Implementation of a SOC Lab for Real-Time Threat Detection and Incident Response using Wazuh SIEM

Phase 1: SIEM Server Deployment

Phase 1 covers the deployment of the central Wazuh SIEM Manager within a secure virtualized environment. By configuring the necessary network protocols and verifying the health of core security services, we establish a functional 'intelligence center' capable of receiving and analyzing security telemetry from across the laboratory network.

Environment Provisioning

- **Virtualization:** Oracle VM VirtualBox.
- **Operating System:** Ubuntu 22.04 LTS (x86_64).
- **Resources:** 3GB RAM and 2 vCPUs.
- **Networking:** Bridged Adapter Mode, enabling the server to be reachable at 192.168.254.132.

Figure 1: Virtual machine hardware and network configuration for the Wazuhmanager.

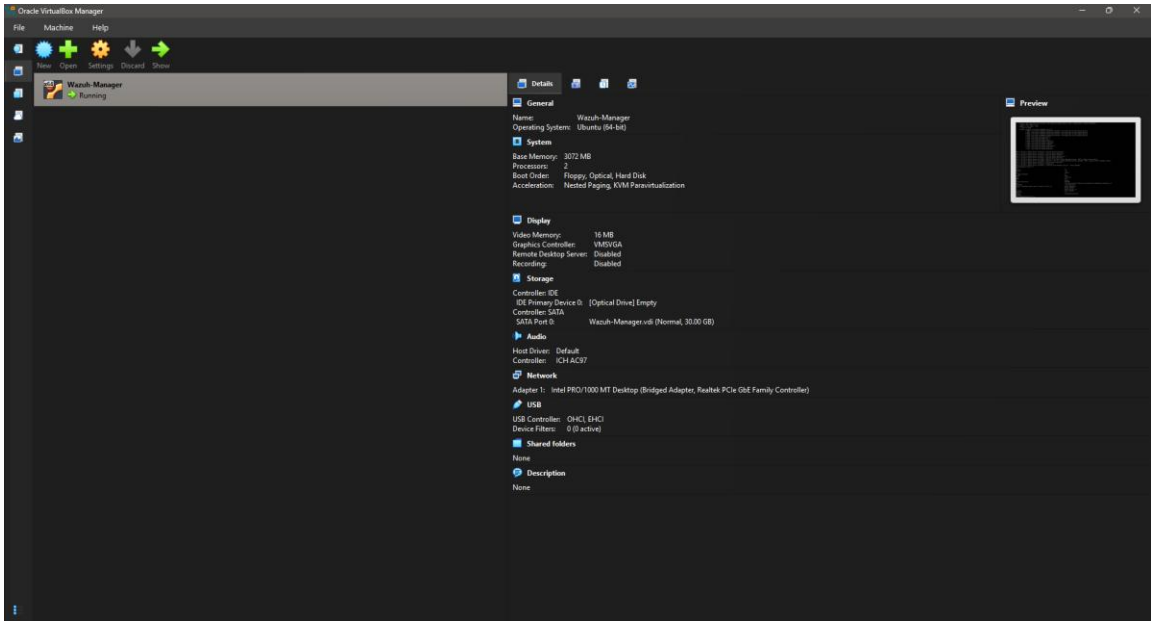
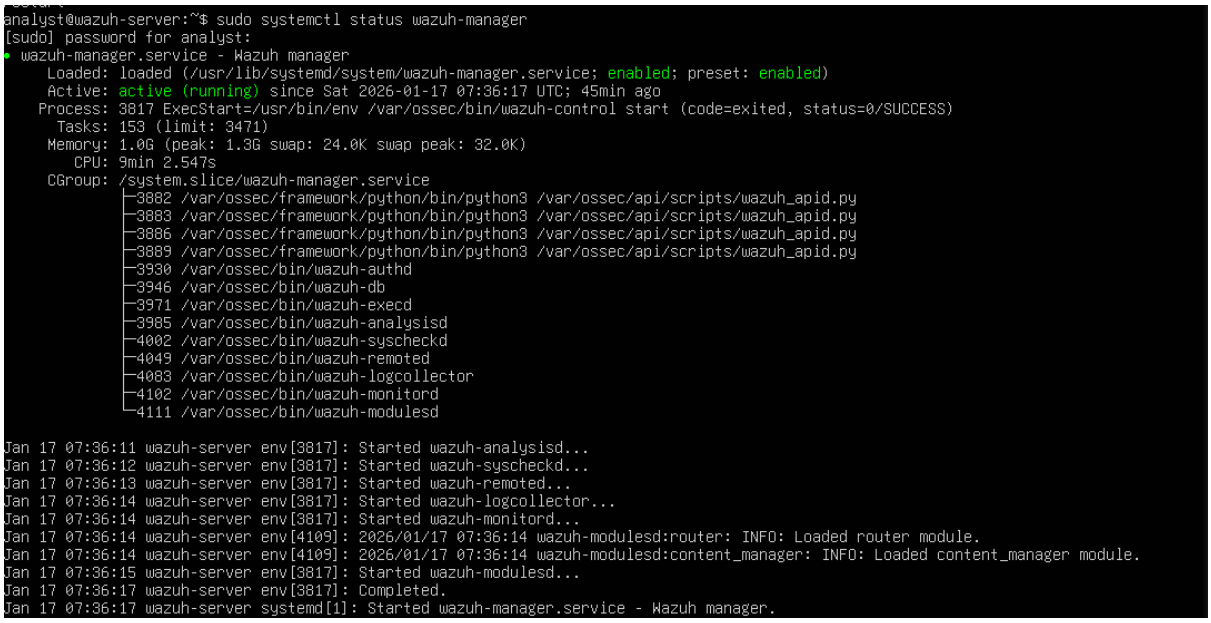


Figure 1 illustrates the hardware and network infrastructure configured within the Oracle VM VirtualBox hypervisor to support the Wazuh SIEM manager. The virtual machine was provisioned with 3GB of RAM and 2 vCPUs to balance system performance with the available host resources. Crucially, the network was configured in Bridged Adapter mode. This configuration allows the Ubuntu guest operating system to obtain a dedicated IP address (192.168.254.132) on the physical local area network, ensuring seamless communication between the Wazuh manager and the Windows host machine that will act as the security agent.

Service Initialization & Verification

After the environment was provisioned, the operational status of the core analysis engine was verified via the Command Line Interface (CLI). This step is critical to ensure that the Wazuh manager is ready to receive and decode security telemetry from remote endpoints.

Figure 2: Verification of the wazuh-manager service status via CLI.



As demonstrated in Figure 2, the wazuh-manager service is confirmed as active (running). This output confirms that all internal sub-daemons, such as the analysis daemon (analysisd), the log collector (logcollector), and the remote registration service (remoted) have successfully initialized and are functional.

Dashboard API Connectivity

The final step of the server setup involved validating the communication between the web-based Wazuh Dashboard and the backend Wazuh Manager. This 'handshake' is performed via the Wazuh API.

Figure 3: Successful Wazuh API connection and dashboard synchronization.

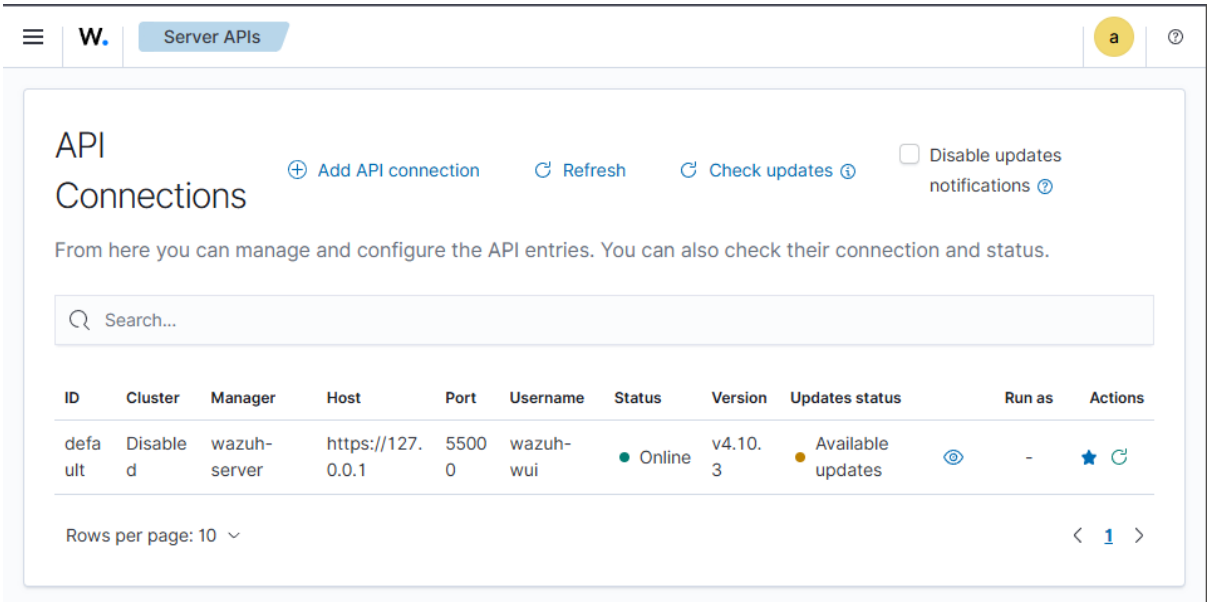


Figure 3 shows the successful 'Online' status of the API connection. While a notification regarding update checks was present due to localized network restrictions, the 'Online' status confirms that the frontend can securely retrieve and visualize security events from the manager. With this verification complete, the SIEM server is officially ready for agent enrollment.

Phase 2: Agent Enrollment and Endpoint Integration

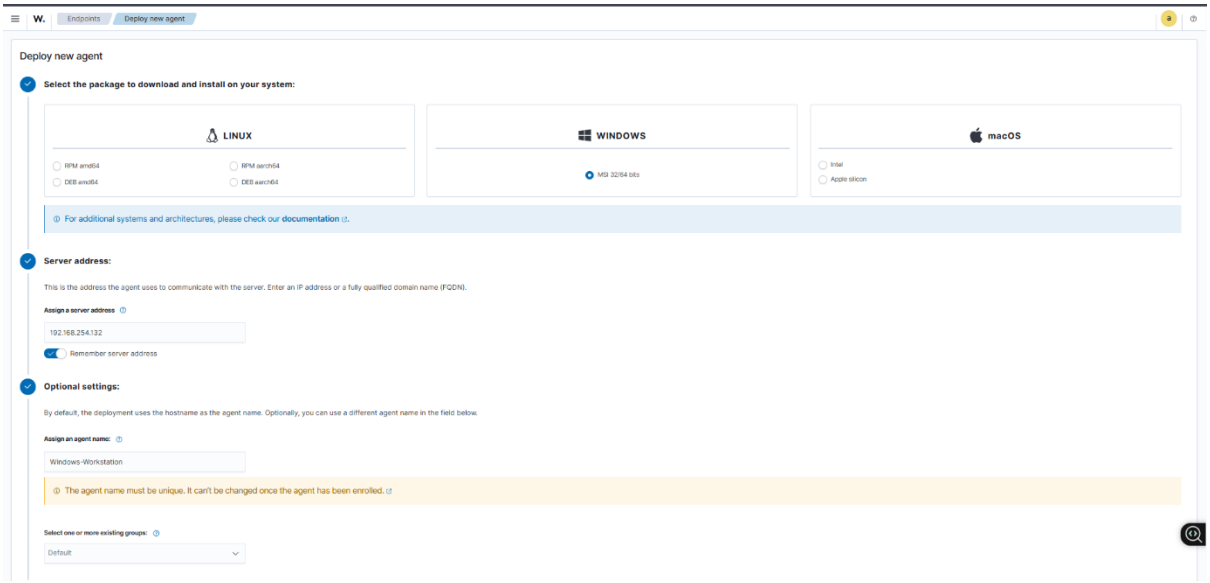
With the Wazuh Manager fully operational, the second phase of the project involved enrolling the Windows host machine as a monitored endpoint. This process integrates the host into the SIEM environment, allowing for real-time log collection and security monitoring.

Generating the Deployment Script

To integrate the Windows host, the Wazuh interactive deployment wizard was utilized. The configuration was specifically set to the Windows operating system, targeting the MSI 32/64 bits package. The Manager IP was explicitly defined as

192.168.254.132, and the agent was assigned the name 'Windows-Workstation' to ensure clear identification within the SIEM dashboard.

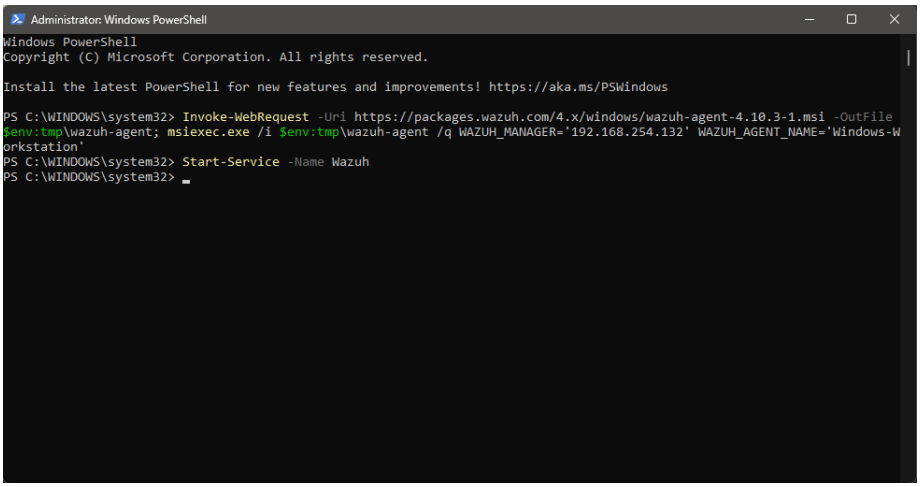
Figure 4: Configuration of the Windows agent deployment and optional naming settings.



Endpoint Installation via PowerShell

The generated deployment script was executed on the target Windows host using an administrative PowerShell session.

Figure 5: Administrative PowerShell execution of the Wazuh agent installation and service initiation.



As shown in Figure 5, the command successfully downloaded the agent package from the Wazuh repository and performed a silent installation with the pre-defined manager parameters. Following the installation, the Start-Service -Name Wazuh command was executed to initialize the agent service and begin the encrypted handshake with the remote manager.

Agent Connectivity Verification

As demonstrated in the dashboard verification phase, the agent enrollment was successful.

Figure 6: Wazuh 'Endpoints' dashboard confirming the successful active enrollment of the Windows agent.

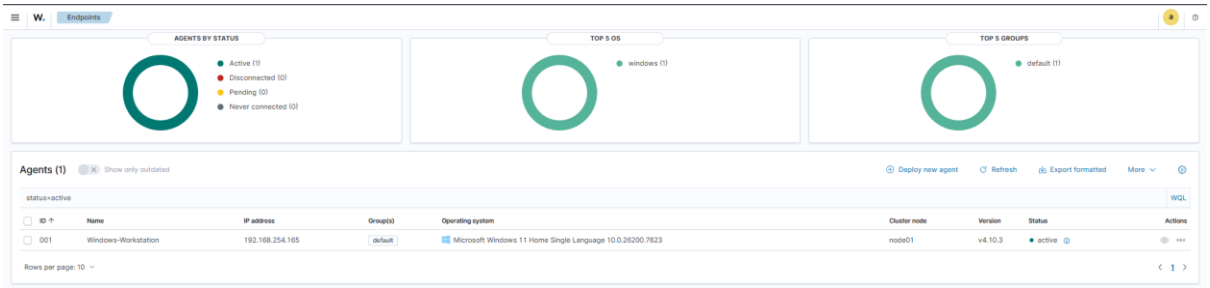


Figure 6 confirms that the Windows host is now listed as 'active', with the SIEM successfully identifying the operating system as Microsoft Windows 11 Home Single Language. This 'Active' status validates that the manager is receiving real-time heartbeats and security telemetry from the endpoint.

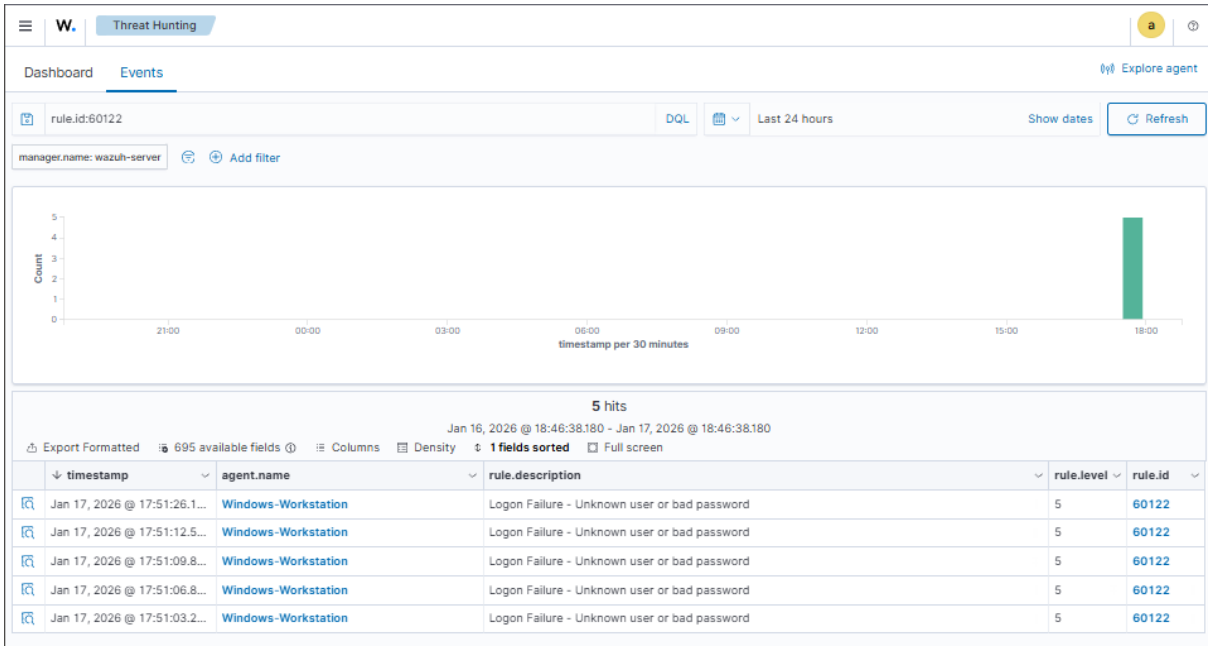
Phase 3: Security Monitoring and Attack Simulation

The final phase of the project demonstrates the SIEM’s capability to detect and alert on active threats in real-time. To evaluate the detection thresholds of the Wazuh Manager, a brute-force attack was simulated on the Windows-Workstation endpoint.

This activity generated multiple 'Audit Failure' events which were harvested by the Wazuh agent and forwarded to the manager for analysis.

Real-Time Alert Detection

Figure 7: Wazuh 'Threat Hunting' dashboard displaying real-time alerts for the simulated brute-force attack.



As shown in Figure 7, the Wazuh Manager successfully triggered high-severity alerts for the failed authentication attempts. The dashboard reflects five distinct 'hits' for Rule ID 60122 (Logon Failure), confirming that the manager is successfully processing telemetry from the Windows agent.

Forensic Event Analysis

A granular review of the security events was conducted to verify the accuracy of the logs.

Figure 8: Detailed forensic metadata of a failed logon event extracted from the Windows endpoint.

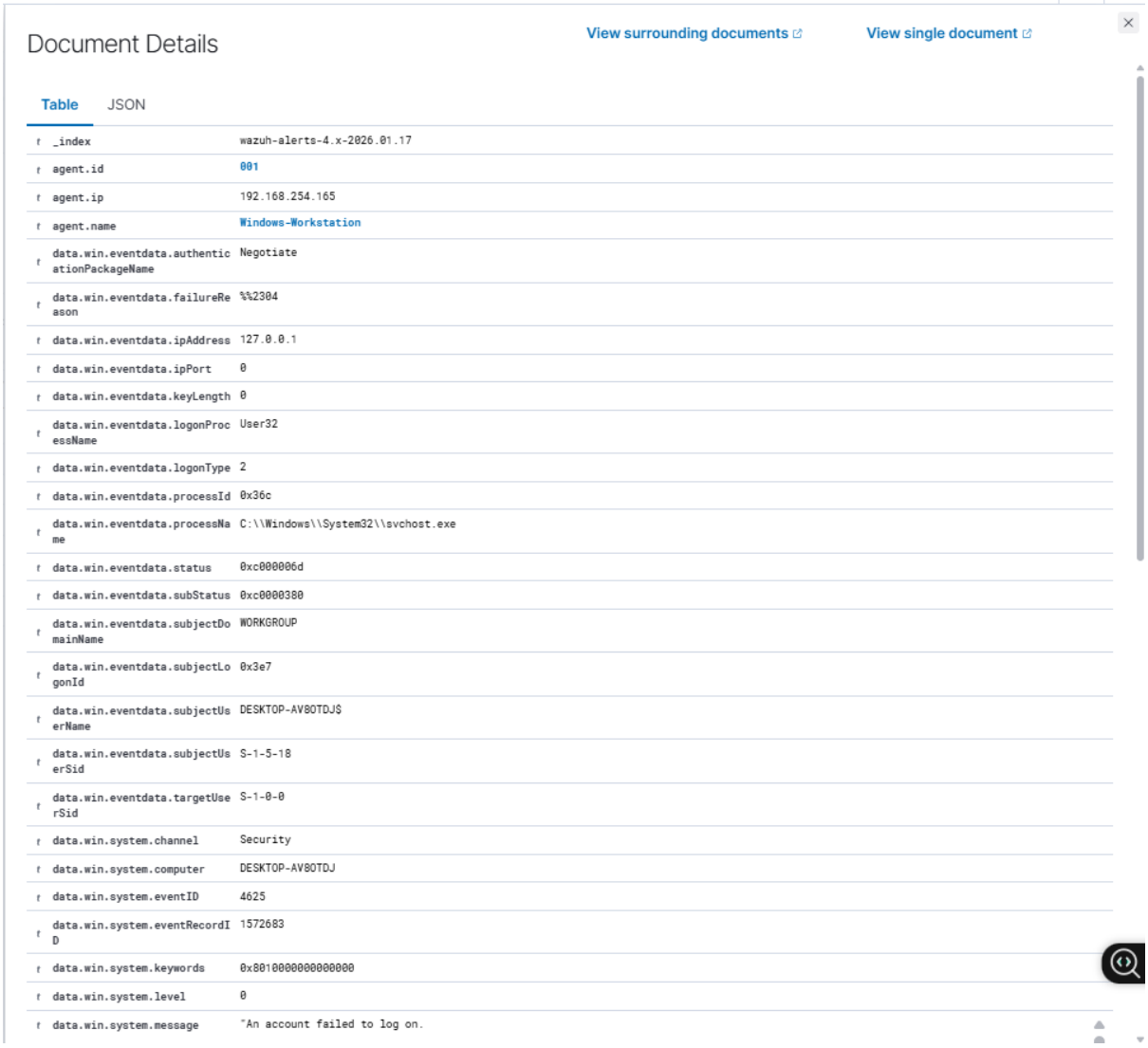
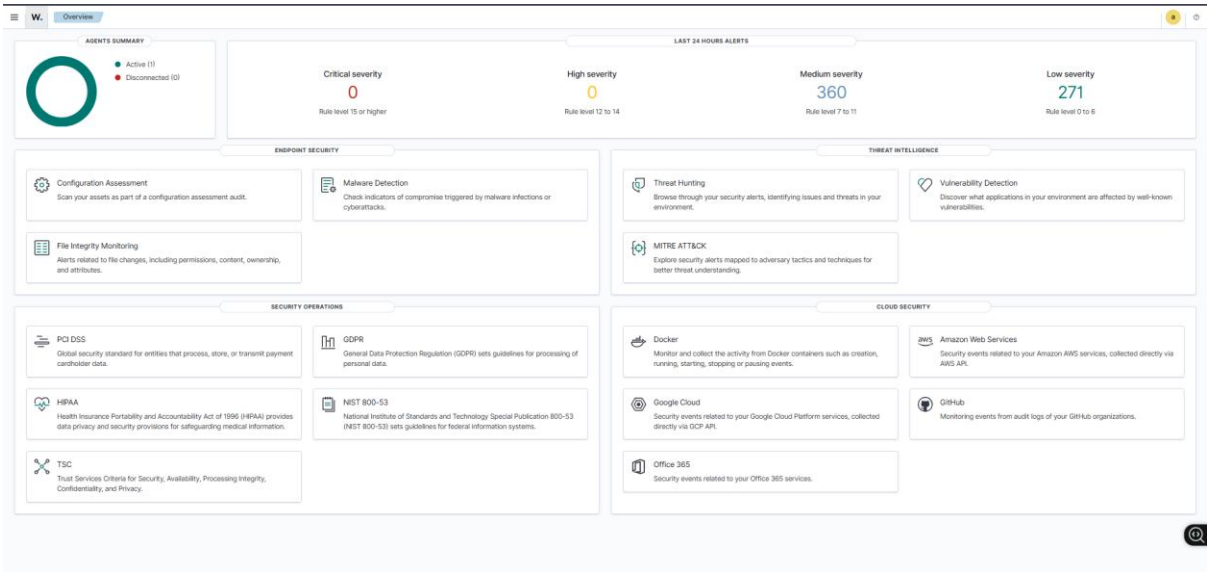


Figure 8 illustrates the detailed metadata extracted from the Windows host, including the specific Windows Event ID 4625 and the source agent IP address (192.168.254.165). This forensic depth allows SOC analysts to identify the exact origin and nature of a potential breach.

Global Security Posture

The cumulative impact of the security monitoring is visualized in the final Overview dashboard.

Figure 9: Security Events Overview dashboard illustrating the final operational status of the SOC lab.



As demonstrated in Figure 9, the SIEM maintains an 'Active' status for the enrolled agent and provides a summary of all recent alerts by severity. This consolidated view confirms that the lab infrastructure is fully operational and capable of providing comprehensive visibility across the network.

Conclusion and Lessons Learned

The implementation of this SOC Laboratory successfully demonstrated the deployment of an enterprise-grade SIEM solution within a virtualized environment. Through the three phases of this project, the primary objectives of centralizing security telemetry and establishing real-time threat detection were achieved.

1.) Technical Achievements

- Infrastructure Orchestration: Successfully provisioned a Linux-based security manager using bridged networking to allow cross-platform communication between virtualized and physical hosts.

- **Endpoint Integration:** Enrolled a Windows 11 host as a security sensor, ensuring the encrypted transmission of local event logs to the central analysis engine.
- **Incident Detection:** Validated the system's operational effectiveness by simulating a brute-force attack, which was successfully identified and categorized by the SIEM's rule engine.

2.) Key Learnings and Troubleshooting

- **Resource Management:** Operating the Wazuh stack on 3GB of RAM highlighted the importance of service orchestration. I learned that resource-intensive services like the Wazuh Indexer require careful sequencing during system reboots to prevent database failures.
- **Connectivity and Networking:** The project reinforced my understanding of bridged networking and the necessity of verifying API handshakes between the frontend dashboard and backend manager.
- **Forensic Analysis:** Through the simulation phase, I gained practical experience in analyzing Windows Event IDs (specifically ID 4625) and understanding how SIEM rules transform raw logs into actionable security alerts.

3.) Final Reflection

This project provided invaluable hands-on experience in the field of Cyber Security Operations. By building this lab from the ground up, I have developed a deeper understanding of how modern SOCs monitor for unauthorized access and the critical role that log aggregation plays in incident response.