NED University of Engineering & Technology

Vulnerability Assessment & Reverse Engineering

PROJECT REPORT:

Threat Detection Using Open Source SIEM

Name	***Roll Number***
1. *Abdullah Khalid* ***(Group leader)***	*CR-22027*
2. *Sheheryar Amir*	*CR-22008*
3. *Ayesha Noor*	*CR-22004*
4. *Maryam Khan*	*CR-22021*

Introduction:

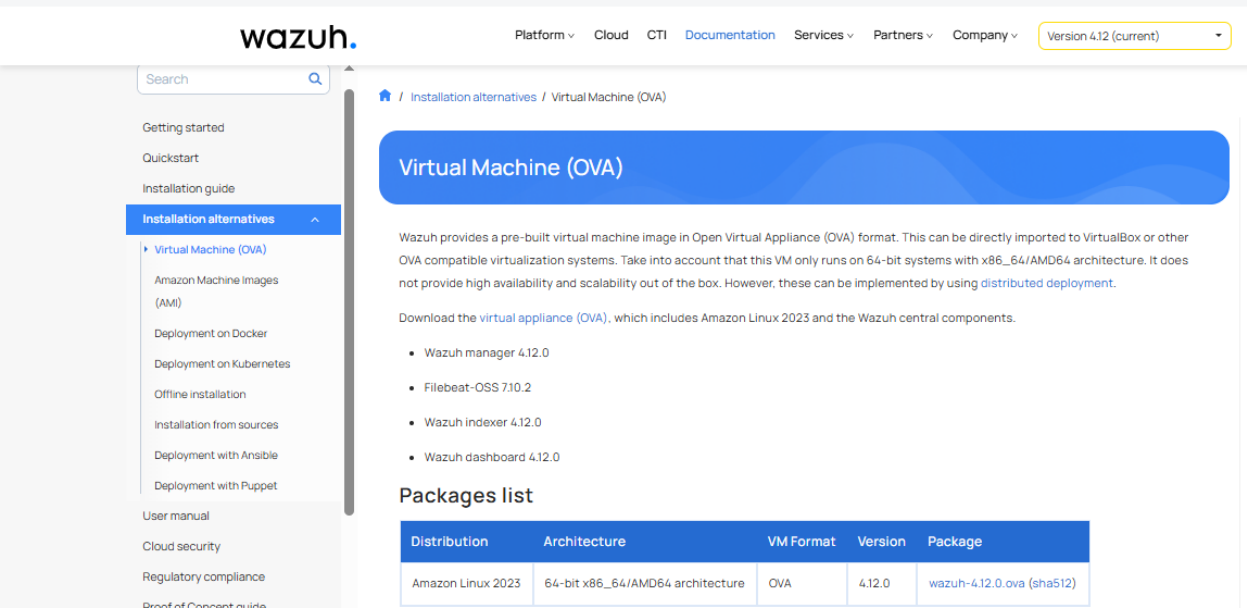
Wazuh is an open-source security monitoring platform that offers intrusion detection, log analysis, vulnerability detection, and compliance monitoring. In this report, we document the steps for downloading, installing, and customizing Wazuh on a VirtualBox virtual machine.

Wazuh Installation:

To simplify the installation process, I chose to use the official Wazuh OVA (Open Virtual Appliance) file, which provides a pre-configured virtual machine with Wazuh already installed and ready to run.

Steps to Download:

1. I visited the official Wazuh website at: https://documentation.wazuh.com/current/deployment-options/virtual-machine/virtual-machine.html
2. Under the "Wazuh OVA" section, I selected the appropriate version (e.g., Wazuh 4.9) and clicked the download link.
3. The OVA file (e.g., wazuh-4.9.0.ova) was approximately 2.5 GB in size.
4. The file was downloaded and saved locally to my machine in the **Downloads** folder.

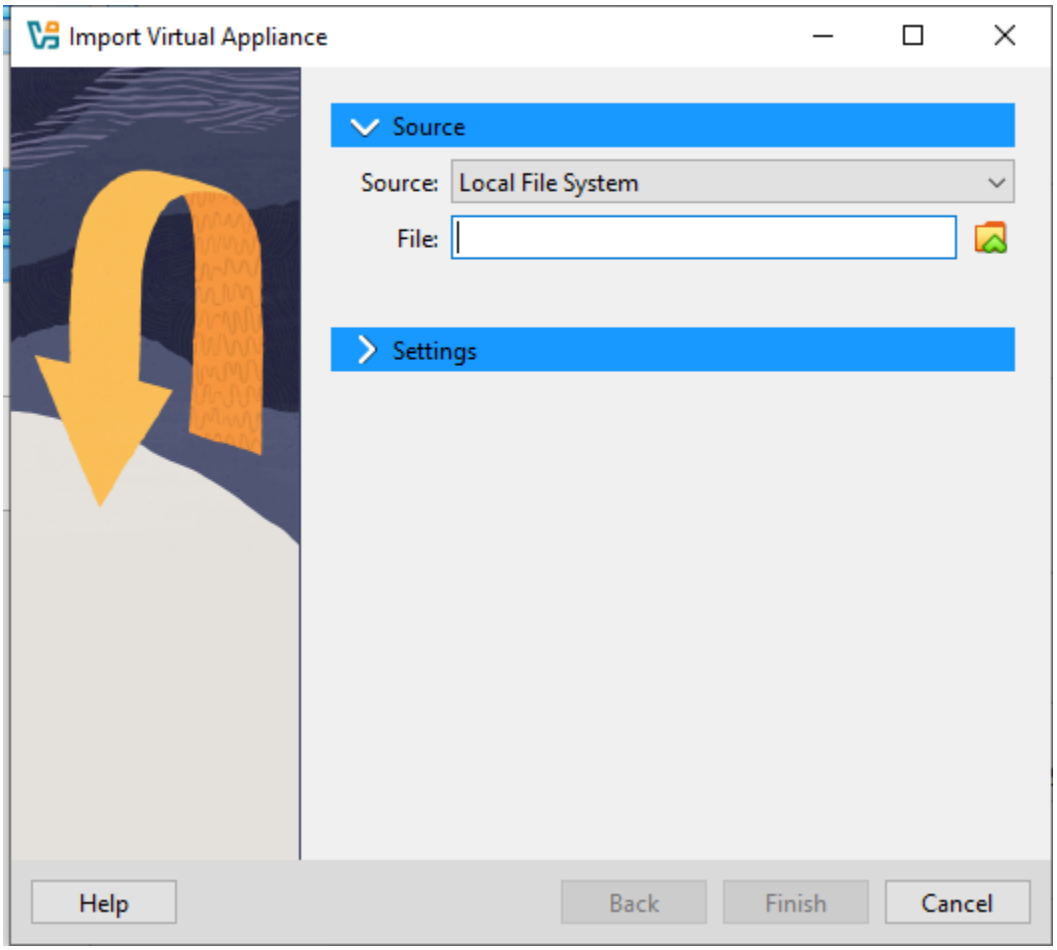


Setting Up Wazuh in Our VM:

After downloading the Wazuh 4.9 OVA file, I proceeded to import and configure it in Oracle VirtualBox. The process was straightforward and allowed me to quickly deploy a fully functional Wazuh environment.

Steps to Import and Set Up:

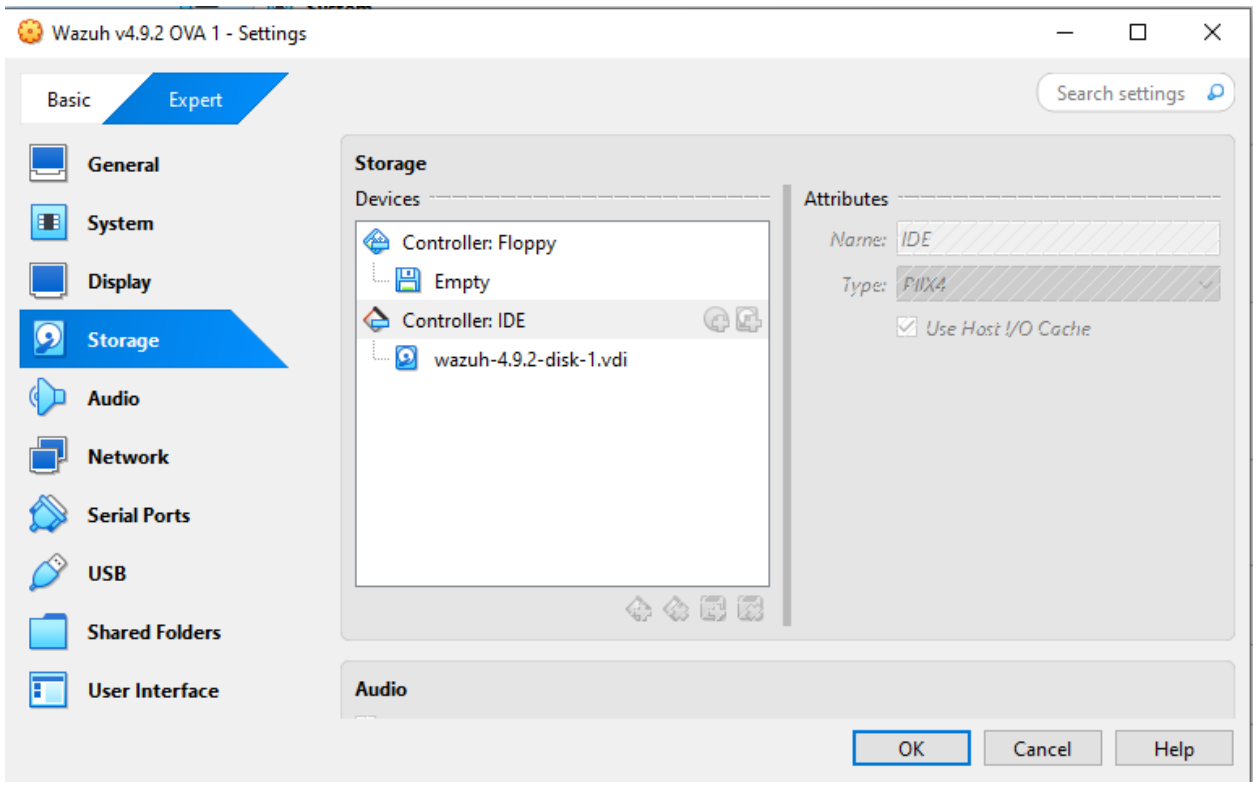
1. **Open VirtualBox** on the host system.
2. From the top menu, go to **File > Import Appliance**.
3. Click **Choose** and select the downloaded OVA file (wazuh-4.9.0.ova) from the local directory.
4. Click **Next**, review the virtual machine settings (CPU, RAM, disk space), and click **Import**.



5. After import, the Wazuh virtual machine appeared in the VirtualBox Manager.
6. I then selected the VM and clicked **Start** to boot it up.

Default VM Settings:

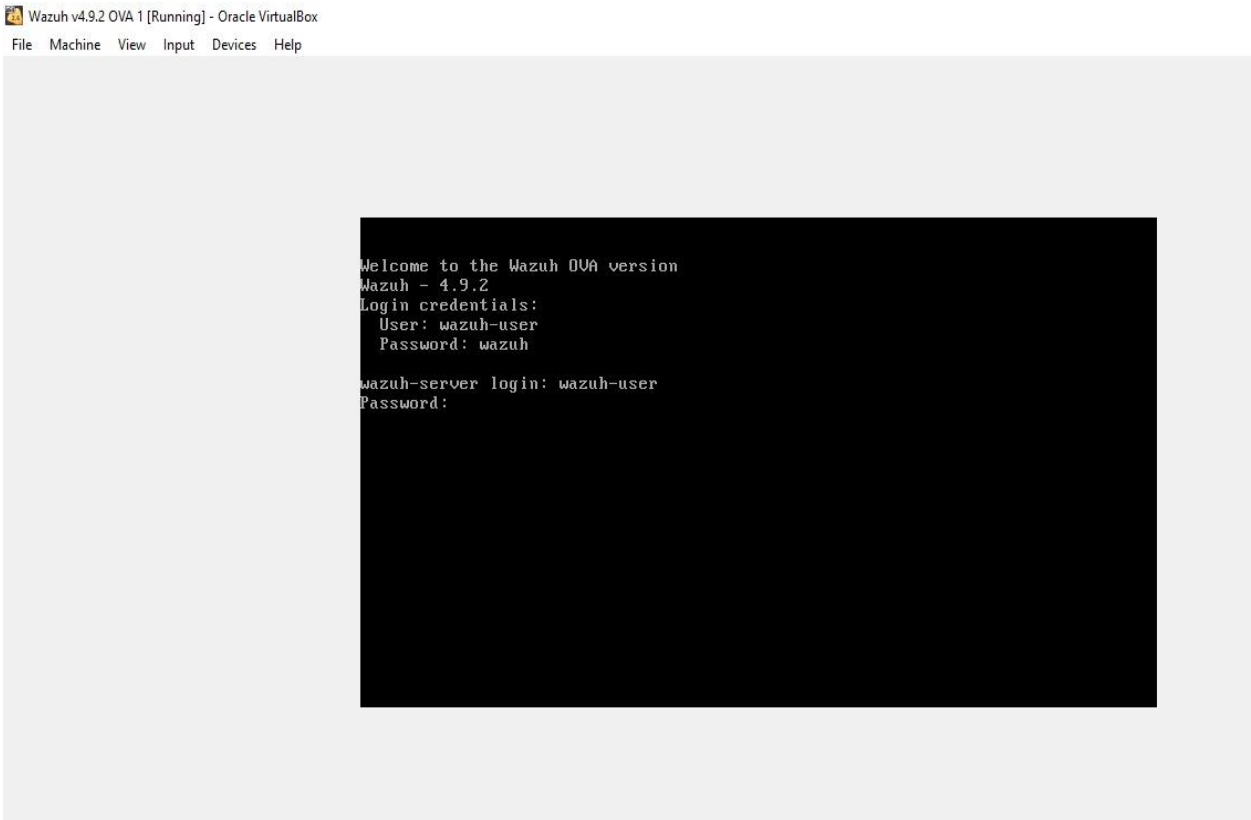
- **RAM**: 4 GB
- **CPUs**: 2
- **Disk Space**: 40 GB (dynamically allocated)
- **Network Adapter**: Bridged Adapter (or NAT, depending on your setup.



After successfully importing and starting the Wazuh 4.9 OVA virtual machine in VirtualBox, I needed the server's IP address to access Wazuh services from my host system.

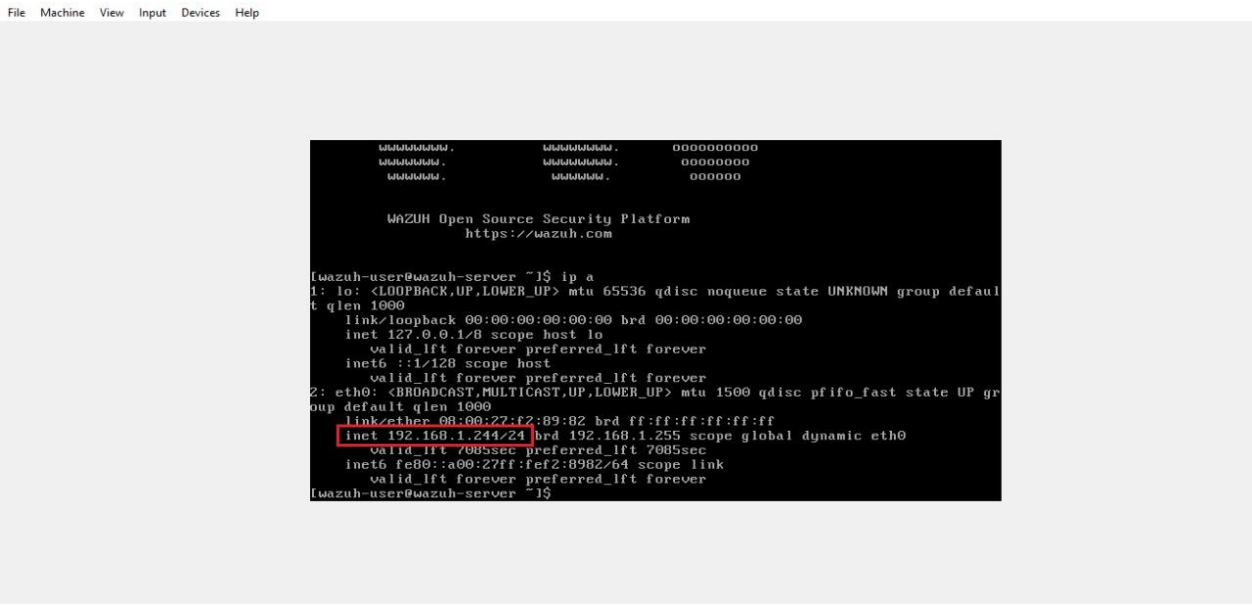
Steps Taken:

1. Once the VM finished booting, I logged into the terminal using the default credentials if prompted.



2. I ran the following command to check the assigned IP address:

   “” ip a ””



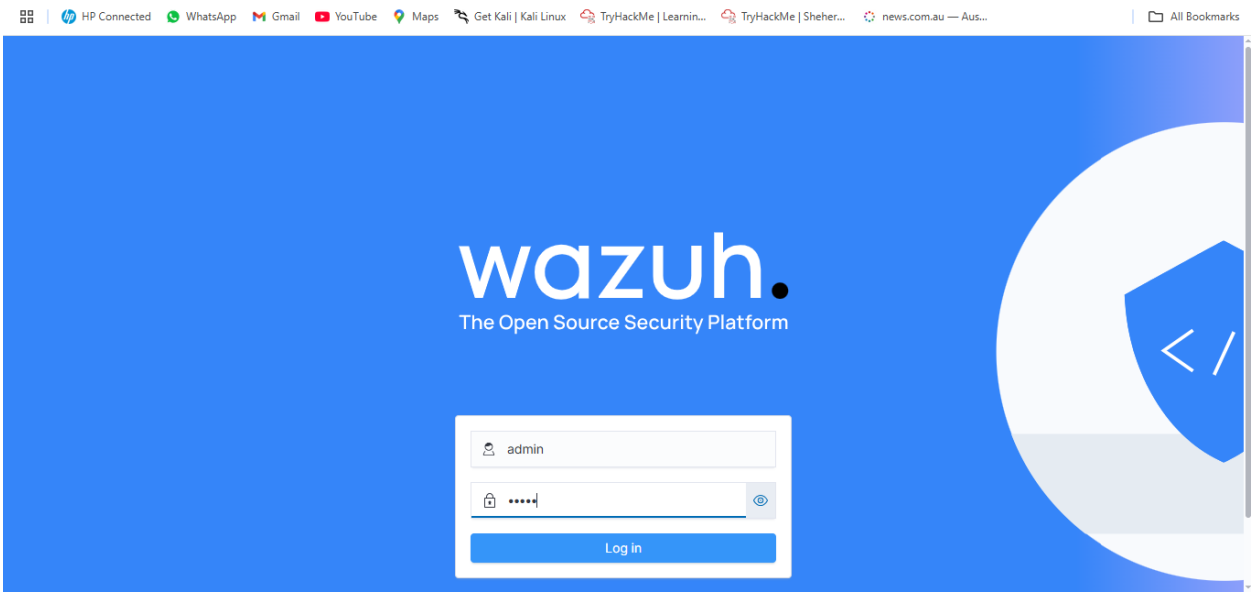
After retrieving the IP address (192.168.1.244) from the Wazuh server terminal, I accessed the Wazuh dashboard through a web browser on my host machine.

Steps Taken:

1. I opened a web browser and entered the following URL: https://192.168.1.244:5601
2. Since the Wazuh server uses a self-signed SSL certificate, I received a browser warning. I clicked “Advanced” and proceeded to the site.
3. The Wazuh login page appeared.

Login Credentials Used:

- **Username**: admin
- **Password**: admin

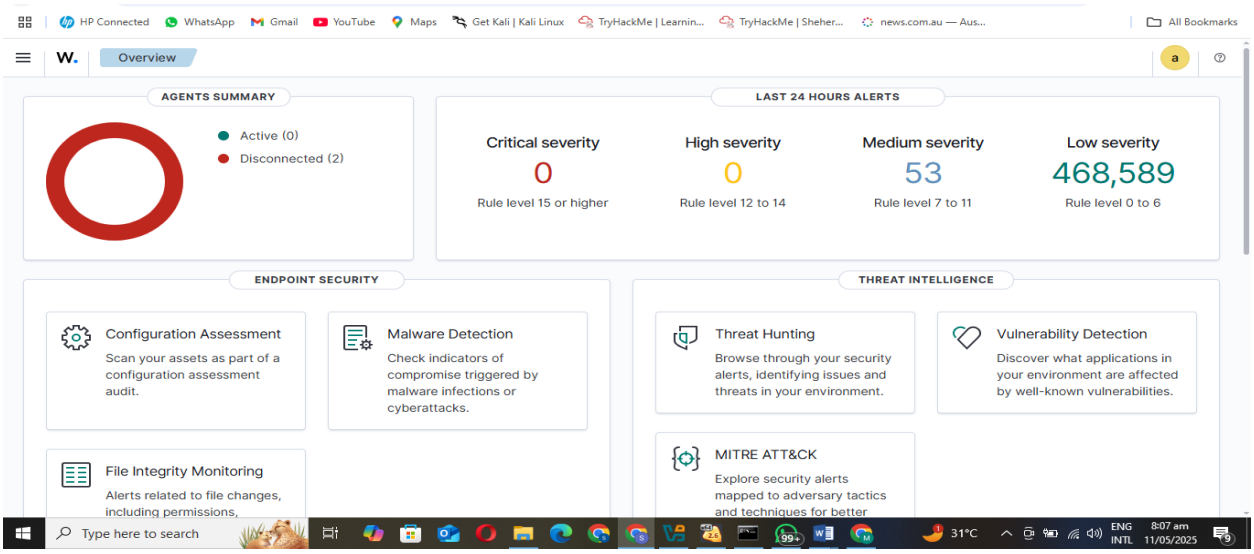


Successful Dashboard Access

After logging in with the provided credentials (wazuh-user / wazuh), I successfully accessed the Wazuh 4.9 dashboard. The web interface loaded correctly and displayed the default monitoring panels, including:

- **Overview of agent status**
- **Security alerts and logs**
- **System performance metrics**
- **Menu options for configuration and management**

This confirmed that the Wazuh server was running properly and the environment was ready for further customization or agent deployment.



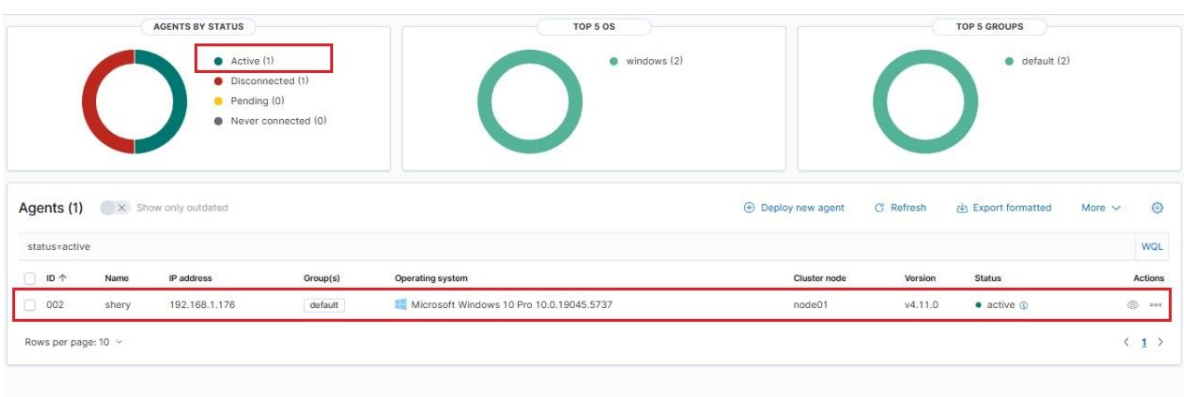
Agent Deployment Status

After logging into the Wazuh dashboard, I verified that the agent had been successfully deployed and was active.

Status Overview:

- The dashboard displayed the agent as **"Active"**
- Status indicators confirmed the agent was **connected to the Wazuh manager**
- The message shown was:
  "Agent is active "

This confirms that the agent is functioning correctly and will begin sending system and security event data to the Wazuh manager for analysis and visualization.



Next Steps:

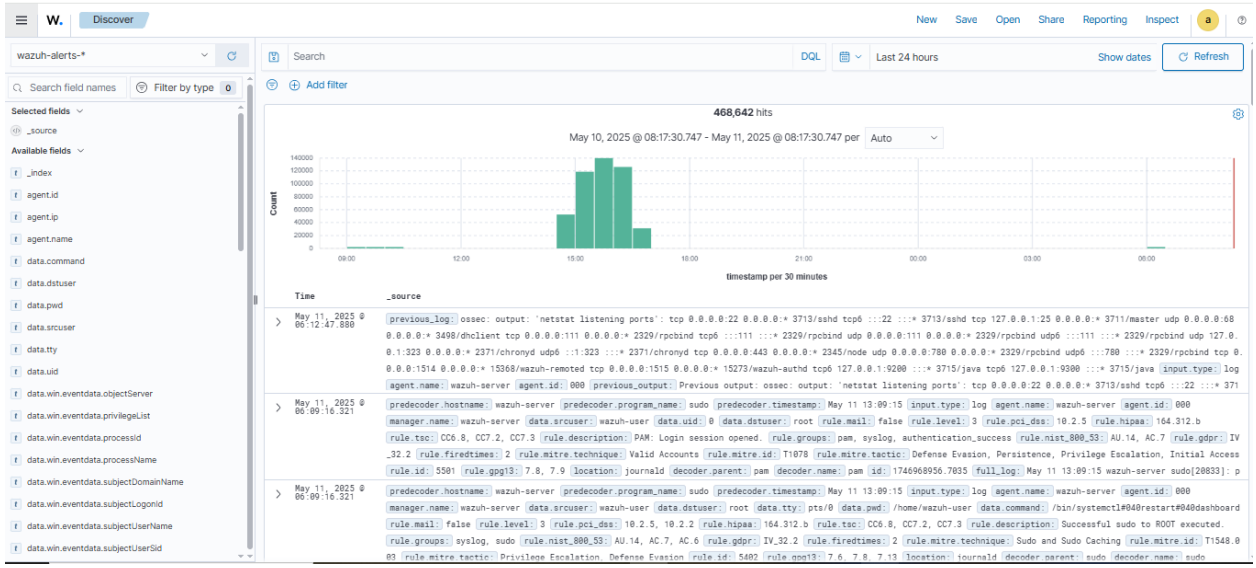
- Monitor real-time logs under the “Security Events” or “Discover” tab
- Customize rules, decoders, or alerts
- Integrate additional agents or configure log collection sources.

Viewing Logs in the Discover Tab

After confirming that the agent was active, I navigated to the **Discover** tab within the Wazuh dashboard to begin analyzing the log data being collected.

Steps Taken:

1. From the left-hand menu on the Wazuh dashboard, I clicked on **"Discover"**.
2. The page loaded successfully and displayed a continuous stream of log entries coming from the deployed agent.



Custom Rule: Brute Force / Authentication Failure Detection

To enhance the detection capabilities of my Wazuh setup, I created a custom rule designed to identify brute-force attacks or repeated authentication failures.

Objective:

Detect multiple failed login attempts within a short period, which is a common indicator of brute-force attacks.

Steps Taken:

1. I accessed the Wazuh manager's file system through the terminal.
2. Navigated to the custom rules directory:

```
[wazuh-user@wazuh-server ~]$ sudo nano /var/ossec/etc/rules/local_rules.xml
[wazuh-user@wazuh-server ~]$
```

- I added the following custom rule:

```
wazuh-user@wazuh-server:~
  GNU nano 2.9.8                    /var/ossec/etc/rules/local_rules.xml

<group name="shery">
  <rule id="100001" level="10" frequency="6" timeframe="60">
  <if_matched_sid>5760</if_matched_sid>
  <frequency>5</frequency>
  <timeframe>60</timeframe>
  <same_source_ip/>
  <same_srcip/>
  <description>SSH brute force observed on wazuh server</description>
  </rule>
</group>

File Name to Write: /var/ossec/etc/rules/local_rules.xml
^G Get Help     M-D DOS Format     M-A Append     M-B Backup File
^C Cancel       M-M Mac Format     M-P Prepend    ^T To Files
```

Result:

- The rule became active and was visible in the Wazuh dashboard.
- I simulated multiple failed login attempts, via Command Line

```
C:\Users\DELL>ssh wazuh-user@192.168.1.244
wazuh-user@192.168.1.244's password:
Permission denied, please try again.
wazuh-user@192.168.1.244's password:
Permission denied, please try again.
wazuh-user@192.168.1.244's password:
wazuh-user@192.168.1.244: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

Alert Triggered: Brute Force Detection

After configuring the custom rule for detecting brute-force or authentication failure attempts, I tested it by simulating multiple failed login attempts.

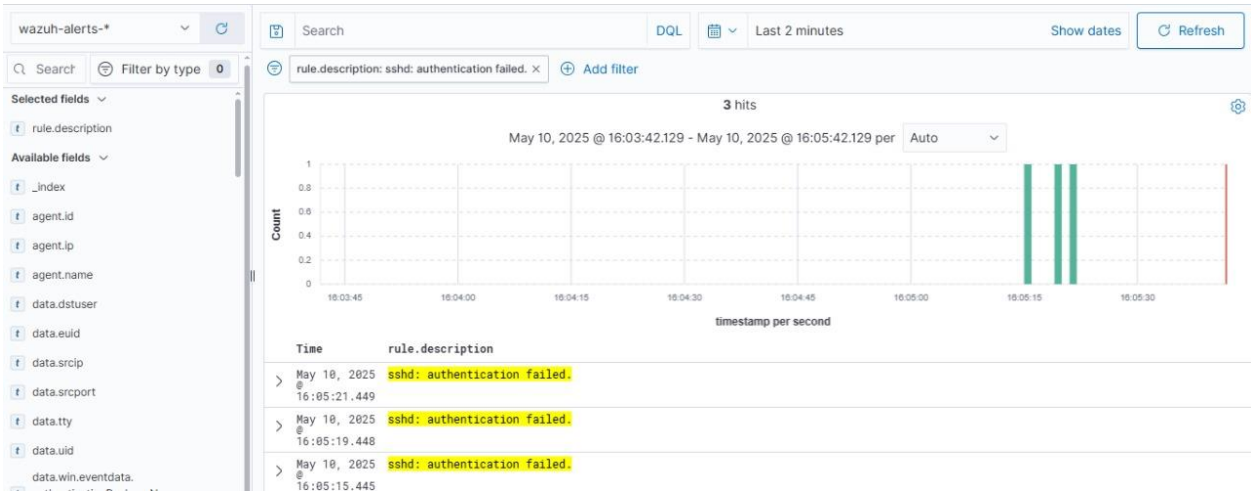
Test Procedure:

- I attempted to log in with incorrect credentials multiple times within a short time frame (3 attempts in under 60 seconds) from the same IP address.

Result:

- The custom rule was triggered successfully.
- An alert appeared in the Wazuh dashboard under the **"Security Events"** or **"Alerts"** section.
- The alert message displayed:

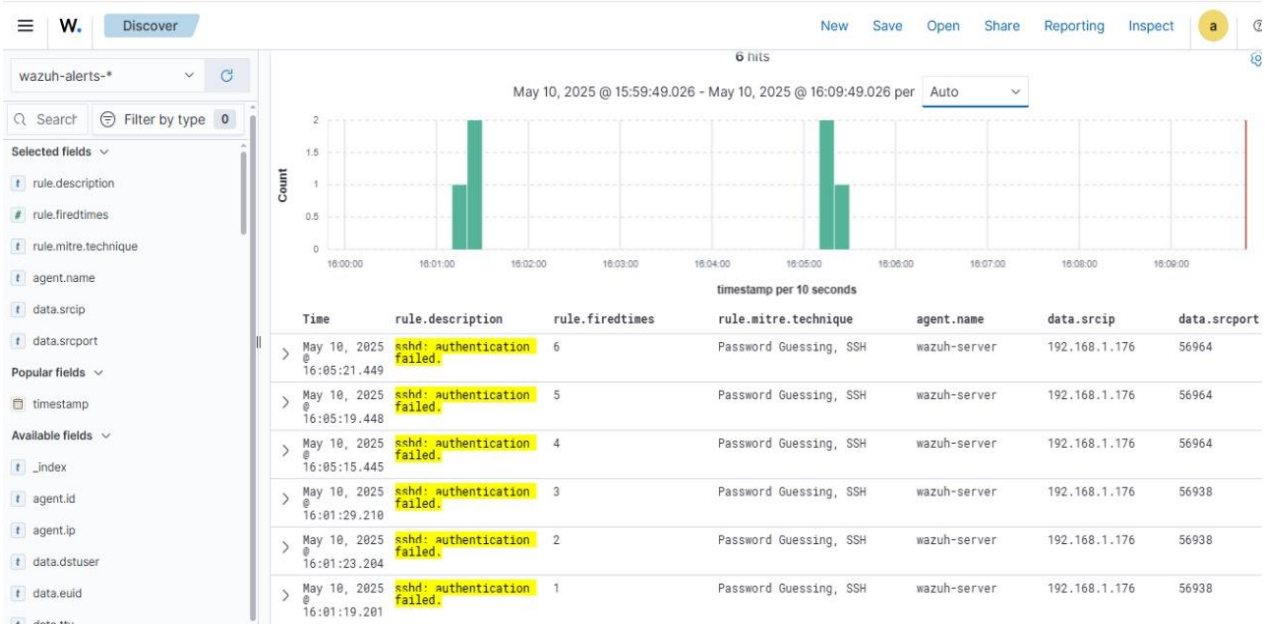
  "sshd: authentication failed"



Additional details in the alert included:

- **Source IP:** 192.168.1.176
- **Timestamp:** May 10 @ 16:04 PM
- **Rule ID**: 100100
- **Severity Level**: 10
- **Rule Fired Times**: 6

This confirmed that the Wazuh server is actively monitoring and detecting suspicious authentication behavior based on the custom rule created.



Custom Dashboard for SSH Authentication Alerts

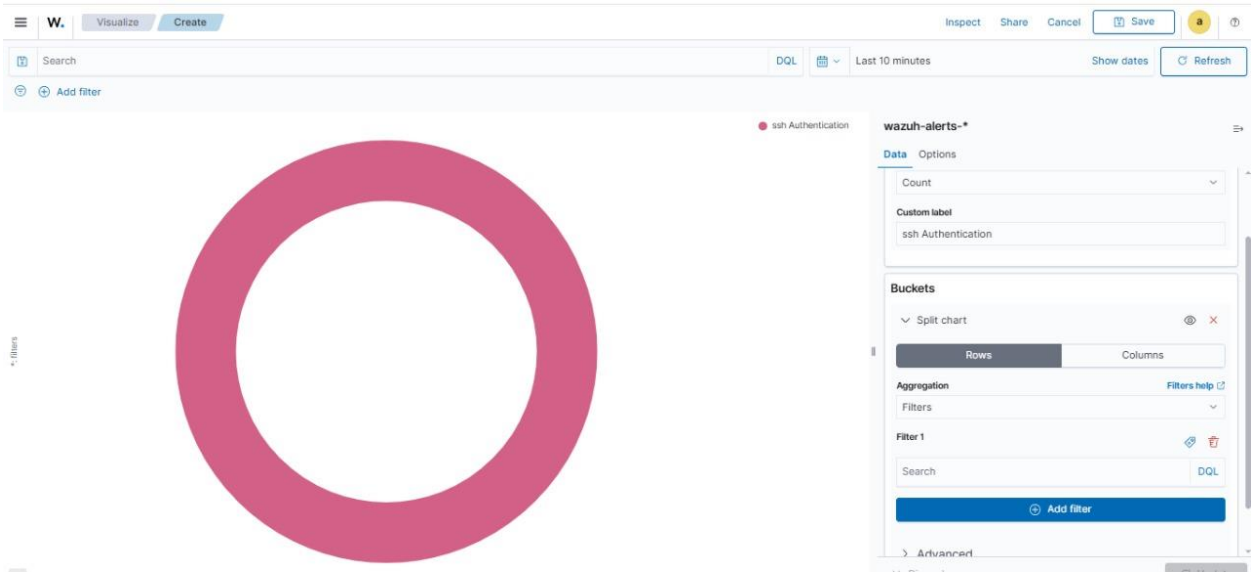
To improve monitoring and quickly detect suspicious SSH activity, I created a custom dashboard in Wazuh focused specifically on **SSH authentication events**.

Purpose of the Dashboard:

- **Centralized View**: Provides a dedicated visual representation of SSH login activity, separating it from general system alerts.
- **Real-Time Monitoring**: Helps in spotting brute-force attacks or unusual SSH login attempts as they happen.
- **Security Incident Response**: Speeds up analysis and decision-making by highlighting authentication trends and potential attack patterns.
- **Visualization**: The donut chart visualization makes it easy to understand the frequency and severity of SSH-related alerts at a glance.

Dashboard Features:

- Pulls data from the wazuh-alerts-* index.
- Filters specifically for events labeled as **SSH Authentication**.
- Uses count aggregation to visualize how often these events occur over a selected time range.



Categorizing SSH Alerts by Severity

To further refine my SSH alert dashboard and improve threat analysis, I categorized the authentication alerts based on **severity levels** — specifically into **Medium** and **Low** severity.

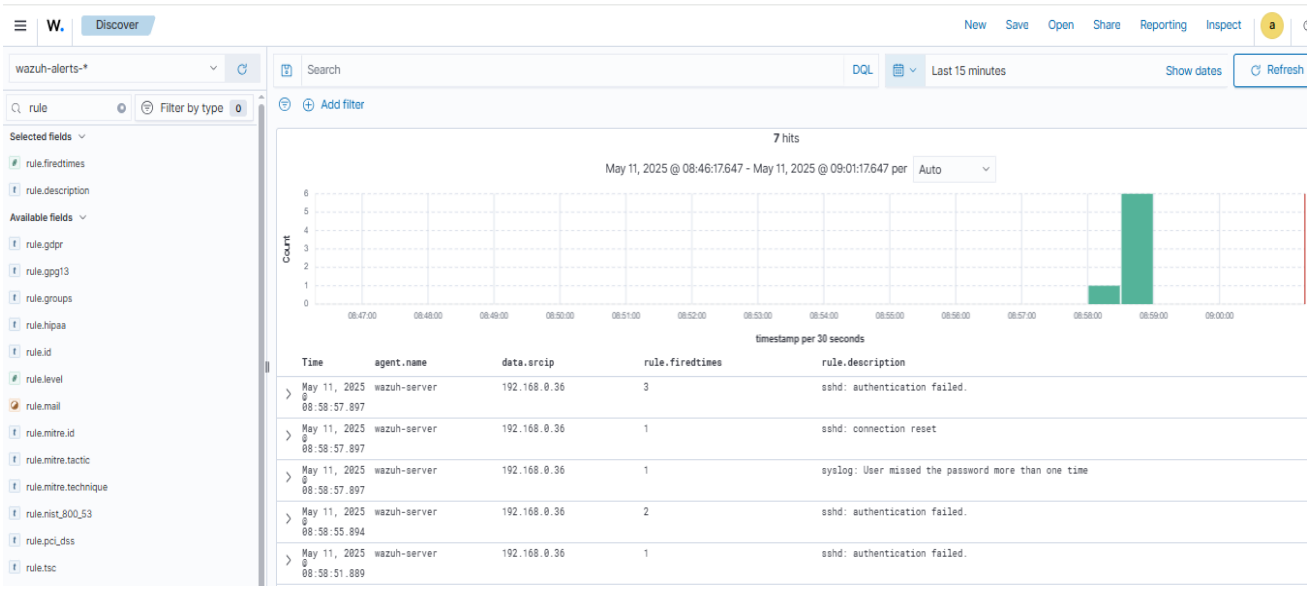
Purpose:

- **Prioritization**: Helps identify which alerts require immediate attention (e.g., multiple failures) versus routine monitoring (e.g., single failed attempt).
- **Better Incident Response**: Enables faster decision-making by focusing on higher-risk activities first.
- **Improved Visualization**: Makes the dashboard cleaner and more informative through color-coded or grouped severity charts.

Implementation:

- I applied filters within the dashboard to split SSH-related alerts by their assigned severity levels from Wazuh (based on rule configuration).
- The donut chart or other visualizations now reflect:
    - **Medium severity**: e.g., multiple failed logins from the same IP within a short time frame.

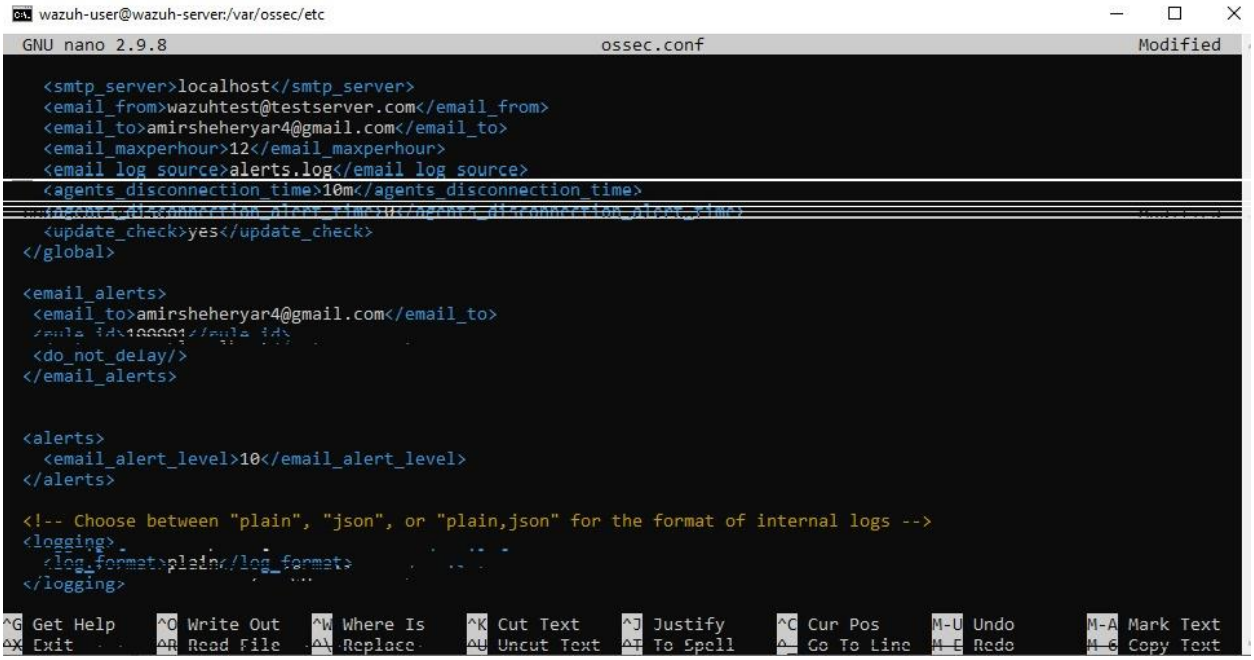
- This categorization provides a clear visual breakdown of potential threats and improves the efficiency of the monitoring process.



Custom Email Alerts Configuration in Wazuh

To enable custom email alerts in Wazuh, modifications were made to the ossec.conf file. The <**amirsheheryar4@gmail.com**>, <**wazuhtest@testserver.com>,** and <**12** > fields were configured to define the sender and receiver email addresses, and to limit the number of alerts per hour.

- Defined <agents_disconnection_time> to 10 minutes to detect inactive agents.
- Enabled <update_check> to allow automatic update checks.
- Ensured email notifications are sent for critical Wazuh events.



Conclusion

*In this project, I successfully downloaded, installed, and customized **Wazuh 4.9** using a pre-configured OVA file in **VirtualBox**. The process involved deploying the virtual machine, retrieving the server IP, and accessing the Wazuh dashboard using default credentials.*

*Once inside the dashboard, I confirmed that the Wazuh agent was active and transmitting logs. I created a **custom brute-force detection rule** to identify repeated authentication failures, which triggered successfully and generated alerts within the system.*

To enhance monitoring and visibility, I:

- *Accessed and analyzed real-time logs via the **Discover** tab.*
- *Built a **custom dashboard** focused on **SSH authentication alerts**.*
- ***Categorized alerts by severity** (Medium and Low) to support better prioritization and response.*

Overall, this setup demonstrates a functional and customized Wazuh environment capable of detecting and visualizing security threats, particularly brute-force attacks, in a virtual lab setting.