Internal Network Threat Detection

SNA

Ahmed Baha Eddine Alimi
Egor Lazutkin
Yusuf Abdughafforzoda
Anvar Gelimov

May 2025

1 Goal/Tasks of the Project

Goal

This project establishes a scalable, containerized security monitoring environment for internal enterprise networks. Leveraging the Wazuh Security Information and Event Management (SIEM) platform integrated with the ELK Stack (Elasticsearch, Logstash, Kibana), our solution enables real-time log collection, threat detection, and visual analytics to identify and respond to insider and external attack vectors.

Responsibilities

- **Ahmed Baha Eddine Alimi & Egor Lazutkin:** Wazuh architecture, custom detection rule development, SSH monitoring, DevSecOps pipeline design.
- **Yusuf Abdughafforzoda & Anvar Gelimov:** Docker Compose orchestration, ELK Stack configuration, report compilation, demo video.

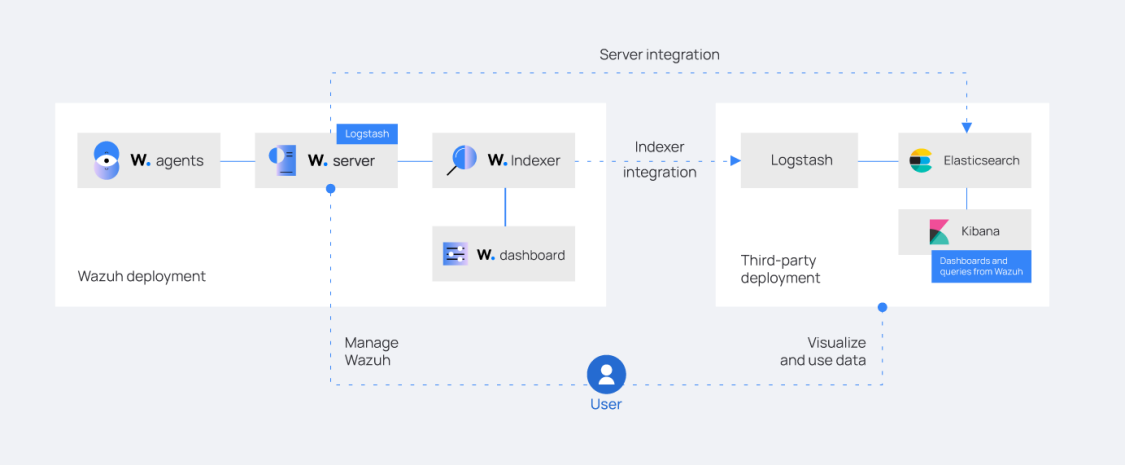
2 Execution Plan/Methodology

Plan for the Solution

1. **Requirement Analysis:** Catalog internal threat scenarios (e.g. SSH brute-force, port scans) and define monitoring objectives.
2. **Architecture Design:** Architect a microservices-based stack combining Wazuh manager, Elasticsearch nodes, Logstash pipelines, and Kibana interfaces.
3. **Environment Setup:** Deploy Wazuh agents and ELK services using Docker Compose for consistent, portable environments.
4. **Threat Simulation:** Develop automated scripts to simulate SSH brute-force attacks and nmap-based reconnaissance at configurable volumes.
5. **Monitoring Implementation:** configure Logstash filters, and build Kibana alerts and dashboards for actionable insights.

Planned Infrastructure

- **Wazuh SIEM:** Central threat detection with endpoint visibility, integrity monitoring, and alert orchestration.
- **ELK Stack:** Scalable log indexing (Elasticsearch), pipeline-based parsing (Logstash), and interactive dashboards (Kibana).
- **Docker Compose:** Single-command orchestration of all components, ensuring environment parity between development and testing.
- **CI/CD Pipeline:** Automated builds, vulnerability scanning (Trivy), and continuous deployment to streamline updates and enforce security policies.



- Agents collect system logs and forward them to the Wazuh Manager.
- Manager applies detection rules and forwards events to Logstash.
- Logstash parses and enriches logs for Elasticsearch.
- Elasticsearch stores and indexes events. Kibana visualizes data.

3 Development of Solution/Tests as the PoC

Explanation of Solution

- **Wazuh Implementation:** Installed Wazuh manager and agents, fine-tuned detection rules for SSH brute-force and anomalous logins, and integrated with AlertManager.
- **ELK Integration:** Created Logstash configuration files, set Elasticsearch index templates, and developed Kibana visualizations for event trends.
- **Threat Simulation:** Utilized Hydra for SSH brute-force tests and nmap scripts for port scanning; parameterized scripts to generate varying traffic patterns.
- **Containerization:** Defined Docker Compose YAML with healthchecks, volume mounts, and network isolation for each service.
- **DevSecOps:** Embedded Trivy vulnerability scanning in the CI/CD workflow, ensuring container images meet security benchmarks prior to deployment.

Proof of Concept (PoC)

- Successfully detected simulated SSH brute-force attacks with Wazuh
- Kibana dashboards displayed real-time attack patterns
- Docker Compose deployment succeeded with automated health monitoring
- Implemented CI/CD pipeline with security scanning

4 Difficulties Faced, New Skills Acquired

Difficulties Faced

1. **Complex Logstash Configuration:** Crafting filter chains to parse multi-source logs without dropping critical fields.
2. **Alert Tuning:** Balancing sensitivity to avoid alert fatigue while capturing low-frequency anomalies.
3. **Resource Optimization:** Managing Elasticsearch heap and Kibana memory overhead to maintain performance under load.
4. **Realistic Threat Emulation:** Ensuring attack scripts mimic genuine adversary behavior without destabilizing the environment.

New Skills Acquired

- **Custom Wazuh Rule Development:** Writing XML-based rules and decoders for nuanced threat detection.
- **Advanced ELK Management:** Designing index lifecycle policies and enriching logs with metadata.
- **Docker Security Best Practices:** Implementing minimal base images, non-root containers, and immutable tags.
- **CI/CD Security Integration:** Automating vulnerability assessments and compliance checks in GitLab CI/CD.
- **Threat Simulation Engineering:** Parameterizing attack workflows and capturing forensic logs for analysis.

5 Conclusion

The Internal Network Threat Detection project validates the effectiveness of a Wazuh-driven SIEM integrated with the ELK Stack in detecting and visualizing internal threat activities. Our containerized deployment ensures rapid provisioning and consistent testing, while the DevSecOps pipeline enforces security standards throughout development. Future work will expand threat coverage to include cloud-native logs, integrate network-level IDS (e.g., Zeek), and implement automated remediation playbooks for end-to-end security orchestration.

Repository: `https://github.com/3llimi/ThreatHound`
Demo Video: Link