Security Monitoring & SIEM Implementation Report

Author: ZtotheZ

This report showcases my firsthand experience in implementing a Security Information and Event Management (SIEM) solution using Elastic Stack. It highlights my ability to configure, monitor, and automate security event detection across Windows and Linux environments. The project involved real-world attack simulations, log enrichment, and automated response mechanisms using Python and PowerShell. Through this work, I have demonstrated proficiency in cybersecurity monitoring, threat detection, and incident response.

Project Overview

In today's evolving cyber landscape, **real-time monitoring and automated incident response** are crucial for defending against security threats. To strengthen security operations, a **SIEM-based monitoring system** was implemented using the **Elastic Stack (ELK)**. This system was designed to provide **continuous log collection, advanced threat detection, and automated response mechanisms**, ensuring robust security for both **Ubuntu and Windows environments**.

The project focused on **collecting and analysing security-related logs** from critical endpoints. By leveraging **Filebeat, Winlogbeat, and Sysmon**, logs from authentication events, system activity, and privilege escalations were ingested into **Elasticsearch**, allowing deep visibility into potential security threats.

To enhance situational awareness, **custom SIEM detection rules** were configured to identify **brute-force attacks, unauthorized access attempts, and suspicious script executions**. When threats were detected, **Slack notifications** were triggered in real time, allowing the security team to react immediately. Additionally, **Python and PowerShell automation scripts** were developed to **block attackers**, ensuring that the system could respond to incidents autonomously.

Through extensive **testing and validation**, the monitoring system successfully detected security events, provided instant alerts, and executed automated actions. The **Ubuntu firewall automation worked seamlessly**, while Windows security response policies were manually evaluated to explore further enhancements.

Building the SIEM Monitoring System

The **Elastic Stack (ELK)** provided the backbone for security monitoring, enabling **log collection, threat detection, and automated response**. Implementing an efficient **SIEM framework** required configuring **log collectors, dashboards, and security rules** to ensure real-time visibility into system activities.

Collecting Security Logs from Linux and Windows

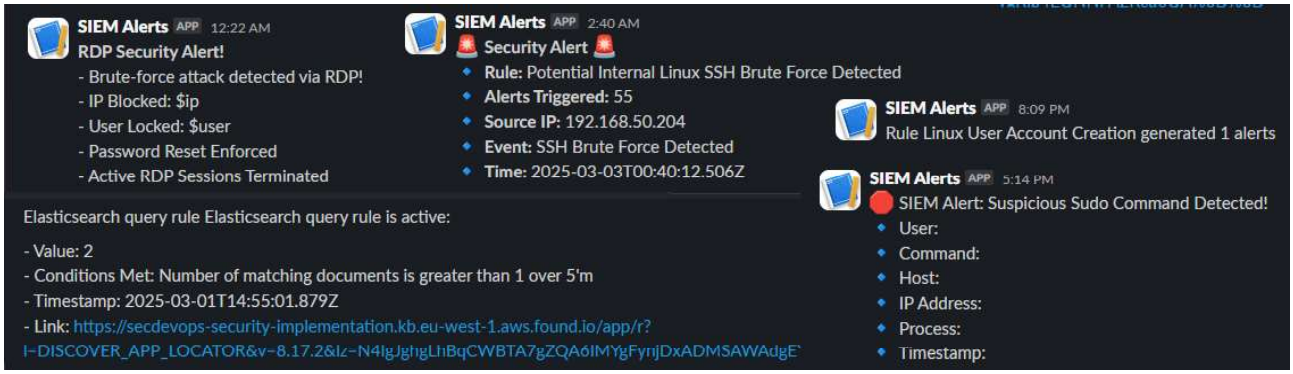
The first step in building this monitoring system was ensuring that security logs were **collected, structured, and indexed properly**.

On **Ubuntu systems**, Filebeat was configured to extract authentication and Sudo activity logs. These logs were critical for detecting **failed login attempts, unauthorized privilege escalations, and suspicious command executions**. The collected data was processed using **custom Grok parsers**, ensuring that logs were structured correctly before being stored in **Elasticsearch** for querying and visualization.

For **Windows environments**, a combination of **Winlogbeat and Sysmon** was used. These tools provided deep insight into **system activity, login events, PowerShell execution, and process behaviour**. Sysmon allowed tracking of **newly created processes, network connections, and registry modifications**, helping detect anomalies that could indicate **a potential compromise**.

Once logs were ingested into **Elasticsearch**, they were stored in structured indices, such as logs-windows.security-* and Filebeat-*, allowing SIEM rules to analyse patterns and detect suspicious behaviour effectively.

Figure 1: Real-Time Security Alert in Slack for SSH Brute-Force Detection and others



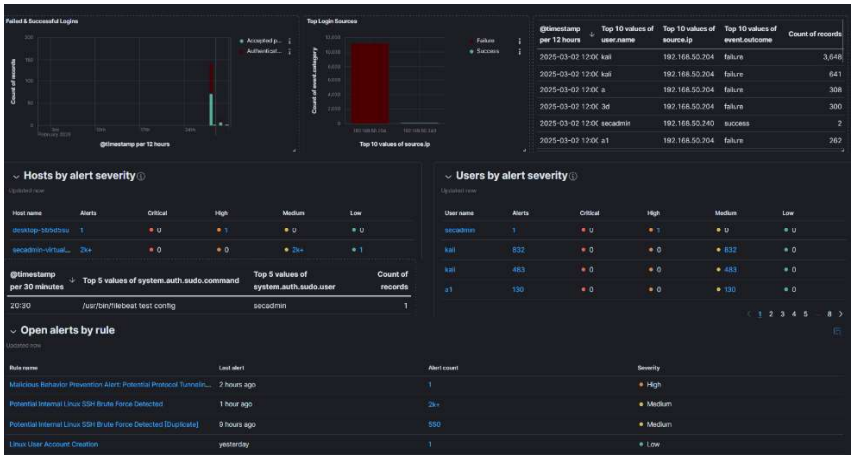
Visualizing Threats with Kibana Dashboards

To make sense of the collected data, **custom Kibana dashboards** were created to provide **real-time visualization of security events**. These dashboards allowed security teams to:

- Monitor **failed login tries across multiple endpoints**.
- Track **Sudo privilege escalation tries** on Linux machines.
- Detect **suspicious PowerShell executions** and **unauthorized network activity**.

One of the most valuable dashboards was the **Brute-Force Attack Monitoring Dashboard**, which correlated multiple failed logins tries across Windows and Linux environments. Another critical dashboard focused on **Sudo activity tracking**, highlighting instances where users executed potentially risky commands with elevated privileges.

Figure 2: Kibana Dashboard Showing Brute-Force SSH and other malicious attempts



Automated Incident Response: Blocking Threats in Real Time

While detecting threats is essential, **responding to them quickly is even more critical**. To ensure initiative-taking defence, **automated response mechanisms** were developed using **Python for Linux-based actions** and **PowerShell for Windows-based actions**.

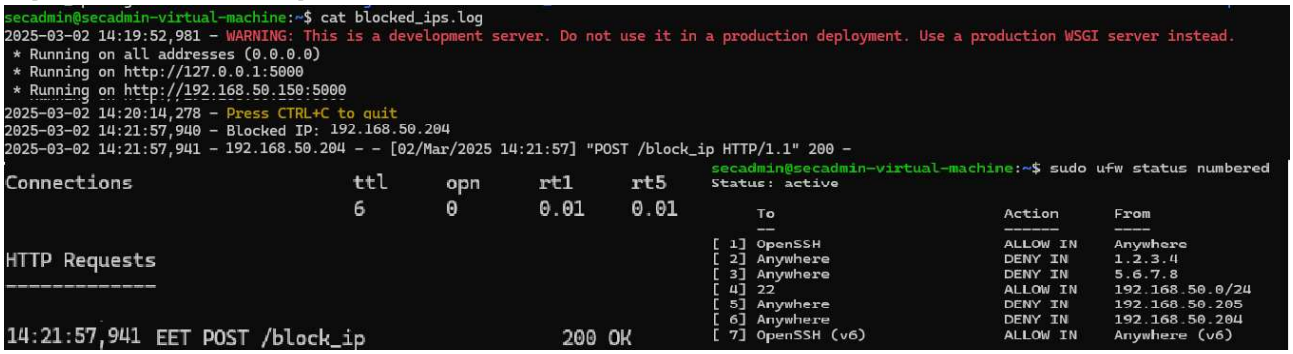
Linux Firewall Automation (Python Script)

On Ubuntu, a **Python script** was created to **automatically block IP addresses** of attackers trying brute-force SSH login attempts. Whenever the SIEM system detected multiple failed SSH logins, the script:

1. **Added the attacker's IP to the UFW (Uncomplicated Firewall) blocklist**, preventing further login attempts.
2. **Sent a Slack notification** to the security team, confirming that the IP had been blocked.

This automation ensured that attacks were mitigated instantly without requiring manual intervention.

Figure 3: UFW Blocking a Malicious IP After SIEM Alert



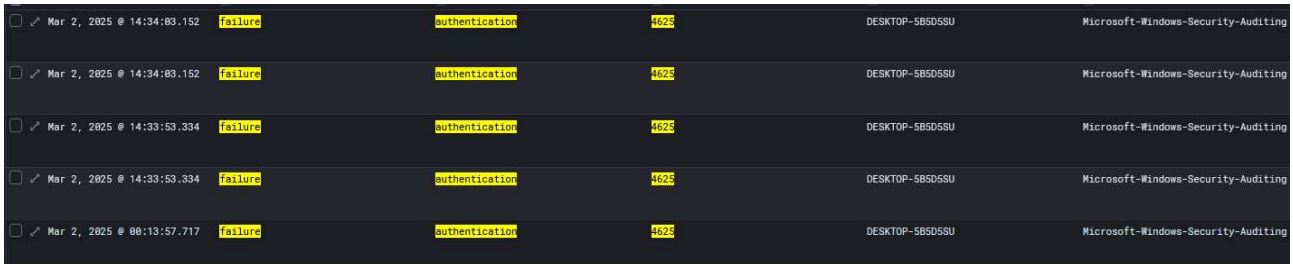
Windows Firewall Automation (PowerShell Script)

For Windows systems, **PowerShell scripts** were developed to:

- **Block attacker IPs in Windows Firewall** to prevent RDP brute-force attacks.
- **Lock compromised user accounts and enforced password resets**.
- **Terminate active RDP sessions** to disrupt potential lateral movement by attackers.

During testing, the **Windows Firewall blocking rules were confirmed manually**, ensuring they could be enforced effectively. Future iterations will refine automated enforcement of **account lockouts and session terminations**.

Figure 4: SIEM Alert Detecting RDP Brute-Force Attempts



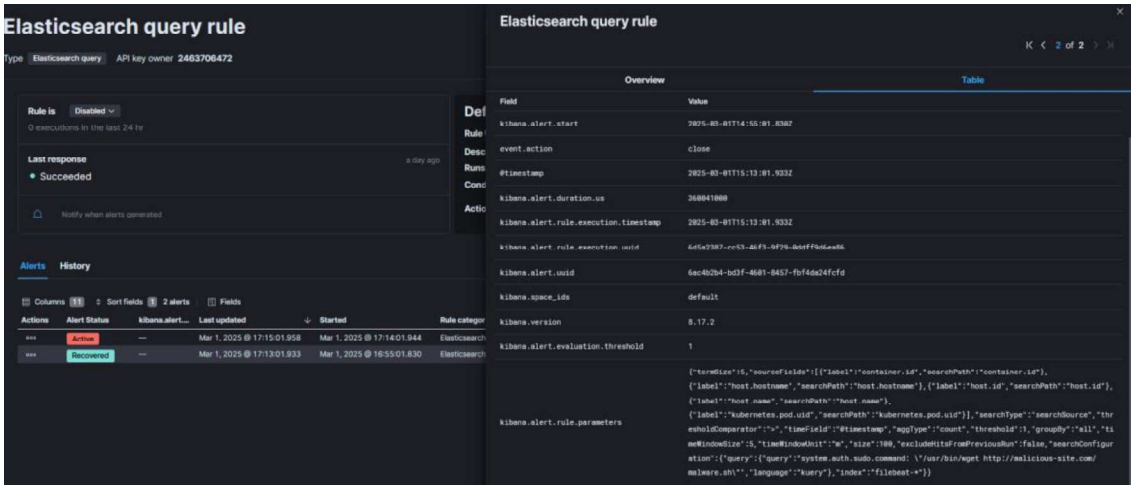
Attack Simulations & Testing

To confirm the effectiveness of **threat detection and automation**, controlled **attack simulations** were performed.

Simulated Linux Attacks

- **SSH Brute-Force Attack:** A Hydra brute-force attack was executed, successfully triggering SIEM alerts and automatically **blocking the attacker's IP** in UFW.
- **Privilege Escalation (Sudo su -):** SIEM correctly showed the unauthorized escalation attempt and sent **Slack alerts** for investigation.
- **Malicious Script Execution (wget)**: Attempts to download an external script were logged and flagged in Kibana.

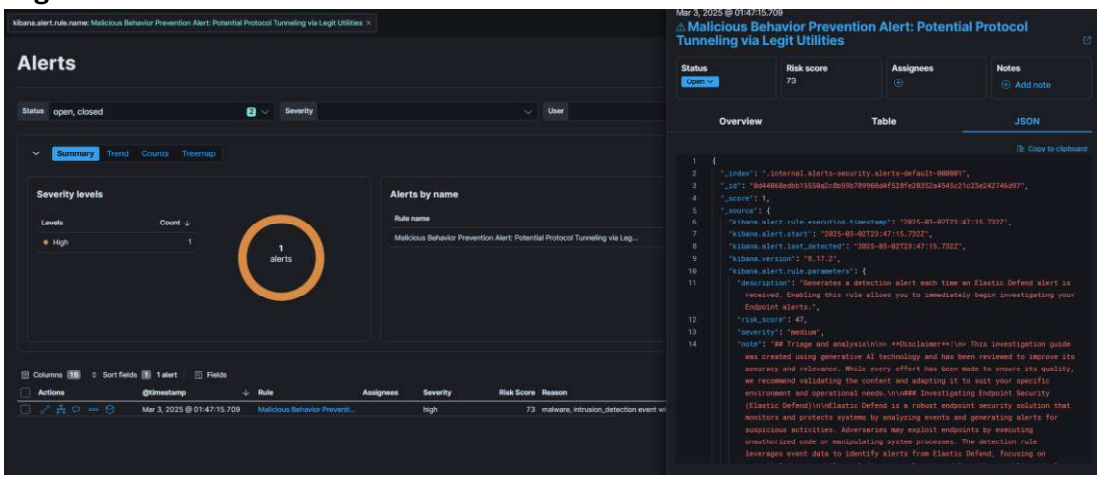
Figure 5: SIEM Alert Detecting Malicious Script Execution (wget)



Simulated Windows Attacks

- **RDP Brute-Force Attack:** SIEM successfully detected multiple failed login attempts, logging them for correlation with security policies.
- **Unauthorized PowerShell Execution:** PowerShell logs captured suspicious script activity, confirming SIEM's ability to **detect command-based attacks**.

Figure 6: PowerShell Execution Alert in SIEM



Conclusion & Future Improvements

This project successfully established a **real-time security monitoring framework** that integrates **SIEM (Elastic Stack), automation (Python, PowerShell), and event-driven alerting (Slack)** to detect and mitigate threats.

Key Takeaways:

- **Ubuntu firewall automation** successfully blocked threats in real-time.
- **Windows security policies were evaluated manually, with future enhancements planned for automated execution.**
- **SIEM alerting and detection** effectively identified brute-force attacks and privilege escalation attempts.

Future Enhancements:

- **Improving automated enforcement** for Windows security responses.
- **Integrating threat intelligence feeds** to enhance SIEM detection capabilities.
- **Expanding monitoring to cloud environments (AWS, Azure).**

This implementation showcases **expertise in SIEM engineering, security automation, and initiative-taking defence strategies**, demonstrating strong skills in **threat detection and response**.

Security-Monitoring-SIEM-Setup-Config-Files