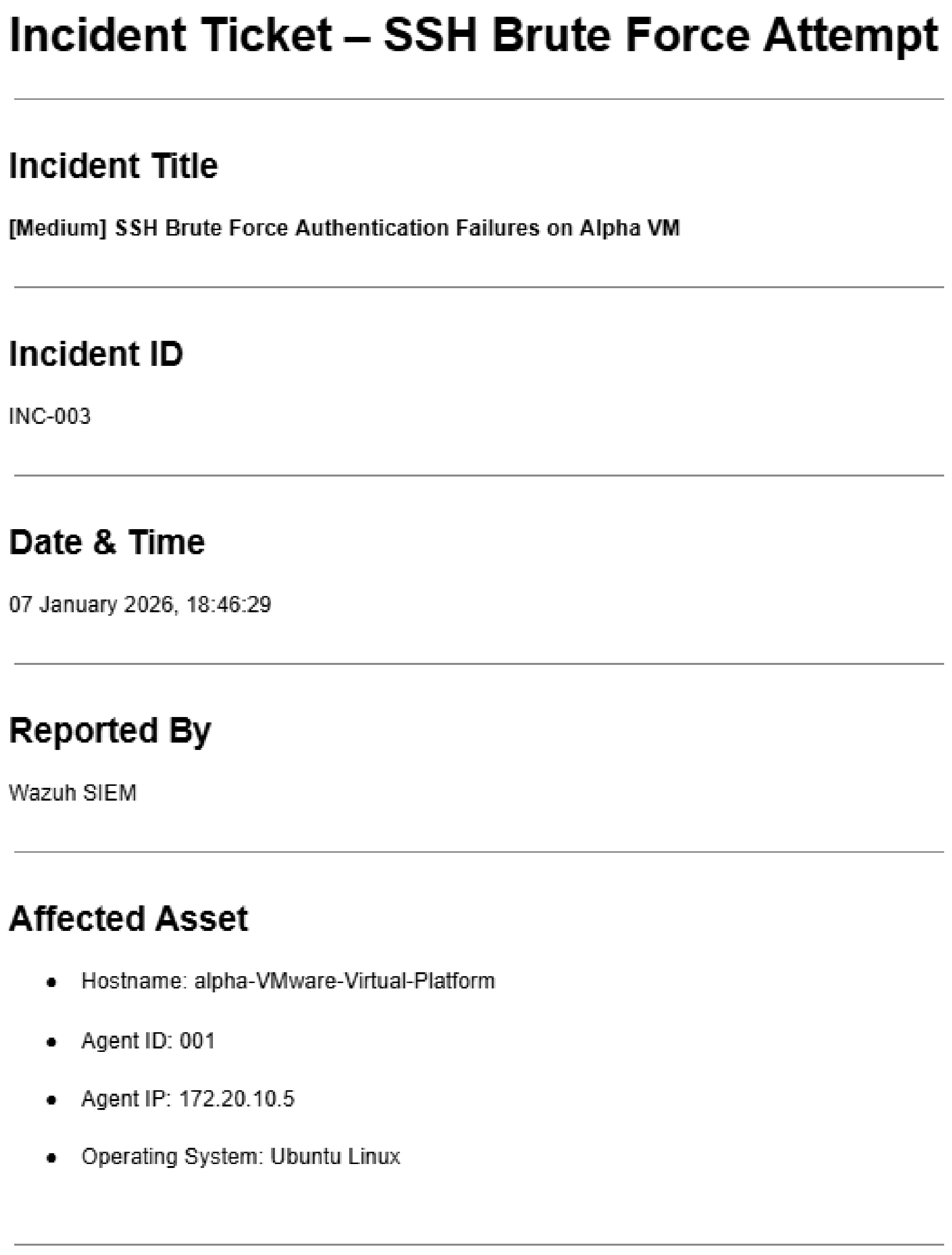
```
W: Some index files failed to download. They have been ignored, or old ones used instead.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done

No apt package "seclists", but there is a snap with that name.
Try "snap install seclists"

E: Unable to locate package seclists
alpha@alpha-VMware-Virtual-Platform:~$ for i in {1..15}; do ssh wronguser@localhost; done
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
wronguser@localhost: Permission denied (publickey,password).
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
wronguser@localhost: Permission denied (publickey,password).
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
wronguser@localhost: Permission denied (publickey,password).
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
wronguser@localhost: Permission denied (publickey,password).
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
wronguser@localhost: Permission denied (publickey,password).
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
wronguser@localhost: Permission denied (publickey,password).
wronguser@localhost's password:
```



	Time	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
›	Jan 7, 2026 @ 12:46:29.745	001	alpha-VMware-Virtual-Platform	T1110	Credential Access	syslog: User missed the password more than one time	10	2502
›	Jan 7, 2026 @ 12:46:15.768	001	alpha-VMware-Virtual-Platform	T1110	Credential Access	PAM: Multiple failed logins in a small period of time.	10	5551
›	Jan 7, 2026 @ 12:46:09.880	001	alpha-VMware-Virtual-Platform	T1110	Credential Access	syslog: User missed the password more than one time	10	2502
›	Jan 7, 2026 @ 12:45:53.743	001	alpha-VMware-Virtual-Platform	T1110.001	Credential Access	PAM: User login failed.	5	5503

Incident Ticket Creation

Detected security alerts were formally documented by creating incident tickets to ensure structured tracking and management of security incidents. Each incident ticket captured essential information, including the incident title, severity level, affected systems, identified indicators of compromise, and an initial analytical assessment. This structured documentation enables consistent incident handling, supports effective escalation to higher SOC tiers, and facilitates coordination among security teams. Additionally, maintaining detailed incident tickets ensures traceability and accountability throughout the entire incident lifecycle.



Incident Ticket – SSH Brute Force Attempt

Incident Title

[Medium] SSH Brute Force Authentication Failures on Alpha VM

Incident ID

INC-003

Date & Time

07 January 2026, 18:46:29

Reported By

Wazuh SIEM

Affected Asset

- Hostname: alpha-VMware-Virtual-Platform
- Agent ID: 001
- Agent IP: 172.20.10.5
- Operating System: Ubuntu Linux

Incident Category

Unauthorized Access Attempt / Brute Force Attack

Severity

Medium
(Rule Level: 10)

Status

Open

Incident Description

Wazuh detected multiple failed SSH authentication attempts on the Alpha Ubuntu virtual machine. The alerts indicate repeated password failures recorded by the SSH daemon, suggesting a brute-force attack against the SSH service. The activity was identified through system authentication logs and analyzed by the Wazuh manager.

No successful SSH login was observed during the incident timeframe.

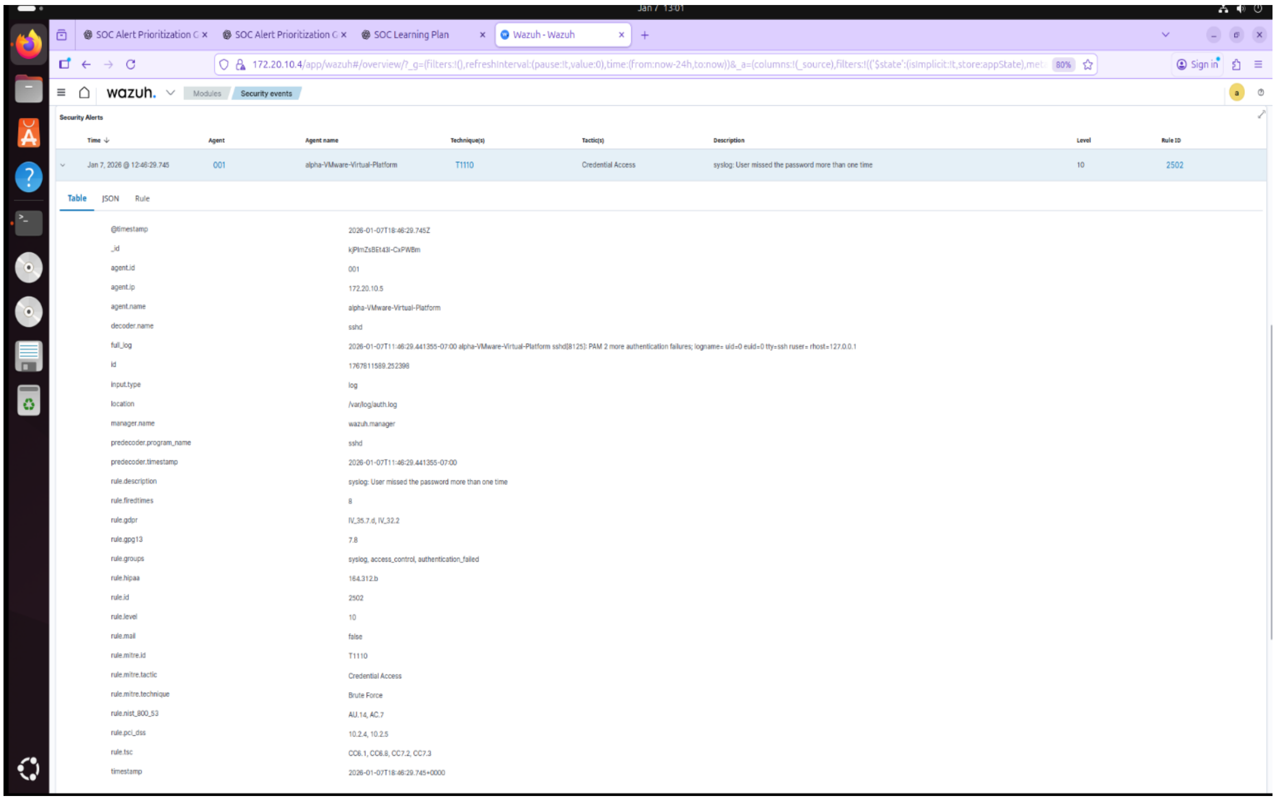
Detection Details

- Detection Tool: Wazuh Agent
- Decoder Name: sshd
- Log Source: /var/log/auth.log
- Rule ID: 2502
- Rule Description: User missed the password more than one time

Alert Generation and Detection

The process starts when the agent machine generates logs based on activities happening on the system, such as a user trying to log in via SSH. These logs are sent to the Wazuh manager, which compares the raw data against a set of pre-defined security rules. For this task, the manager detected a series of failed login attempts that matched rule ID 2502, which is specifically for users missing their password more than once.

Once a rule is triggered, the system generates a formal alert in the dashboard. This alert includes important technical details like the rule level, which was set to 10 for high severity, and the MITRE ATT&CK ID T1110 to show it was a brute force attempt. This automation is what allows a SOC to detect attacks in real-time instead of manually searching through thousands of log files.

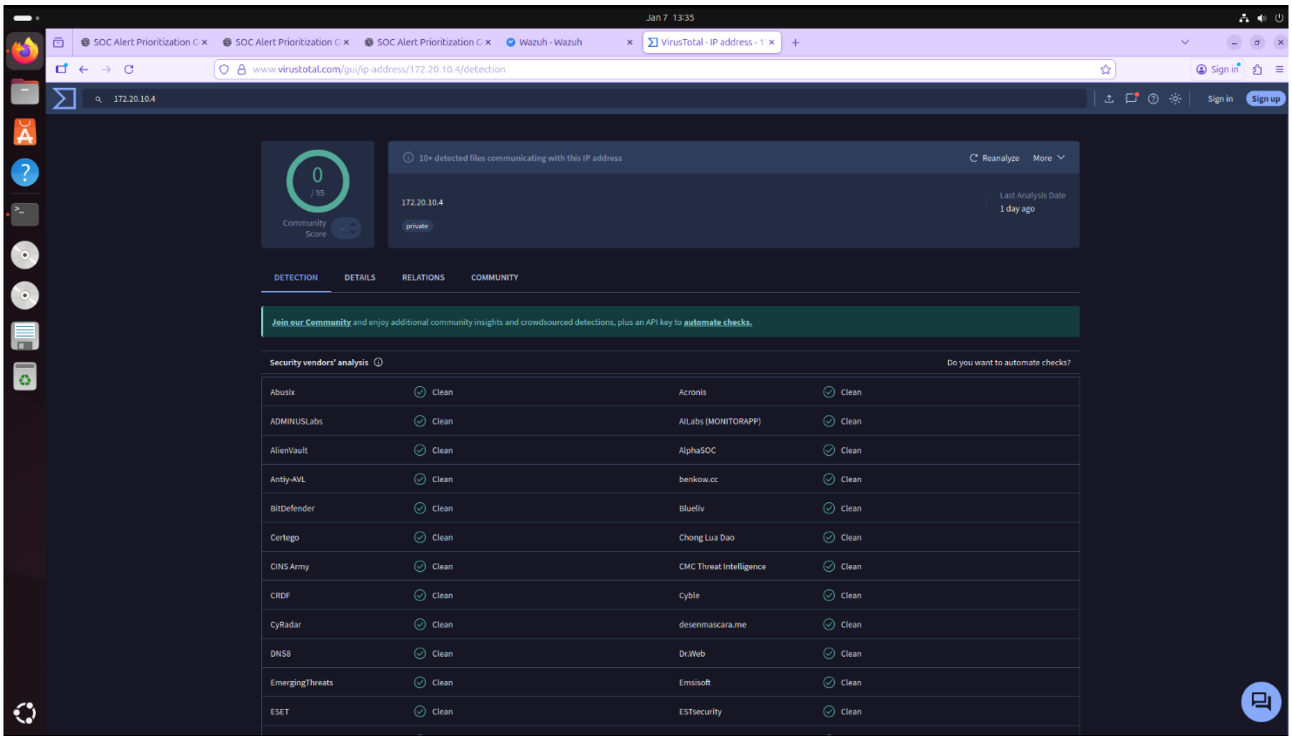


```
172.20.10.4/app/wazuh#/overview/?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now-24h,to:now))&_a=(columns:!(_source),filters:!(('$state':(isImplicit:!t,store:appState),met
```

Security Alerts

Time	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jan 7, 2026 @ 12:46:29.745	001	alpha-VMware-Virtual-Platform	T1110	Credential Access	syslog: User missed the password more than one time	10	2502

@timestamp	2026-01-07T18:46:29.745Z
_id	kjPlmZsBEt43l-CxPWBm
agent.id	001
agent.ip	172.20.10.5
agent.name	alpha-VMware-Virtual-Platform
decoder.name	sshd
full_log	2026-01-07T11:46:29.441355-07:00 alpha-VMware-Virtual-Platform sshd[8125]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=127.0.0.1
id	1767811589.252398
input.type	log
location	/var/log/auth.log
manager.name	wazuh.manager
predecoder.program_name	sshd
predecoder.timestamp	2026-01-07T11:46:29.441355-07:00
rule.description	syslog: User missed the password more than one time
rule.firedtimes	8
rule.gdpr	IV_35.7.d, IV_32.2
rule.gpg13	7.8
rule.groups	syslog, access_control, authentication_failed
rule.hipaa	164.312.b
rule.id	2502
rule.level	10
rule.mail	false
rule.mitre.id	T1110
rule.mitre.tactic	Credential Access
rule.mitre.technique	Brute Force
rule.nist_800_53	AU.14, AC.7
rule.pci_dss	10.2.4, 10.2.5
rule.tsc	CC6.1, CC6.8, CC7.2, CC7.3
timestamp	2026-01-07T18:46:29.745+0000



www.virustotal.com/gui/ip-address/172.20.10.4/detection

172.20.10.4

0 / 95 Community Score

10+ detected files communicating with this IP address

172.20.10.4

private

Last Analysis Date 1 day ago

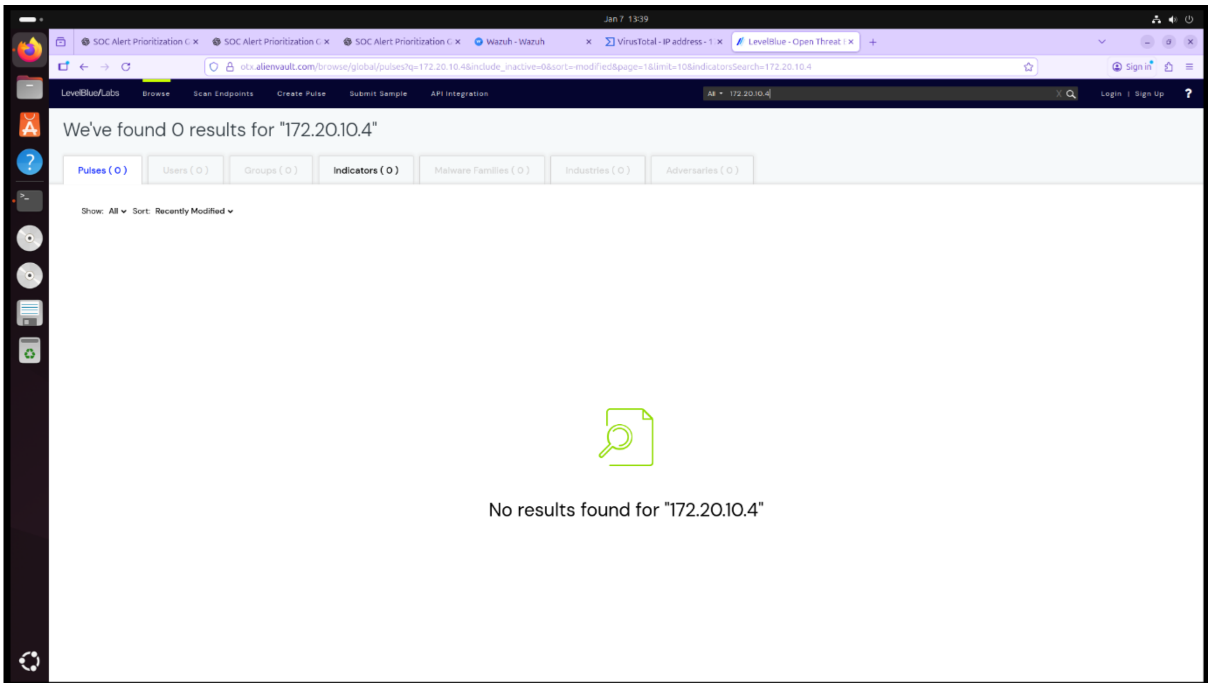
DETECTION DETAILS RELATIONS COMMUNITY

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

Security vendors' analysis

Do you want to automate checks?

Abusix	Clean	Acronis	Clean
ADMINUSLabs	Clean	AILabs (MONITORAPP)	Clean
AlienVault	Clean	AlphaSOC	Clean
Antiy-AVL	Clean	benkow.cc	Clean
BitDefender	Clean	Bluellv	Clean
Certego	Clean	Chong Lua Dao	Clean
CINS Army	Clean	CMC Threat Intelligence	Clean
CRDF	Clean	Cyble	Clean
CyRadar	Clean	desenmascara.me	Clean
DNS8	Clean	Dr.Web	Clean
EmergingThreats	Clean	Emsisoft	Clean
ESET	Clean	ESTsecurity	Clean



otx.alienvault.com/browse/global/pulses?q=172.20.10.4&include_inactive=0&sort=-modified&page=1&limit=10&indicatorsSearch=172.20.10.4

We've found 0 results for "172.20.10.4"

Pulses (0) Users (0) Groups (0) Indicators (0) Malware Families (0) Industries (0) Adversaries (0)

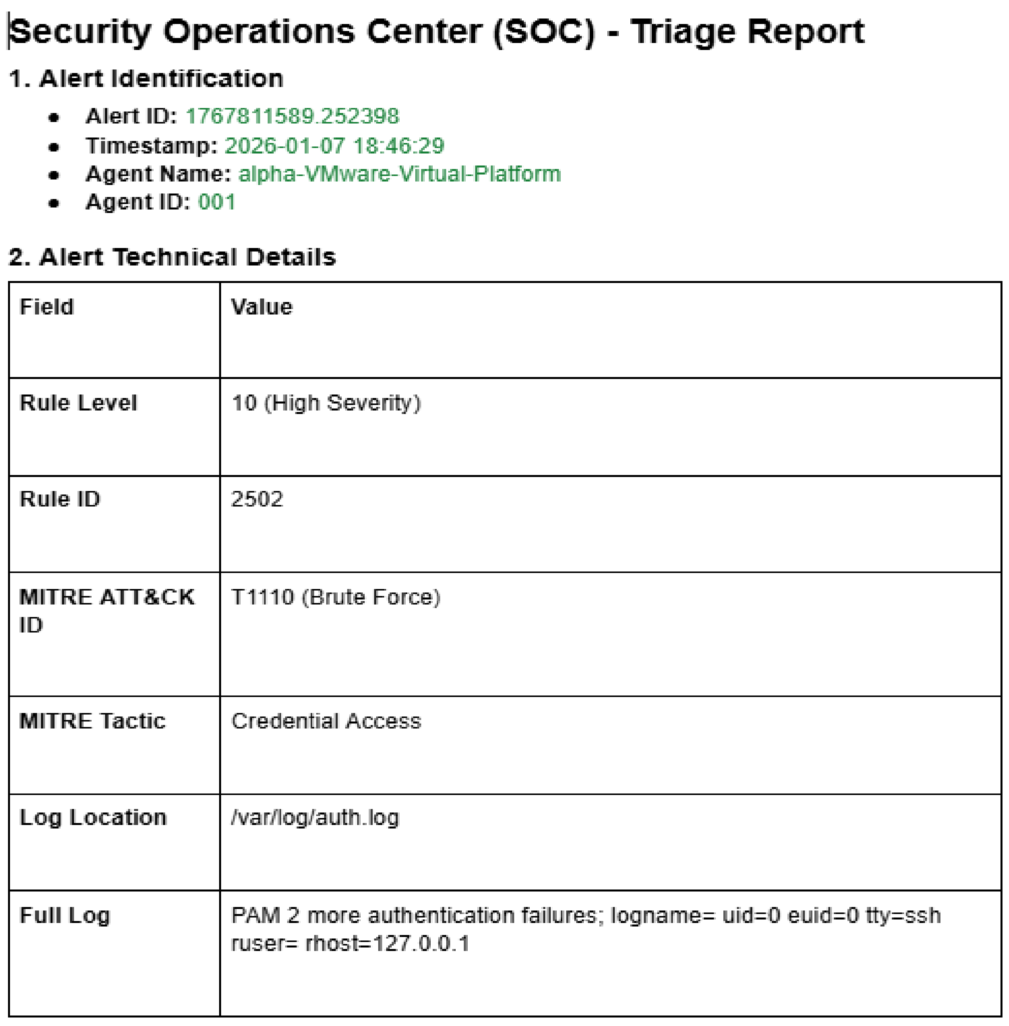
Show: All Sort: Recently Modified

No results found for "172.20.10.4"

Incident Response and Documentation

After performing the triage, the next step was to record all the actions taken and decide how to handle the situation. Since the logs showed that the brute-force attempts did not result in a successful login, the system was not actually broken into, so the response was focused on documenting the event and suggesting ways to stop it from happening again.

A formal incident report was created to keep a clear record of the event for the SOC team. This documentation includes a summary of the attack, a timeline of when the failed logins started, and a list of steps taken to verify the alert. The report also features a lessons learned section to help improve the security of the agent, such as recommending stronger password policies or checking for automated scripts that might be causing the local login failures.



Security Operations Center (SOC) - Triage Report

1. Alert Identification

- **Alert ID:** 1767811589.252398
- **Timestamp:** 2026-01-07 18:46:29
- **Agent Name:** alpha-VMware-Virtual-Platform
- **Agent ID:** 001

2. Alert Technical Details

Field	Value
Rule Level	10 (High Severity)
Rule ID	2502
MITRE ATT&CK ID	T1110 (Brute Force)
MITRE Tactic	Credential Access
Log Location	/var/log/auth.log
Full Log	PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=127.0.0.1

3. Triage Documentation Table

Alert ID	Description	Source IP	Target	Severity	Analyst Verdict
003	SSH brute-force authentication failure	127.0.0.1	VM2 (alpha-VMware)	Medium	**True Positive**

4. Analyst Triage Decision & Reasoning

Verdict: True Positive

Detailed Reasoning:

- **Authentication Activity:** The logs explicitly show multiple failed login attempts (rule.firedtimes: 8) via SSH.
- **Verification:** The full_log field confirms that the Pluggable Authentication Module (PAM) triggered an alert for repeated authentication failures.
- **Source Analysis:** The source IP is 127.0.0.1 (localhost), suggesting the attempts may be originating from an internal script, a user already on the system, or a misconfigured service on the local machine.
- **Impact Assessment:** While the attempts were unsuccessful (no compromise detected), the activity is confirmed as a real brute-force pattern and not a system glitch, justifying the "True Positive" classification.

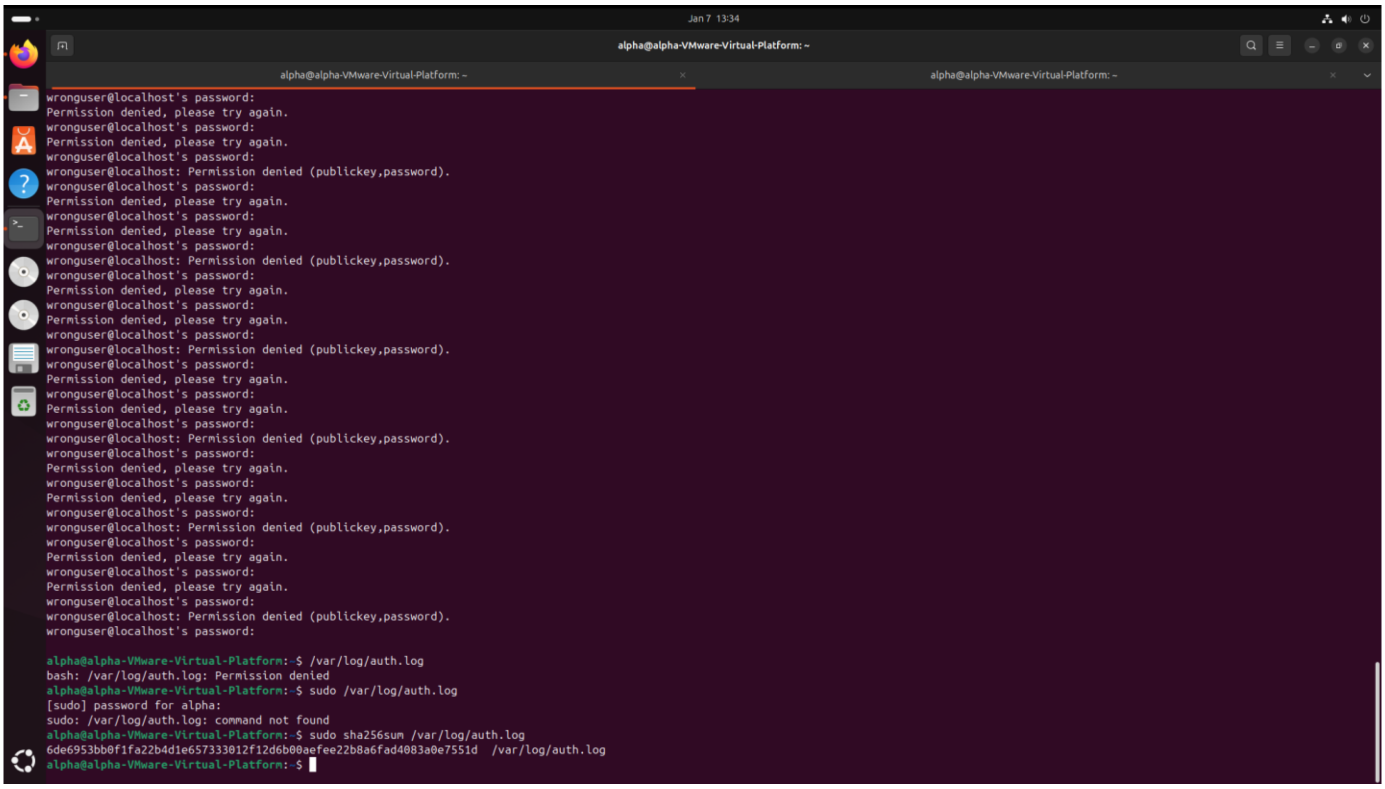
Evidence Preservation and Chain of Custody

After completing the response, the final part of the process involves making sure that all information gathered is kept safe and documented correctly. This is known as evidence preservation and chain of custody, which is essential if the incident ever needs to be reviewed or used for a legal case.

During the lab, I practiced these steps to ensure the data was handled properly:

- Identifying evidence: I picked out the important logs and alert details from the ssh attack to save for the report.
- Data collection: I used tools to gather the system information while it was still running to make sure no data was lost.
- Hashing: I learned that every piece of evidence should have a unique digital fingerprint, or hash, to prove that it hasn't been changed since it was collected.
- Documentation: I recorded who collected the data, when it was taken, and what tools were used, which creates a clear chain of custody.

By following these rules, the SOC ensures that the investigation is reliable and that the evidence can be trusted by anyone who reviews the case later.


Jan 7 13:34
alpha@alpha-VMware-Virtual-Platform: ~
alpha@alpha-VMware-Virtual-Platform: ~
alpha@alpha-VMware-Virtual-Platform: ~

```
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
wronguser@localhost: Permission denied (publickey,password).
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
wronguser@localhost: Permission denied (publickey,password).
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
wronguser@localhost: Permission denied (publickey,password).
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
wronguser@localhost: Permission denied (publickey,password).
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
wronguser@localhost: Permission denied (publickey,password).
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
Permission denied, please try again.
wronguser@localhost's password:
wronguser@localhost: Permission denied (publickey,password).
wronguser@localhost's password:

alpha@alpha-VMware-Virtual-Platform:~$ /var/log/auth.log
bash: /var/log/auth.log: Permission denied
alpha@alpha-VMware-Virtual-Platform:~$ sudo /var/log/auth.log
[sudo] password for alpha:
sudo: /var/log/auth.log: command not found
alpha@alpha-VMware-Virtual-Platform:~$ sudo sha256sum /var/log/auth.log
6de6953bb0f1fa22b4d1e657333012f12d6b00aefee22b8a6fad4083a0e7551d  /var/log/auth.log
alpha@alpha-VMware-Virtual-Platform:~$
```

Capstone Project: End-to-End SOC Workflow

The capstone task demonstrated the complete SOC workflow from attack simulation to final reporting. A security event was generated, detected, triaged, responded to, and documented end-to-end. A detailed incident report and a non-technical management briefing were prepared, highlighting both technical analysis and professional communication skills required in SOC operations.

Beyond just detecting the attack, this project emphasized the importance of proper documentation and evidence preservation. Learning how to create incident reports and maintain a chain of custody ensures that the work of a SOC analyst is accurate and useful for future security improvements. Overall, this task provided the hands-on experience needed to understand the daily responsibilities and technical skills required in a modern Security Operations Center.

Conclusion

This week's assignment provided a comprehensive understanding of both the theoretical and practical aspects of Security Operations Center (SOC) operations. Through structured learning and hands-on activities, key concepts such as alert prioritization, incident classification, alert triage, incident response, and evidence preservation were studied and applied in a simulated SOC environment. The use of a SIEM-based lab enabled practical exposure to real-world security monitoring and incident handling processes.

The practical implementation reinforced theoretical knowledge by demonstrating how security alerts are generated, analyzed, documented, and resolved in an operational SOC setting. Additionally, the assignment emphasized the importance of proper documentation, evidence integrity, and professional communication throughout the incident lifecycle. Overall, this week's work strengthened foundational SOC skills and contributed to a deeper understanding of incident handling workflows, preparing for more advanced security operations and real-world SOC responsibilities.