Threat Visibility for FireOps Systems

Security Monitoring Environment Project

Team Name: DefenseGrid

Team leader: Adnan Mardini

Team members: Peter Abiya | Oluwanifemi Aduraponmile | Oneneobari Obe | Chancelle Ahinon
Project: FireOps Systems Security Monitoring

Overview

The FireOps security monitoring environment was designed to simulate an enterprise-level security infrastructure comprising Windows endpoints, Ubuntu monitoring servers, and cloud connectivity through AWS.

The environment leveraged Wazuh as a centralized SIEM platform for log aggregation and threat detection, OpenSearch for log storage and analysis, and Wazuh Dashboard for alert visualization. Tailscale provided secure encrypted networking across all distributed components.

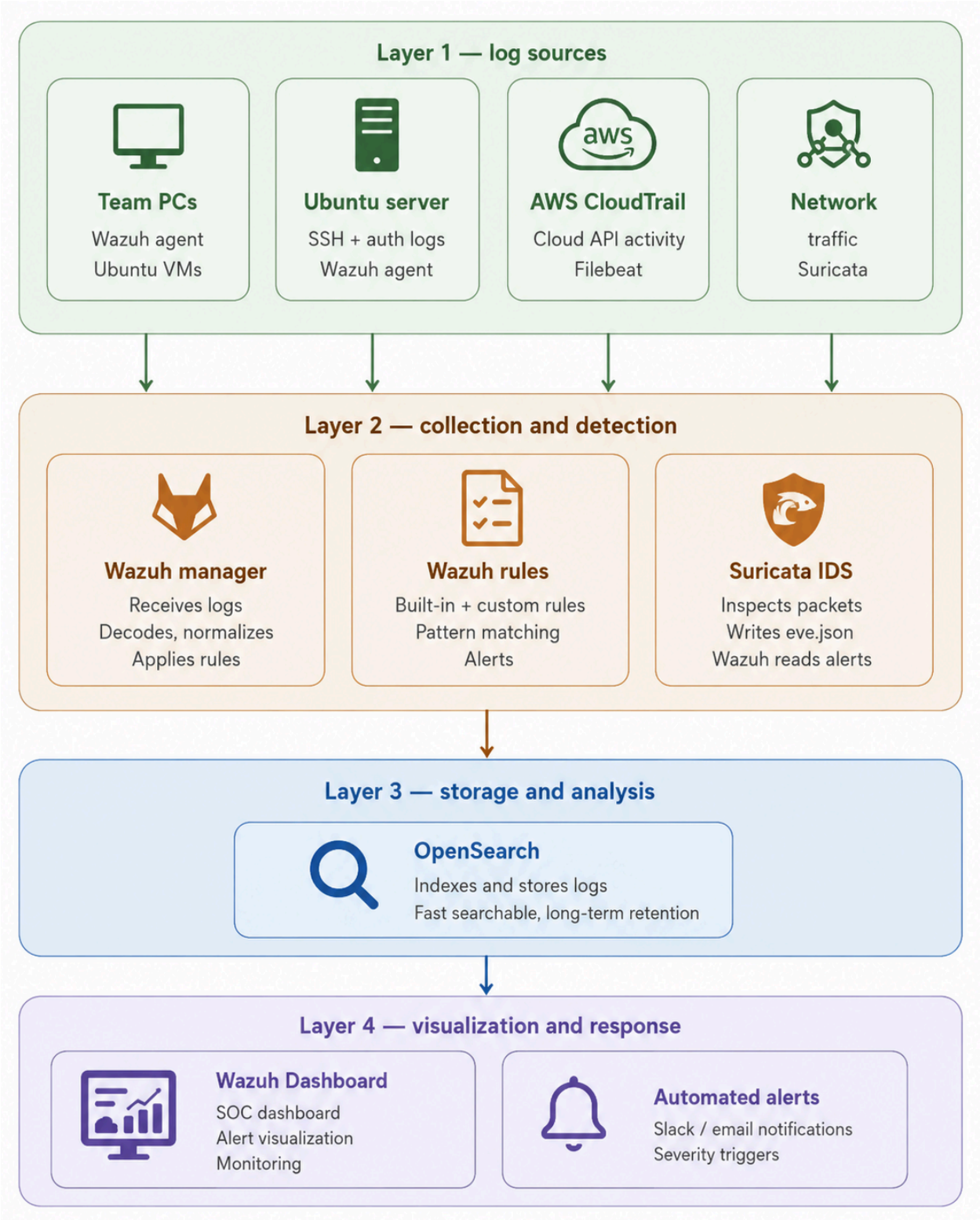
Security simulations, including brute-force SSH attacks using Hydra, Port scanning using nmap and privileged command execution, were conducted to evaluate detection and response capabilities.

The monitoring system successfully identified and logged authentication failures, configuration changes, and administrative operations, demonstrating effective visibility into potential security events.

Alerts were triaged and categorized by severity to prioritize response actions, and incident handling followed the structured lifecycle of preparation, detection and analysis, containment, eradication, recovery, and lessons learned as recommended by the NIST Cybersecurity Framework.

The exercise validated the capability of the monitoring infrastructure to detect and respond to simulated threats while highlighting the importance of strong authentication controls, least-privilege policies, and continuous monitoring for proactive risk mitigation in organizational environments.

FireOps Architecture Design



The system architecture was designed to simulate an enterprise security monitoring environment consisting of Windows host machines, Ubuntu virtual machines acting as monitoring servers, and external connectivity through AWS CloudTrail and Azure Cloud.

Log Sources (Layer 1): - Windows endpoints generating security events and authentication logs - Ubuntu servers with SSH and system activity - AWS CloudTrail for cloud API activity - Network traffic through Suricata IDS

Collection & Detection (Layer 2): - Wazuh Manager receiving all agent logs from endpoints - Wazuh Rules engine applying built-in and custom detection rules - Suricata IDS inspecting network packets and generating alerts

Storage & Analysis (Layer 3): - OpenSearch indexing and storing all logs for fast search and long-term retention - Time-series database optimized for security log analysis

Visualization & Response (Layer 4): - Wazuh Dashboard providing SOC and executive dashboards - Automated alerts via email for critical events - Incident management and tracking

Infrastructure: - Tailscale mesh VPN connecting all components securely (100.x.x.x overlay network) - Docker containerization for deployment - Ubuntu as the base operating system

---

Installation of Wazuh and OpenSearch

Wazuh was deployed on the Ubuntu monitoring server using Docker containers alongside OpenSearch for log storage.

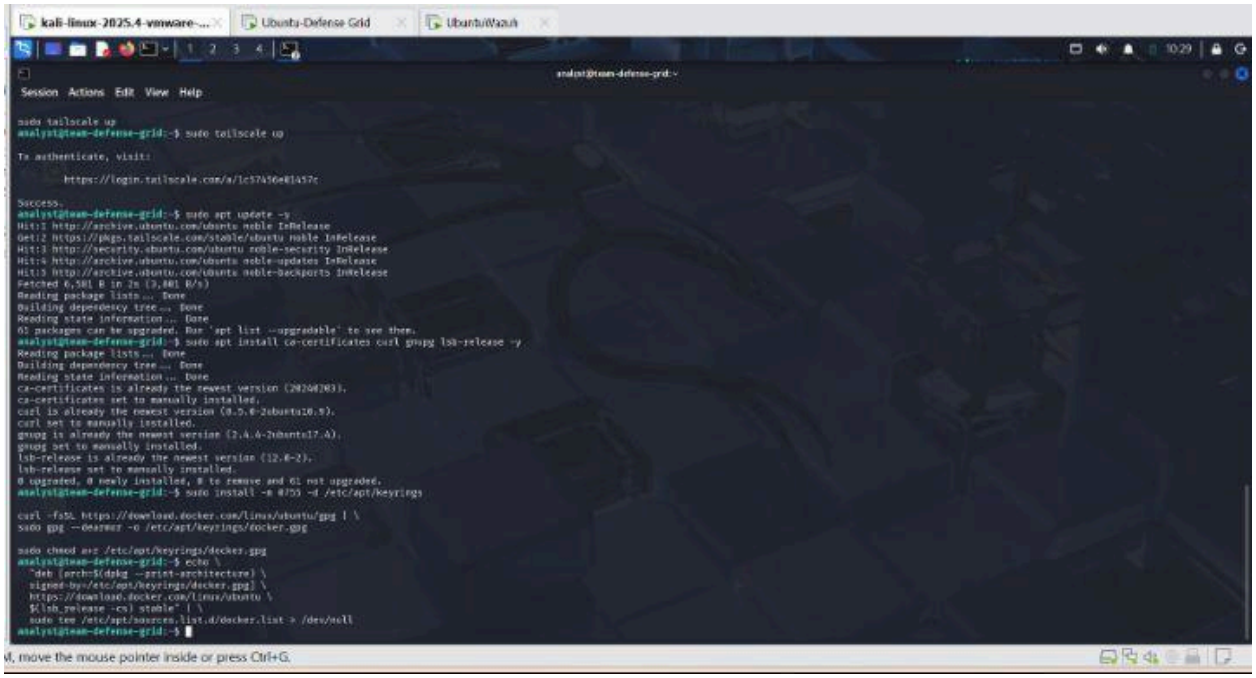
System Preparation

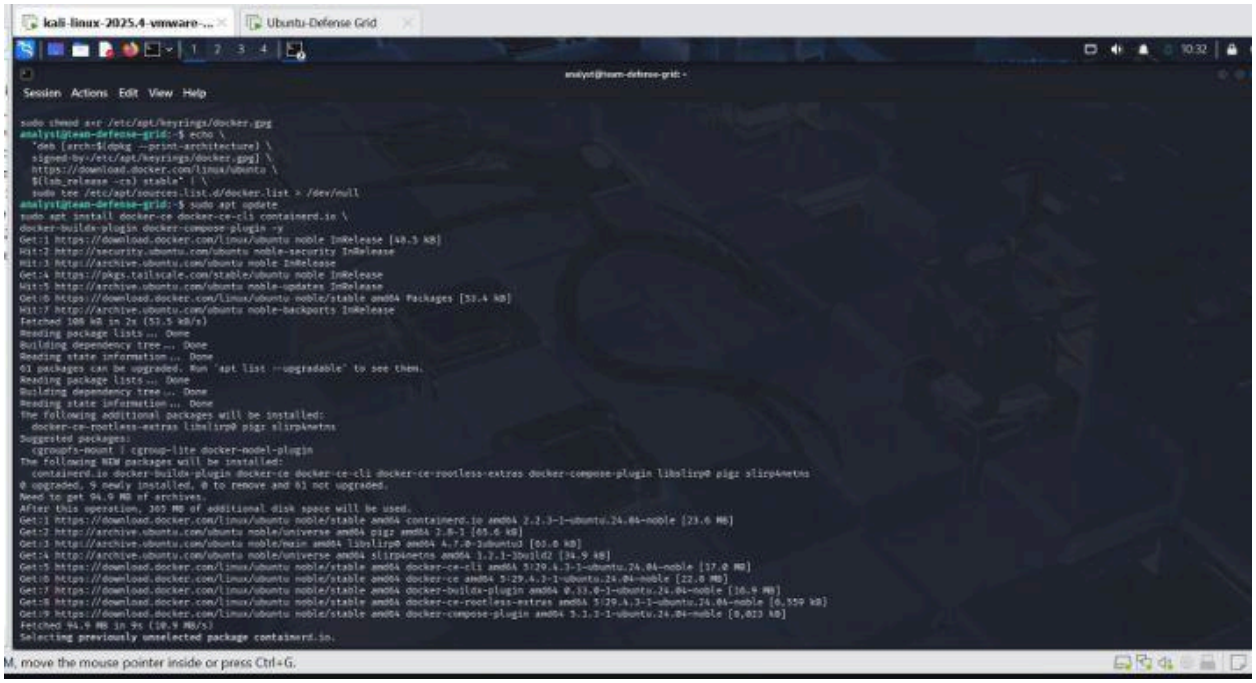
```
sudo apt update
sudo apt upgrade -y
```

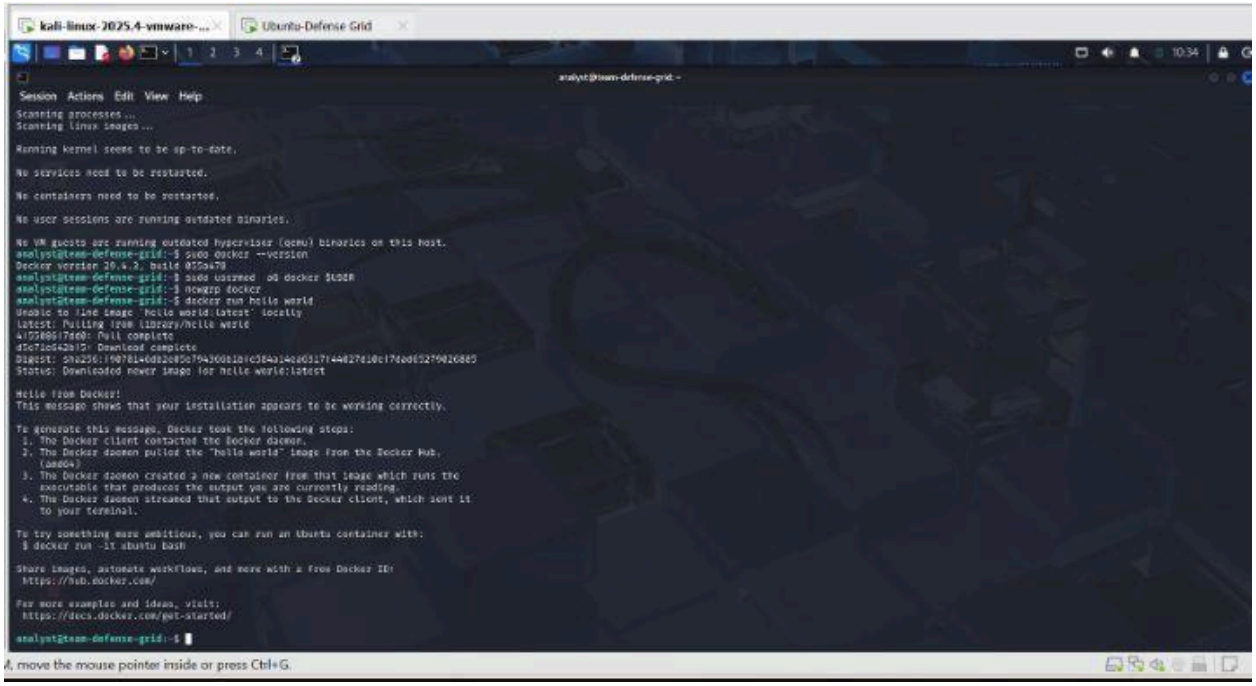
Wazuh Installation via Docker

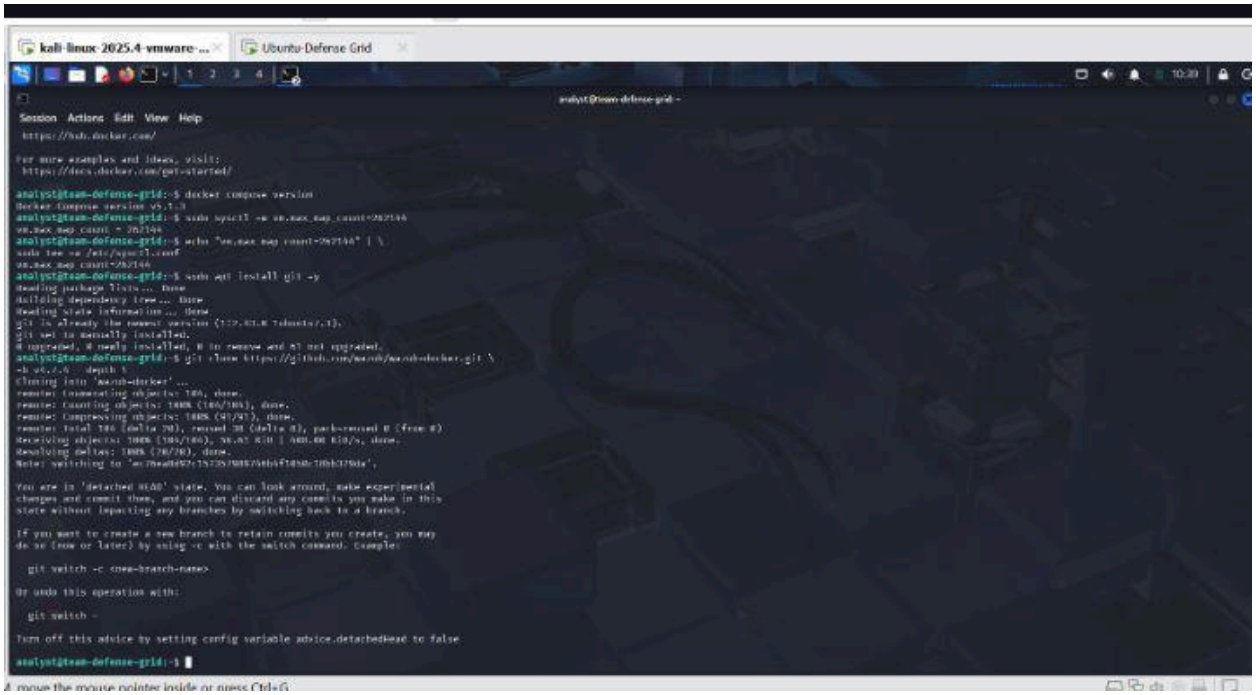
```
git clone https://github.com/wazuh/wazuh-docker.git -b v4.7.4 --depth 1
cd wazuh-docker/single-node
```

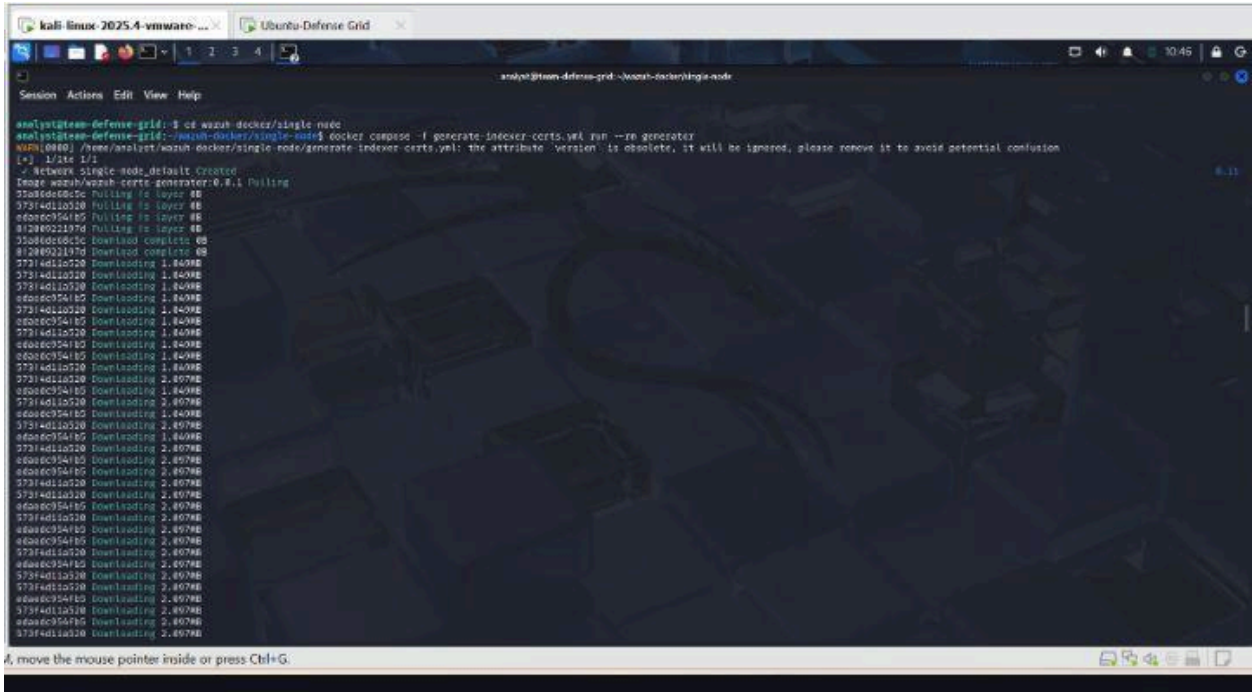
This automatically installs: - **Wazuh Manager** - Log collection and processing - **OpenSearch** - Log storage and indexing - **Wazuh Dashboard** - Visualization and alerting

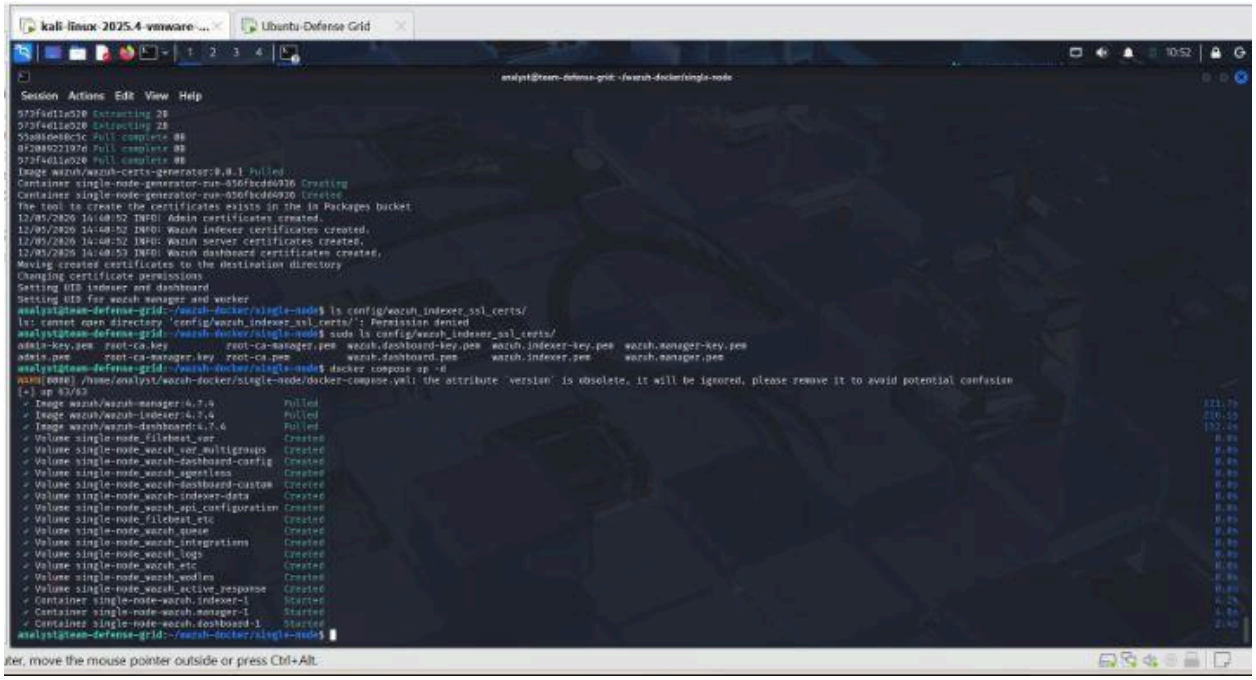

kali-linux-2025.4-vmware-...
Ubuntu-Defense Grid
Ubuntu-Wazuh
Session Actions Edit View Help
sudo tailscale up
analyst@team-defense-grid:~$ sudo tailscale up
To authenticate, visit:
https://login.tailscale.com/a/1c57a50e81457c
Success.
analyst@team-defense-grid:~$ sudo apt update -y
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Get:2 https://pkgs.tailscale.com/stable/ubuntu noble InRelease
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 6,501 B in 2s (3,801 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
61 packages can be upgraded. Run 'apt list --upgradable' to see them.
analyst@team-defense-grid:~$ sudo apt install ca-certificates curl gnupg lsb-release -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.6).
curl set to manually installed.
gnupg is already the newest version (2.4.4-2ubuntu17.4).
gnupg set to manually installed.
lsb-release is already the newest version (12.0-2).
lsb-release set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 61 not upgraded.
analyst@team-defense-grid:~$ sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \
sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
analyst@team-defense-grid:~$ echo \
"deb [arch=$(dpkg --print-architecture) \
signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
analyst@team-defense-grid:~$
M, move the mouse pointer inside or press Ctrl+G.

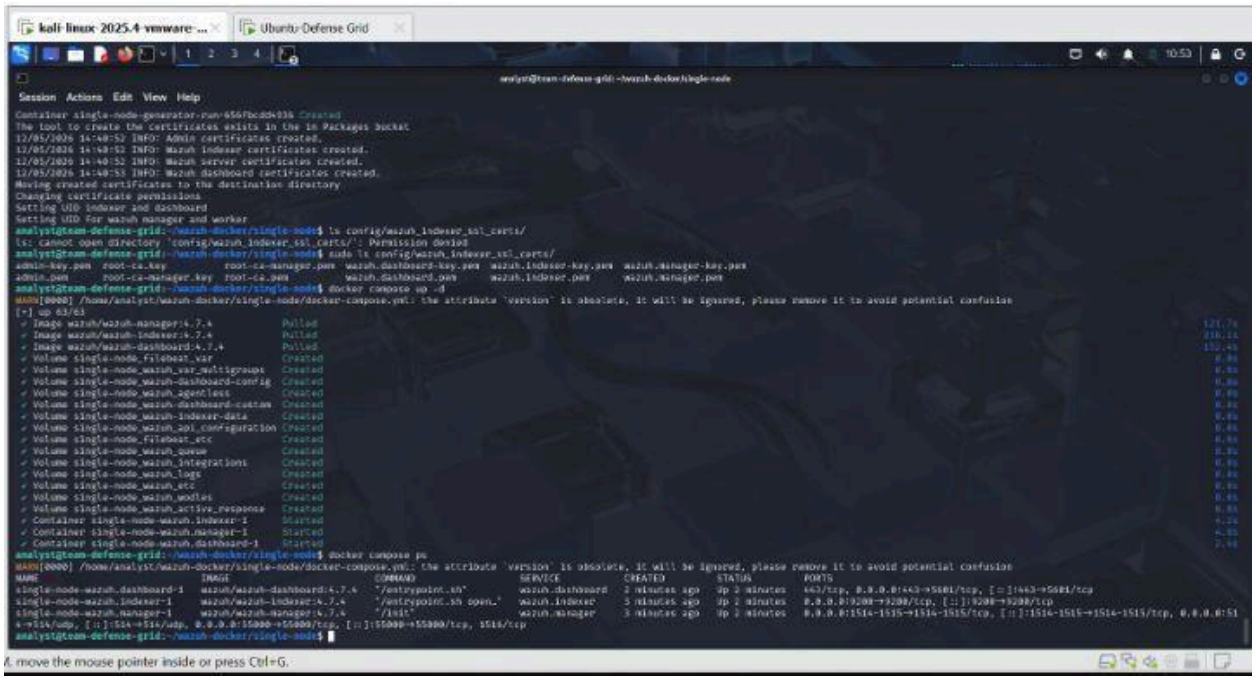

kali-linux-2025.4-vmware-...
Ubuntu-Defense Grid
Session Actions Edit View Help
sudo chmod a+r /etc/apt/keyrings/docker.gpg
analyst@team-defense-grid:~$ echo \
"deb [arch=$(dpkg --print-architecture) \
signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
analyst@team-defense-grid:~$ sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io \
docker-buildx-plugin docker-compose-plugin -y
Get:1 https://download.docker.com/linux/ubuntu noble InRelease [48.5 kB]
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble InRelease
Get:4 https://pkgs.tailscale.com/stable/ubuntu noble InRelease
Hit:5 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [53.4 kB]
Hit:7 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 106 kB in 2s (53.5 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
61 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
docker-ce-rootless-extras libslirp0 pigz slirp4netns
Suggested packages:
cgroupfs-mount | cgroup-lite docker-model-plugin
The following NEW packages will be installed:
containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libslirp0 pigz slirp4netns
0 upgraded, 9 newly installed, 0 to remove and 61 not upgraded.
Need to get 94.9 MB of archives.
After this operation, 365 MB of additional disk space will be used.
Get:1 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 2.2.3-1~ubuntu.24.04~noble [23.6 MB]
Get:2 http://archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 kB]
Get:5 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-cli amd64 5:29.4.3-1~ubuntu.24.04~noble [17.0 MB]
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce amd64 5:29.4.3-1~ubuntu.24.04~noble [22.8 MB]
Get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-buildx-plugin amd64 0.33.0-1~ubuntu.24.04~noble [16.9 MB]
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-rootless-extras amd64 5:29.4.3-1~ubuntu.24.04~noble [6,559 kB]
Get:9 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-compose-plugin amd64 5.1.3-1~ubuntu.24.04~noble [8,033 kB]
Fetched 94.9 MB in 9s (10.9 MB/s)
Selecting previously unselected package containerd.io.
M, move the mouse pointer inside or press Ctrl+G.

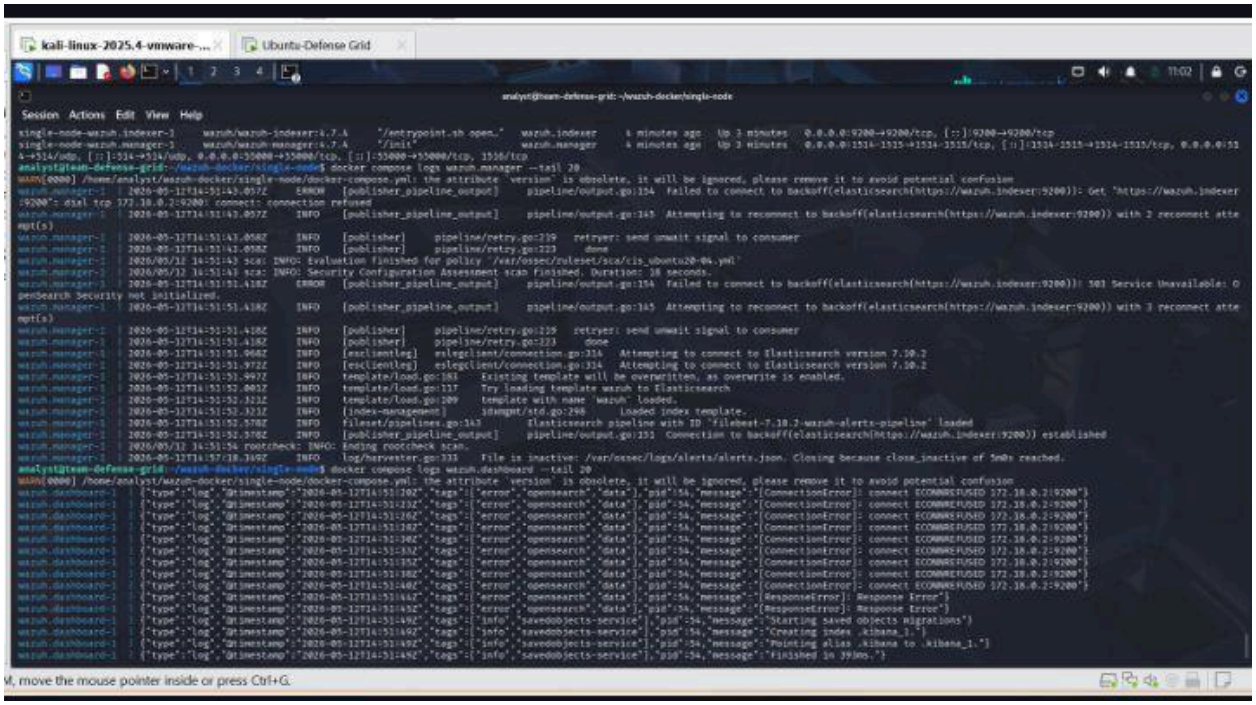

kali-linux-2025.4-vmware-...
Ubuntu-Defense Grid
Session Actions Edit View Help
Scanning processes...
Scanning Linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
analyst@team-defense-grid:~$ sudo docker --version
Docker version 28.4.2, build 0725a78
analyst@team-defense-grid:~$ sudo usermod -aG docker $USER
analyst@team-defense-grid:~$ newgrp docker
analyst@team-defense-grid:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e7790b17b6b0: Pull complete
d7e71b642b15: Download complete
Digest: sha256:f9078146db2085e794366b10fc8f4a12eec087174482f410ec7fe84d5275026885
Status: Downloaded newer image for hello-world:latest
Hello from Docker!
This message shows that your installation appears to be working correctly.
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the
executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
to your terminal.
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
For more examples and ideas, visit:
https://docs.docker.com/get-started/
analyst@team-defense-grid:~$
4, move the mouse pointer inside or press Ctrl+G.


kali-linux-2025.4-vmware-...
Ubuntu-Defense Grid
Session Actions Edit View Help
https://hub.docker.com/
For more examples and ideas, visit:
https://docs.docker.com/get-started/
analyst@team-defense-grid:~$ docker compose version
Docker Compose version v5.1.3
analyst@team-defense-grid:~$ sudo sysctl -w vm.max_map_count=262144
vm.max_map_count = 262144
analyst@team-defense-grid:~$ echo "vm.max_map_count=262144" | \
sudo tee -a /etc/sysctl.conf
vm.max_map_count=262144
analyst@team-defense-grid:~$ sudo apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.3).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 67 not upgraded.
analyst@team-defense-grid:~$ git clone https://github.com/marcb/marcb-docker.git \
-b v0.2.6 --depth 1
Cloning into 'marcb-docker'...
remote: Enumerating objects: 146, done.
remote: Counting objects: 100% (146/146), done.
remote: Compressing objects: 100% (47/47), done.
remote: Total 146 (delta 70), reused 38 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (146/146), 56.41 KiB | 460.00 KiB/s, done.
Resolving deltas: 100% (70/70), done.
Note: switching to 'ac70ea8d52c15735788570ab1f1858c10bb3750a'.
You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.
If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:
git switch -c <new-branch-name>
Or undo this operation with:
git switch -
Turn off this advice by setting config variable advice.detachedHead to false
analyst@team-defense-grid:~$
4, move the mouse pointer inside or press Ctrl+G.

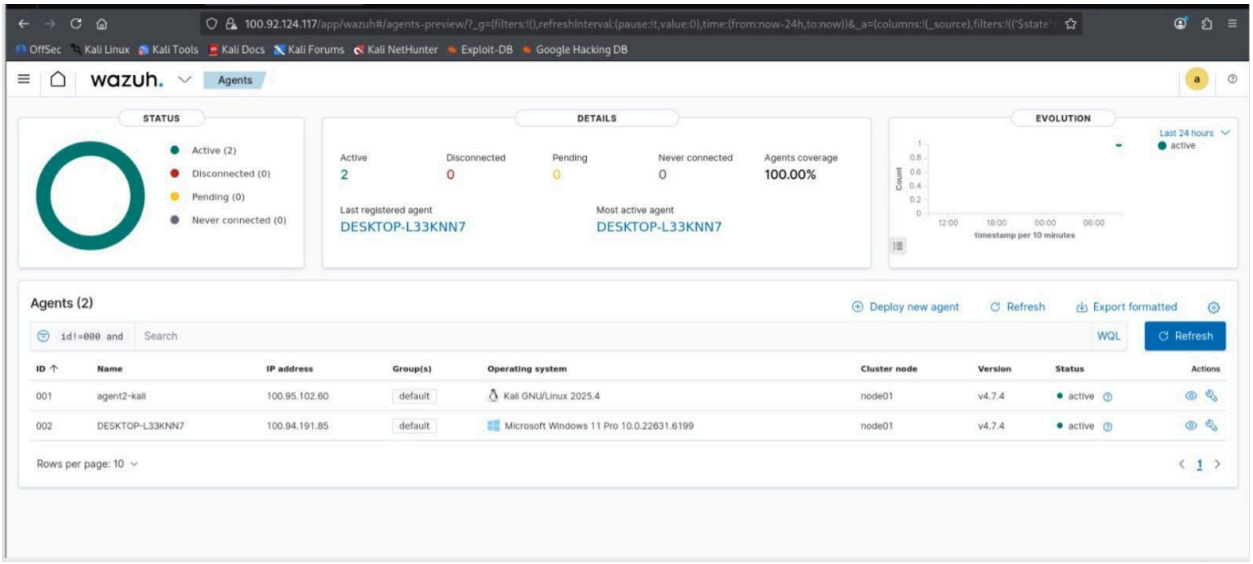

kali-linux-2025.4-vmware-...
Ubuntu-Defense Grid
Session Actions Edit View Help
analyst@team-defense-grid:~$ cd wazuh-docker/single-node
analyst@team-defense-grid:~/wazuh-docker/single-node$ docker compose -f generate-indexer-certs.yml run --rm generator
[+] 1/1te 1/1
✓ Network single-node_default Created
Image wazuh/wazuh-certs-generator:0.0.1 Pulling
, move the mouse pointer inside or press Ctrl+G.


kali-linux-2025.4-vmware-...
Ubuntu-Defense Grid
Session Actions Edit View Help
Image wazuh/wazuh-certs-generator:0.0.1 Pulled
Container single-node-generator-run-056fbcd66936 Creating
Container single-node-generator-run-056fbcd66936 Created
The tool to create the certificates exists in the in Packages bucket
12/05/2026 14:40:52 INFO: Admin certificates created.
12/05/2026 14:40:52 INFO: Wazuh indexer certificates created.
12/05/2026 14:40:52 INFO: Wazuh server certificates created.
12/05/2026 14:40:53 INFO: Wazuh dashboard certificates created.
Moving created certificates to the destination directory
Changing certificate permissions
Setting UID indexer and dashboard
Setting UID for wazuh manager and worker
analyst@team-defense-grid:~/wazuh-docker/single-node$ ls config/wazuh_indexer_ssl_certs/
ls: cannot open directory 'config/wazuh_indexer_ssl_certs/': Permission denied
analyst@team-defense-grid:~/wazuh-docker/single-node$ sudo ls config/wazuh_indexer_ssl_certs/
admin-key.pem root-ca.key root-ca-manager.pem wazuh.dashboard-key.pem wazuh.indexer-key.pem wazuh.manager-key.pem
admin.pem root-ca-manager.key root-ca.pem wazuh.dashboard.pem wazuh.indexer.pem wazuh.manager.pem
analyst@team-defense-grid:~/wazuh-docker/single-node$ docker compose up -d
[+] up 63/63
✓ Image wazuh/wazuh-manager:4.7.4 Pulled
✓ Image wazuh/wazuh-indexer:4.7.4 Pulled
✓ Image wazuh/wazuh-dashboard:4.7.4 Pulled
✓ Volume single-node_filebeat_var Created
✓ Volume single-node_wazuh_var_multigroups Created
✓ Volume single-node_wazuh-dashboard-config Created
✓ Volume single-node_wazuh_agentless Created
✓ Volume single-node_wazuh-dashboard-custom Created
✓ Volume single-node_wazuh-indexer-data Created
✓ Volume single-node_wazuh_api_configuration Created
✓ Volume single-node_filebeat_etc Created
✓ Volume single-node_wazuh_queue Created
✓ Volume single-node_wazuh_integrations Created
✓ Volume single-node_wazuh_logs Created
✓ Volume single-node_wazuh_etc Created
✓ Volume single-node_wazuh_wodles Created
✓ Volume single-node_wazuh_active_response Created
✓ Container single-node-wazuh.indexer-1 Started
✓ Container single-node-wazuh.manager-1 Started
✓ Container single-node-wazuh.dashboard-1 Started
analyst@team-defense-grid:~/wazuh-docker/single-node$
ter, move the mouse pointer outside or press Ctrl+Alt.





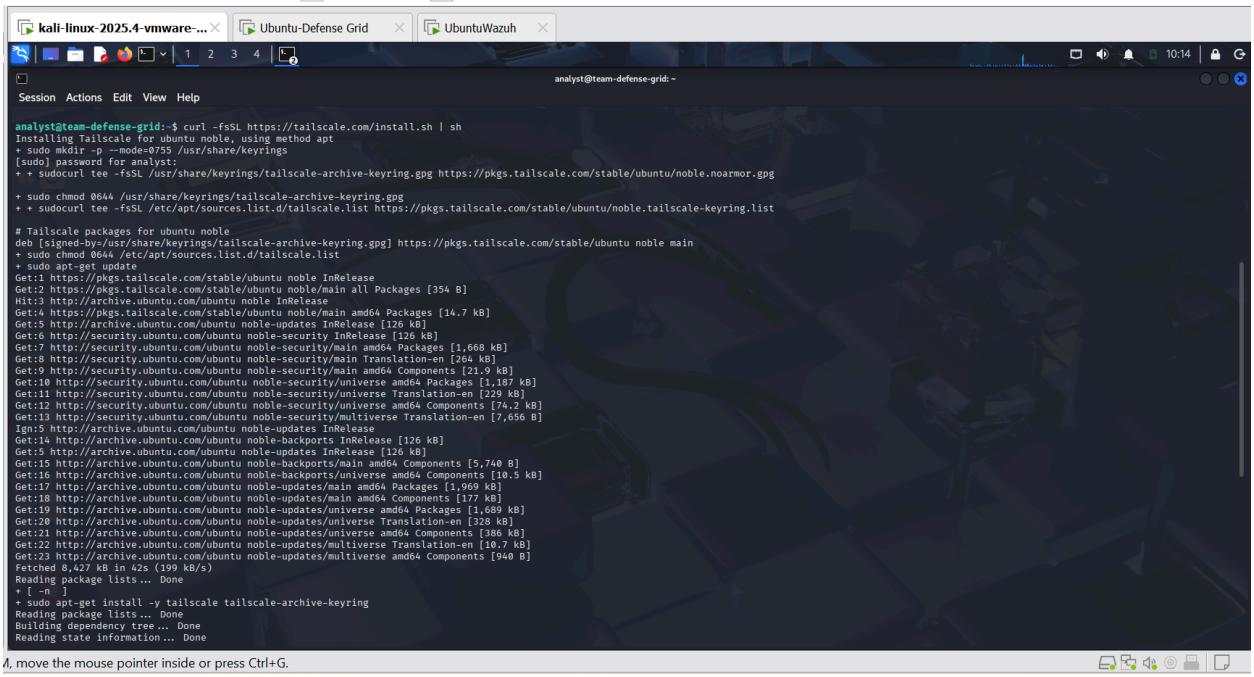
Access Wazuh Dashboard at: `https://server_ip`

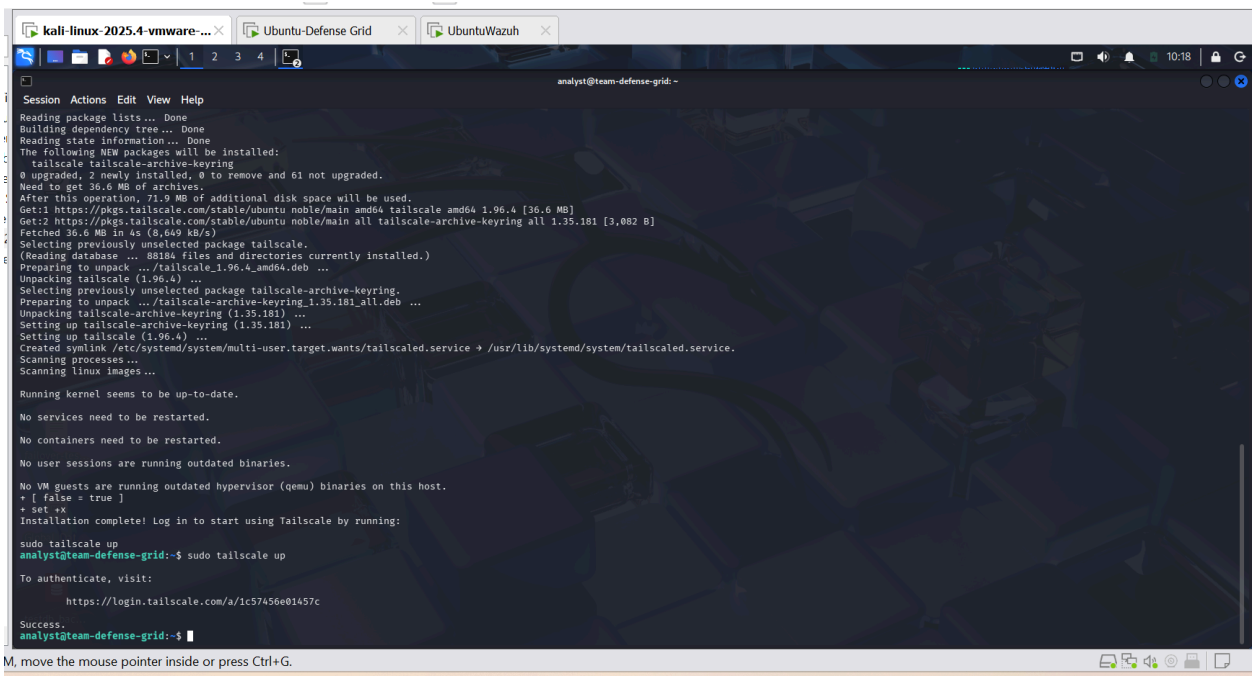


Tail scale Integration

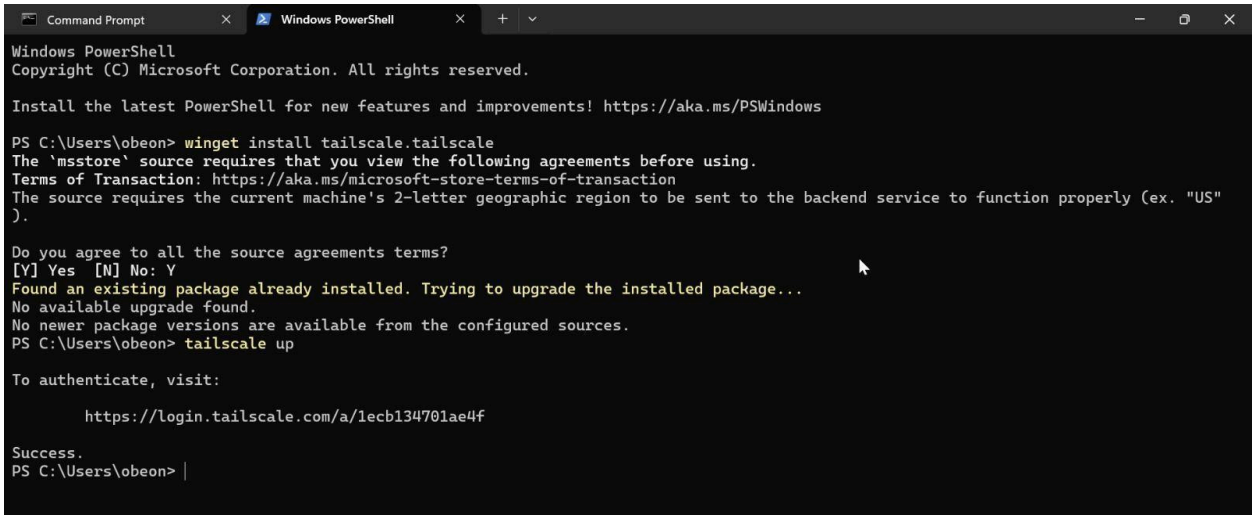
Each component receives a Tailscale IP for secure communication:

- Tailscale installation on Ubuntu wuzuh server

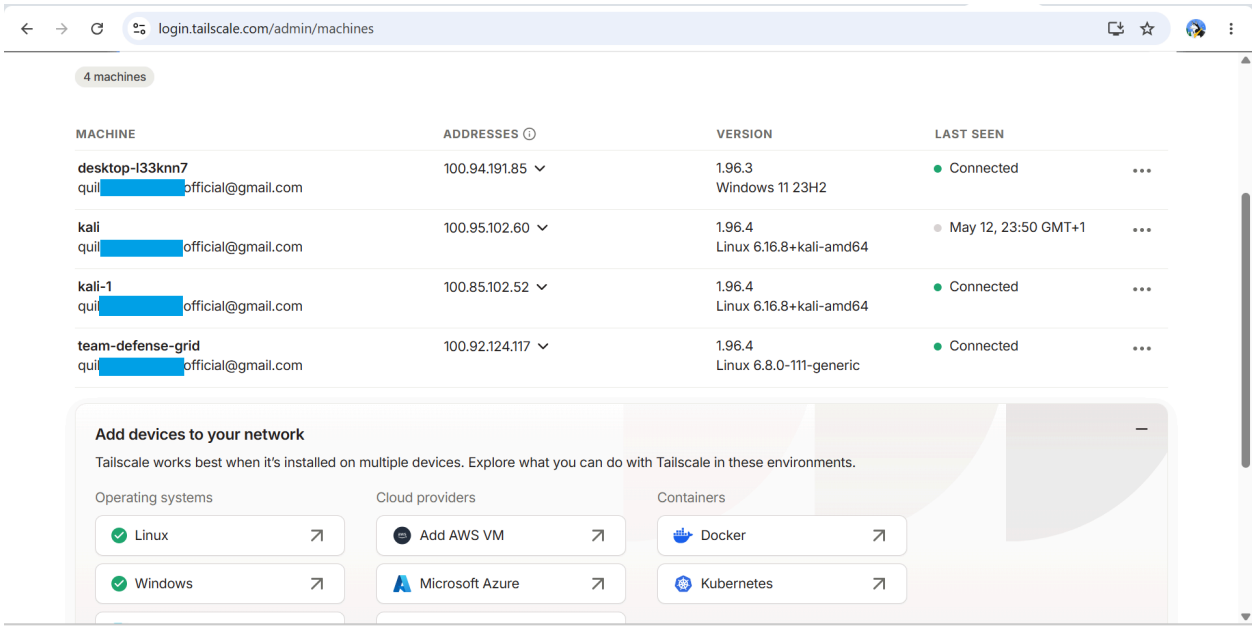


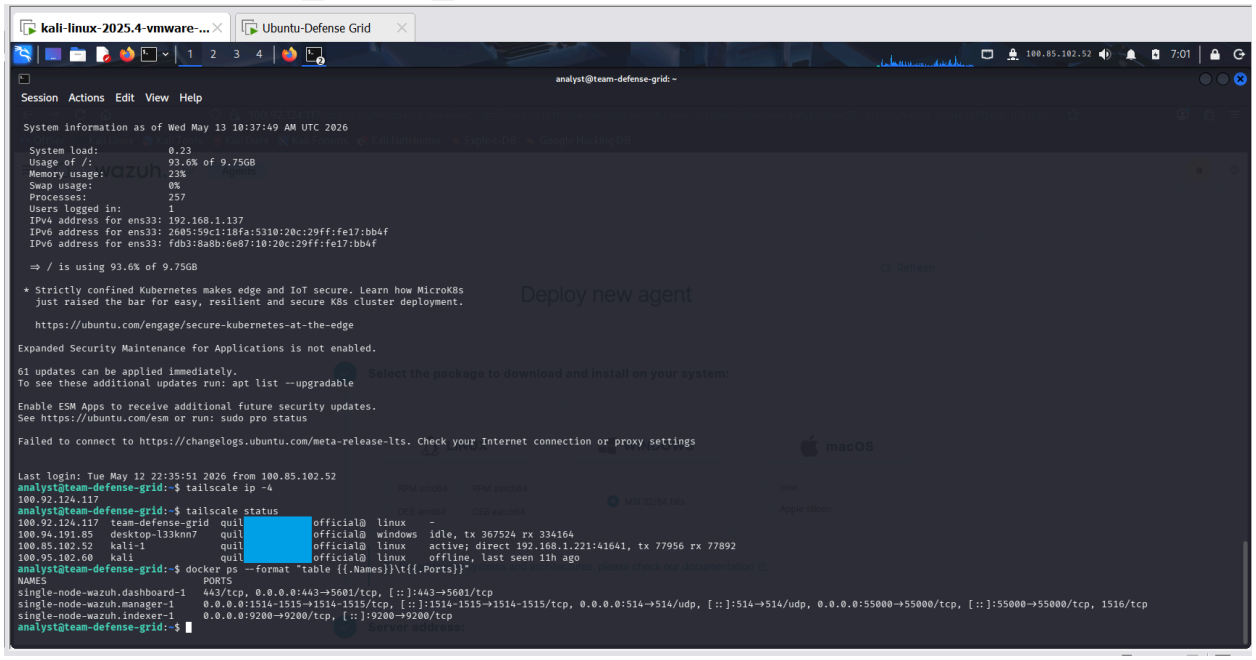


- Tailscale installation on window endpoint



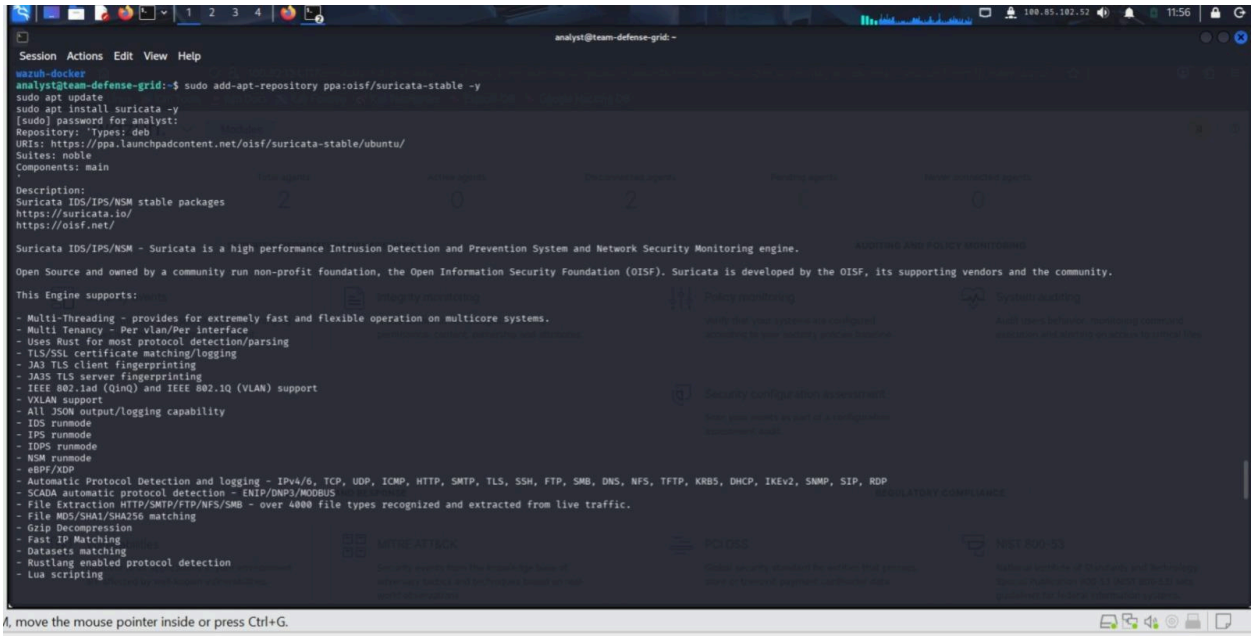
- *After installation of tail scale on the different end points*

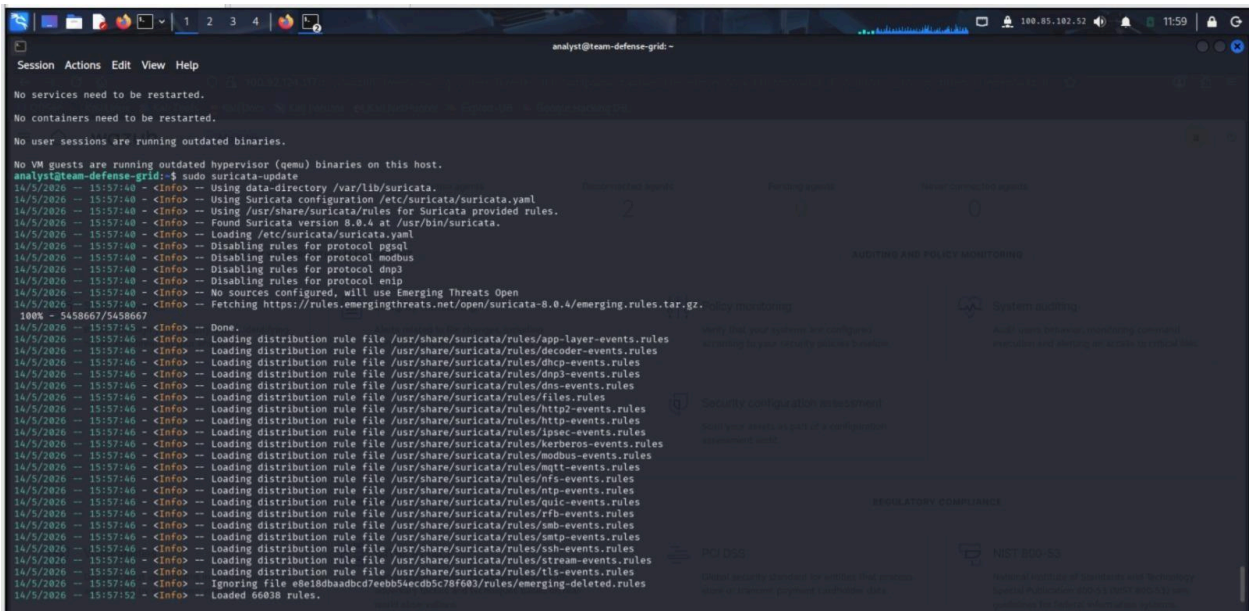




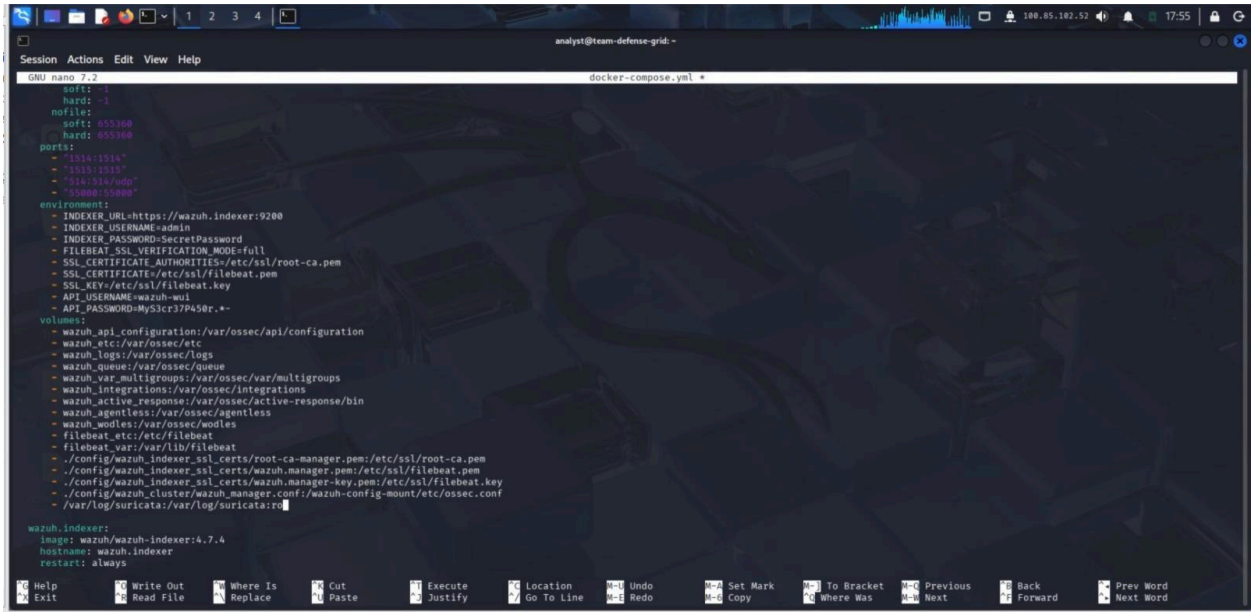
Installation of Suricata on the Ubuntu server

The different steps for the implementation of Suricata as a Network Intrusion Detection System (NIDS) on an Ubuntu Server system, alongside its log-forwarding integration into a Docker-based Wazuh SIEM deployment.

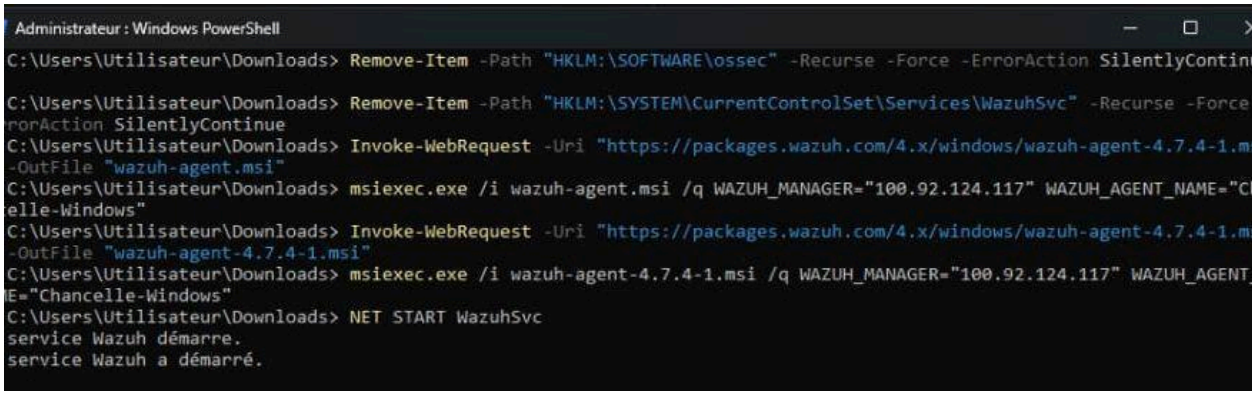




To enable the centralized Wazuh Manager engine to parse, analyze, and alert on Suricata telemetry, the host engine's logs must be mounted inside the target container framework.



Wazuh Agent Installation and Configuration



```
strateur : Windows PowerShell
ers\Utilisateur\Downloads> NET START WazuhSvc
ce Wazuh démarre.
ce Wazuh a démarré.

ers\Utilisateur\Downloads> Get-Service WazuhSvc

 Name               DisplayName
 ----               -----------
 WazuhSvc           Wazuh
```

```
Administrateur : Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> NET START WazuhSvc
Le service demandé a déjà été démarré.

Vous obtiendrez une aide supplémentaire en entrant NET HELPMSG 2182.

PS C:\WINDOWS\system32> Get-Content "C:\Program Files (x86)\ossec-agent\ossec.log" -Tail 30
2026/05/14 12:41:07 wazuh-agent: INFO: Trying to connect to server ([100.92.124.117]:1514/tcp).
2026/05/14 12:41:28 wazuh-agent: ERROR: (1216): Unable to connect to '[100.92.124.117]:1514/tcp': 'Une tentative de connexion a échoué car le parti connecté n'a pas répondu convenablement au-delà d'une certaine durée ou une connexion établie a échoué car l'hôte de connexion n'a pas répondu.'.
2026/05/14 12:41:38 wazuh-agent: INFO: Trying to connect to server ([100.92.124.117]:1514/tcp).
2026/05/14 12:41:41 wazuh-agent: INFO: (4102): Connected to the server ([100.92.124.117]:1514/tcp).
2026/05/14 12:41:41 wazuh-agent: INFO: Server responded. Releasing lock.
2026/05/14 12:41:41 wazuh-agent: INFO: Agent is now online. Process unlocked, continuing...
2026/05/14 12:41:41 wazuh-agent: INFO: Agent is now online. Process unlocked, continuing...
2026/05/14 12:41:41 sca: INFO: Security Configuration Assessment scan finished. Duration: 2906 seconds.
2026/05/14 12:41:41 wazuh-agent: INFO: Agent is now online. Process unlocked, continuing...
2026/05/14 12:41:43 wazuh-agent: INFO: Agent is now online. Process unlocked, continuing...
2026/05/14 12:41:43 wazuh-agent: INFO: Agent is now online. Process unlocked, continuing...
2026/05/14 12:41:44 wazuh-agent: INFO: Agent is now online. Process unlocked, continuing...
2026/05/14 12:41:45 wazuh-agent: INFO: Agent is now online. Process unlocked, continuing...
```

Creation of Custom Detection Rules

Custom Wazuh rules were created to detect FireOps-specific threats beyond built-in rules.

Rule 1: SSH Brut-Force

To evaluate the effectiveness of the monitoring environment, a simulated brute-force attack was conducted against the SSH service.

```
Session  Actions  Edit  View  Help
 GNU nano 7.2                    /var/lib/docker/volumes/single-node_wazuh_etc/_data/rules/fireops_rules.xml
<!--
    FireOps Systems - Custom Detection Rules
    Category: Authentication & Access Control
-->

<group name="fireops,authentication,">

  <!-- Rule 1: SSH Brute Force -->
  <rule id="100001" level="10" frequency="5" timeframe="60">
    <if_matched_sid>5503</if_matched_sid>
    <description>FireOps: Brute Force - Multiple failed SSH login attempts detected</description>
    <mitre>
      <id>T1110</id>
    </mitre>
    <group>authentication_failed,brute_force,</group>
  </rule>
```

Rule 2: After-hours login

Detecting after-hours logins on Wazuh is highly actionable because Wazuh ships with built-in time-dependent rules.

Rule 3: Sudo usage

Wazuh uses privilege escalation rules to monitor and alert on sudo usage, privilege escalation, and root command execution. These rules detect unauthorized administration attempts and log successful executions.Built-in Alerting RulesWazuh tracks privilege changes through default syslog.

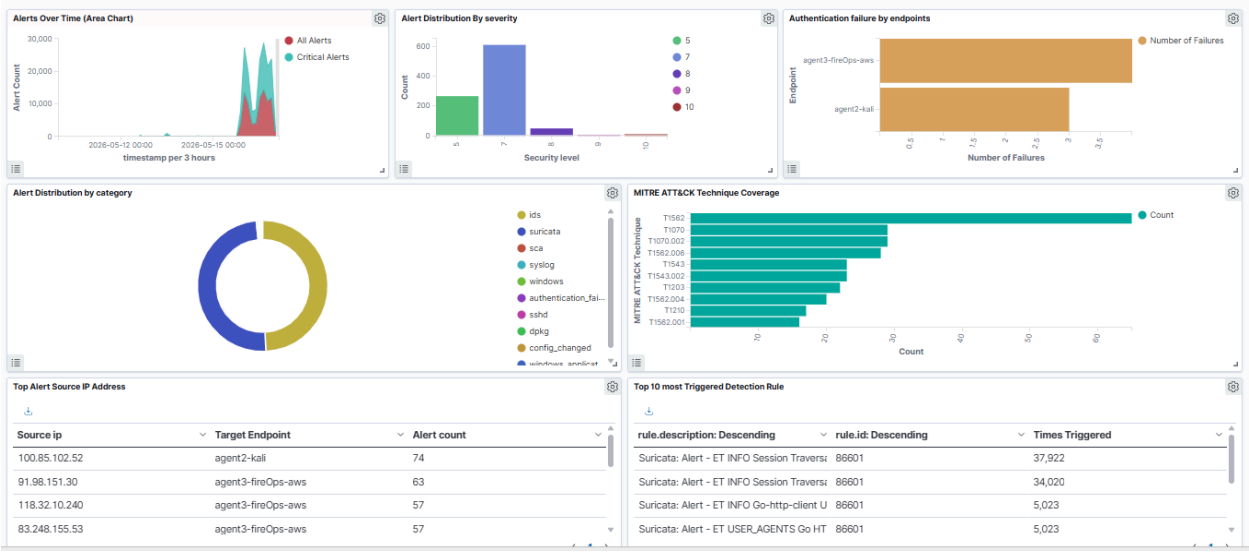
Security Event Monitoring and Centralized Log Collection

All security events from simulated attacks, cloud activity, and endpoint logs were aggregated in Wazuh Dashboard for centralized visibility.

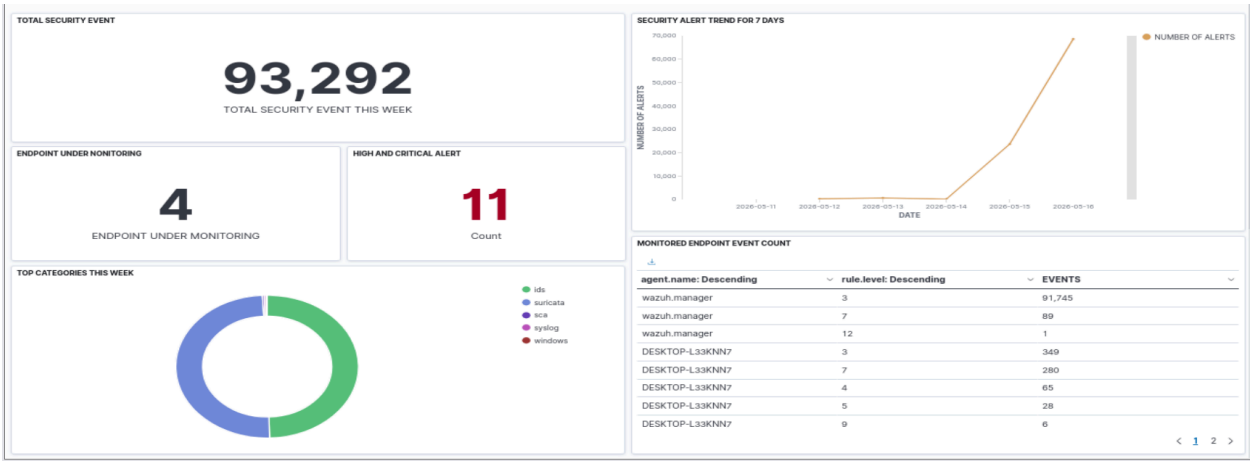
Centralized Logging Benefits

- Single pane of glass for all security events
- Cross-system correlation of related alerts
- Rapid identification of attack patterns
- Historical data for forensics

Dashboard 1- Technical view



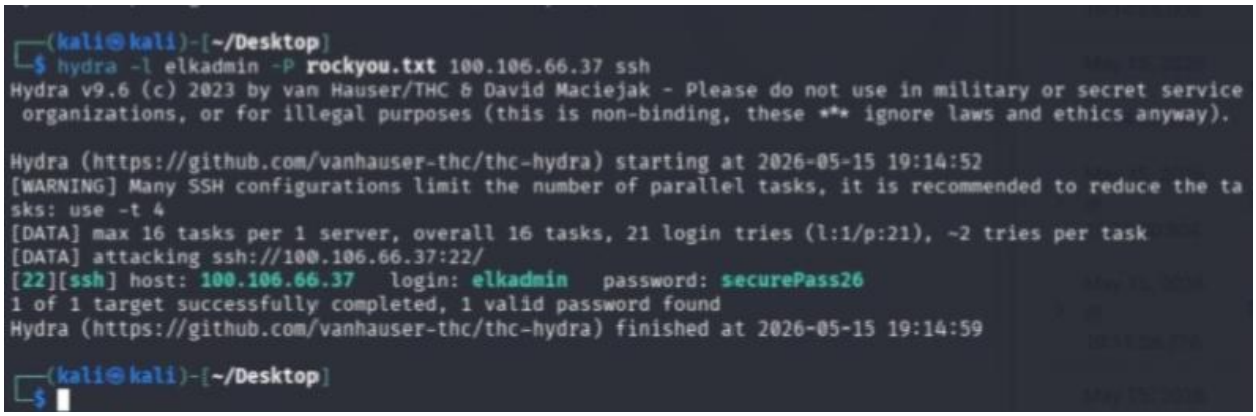
Dashboard 2- Executive view



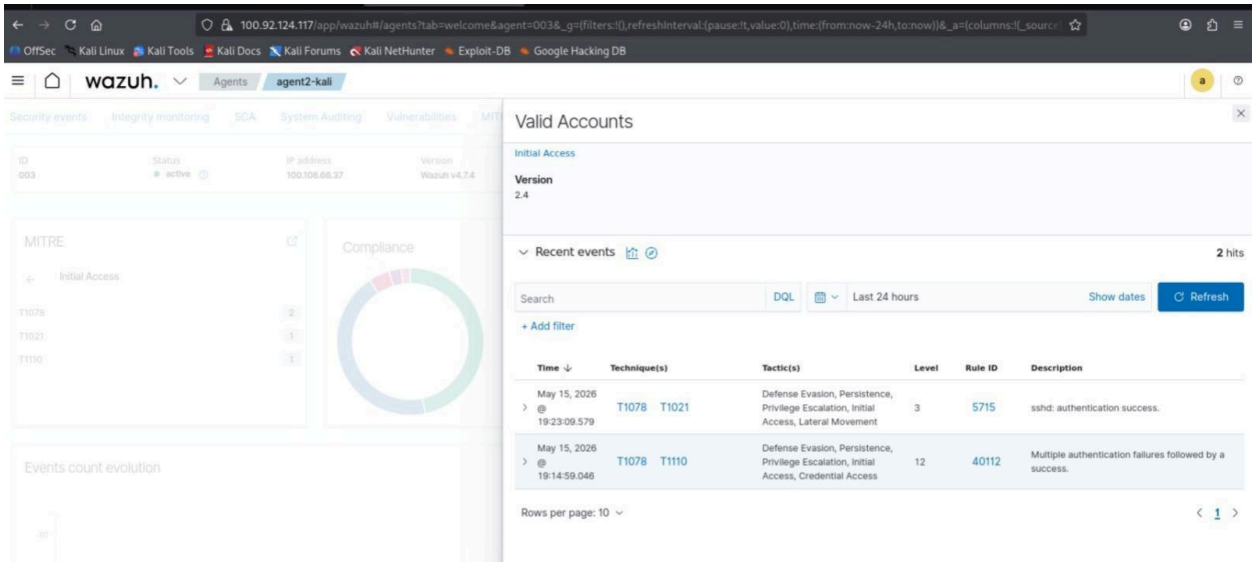
Threat simulation

- ### Brute-Force Attack

An automated dictionary attack was launched from the attacker platform targeting the SSH daemon on the target server using the `rockyou.txt` wordlist.

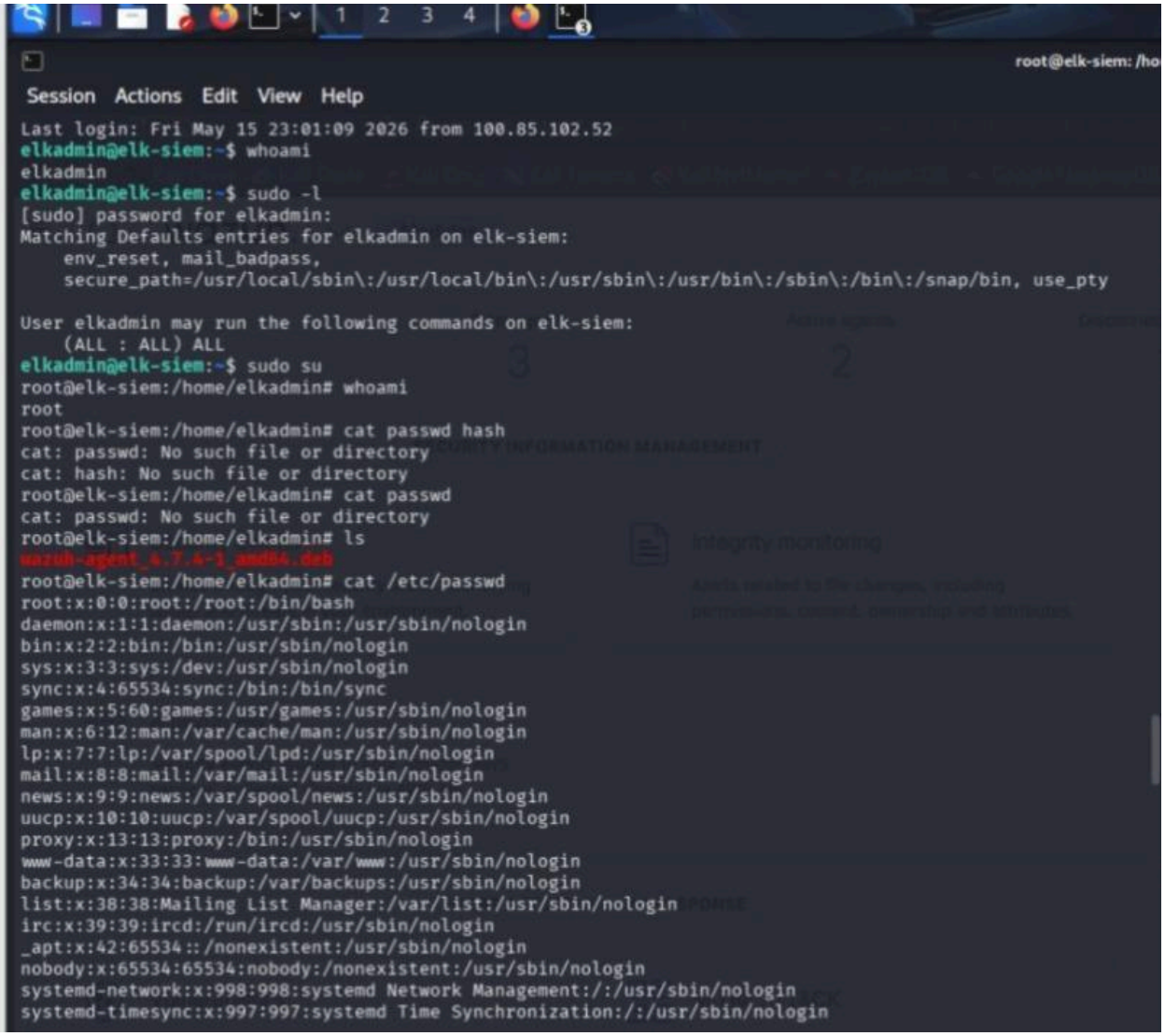


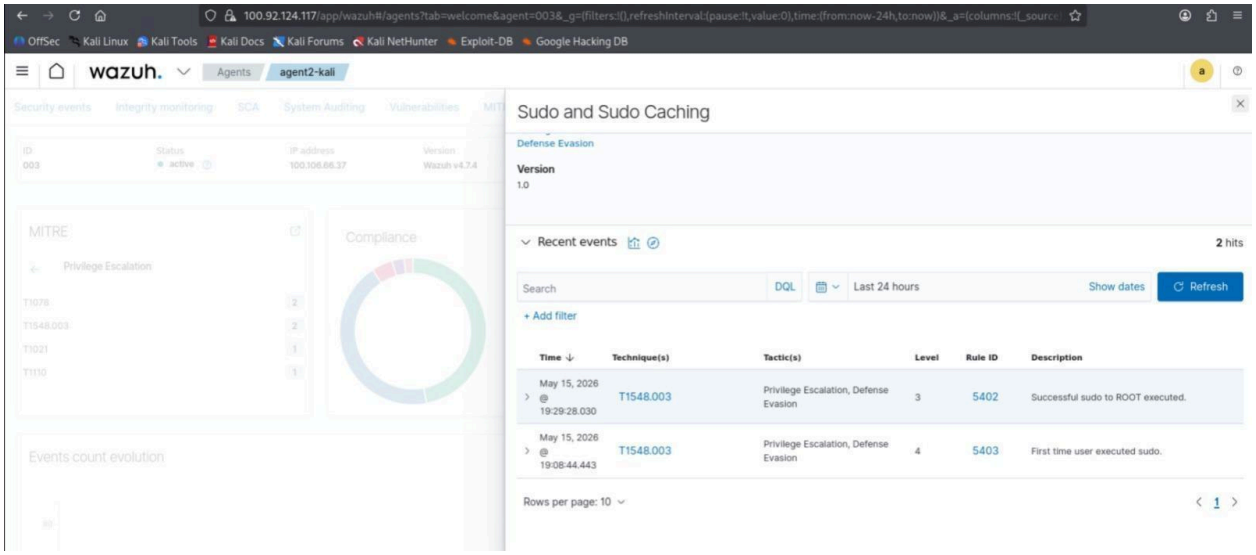
Attack Execution
hydra -l admin -P /path/to/passwords.txt 100.xx.xx.xx ssh
This generated multiple failed SSH login attempts (10+ in 50 seconds)



- **Privilege escalation**

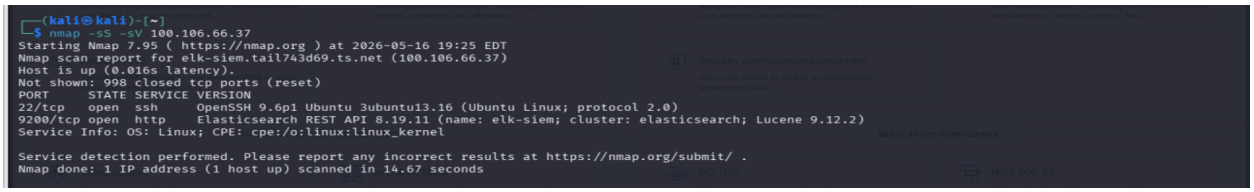
Once authenticated, the attacker checked local constraints, abused misconfigured sudo binaries, and gathered administrative intelligence

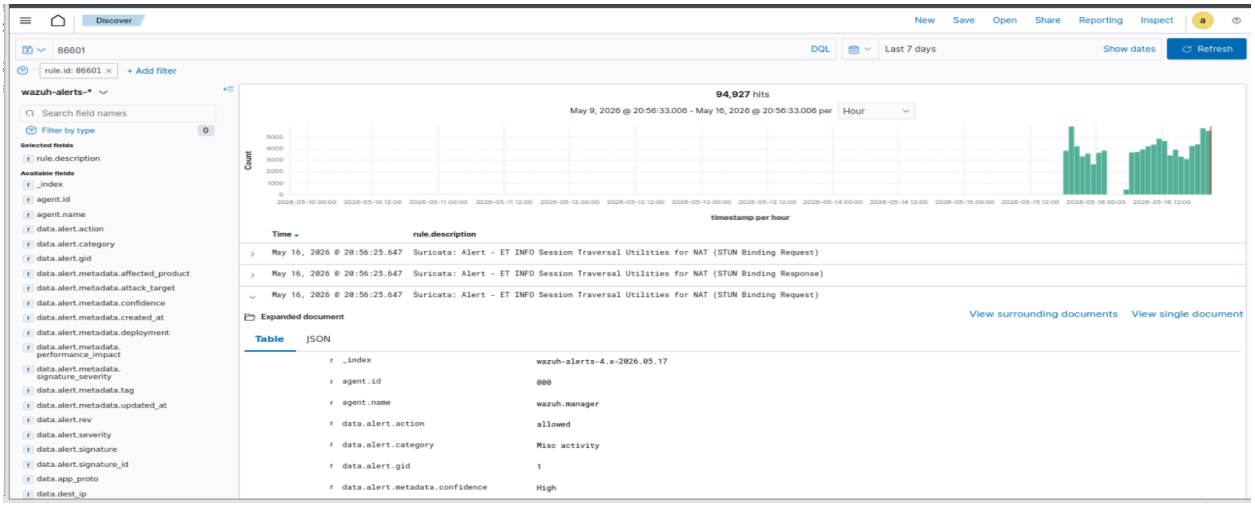

root@elk-siem: /ho
Session Actions Edit View Help
Last login: Fri May 15 23:01:09 2026 from 100.85.102.52
elkadmin@elk-siem:~$ whoami
elkadmin
elkadmin@elk-siem:~$ sudo -l
[sudo] password for elkadmin:
Matching Defaults entries for elkadmin on elk-siem:
env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty
User elkadmin may run the following commands on elk-siem:
(ALL : ALL) ALL
elkadmin@elk-siem:~$ sudo su
root@elk-siem:/home/elkadmin# whoami
root
root@elk-siem:/home/elkadmin# cat passwd hash
cat: passwd: No such file or directory
cat: hash: No such file or directory
root@elk-siem:/home/elkadmin# cat passwd
cat: passwd: No such file or directory
root@elk-siem:/home/elkadmin# ls
wazuh-agent_4.7.4-1_amd64.deb
root@elk-siem:/home/elkadmin# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin


wazuh.
Agents
agent2-kali
Sudo and Sudo Caching
Defense Evasion
Version
1.0
Recent events
2 hits
Search
DQL
Last 24 hours
Show dates
Refresh
+ Add filter
Time
Technique(s)
Tactic(s)
Level
Rule ID
Description
May 15, 2026 @ 19:29:28.030
T1548.003
Privilege Escalation, Defense Evasion
3
5402
Successful sudo to ROOT executed.
May 15, 2026 @ 19:08:44.443
T1548.003
Privilege Escalation, Defense Evasion
4
5403
First time user executed sudo.
Rows per page: 10
MITRE
Compliance
Events count evolution

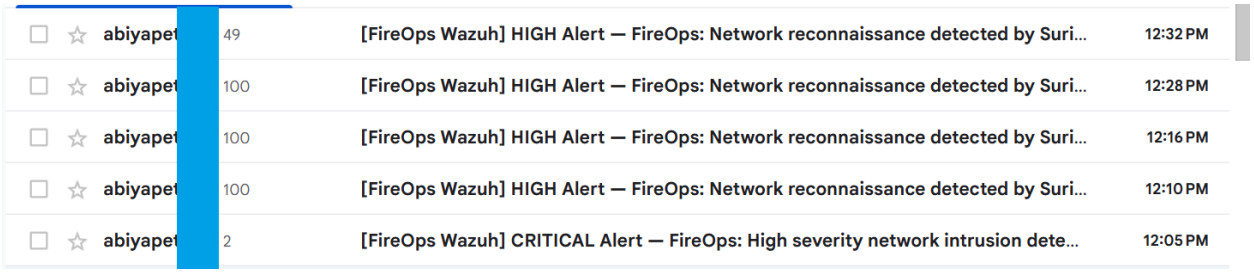
- **Reconnaissance via Port Scanning**

Before exploitation, the attacker conducted active reconnaissance, performed a TCP SYN stealth scan with service version detection, and identified open ports and running services using nmap -sS -sV 100.106.66.37.





- **EMAIL ALERT RECEIVED**



Incident Response Plan — (MITRE ATT&CK Framework | NIST Incidence Response Lifecycle)

1. OPERATIONAL OBJECTIVE

This handbook defines standardized procedures for responding to security incidents while ensuring: Continuous system availability during incidents.

Controlled and minimal disruption containment Fast detection-to-response lifecycle Consistent decision-making across SOC analysts Preservation of critical business operations

2. RESPONSE MODEL (6 PHASES)

All incidents follow this lifecycle:

- Preparation
- Identification
- Containment
- Eradication
- Recovery
- Lessons Learned

3. INCIDENT RESPONSE PLAYBOOKS

3.1 SSH BRUTE FORCE ATTACK

Preparation

- Enable SSH authentication logging
- Configure Wazuh rules for failed login detection
- Deploy Fail2Ban for automatic response
- Restrict SSH exposure to trusted networks
- Maintain backup administrative access path

Identification

- Detect repeated failed SSH login attempts
- Identify attacking IP addresses
- Verify whether login success occurred
- Assess SSH service stability and load impact

Containment options

Option 1: Fail2Ban Automatic Protection (Preferred Default)

➔ Use when:

- Brute-force attack is automated
- Multiple failed login attempts detected
- SSH service must remain available

Response Steps

1. Verify Fail2Ban service is running:

```
sudo systemctl status Fail2ban
```

2. Check banned IP addresses:

```
sudo fail2ban-client status sshd
```

3. Confirm attacker IP appears in banned list.
4. Monitor whether failed login attempts decrease.
5. Continue monitoring Wazuh alerts for new attacker sources.

➔ Why this option

- Automatically protects SSH service
- Preserves system availability
- Requires minimal operational disruption

Option 2: SSH Rate Limiting

➔ Use when

- Attack originates from multiple IP addresses
- Full blocking may affect legitimate users
- Service availability must remain uninterrupted

Response Steps

1. Apply connection rate limit:

```
sudo iptables -A INPUT -p tcp --dport 22 -m
conntrack --ctstate NEW -m recent --set
```

2. Limit repeated connections:

```
sudo iptables -A INPUT -p tcp --dport 22 -m
conntrack --ctstate NEW -m recent --update
 --seconds 60 --hitcount 10 -j DROP
```

3. Verify rule is active:

```
sudo iptables -L
```

4. Monitor SSH performance and failed login activity.

➔ Why this option
- Slows attack without fully denying access
- Reduces risk of service interruption

Option 3: Switch to SSH Key Authentication Only

➔ Use when
- Password attacks continue despite mitigation
- Credential guessing is confirmed
- SSH keys are already deployed for administrators and authorized employees
- Secure key distribution process has been completed successfully

Response Steps

1. Verify all authorized administrators and employees have valid SSH key pairs configured.
2. Confirm public keys are already distributed to target servers:

```
cat ~/.ssh/authorized_keys
```

3. If keys are not yet distributed, securely generate and distribute SSH keys before disabling passwords:

```
ssh-keygen -t rsa -b 4096
```

4. Copy public key to target server:

```
ssh-copy-id user@server-ip
```

5. Test SSH login using key authentication:

```
ssh user@server-ip
```

6. Edit SSH configuration:

```
sudo nano /etc/ssh/sshd_config
```

7. Disable password authentication:

```
PasswordAuthentication no
```

8. Restart SSH service:

```
sudo systemctl restart ssh
```

9. Verify users can successfully authenticate using SSH keys only

➔ Why this option

- Removes password attack vector entirely
- Maintains secure administrative access
- Prevents accidental lockout of legitimate employees
- Preserves operational availability during containment

Operational Warning

Do not disable password authentication before validating SSH key distribution across all authorized users and systems. Failure to verify access paths may result in administrative lockout and business disruption.

Option 4: Firewall IP Blocking (Last Resort)

➔ Use when

- Single confirmed malicious attacker
- Attack remains aggressive after other controls

Response Steps

1. Block attacker IP using UFW:

```
sudo ufw deny from <attacker-ip>
```

2. Verify firewall rule:

```
sudo ufw status
```

3. Monitor whether attack traffic stops.

➔ Why this option

- Immediate suppression of confirmed attacker
- Useful for targeted attacks

Eradication steps

- Disable password authentication permanently
- Rotate exposed credentials
- Remove unauthorized accounts
- Harden SSH configuration

Steps:

- Disable root login:

```
PermitRootLogin no
```

- Restart SSH service:

```
sudo systemctl restart ssh
```

- Review users:

```
cat /etc/passwd
```

- Remove suspicious account:

```
sudo userdel <username>
```

Recovery

- Restore secure SSH access
- Validate no unauthorized sessions exist
- Continue heightened monitoring for 48 hours

Steps:

- Check active SSH sessions:

```
who
```

- Review authentication logs:

```
sudo tail -f /var/log/auth.log
```

Confirm Wazuh alerts normalize.

Lessons Learned

- Fail2Ban provides safer automated protection than manual blocking
- Rate limiting preserves availability during attacks
- SSH key authentication significantly reduces exposure

3.2 PRIVILEGE ESCALATION ATTACK

Preparation

- Enable sudo logging
- Forward logs to Wazuh
- Restrict unnecessary sudo access
- Maintain emergency administrator account

Identification

- Detect sudo abuse or root shell activity
- Identify privilege enumeration attempts
- Verify affected users and commands executed

Containment options

Option 1: Terminate Active Elevated Session

➔ Use when
- Unauthorized root shell is active
- Immediate privileged activity detected

Response Steps

- Identify active sessions:

```
who
```

- Identify suspicious processes:

```
ps aux | grep sudo
```

- Terminate malicious process:

```
sudo kill -9 <PID>
```

- Monitor for session re-establishment.

➔ Why this option

- Stops immediate administrative abuse
- Prevents further system modification

Option 2: Revoke Sudo Privileges (Preferred Default)

➔ Use when

- Suspicious activity exists without confirmed compromise
- Misconfigured privileges suspected

Response Steps

- Open sudoers configuration safely:

```
sudo visudo
```

- Remove unnecessary sudo permissions.
- Verify affected user groups:

```
groups <username>
```

- Remove from sudo group if required:

```
sudo deluser <username> sudo
```

➔ Why this option

- Limits escalation without disabling account
- Preserves operational continuity

Option 3: Suspend User Account

➔ Use when
- Confirmed malicious behavior
- High risk of lateral movement

Response Steps

- Lock account:

```
sudo passwd -l <username>
```

- Verify account lock:

```
sudo passwd -l <username>
```

- Monitor for additional suspicious accounts.

➔ Why this option
- Fully isolates compromised account
- Prevents continued access

Eradication steps

- Review sudoers file:

```
sudo visudo
```

- Audit privileged users:

```
getent group sudo
```

- Reset passwords if needed:

```
sudo passwd <username>
```

- Review system modifications:

```
sudo ausearch -m USER_CMD
```

Recovery steps

- Restore approved privileges carefully.
- Validate system integrity:

```
sudo debsums
```

- Continue elevated monitoring in the Wazuh dashboard.

Lessons Learned

- Excessive sudo permissions increase compromise impact
- Session termination is safer than full shutdown
- Least privilege reduces operational risk

3.3 NETWORK RECONNAISSANCE (PORT SCANNING)

Preparation

- Deploy Suricata monitoring on all interfaces
- Enable SYN scan detection rules
- Maintain approved port exposure baseline
- Centralize logs in Wazuh dashboard

Identification

- Detect scanning activity
- Identify source and target systems
- Determine scan intensity and scope

Containment options

Option 1: Traffic Rate Limiting (Preferred Default)

➔ Use when
- Scan source is uncertain
- Maintaining service availability is critical

Response steps

- Apply rate limiting:

```
sudo iptables -A INPUT -p tcp --syn -m limit
--limit 5/s -j ACCEPT
```

- Verify rules:

```
sudo iptables -L
```

- Monitor service stability and alert volume.

➔ Why this option

Slows scans without interrupting legitimate traffic.

Option 2: Firewall Blocking

➔ Use when
- External malicious scanner confirmed
- Aggressive scan detected

Response steps

- Block source IP:

```
sudo ufw deny from <scanner-ip>
```

- Verify rule:

```
sudo ufw status
```

- Confirm scan traffic stops.

➔ Why this option

Immediate containment for confirmed attacker.

Option 3: Service Restriction / Hardening

➔ Use when
- Scan exposes unnecessary open services
- Exposure reduction is required

Response steps

- Identify open ports:
-

```
sudo ss -tulnp
```

- Disable unnecessary service:

```
sudo systemctl stop <service>
```

- Disable service permanently:

```
sudo systemctl disable <service>
```

- Close unnecessary ports:

```
sudo ufw deny <port>
```

➔ Why this option
- Reduces long-term attack surface
- Prevents future exploitation

Recovery steps

- Verify required services remain accessible:

```
sudo systemctl status <service>
```

- Confirm the monitoring pipeline is operational.
- Continue monitoring for follow-up scans.

Lessons Learned

- Rate limiting preserves availability better than aggressive blocking
- Reducing exposed services improves long-term resilience
- Monitoring all interfaces is essential for visibility

4. SOC OPERATING PRINCIPLES

- Availability is preserved unless compromise is confirmed
- Automated controls are preferred over manual blocking
- Blocking is a controlled escalation, not a default action
- Detection must always precede containment
- Security controls must not disrupt critical services unnecessarily
- Response actions must scale with attack confidence level

Conclusion

The FireOps monitoring environment successfully detected all three simulated attack scenarios across endpoint, network, and cloud layers in near real time. The combination of Wazuh for endpoint and authentication monitoring, Suricata for network intrusion detection, and AWS CloudTrail for cloud-layer visibility delivered the centralized threat detection capability the project required. The NIST lifecycle and MITRE ATT&CK mappings applied throughout confirm that our detection and response workflow meets enterprise-grade standards, and the lessons identified across each scenario provide a clear roadmap for strengthening FireOps's security posture going forward.