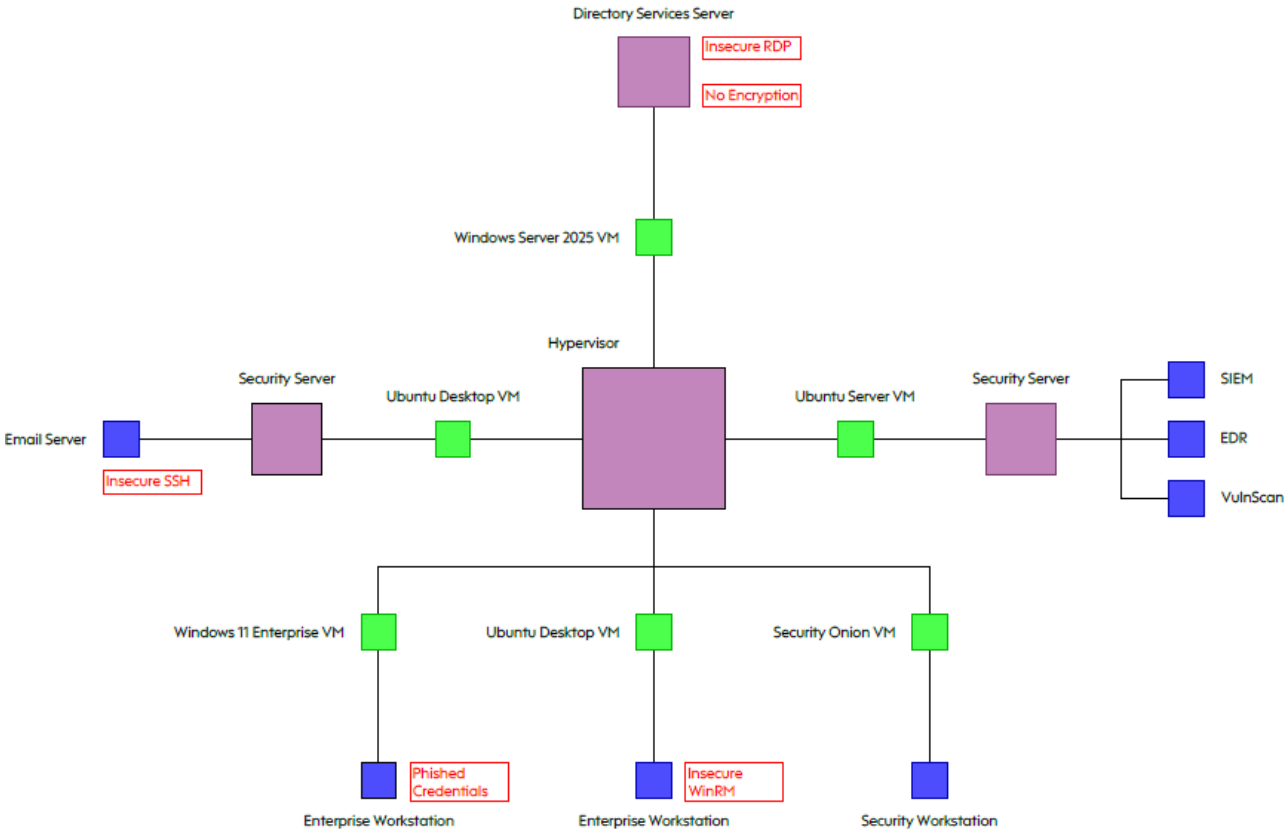
Cybersecurity Homelab: SIEM & Detection Setup Documentation:

This guide provides step-by-step instructions with screenshots for key steps. Optional explanations are highlighted in red for clarity and can be skipped.

This project is a direct continuation from the first project, **Cybersecurity Homelab: Building the Environment.**

The purpose of this project is to simulate common network vulnerabilities and configure the SIEM to accurately detect and respond to these security threats.

Below is a diagram of the network architecture and the vulnerabilities that will be applied highlighted in red:



Step 1: Provision Insecure SSH on Security Server 1

1. Start Security Server 1 VM

- Boot the VM and log in.

- Open a terminal. Prepare the VM for SSH installation and configuration.

2. Install & Enable SSH

```
sudo apt update
sudo apt install openssh-server -y
```

sudo systemctl start ssh
sudo systemctl enable ssh
sudo systemctl status ssh

Installs the OpenSSH server, starts the service, and ensures it runs on boot. Verifying status confirms it is active.

3. Edit SSH Configuration

sudo nano /etc/ssh/sshd_config

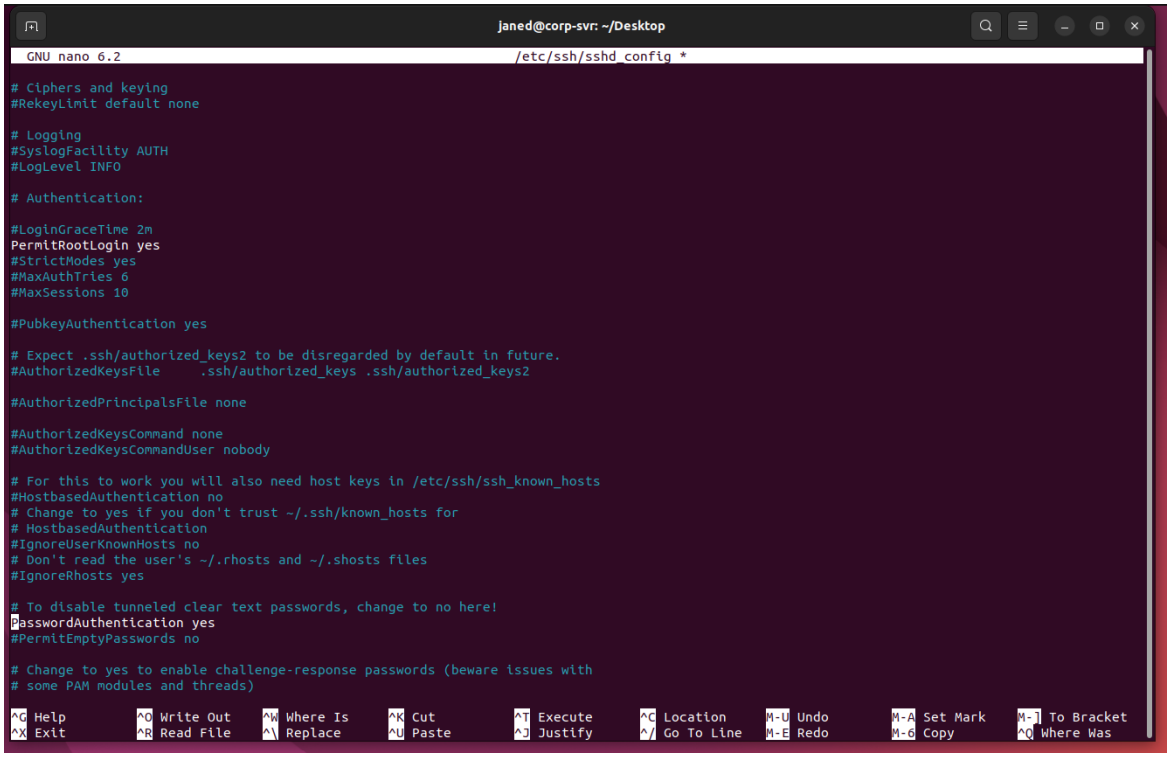
Make the following changes:

- PasswordAuthentication yes

Allows password-based logins, vulnerable to brute-force attacks.

- PermitRootLogin yes

The SSH configuration file controls the server's authentication and access settings. Editing it introduces insecure settings for lab purposes.



4. Restart SSH Service

sudo systemctl restart ssh

Applies the new insecure configuration.

5. Set Weak Root Password

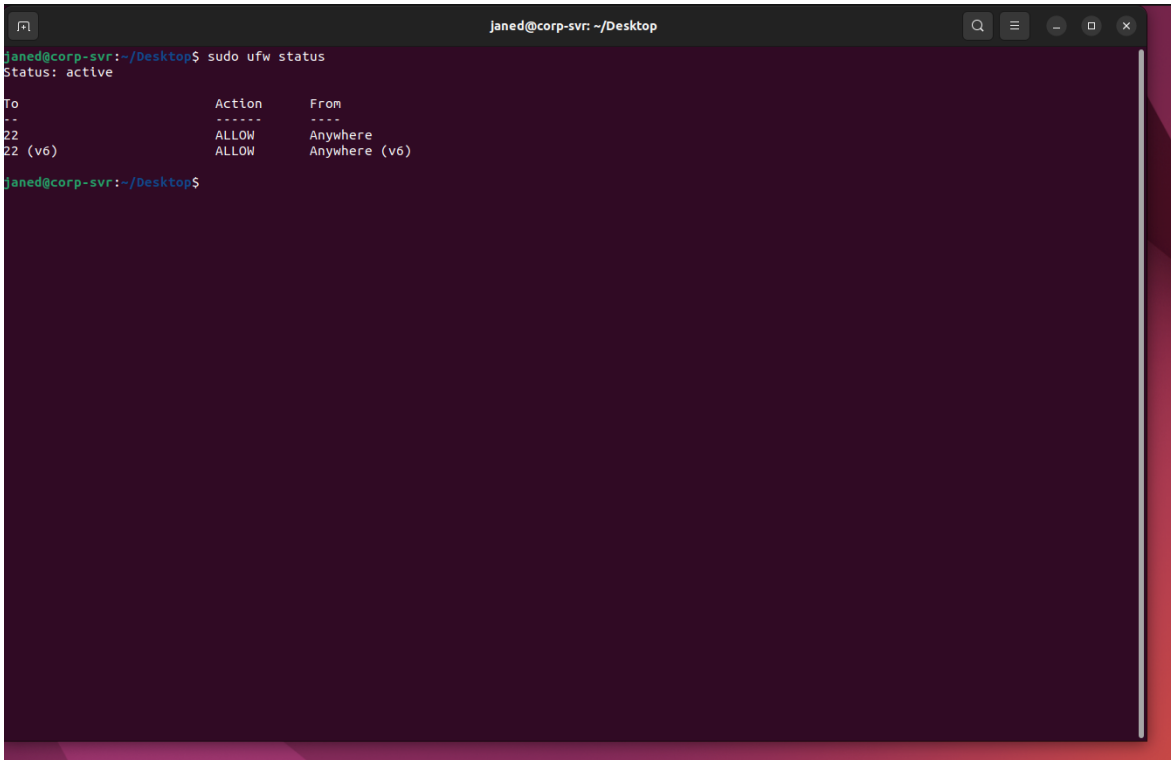
sudo passwd root

Set a predictable root password, making the system easily exploitable.

5. Configure Firewall

sudo ufw allow 22
sudo ufw status

Opens port 22 for SSH access. This simulates a network-exposed server, increasing exposure for testing detection and alerts.



6. Take Snapshot

To capture the vulnerable state to allow rollback and repeatable testing.

<u>Step 2: Create Insecure WinRM On Enterprise Workstation 2:</u>

1. Start Enterprise Workstation 2 VM
- Boot the VM and log in.

- Open a terminal. Prepare the VM for service configuration.

2. Install & Enable SSH

```
sudo apt update
sudo apt install openssh-server -y
sudo systemctl start ssh
sudo systemctl enable ssh
sudo systemctl status ssh
```

Installs OpenSSH, starts it, and ensures it runs on boot. Verifying status confirms the service is active.

3. Edit SSH Configuration

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

PasswordAuthentication yes → allows password-based logins (vulnerable to brute-force).

PermitRootLogin yes → permits direct root login (gives full control if password is obtained).

The SSH config is the server's control center — editing these fields intentionally weakens auth for lab testing.

4. Restart SSH Service

```
sudo systemctl restart ssh
```

Applies the new configuration.

5. Set Weak Root Password

```
sudo passwd root
```

Sets a predictable/weak root password so the system is easily exploitable for attack simulation.

6. Open SSH in Firewall

```
sudo ufw allow 22
sudo ufw status
```

Opens port 22 so the host is reachable from the lab network. This increases exposure for attack testing (intentionally).

7. Verify SSH Running

```
sudo systemctl status ssh
```

Confirm active (running).

8. Take Snapshot

Creates a restore point so you can repeat or revert the exercise.

Step 3: Create Insecure WinRM On Enterprise Workstation 1:

1. Start the VM & open an elevated PowerShell

- Right-click Windows PowerShell → Run as administrator.

2. Run the insecure configuration commands

```
powershell -ExecutionPolicy Bypass
Enable-PSRemoting -Force
winrm quickconfig -transport:https
Set-Item WSMan:\localhost\Client\TrustedHosts -Value '*'
net localgroup "Remote Management Users" /add Administrator
Restart-Service WinRM
```

Enable-PSRemoting -Force and winrm quickconfig -transport:https enable WinRM listeners so remote PowerShell sessions are possible.

-ExecutionPolicy Bypass lets arbitrary scripts run (useful for lab payloads).

TrustedHosts = '' trusts all remote hosts (highly insecure — allows MITM/credential relay).*

Adding Administrator to Remote Management Users expands remote access rights.

Restarting WinRM applies changes.

3. Firewall / Network Profile fix (if Enable-PSRemoting errors)

If Enable-PSRemoting fails with a message about a Public network profile preventing firewall exception creation, run:

```
Get-NetConnectionProfile | Format-Table Name, InterfaceAlias, InterfaceIndex, NetworkCategory -AutoSize

Set-NetConnectionProfile -InterfaceIndex <InterfaceIndex> -NetworkCategory Private
```

Rerun:

```
Enable-PSRemoting -Force
```

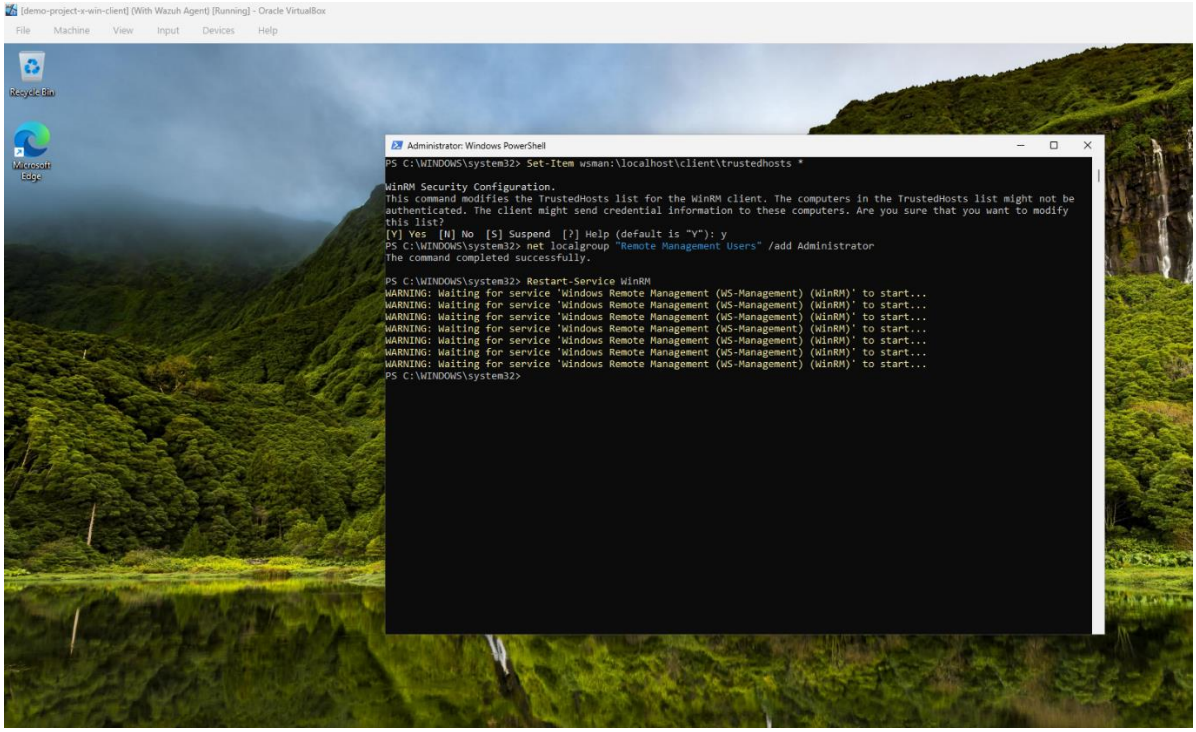
4. Verify WinRM & listeners

```
Get-Service WinRM
winrm enumerate winrm/config/listener
Test-WsMan -ComputerName localhost
```

Get-Service should report Running.

winrm enumerate ... shows listeners on ports 5985/5986.

Test-WsMan -ComputerName localhost tests local WinRM.



5. Take Snapshot

Preserve the vulnerable state for repeatable attack/testing and safe rollback.

<u>Step 4: Enable Insecure RDP On Directory Services Server:</u>

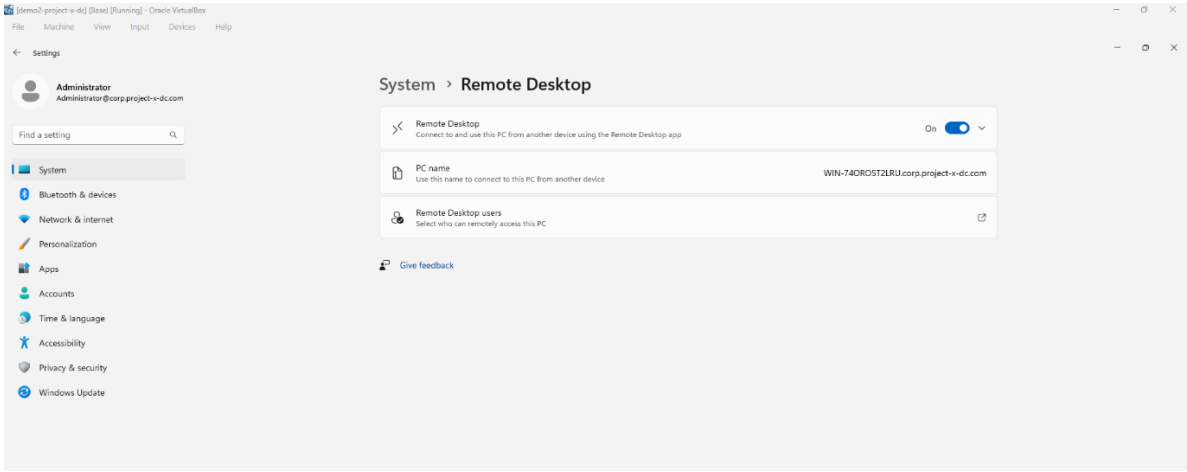
1. Start the Domain Controller VM

- Boot project-x-dc and sign in as CORP\Administrator (or equivalent admin account).

2. Enable Remote Desktop (GUI)

- Open Settings → System → Remote Desktop.
- Toggle Enable Remote Desktop → Confirm when prompted. This enables the RDP server (listening on TCP port 3389).

RDP provides a graphical remote session; enabling it simulates an exposed administrative surface often targeted by attackers.

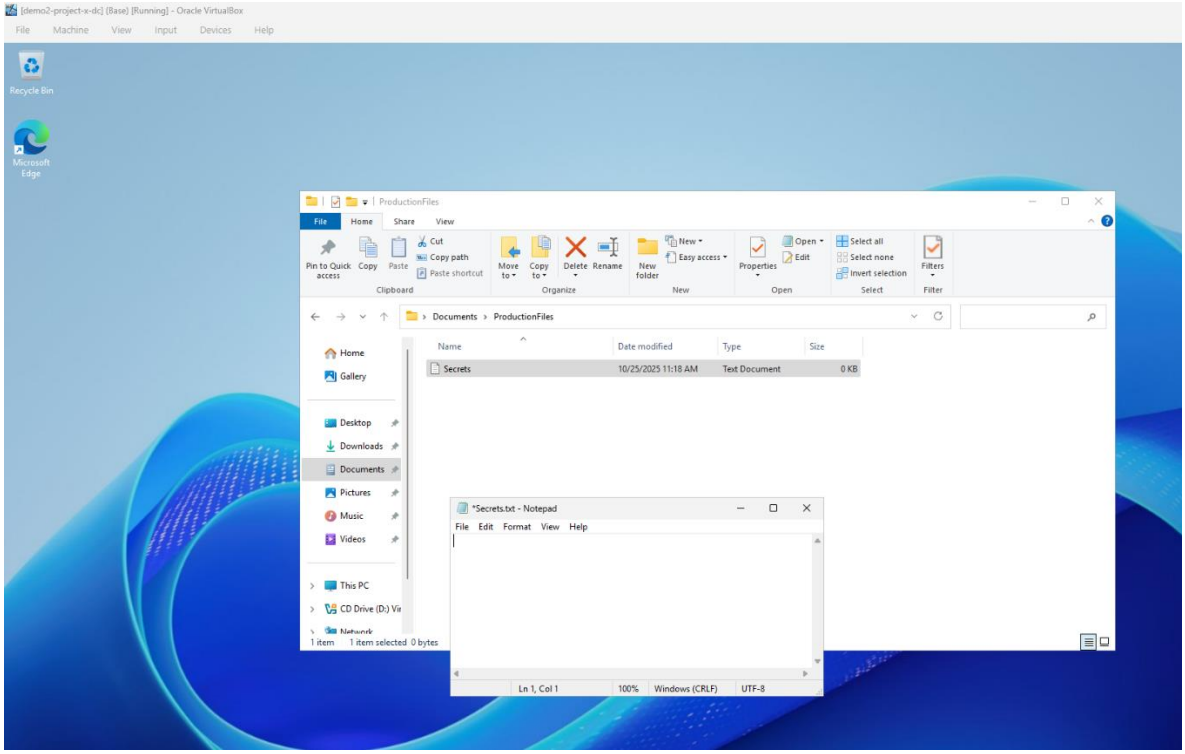


Step 5: Create a Sensitive "Emulated" File on the Domain Controller:

1. Create the folder and sensitive file

- Open File Explorer → C:\Users\Administrator\Documents
- Right-click → New → Folder → name it ProductionFiles.
- Open the ProductionFiles folder → Right-click → New → Text Document → rename to secrets.txt.
- Open secrets.txt, type a short sample secret → Save → Close.

Provides a simple, easily located target for File Integrity Monitoring and exfiltration exercises.



2. Take Snapshot

Allows you to revert to the state where the file exists for repeated exfiltration/detection tests.

Step 6: Setting Up Detections & Alerts For SSH Authentication Attempt Failures:

1. Access Wazuh Dashboard

- Open internet browser and type http://localhost

- Log into Wazuh

2. Generate SSH Authentication Failures (Test)

- Ensure project-x-linux-client agent is active in Server Management → Endpoint Summary.

- Open a terminal on the Linux client and type ssh root@10.0.0.101

- Accept the host key by typing yes

- Deliberately type wrong passwords multiple times to create failed SSH login events.

This produces log entries that Wazuh can detect using its default SSH authentication failure rules.

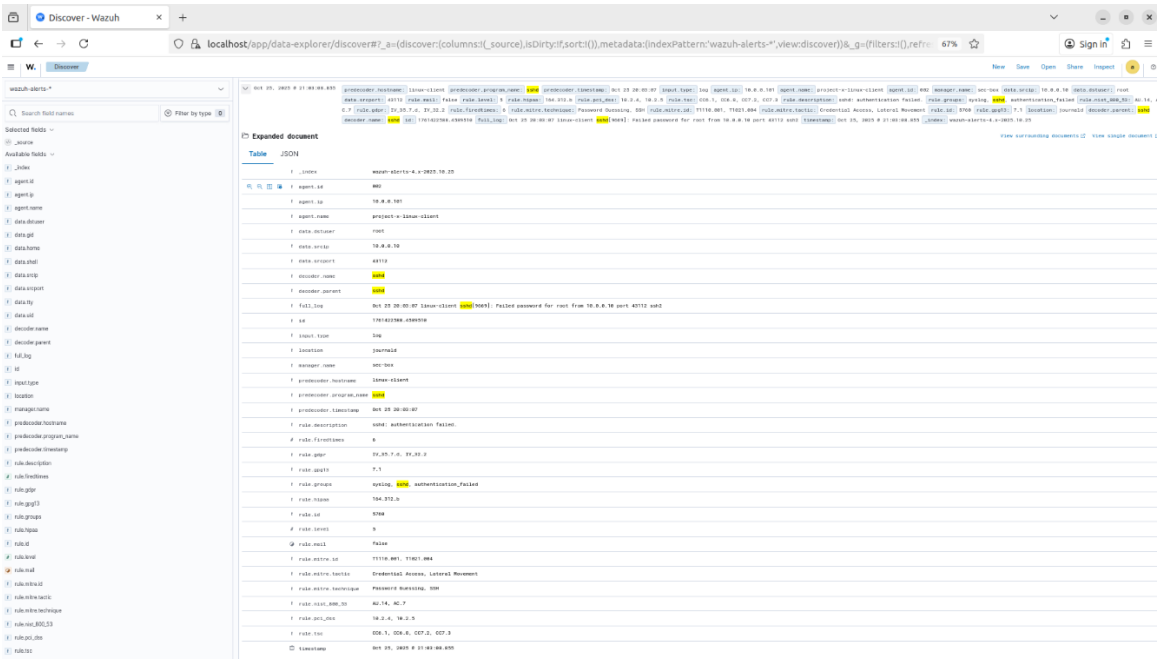
3. Explore Wazuh Logs

- In Wazuh Dashboard → Explore → Discover.

- Search for sshd to view authentication failure logs.

- Confirm logs show failed attempts.

Ensures Wazuh is correctly receiving and parsing SSH authentication failure events.



4. Create Alert Monitor

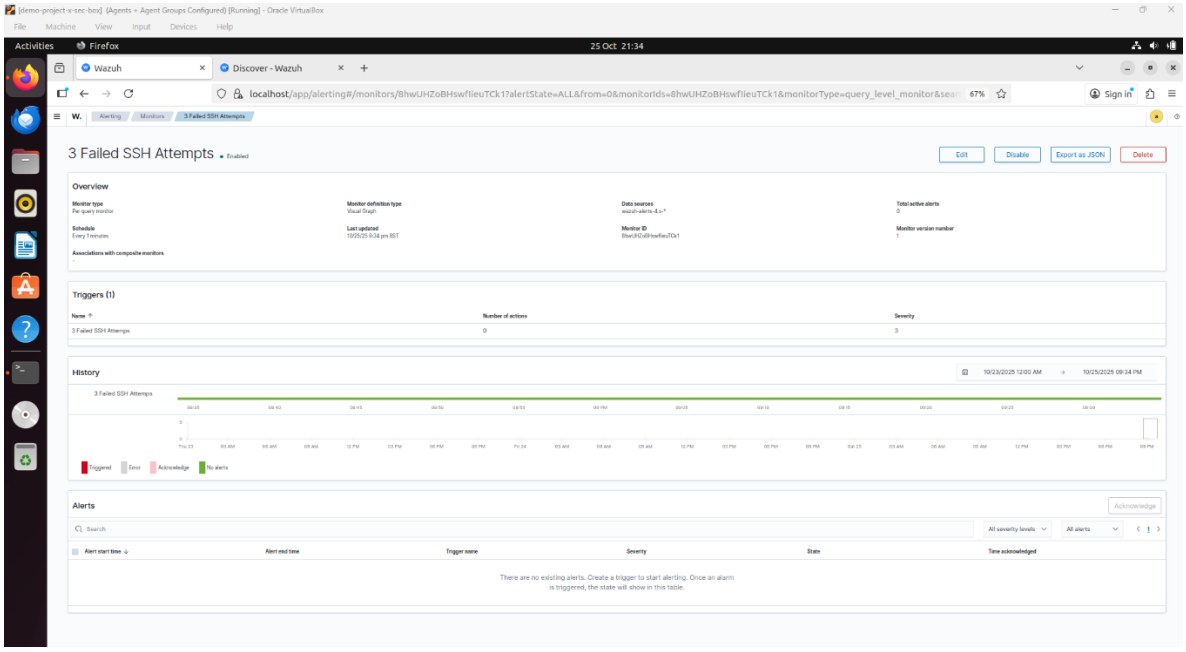
- Navigate to Explore → Alerting → Create Monitor.

- Configure the monitor:
    - Name: 3 Failed SSH Attempts
    - Monitor type: Per query monitoring
    - Defining method: Visual editor
    - Frequency: Run every 1 minute
    - Index pattern: wazuh-alerts-4.x-*
    - Time field: @timestamp
- Define query data filters:
    - decoder.name is sshd
    - rule.groups contains authentication_failed
- Add a trigger under the monitor:
    - Trigger name: 3 Failed SSH Attempts
    - Severity: 3 (Medium)
    - Trigger Condition: Above 2

Select create to save the monitor.

Filters logs specifically for SSH authentication failures, ignoring unrelated events.

Generates an alert only when multiple authentication failures occur, reducing noise while detecting brute-force attempts.



5. Take Snapshot

Preserves configuration for repeatable testing and rollback if needed.

Step 7: Configure Wazuh Alert for Successful WinRM Logons

1. Access Wazuh Rules

- In Wazuh Dashboard, navigate to Rules → Search.

- Search for rule 60106 (Windows Logon Success).

This default rule tracks successful Windows logins, allowing us to detect when accounts authenticate to endpoints. Event ID 4624 represents successful logons, while 4625 represents failed attempts.

2. Create Alert Monitor

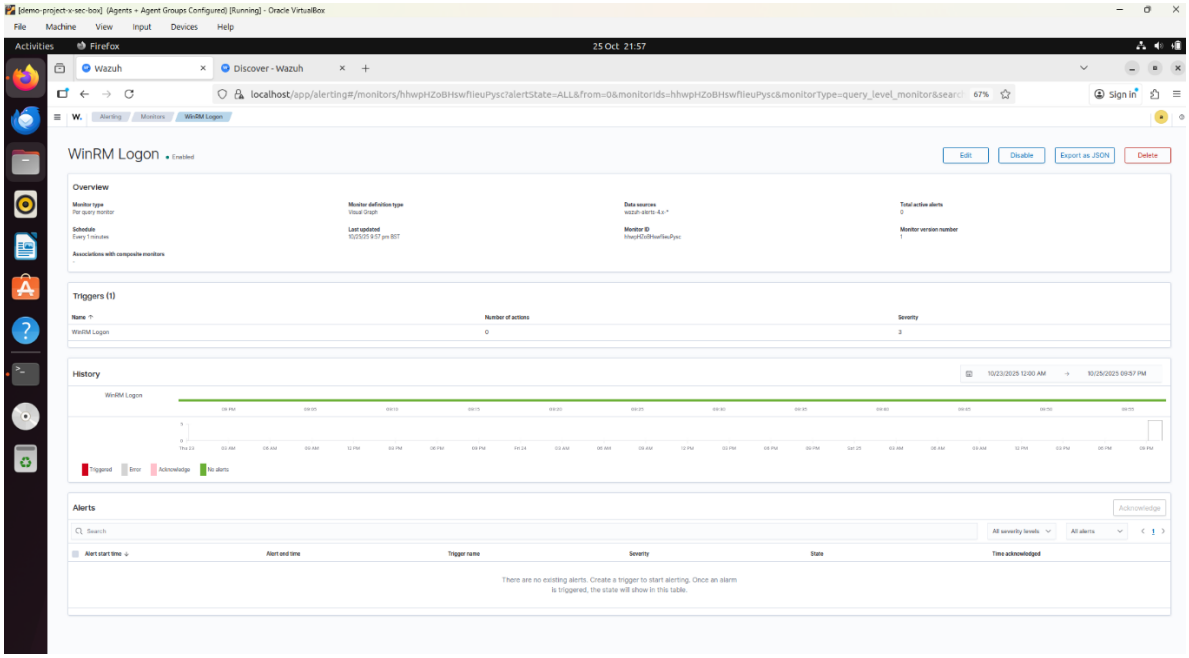
- Navigate to Explore → Alerting → Create Monitor.

- Configure the monitor:
    - Name: WinRM Logon
    - Monitor type: Per query monitoring
    - Defining method: Visual editor
    - Frequency: Run every 1 minute
    - Index pattern: wazuh-alerts-4.x-*
    - Time field: @timestamp
- Define query data filters:
    - data.win.eventdata.logonProcessName is Kerberos
    - data.win.system.eventID is 4624
- Add a trigger under the monitor:
    - Trigger name: WinRM Logon
    - Severity: 3 (Medium)
    - Trigger Condition: Above 1

Select create to save the monitor.

Filters log events to successful WinRM logins over Kerberos, focusing alerts on relevant authentication activity.

Generates alerts when multiple successful logins are detected, signaling potential unauthorized or suspicious access.



3. Take Snapshot

Preserves configuration for repeatable testing and rollback if needed.

<u>Step 8: Setting Up Alert To Monitor Sensitive File In Directory Services Server:</u>

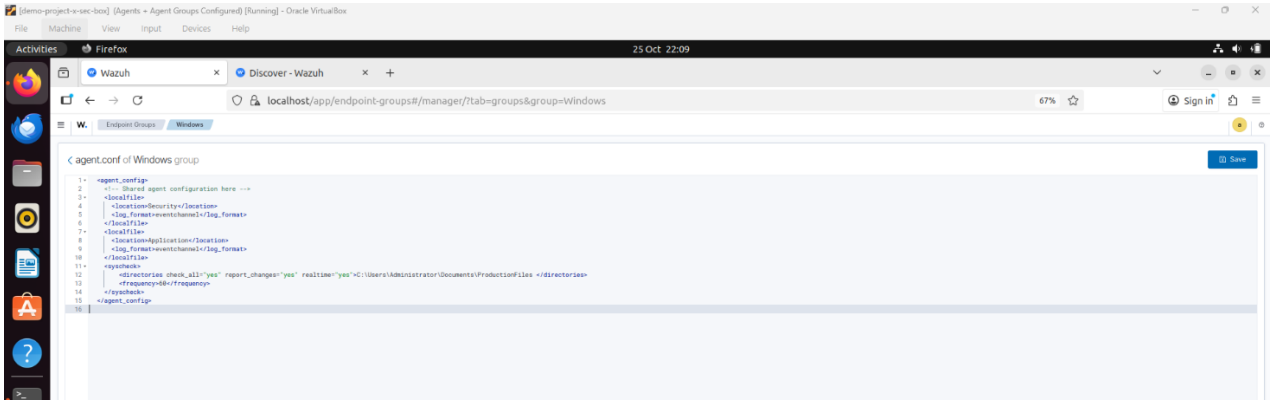
1. Configure File Integrity Monitoring (FIM) on Windows Endpoints

- Navigate to Endpoint Groups → Windows → Files → agent.conf

- Add the following syscheck block:

```
<syscheck>
  <directories check_all="yes" report_changes="yes" realtime="yes">
    C:\Users\Administrator\Documents\ProductionFiles
  </directories>
  <frequency>60</frequency>
</syscheck>
```

Monitors the folder where secrets.txt is stored, reporting any changes in real-time or at the specified interval.



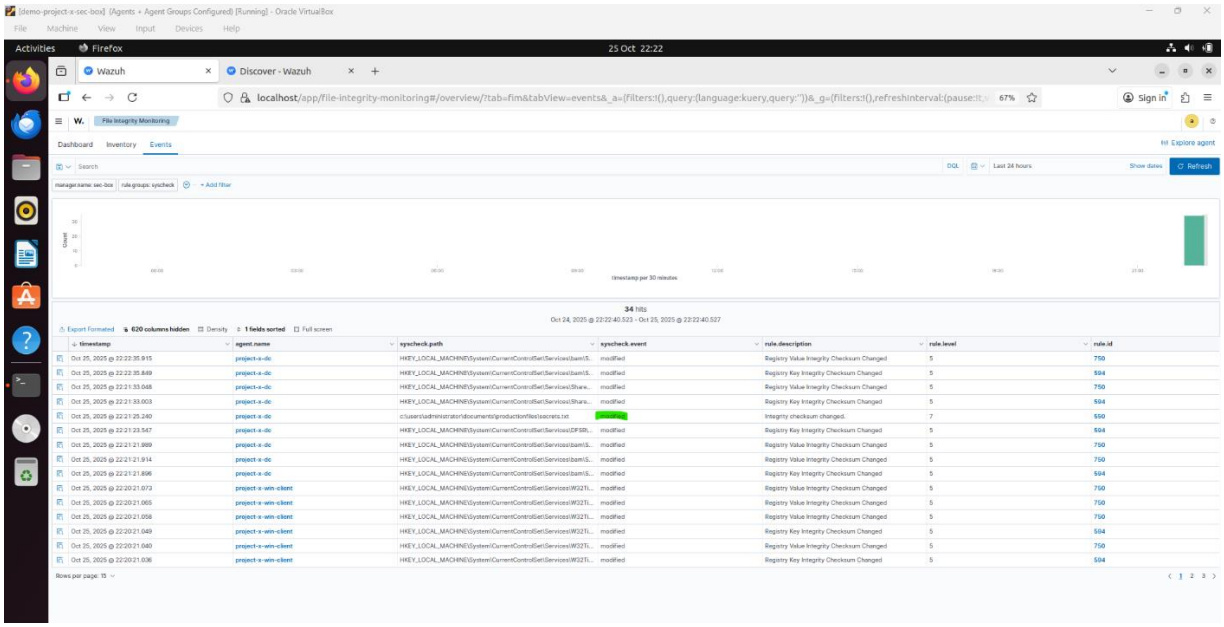
- Restart Wazuh to apply the configuration:

- On Security Server 2 VM: sudo systemctl restart wazuh-manager

- On Directory Services VM:

NET STOP Wazuh
NET START Wazuh

- Modify secrets.txt on the Directory Services VM. Verify the modification is visible in Endpoint Security → File Integrity Monitoring → Events Tab.



2. Create Local Rule for File Alert

- Navigate to Server Management → Rules.

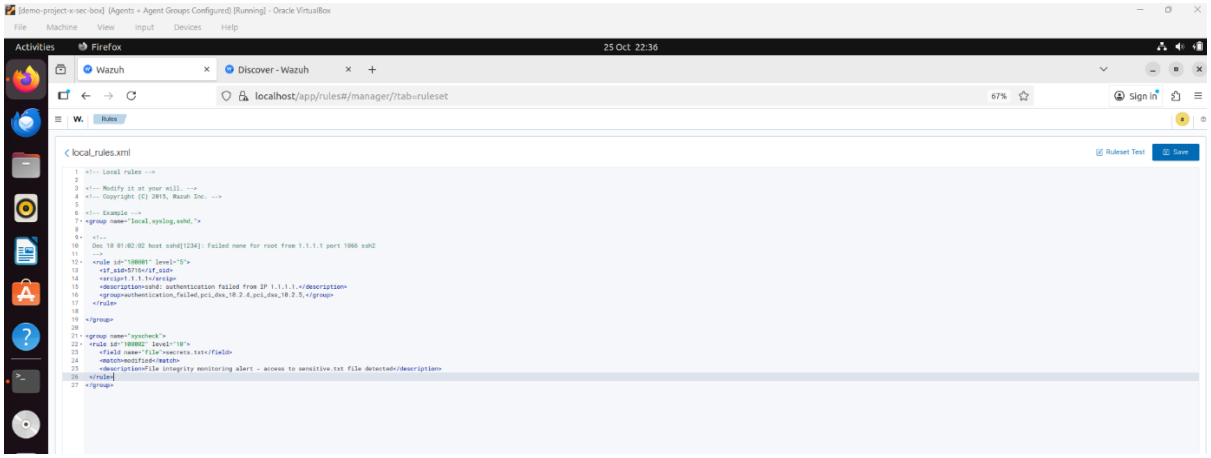
- Open local_rules.xml.

- Add the following block at the end (after the default rules):

```
<group name="syscheck">
 <rule id="100002" level="10">
   <field name="file">secrets.txt</field>
   <match>modified</match>
   <description>File integrity monitoring alert - access to secrets.txt detected</description>
 </rule>
</group>
```

Creates a high-severity rule (level 10) that triggers when secrets.txt is modified.

- Restart Wazuh to apply local rules.

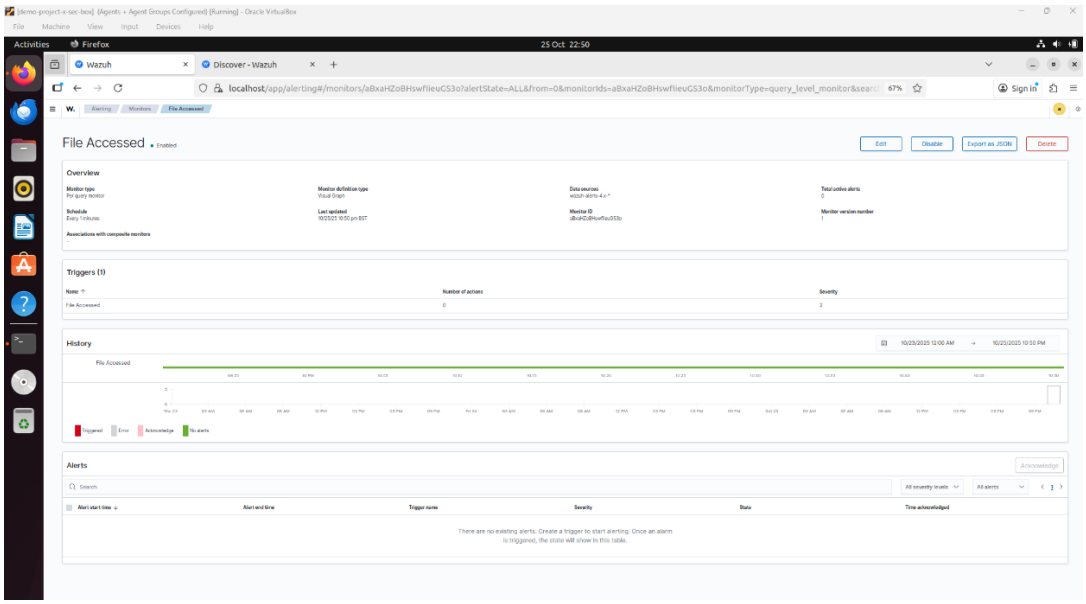


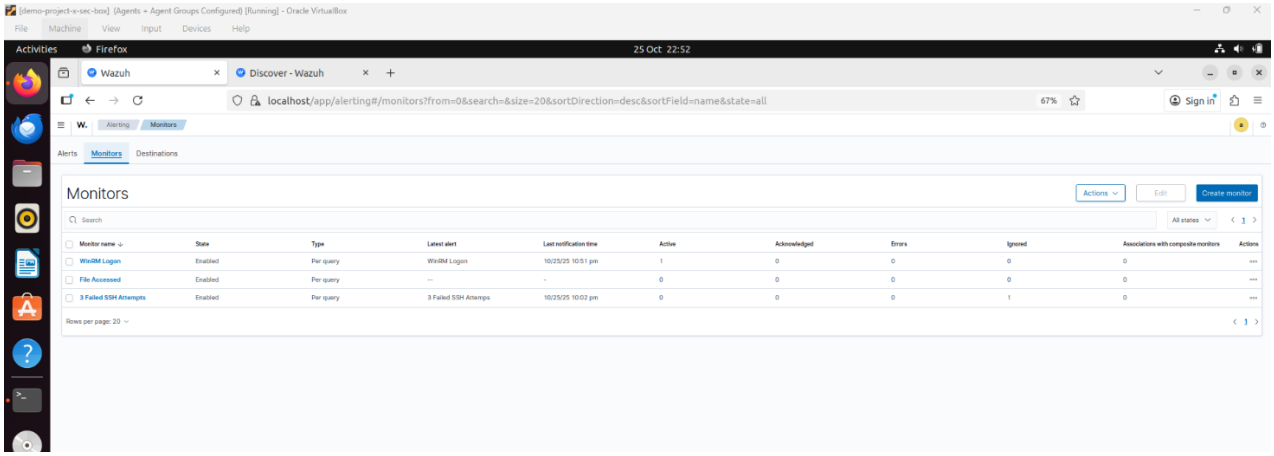
3. Create Alert Monitor for File Modification

- Navigate to Explore → Alerting → Create Monitor.

- Configure the monitor:
    - Name: File Accessed
    - Monitor type: Per query monitoring
    - Defining method: Visual editor
    - Frequency: Run every 1 minute
    - Index pattern: wazuh-alerts-4.x-*
    - Time field: @timestamp
- Define query data filters:
    - full_log contains secrets.txt
    - syscheck.event is modified
- Add a trigger under the monitor:
    - Trigger name: File Accessed
    - Severity: 2 (High)
    - Trigger Condition: Above 1

Select create to save the monitor.





5. Take Snapshot

Preserves the file monitoring configuration and ensures repeatable testing.