Supplementary Technical Report

Tool Constraints, Substitutions & Implementation Decisions

1. Overview

The original task objective included the implementation of Wazuh as the SIEM solution and the use of MITRE Caldera for adversary emulation. However, due to infrastructure constraints, architectural limitations, and compatibility issues within the lab environment, certain adjustments were made.

This report documents the technical reasoning behind those decisions and demonstrates that the learning objectives were still successfully achieved.

2. Wazuh Deployment Limitation & Elastic Stack Substitution

2.1 Initial Objective

The initial plan was to deploy:

- Wazuh Manager
- Wazuh Agent
- Filebeat
- Elasticsearch
- Kibana

This would provide full Wazuh-based detection, alerting, and rule-based monitoring.

2.2 Technical Constraint Encountered

The lab environment was running on:

- macOS host
- Ubuntu VM (ARM64 architecture)
- 4GB RAM allocation

During deployment, the following issues were observed:

- Memory utilization exceeded 75% when Elasticsearch + Kibana + Wazuh Manager were active.
- System entered read-only filesystem state due to resource exhaustion.
- dpkg operations failed with "Read-only file system" errors.
- NodeJS and Go version conflicts impacted Caldera build.
- Multiple services competed for limited RAM (Wazuh + Elastic + Caldera).

Wazuh requires:

- Stable 4GB+ RAM
- Consistent disk write access
- Proper kernel support

Given the constrained virtual environment, running Wazuh alongside Elastic Stack and other tools caused instability.

2.3 Decision Justification

Since Wazuh internally relies on Elasticsearch and Kibana for indexing and visualization, the core detection functionality could still be achieved using:

- Filebeat (log shipper)
- Elasticsearch (log storage & indexing)
- Kibana (visualization & detection)

Instead of rule-based Wazuh alerts, detection was achieved via:

- SSH log monitoring
- Failed login pattern analysis
- Lens visualization
- Timestamp correlation

This preserved the core SIEM objective:
Log ingestion → Detection → Visualization → Incident Validation

Therefore, Elastic Stack was used as the primary SIEM implementation.

3. MITRE Caldera Partial Deployment

3.1 Initial Objective

Caldera was intended to simulate adversary TTPs using:

- Sandcat agents
- Stockpile plugin
- MITRE ATT&CK mapping

3.2 Technical Challenges Encountered

The following constraints were observed:

- ARM64 architecture caused Sandcat binary execution errors.
- Reverse agent payload returned “Exec format error”.
- NodeJS version incompatibility during UI build.
- Go version requirement not met initially.
- Docker module dependency missing.
- SSH tunnel dependency failure due to temporary directory errors.

Although Caldera server started successfully and UI was accessible, full adversary emulation (agent deployment and ability execution) could not be completed due to:

- Architecture mismatch (aarch64 vs x86 expectation)
- VM resource limitations
- Dependency conflicts

3.3 Outcome

Caldera was:

- Successfully installed
- Vue UI built
- Server started
- Web interface accessible

However, adversary ability execution was not fully demonstrated due to architecture and dependency limitations.

4. Learning Objectives Still Achieved

Even though Wazuh and Caldera were partially limited, the core SOC lifecycle was fully implemented:

✔ Reconnaissance

- Nmap service enumeration
- Service exposure identification (Samba 139/445)

✔ Exploitation

- Metasploit exploit: exploit/multi/samba/usermap_script
- Reverse shell obtained
- Root access verified

✔ Detection

- SSH brute-force detected via Elastic
- Log ingestion verified
- Visualization created

✔ Containment

- CrowdSec ban applied
- Ban verified via decision list

✔ SOAR / Case Management

- TheHive case created
- Observables added
- MITRE mapping documented
- Tasks created

✔ Metrics

- MTTD calculated
- MTTR calculated
- Dwell Time documented

Thus, the operational workflow was successfully demonstrated.

5. Professional Reflection

In real-world SOC environments, infrastructure constraints, version incompatibilities, and resource limitations frequently require adaptive decision-making.

This project demonstrates:

- Technical troubleshooting ability
- Tool substitution decision-making
- Architecture awareness (ARM64 vs x86)
- Resource impact analysis
- Prioritization of core objectives over tool dependency

6. Conclusion

While Wazuh and full Caldera adversary execution could not be completely implemented due to VM resource limitations and architecture compatibility issues, the core detection, exploitation, containment, and response objectives of the project were achieved using Elastic Stack, Metasploit, CrowdSec, and TheHive.