Bastion Implementation and Evaluation of an AI Assisted Endpoint Security Monitoring Solution

Using Wazuh Microsoft Sysmon and Locally Hosted AI Analysis

Enes Oguz
Cybersecurity Graduate Capstone Task 2

Project Scope Clarification

The capstone does not entail the development of any technology for EDR system, SIEM system, or AI. The capstone entails the use of already available technology that has been incorporated into working within a laboratory environment in the enterprise environment. The terms "Bastion," "Pawn," and "Rook" only refer to the various components of the architecture.

Architecture Label	Implemented Technology	Role in the Project
Bastion	Wazuh central components: server, indexer, and dashboard	Collects and centralizes logs, creates alerts, supports search and dashboards, manages agents, controls access, and supports event investigation.
Pawn	Wazuh Agent with Microsoft Sysmon on Windows; Wazuh Agent/native logs on Linux	Log collection and consolidation, alert generation, searching and dashboards, agent management, access control, and incident investigation.
Rook	Ollama runtime with a locally hosted model and a controlled alert-analysis workflow	Reviews selected Wazuh alerts and summarizes the evidence, suggests a severity, and recommends investigation steps. It does not make changes to endpoints on its own.

Research Question: How does centralized endpoint telemetry and AI-assisted analysis of security events by leveraging existing security solutions benefit threat detection and incident response in a controlled enterprise setting?

A. Security Problem Under Investigation

The problem under consideration involves the absence of visibility at the end-point level and the degree of manual labor required for security event analysis. Any organization will have computers, servers, accounts, applications, and network services that produce a lot of logs and security events all the time. The fact that such events stay inside systems that need to be checked requires the analyst to look through many systems to determine if it was a regular event or not. Can also make it harder to build a complete timeline of what happened. CISA recommends enabling logs on endpoints, servers, firewalls, applications and cloud services then centralizing those logs and creating alerts for higher‑risk events such as failed logins and privilege escalation (CISA n.d.). NIST SP 800‑61 Rev. 3 Also treats detection and response as

part of the larger cybersecurity risk‑management process. Focuses on making those activities more effective and efficient (Nelson et al. 2025). For this project the solution will use products that already exist. Wazuh will serve as the XDR/SIEM platform Microsoft Sysmon will add more detailed Windows endpoint data and a locally hosted AI workflow will help review selected alerts. The goal is to see whether this combination gives visibility, faster detection quicker analysis and more useful evidence, than the baseline environment.

A1. Importance of the Security Problem, Background, and Environment

Incidents pertaining to security could be easier to understand in context. The failure of one login attempt is not alarming; however, a series of such events, together with the attempt at a privileged login, making an extra account, unusual behavior within PowerShell, and an out-of-the-ordinary network connection could mean a lot more. Without having a holistic perspective of the situation, the analyst would have to review multiple logs. The implementation of a centralized system will facilitate searching and comparing related activities. Centralized logging is promoted by the CISA due to its ability to detect any anomalies and provide for automation where possible. The NIST CSF 2.0 framework categorizes cybersecurity strategies into classes which include, but are not restricted to, Govern, Identify, Protect, Detect, Respond, and Recover. The existing project falls under the following NIST CSF 2.0 categories: Identify, Detect, Respond, and Govern. The existing project involves asset management, event detection and evaluation, incident response, and security metrics (NIST, 2024). The project will run in an isolated enterprise-style lab. The lab will include at least one Windows endpoint, one Linux endpoint, one additional endpoint or server, and a Linux server that hosts the Wazuh central components. Sysmon will be enabled on the Windows endpoint to collect more detailed process and selected network activity. Wazuh agents will send endpoint data to the Wazuh server and indexer, and the Wazuh dashboard will be used to search events, review alerts, manage agents, and view dashboards. Rook will only receive selected alert evidence for AI-assisted review. This lab will not deploy any real malware nor attack any systems except the authorized environment. Lab tests allow for the repetition of tests while leaving production systems untouched. They can even be run both before and after the deployment in order to compare their performance by metrics like detection rate, time of alert, analysis time, completeness of evidence and behavior of false positives.

A2. Documentation Demonstrating the Need for the Solution

NIST Cybersecurity Framework 2.0

NIST CSF 2.0 can be used by organizations of any size or industry. It gives a way to describe cybersecurity outcomes. It supports the parts of this project that deal with asset awareness, event detection, incident response and governance. Because the framework is based on outcomes it also lets the project measure improvement. It does not suggest that one product automatically creates compliance. NIST CSF 2.0 is, from NIST, 2024.

NIST SP 800-61 Revision 3

NIST SP 800-61 Rev. 3 Incident response is set in the context of the risk-management approach to cybersecurity. This tool's goal is to assist in detecting, responding to, and recovering from cyber incidents

in organizations. This model encourages the utilization of detection time, analysis time, availability of evidence, and performance of incident workflow metrics (Nelson et al. 2025).

CISA Logging Guidance

CISA recommends deciding what should be logged. CISA says to enable those logs across systems. CISA advises to centralize them. CISA suggests creating alerts, for higher-risk events. CISA recommends using automated review where. CISA encourages training staff to recognize behavior. CISA recommendations line up directly with this projects use of endpoint data, automated alerting and analyst training (CISA n.d.). I think CISAs recommendations fit perfectly with our project.

Wazuh Vendor Documentation

Wazuh calls its product a free and open-source security platform that has XDR and SIEM features. The way it works includes an endpoint agent and also a Wazuh server, an indexer and a dashboard. The dashboard can be used to look into alerts, control agents make rules and charts create reports set up access based on roles and see MITRE

ATT&CK mappings. Because these features are already part of the product this project can work on putting them into action and checking how well they perform of trying to create a new SIEM as a test. (Wazuh, 2026a; Wazuh, 2026b)

Microsoft Sysmon Documentation

Microsoft describes Sysmon as a Windows service and driver that records activity in the Windows event log. It can record process creation, command-line details, hashes identifiers used for correlation and optional network connections. Sysmon does not decide if an event is malicious on its own. The data is meant to be reviewed by another system, such as a SIEM. That makes it a good telemetry source for Sysmon, in this project.

MITRE ATT&CK

MITRE ATT&CK is an used knowledge base that contains adversary tactics and techniques. I will use selected MITRE ATT&CK techniques to choose and label the test scenarios so that the tests show recognized attack behavior. Wazuh already includes MITRE mappings, for alerts so the project can use the vendors existing mapping instead of making a new classification system (MITRE, 2026; Wazuh, 2026c).

NIST AI Risk Management Framework

Rook is only meant to help the analyst so the artificial intelligence part of the project will follow rules, from the NIST AI RMF and the NIST AI 600-1 Generative AI Profile. In life that means a person will look at the results the original evidence will stay available any doubts will be written down and any AI statements that are not supported will be considered mistakes (Autio et al., 2024).

A3 Root Causes of the Problem

The environment described in A1 is an enterprise-style lab that contains at least one Windows endpoint one Linux endpoint, one additional endpoint or server and a Linux server that hosts the Wazuh central components. The baseline comparison uses logs and a manual review before the Wazuh system the Sysmon tool and the Rook platform are deployed. This design creates gaps in collection gaps in evidence detail and gaps in the analysis workflow. The causes listed below are tied to that baseline and to the activities in H1. Industry guidance supports the need, for these controls. It does not establish that a particular failure has already occurred in this lab. Baseline testing will. Reject each proposed cause before the effect is reported as a finding.

1 Siloed security logs

Windows event logs and Linux authentication and system logs reside on separate lab hosts in the baseline workflow. The analyst must retrieve each source separately to compare the repeated login failures and test account creation in H1 scenarios 1 and 2. The absence of a shared collection and search point is the underlying cause of fragmented visibility. NIST SP 800-92 supports establishing enterprise log management infrastructure and processes (Kent & Souppaya, 2006). Baseline evidence will include the host inventory, log locations, and a record of the retrieval steps and time required for both scenarios. The Sysmon Ingestion and Linux Log Ingestion formative tests will later verify centralized access to those sources.

2 Insufficient endpoint telemetry

The Windows PowerShell and network discovery tests in H1 scenarios 3 and 4 require process, command-line, host, time, and applicable network context. The baseline may lack these fields if the required auditing is not enabled. This configuration gap, rather than the mere use of Windows, is the proposed cause. Microsoft documents Sysmon process creation fields and optional network connection logging; network logging must be explicitly configured (Microsoft, 2026). Baseline evidence will include the active audit configuration and source

event exports showing which required fields are present or absent. The Sysmon Ingestion formative test will compare the same fields after deployment.

3 Event volume relative to manual review capacity

Standard logins, application launches, browsing, file creation, and approved administration in H1 scenario 6 will generate the benign background used for the comparison. Without filtering and with no review queue the analyst has to look through the background to find the seeded test activities. A workload bottleneck is a reason only if the number of events measured and the time spent reviewing back it up; this idea doesn't assume large-scale operations, in a small lab. Evidence will include event counts for a fixed window, the same benign workload before and after deployment, and recorded review time. NIST SP 800-92 provides the log management basis for organizing collection and analysis (Kent & Souppaya, 2006).

4 Manual event correlation

Authentication, account creation, process execution and network activity can show up in sources even when they belong to the same controlled test sequence. In the baseline the analyst must combine these records by host, user, timestamp and any available process identifiers. The absence of a common search workflow is the proposed cause of correlation effort. Baseline evidence will be a manually assembled scenario timeline with its source records and elapsed preparation time. Post-deployment evidence will be the equivalent Wazuh search and timeline. Sysmon process GUIDs provide an additional documented correlation field where applicable (Microsoft, 2026).

5 Dependence on manual detection

The baseline evaluation relies on source-log inspection rather than the proposed centralized Wazuh alert workflow. For repeated authentication failures in H1 scenario 1, the proposed cause of delay is the interval between the event occurring and the analyst reviewing its evidence. Baseline evidence will include the event start time, the time the analyst first identifies the activity, and any native alert already available. Existing endpoint alerts will be recorded rather than assumed absent. Detection Rule Validation and the timed summative scenarios will test whether configured Wazuh rules reduce this interval. This comparison follows the incident detection and analysis focus of NIST SP 800-61 Rev. 3 (Nelson et al., 2025).

6 No shared prioritization procedure in the baseline

Baseline must show how the analyst sets severity for login failures, account creation and routine administrative activity. If no documented decision criteria exist differences in classification may come from the review process of from the evidence itself. The project uses the analyst and administrator roles described in B. It does not assume a SOC. Two analysts have already disagreed. The evidence will include the baseline severity rationale and a comparison, with a predefined scenario classification rubric. Detection Rule Validation will examine the configured Wazuh rule levels; Rook Evidence Test will examine whether the advisory output remains consistent, with the supplied facts. The analyst keeps the decision (Nelson et al. 2025; Autio et al., 2024).

7 Incomplete investigation context

A record of a failed login does not by itself explain associated account changes or process execution. In this lab, missing host, user, time, command-line, or applicable network fields would prevent a complete explanation of H1 scenarios 1 through 4. This is a field availability and retrieval problem that baseline event exports will verify. An evidence checklist will distinguish fields absent at the source from fields present locally but unavailable in the analyst workflow. The Sysmon Ingestion and Linux Log Ingestion formative tests will verify collection; the summative evidence-completeness measure will verify investigation usefulness (Microsoft, 2026; Kent & Souppaya, 2006).

8 Repeated manual analysis tasks

In the baseline the project analyst must read source records summarize the activity assign a classification and identify follow-up checks. Repeating these steps for the supported alert types can raise analysis time. This proposed cause will be examined by timed reviews of the same evidence that will later be given to Rook. The retained evidence will contain analyst notes start and finish times source event identifiers and the final classification. Rook Evidence Test and the final reviews will evaluate accuracy and traceability separate from speed in line with the human review and error management approach in the NIST Generative AI Profile (Autio et al., 2024). The baseline evidence register will connect each cause to its host, scenario, configuration record, event export and measurement. A proposed cause that is not supported by these records will be marked as unconfirmed. No improvement will be credited to it. This keeps the documented limitations of the baseline design, from the results that still need to be measured.

B. Project Stakeholders

Stakeholder	**Implementation Involvement and Needs**	**Effect of Security Problem**	**Influence on Objectives and Outcomes**
Cybersecurity Analyst / SOC Analyst	Describes application scenarios for detection, analyzes Wazuh alerts, examines Rook findings, and confirms the evidence. Needs searchible data, reasonable policies, severity levels, and a reasonable number of false positives.	Adds another delay to investigation and the danger of missing some relevant information. Presents an assessment of how useful the alerts, dashboards, and AI descriptions are for practical purposes.	Helps decide whether the notifications, dashboards, and AI analysis are good enough for practical security use.
System Administrator	Installs agents, enables Sysmon, manages endpoint configuration, and validates resource use. Needs stable agents, documented deployment, and limited endpoint overhead.	Troubleshooting and issues are contributors to administrative overhead because there is inadequate endpoint data.	Has impacts on deployment process, privileges, compatibility, maintenance, and performance requirements.
Network Administrator	Configures the isolated network, firewall, name resolution, and allowed communication between agents and Wazuh.	Low visibility of network can affect investigation of scans and connection analysis.	Affects segmentation, ports, communication, and networking test cases.
Information Security Manager / Sponsor	Determines the importance of the risk and the expected risk acceptance.	Detection being late and incomplete creates operational and security risks.	Approves goals, thresholds, risk management, and the final acceptance of the solution.
End User	Uses monitored endpoints during benign-activity testing. Needs acceptable system performance and appropriate privacy boundaries.	Endpoint compromise can affect user productivity and organizational data.	User impact influences telemetry scope and acceptable Pawn performance overhead.

Stakeholder	Implementation Involvement and Needs	Effect of Security Problem	Influence on Objectives and Outcomes
Project Security Engineer	Installation of Wazuh, Sysmon, and local AI flow as well as rules for detection, testing, documentation, and training. Needs a stable lab environment, well-defined requirements, and constant testing.	Without a baseline, there is no way of proving the improvement. Influences the quality of the implementation, testing repeatability, evidence gathering, and the project timeline.	Directly affects implementation quality, repeatability of the tests, evidence collection, and the project schedule.
Wazuh Project / Vendor	Provides the documented XDR/SIEM platform, updates, installation guidance, and technical documentation.	The organization depends on the vendor platform to provide supported capabilities; the security problem itself does not directly affect the vendor.	Product architecture, supported operating systems, APIs, rules, and licensing constrain the design.
Microsoft Sysinternals	Provides Sysmon and its documented telemetry capabilities.	The problem does not directly affect Microsoft, but Sysmon capability determines available Windows evidence.	Sysmon event design and licensing influence endpoint data collection.
Ollama / Model Provider	Provides the local model runtime and selected model used by Rook.	The problem does not directly affect the provider, but model behavior introduces reliability and licensing considerations.	Runtime capabilities, model license, resource requirements, and output quality influence Rook design.

C. Historical Data Used to Support Project Decision-Making

The project will use two kinds of evidence for decision-making. The first is published industry and vendor documentation showing that the security problem and the selected technologies are established.The second type of data is the baseline that is to be gathered from the lab prior to full integration of the Wazuh centralized logging system.

External Evidence Used as Historical/Reference Data

CISA recommendations for log management and significance of possessing, gathering, analysis, and alerting from security logs. NIST CSF 2.0 and NIST SP 800-61 Rev. 3 recommendations concerning objectives of detection, response, and continuous improvement in cybersecurity.

Wazuh documentation on the capabilities like XDR/SIEM, agent, dashboard, RBAC, rules, events analysis and MITRE ATT&CK. Microsoft Sysmon documentation on the capabilities like process, hash, session and optionally network telemetry. MITRE ATT&CK techniques documentation on real-world and standardized security scenario selection.

Pre-Implementation Baseline Data

I will record a baseline in the same lab before I use centralized Wazuh alerting and investigation. This is not a made up history. It is a pre-implementation measure documented and can be directly compared with the later summative tests.

- Windows Security and System logs of events for successful and failed authentication and account changes, service activity, and other relevant operating system events.
- Sysmon source events Sysmon is enabled for telemetry validation including process creation and configured network events.
- Linux authentication and system logs for login activity and relevant service events.
- Wireshark captures used selectively to validate network activity during controlled scenarios.
- Nmap discovery results for documenting hosts and services inside the authorized lab.
- Manual investigation times showing how long it takes to find source evidence before centralized Wazuh investigation is available.
- Baseline CPU, memory, disk, and network utilization prior to agent deployment for comparison with post-deployment overhead.

Baseline Metrics

Metric	**Use in Decision-Making**
Mean Time to Detect (MTTD)	Elapsed time from the start of a controlled security event until it is identified.
Mean Time to Analyze (MTTA)	Elapsed time from identification/alerting until a preliminary analyst classification is reached.
Detection Coverage	Percentage of predefined scenarios for which the required behavior is detected.
Evidence Completeness	Percentage of required evidence fields available for the test scenario.
Endpoint Visibility	Percentage of in-scope endpoints actively reporting the expected telemetry.
False-Positive Rate	Percentage of executed benign test cases incorrectly flagged as suspicious.
Endpoint Overhead	Difference in average CPU and memory use before and after Wazuh Agent/Sysmon monitoring is enabled.

The baseline can answer many questions about setting up things before the final testing process gets going. For example, it will show whether there is a need to tune detection rules, whether Sysmon is too noisy, whether the endpoint overhead is acceptable, whether Rook is really saving analyst time.

D. Project Requirements

The solution will make use of supported product capabilities wherever feasible. Custom efforts will be restricted to configuration, tuning, reporting, documentation, and a minor integration which passes selected pieces of Wazuh alerts into an AI runtime environment. The objective of the solution is not to duplicate the effort by creating a new Wazuh endpoint agent or AI model.

Functional Requirements

- Enrollment and Identification of all Endpoints Relevant to the Security Scenarios within Wazuh.
- Collection of Windows and Linux security related logs using Wazuh Agents.
- Collection of Sysmon Events on Windows based Endpoints according to Configuration.
- Centralization of Endpoint Telemetry into the Wazuh Platform.
- Availability of Searchable Historical Security Events and Endpoint Inventory.
- Creation of Alerts according to Wazuh Rules for the Defined Security Scenarios.
- Mapping of Relevant Detections in MITRE ATT&CK Framework if Wazuh allows it.
- Dashboards for Endpoint Status, Alerts and Event Investigation.

- Role Based Access Control for Wazuh Dashboard and Administration.
- Forwarding of Selected Evidence Related to Alerts to Rook Workflow.
- Generation of Rook Outputs that Include Event Summary, Severity Suggestion, Explanation based on Evidence, and What's Next in Investigation.
- Retention of Original Wazuh Evidence to Allow Analysts to Validate all Suggestions by Rook.
- Prevention of Rook from Running Automatic Actions on Endpoints.

Security and Nonfunctional requirement

- Use Wazuh supported encryption to encrypt in transit communication.
- Provide least privilege access to dashboard and management controls.
- Protect the central logs from being tampered with by normal endpoints.
- Make sure that all systems in the lab are set to the system time.
- Telemetry shall be limited to what is necessary for the security use cases.
- Document log retention and log removal process for test logs
- Make sure the test environment is isolated from any unauthorized external target.
- Good endpoint performance and low resources consumption on Wazuh server
- Document AI limitations, model/license selection and human review.

D1. Industry-Standard Methodologies Guiding Design and Development

NIST Cybersecurity Framework 2.0

NIST CSF 2.0 will be the main framework for risk management of this project. Govern will address accountability and reporting; Identify will address visibility of assets; Protect will address access control and protection of telemetry; Detect will address monitoring and alerts; Respond will address investigation and escalation; and Recover will assist in lessons learned and maintenance. The CSF being outcome driven makes it an ideal approach to assessing an integrated solution, without suggesting that Wazuh alone provides compliance.

NIST SP 800-61 Rev. 3

NIST SP 800-61 Rev. 3 will be used to guide the incident response phase of this project. This includes activities that relate to preparation, detection, analysis, response, reporting, and improvement. The measurements of the MTTD, MTTA, evidence, and workflow in the project will show if these activities improve their effectiveness.

NIST SP 800-92 Log Management Guidance

The NIST SP 800-92 standard will be utilized for decision-making with respect to the log generation, transmission, centralized storage, access, examination, retention and disposal process. The document is somewhat outdated compared to the tools used in this work, but the fundamental principles of log management are relevant to our problem.

MITRE ATT&CK

The MITRE ATT&CK framework will be applied to choose and classify the security scenarios. Examples of such planned security scenarios are: Brute Force (T1110), Create Account (T1136), PowerShell (T1059.001), and Network Service Discovery (T1046). In case of the existing ATT&CK mapping for Wazuh, it will be used instead of developing another taxonomy for the current project.

NIST AI RMF and NIST AI 600-1

Rook will be considered as an advisory system, and not as a decision-making machine. Evidence will continue to be accessible, a person will analyze the results, the unsupported claims will be classified as mistakes, and Rook will

not be permitted to perform important actions by itself. All this follows from the risk management strategy described in NIST AI RMF.

NIST Secure Software Development Framework

For any case where there may be the need for a small script to transfer selected alerts to the local model, the following NIST SSDF practices shall be applied: dependency management, input validation, secrets management, security testing, and change review. The application of these practices is limited to the custom code developed.

D2. Project Launch, Rollout Phases, Completion Criteria, and Project Management Methodology

This project will follow a series of six two-week sprints using a process flow following the Agile/Scrum approach. Every sprint will be followed by a checkpoint so I can verify that everything works in the current part of the solution. The process is important for the right order – collect the baseline, deploy Wazuh, add endpoint telemetry, detections, Rook, and finally run tests.

Phase	Major Activities	Exit Criteria
Phase 1 – Requirements and Baseline	Document lab environment, current logging, security use cases, test scenarios, baseline network topology, and pre-deployment performance metrics.	Requirements approved; baseline documentation; key performance indicator definitions.
Phase 2 – Wazuh Central Deployment	Install and secure Wazuh manager, indexer, and dashboard on an approved version of Linux. Create user/roles and verify access to the dashboard.	Wazuh central services working; dashboard access; no critical installation errors.
Phase 3 – Endpoint Telemetry Rollout	Deploy Wazuh Agent to a test endpoint, enable Sysmon on a Windows endpoint, validate the telemetry, and then deploy agents to other endpoints.	Deployment of 100% of endpoints scoped; telemetry validation completed; minimal performance impact.
Phase 4 – Detection Engineering and Dashboards	Tune detection rules for selected scenarios, verify ATT&CK mappings, design dashboards/queries, and eliminate known false positives.	All proposed test scenarios have rules/queries validated for known positive and negative tests.
Phase 5 – Rook Integration	Configure approved Rook model locally; design alert processing workflow; maintain links to alerts; require human review.	Rook analyzes selected alert types and provides auditable output, but does not perform any remediation actions.
Phase 6 – Summative Validation and Closure	Repeat the security scenarios, compile KPI results, finish training, document risk areas/deficiencies, and prepare documentation for implementation.	Acceptance criteria evaluated; results recorded; documentation and training complete.

Implementation Completion Criteria

- All in-scope endpoints have been registered in Wazuh.
- Telemetry availability of at least 95% has been met during the test period.
- Windows Sysmon and OS events are indexed in Wazuh based on the test plan.
- Detection and/or investigative technique for each security scenario requirement has been developed.
- Rook can process all supported alert types, and its output is traceable back to the Wazuh evidence.

- RBAC and administration controls work properly.
- Test results and acceptance KPIs have been recorded.
- Network diagram, installation/configuration instructions, analyst procedure, risk register, training material, and maintenance plan have been prepared.

D3. Implementation Risks

Risk	Likelihood	Impact	Mitigation
Wazuh installation or configuration error	Medium	High	Use vendor-supported operating systems and installation guidance; validate each central component before enrolling endpoints.
Agent or Sysmon performance overhead	Medium	Medium	One pilot run on an endpoint; measure before and after CPU/Memory utilization; reduce Sysmon events if overhead and volume become a concern.
Telemetry loss or parsing failure	Medium	High	Source events versus Wazuh ingestion, status monitoring of the agent, and documentation/fixing of parsing/collection issues.
False-positive alerts	High	Medium	Use benign test cases and rule tuning before summative testing.
False-negative detections	Medium	High	Confirm source evidence exists, test known-positive cases, and document unsupported scenarios rather than claiming detection.
AI hallucination or unsupported conclusion	High	High	Provide only structured evidence, preserve original alerts, require human validation, and score factual agreement.
AI model or runtime resource constraints	Medium	Medium	Select a model based on hardware capacity, input only relevant alerts, and check latency.
Unauthorized dashboard access	Low	High	Use Wazuh RBAC, strong authentication, least privilege, and isolated network access.
Sensitive information in telemetry	Medium	High	Synthetic accounts/data in the laboratory, fewer fields collected, and documentation retention.
Storage exhaustion	Medium	Medium	Limit retention for the lab, monitor storage, and avoid unnecessary high-volume events.
Time synchronization error	Low	High	Configure a common time source and validate timestamps before formal testing.
Third-party dependency vulnerability	Medium	High	Record versions, apply supported updates, and document dependency risk.
Model licensing incompatibility	Low	Medium	Select a locally hosted model with a license appropriate for academic use and record the model/license in implementation documentation.
Scope expansion	High	Medium	Limit the project to configuration/integration/evaluation of existing technologies; avoid developing a new SIEM, EDR, SOAR or AI solution.

Risk	Likelihood	Impact	Mitigation
Unsafe test execution	Low	High	Use only owned/authorized isolated systems, benign test commands, Nmap against lab targets, and EICAR instead of operational malware.
Schedule delay	Medium	Medium	Prioritize core Wazuh/Sysmon implementation and treat Rook enhancements as secondary to required detection/evaluation deliverables.

E. Training Approach

The training will be geared towards showing participants how to operate the implemented security tools in their daily operations. This will be done through using the real Wazuh dashboard, the alert management flow designed during the project and the Rook evaluation technique.

Audience	Delivery	Duration	Content
Security Analyst / SOC Analyst	Instructional demonstration, hands-on lab	2 hours	Wazuh Dashboard navigation, endpoint inventory, alert search, ATT&CK mapping, timelines, evidence validation, review of Rook's findings, escalation of incidents, documentation of analyst's decisions.
System / Network Administrator	Guide to technical overview and installation	1 hour	Wazuh Agent install; Sysmon setup; agent groups; connectivity; service management; troubleshooting; resource monitoring; logging; secure communication channels.
Security Manager / Sponsor	Brief reporting demonstration	30 minutes	KPIs; endpoint protection; alert trends; MTTD/MTTA; detection efficacy; AI shortcomings; project success criteria.
End User Awareness	Short briefing as needed	15 minutes	Purpose of endpoint monitoring, acceptable-use expectations in the lab, and reporting unexpected endpoint behavior.

Training is deemed successful if an analyst is able to identify an endpoint, recognize and analyze an alert, trace the alert to its source, analyze the Rook's result, and execute the escalation process. The administrator must also be able to monitor the status of the agents and solve issues relating to telemetry connectivity.

F. Required Resources and Costs

The project will make use of an available computer and open-source evaluation software to ensure there is no new cost for the same at $0. The computer that will be used by the project already exists and will not be considered an additional cost for the project. If the computer lacks enough capacity in memory or space, further cost will be incurred but not within the current scope.

Phase	Resources	Estimated Direct Cost	Cost Source
Phase 1 – Baseline	Existing workstation; Windows/Linux VMs; Wireshark; Nmap; native logs	Existing hardware: $0 incremental. Wireshark: $0. Nmap: $0 for permitted end-user use under its license. Ubuntu: $0.	Wireshark (2026); Nmap Project (2026); Canonical (2026).
Phase 2 – Wazuh Central	Ubuntu Server VM; Wazuh server, indexer, dashboard	Wazuh software: $0; Ubuntu: $0; VM resources: $0 incremental on existing host.	Wazuh (2026a); Canonical (2026).
Phase 3 – Endpoint Telemetry	Wazuh Agent; Microsoft Sysmon; Windows evaluation endpoint	Wazuh Agent: $0; Sysmon: $0; Windows evaluation: no software purchase during evaluation.	Wazuh (2026a); Microsoft (2026); Microsoft Sysinternals (2022).
Phase 4 – Detection/Dashboard	Wazuh dashboard/rules; MITRE ATT&CK; browser	No additional direct software cost.	Wazuh (2026b; 2026c); MITRE (2026).
Phase 5 – Rook	Ollama runtime; locally hosted model with approved license; existing CPU/GPU	Ollama runtime: $0 under MIT license; model: $0 only if a no-cost model with compatible license is selected; existing hardware: $0 incremental.	Ollama (2026).
Phase 6 – Validation/Documentation	Wazuh, Wireshark, Nmap, spreadsheet/documentation tools	No additional direct project cost using already available academic tools.	Same sources as above.

Estimated Budget

Resource	Quantity	Cost	Basis
Existing host workstation	1	$0 incremental	Already owned; no new expenditure.
Ubuntu Server/Desktop	Multiple VMs	$0	Canonical states Ubuntu is free to download and use.
Wazuh platform and agents	1 deployment	$0	Wazuh documentation states Wazuh is free and open source.
Microsoft Sysmon	Windows endpoints	$0	Microsoft Sysinternals states the tools may be installed and used without charge.
Wireshark	1	$0	Wireshark states it is free and open-source software.
Nmap	1	$0	Nmap is available under the Nmap Public Source License for permitted use.
Ollama runtime	1	$0	Ollama source is distributed under the MIT License.
Locally hosted AI model	1	$0 expected	A no-cost model with a compatible license will be selected and documented before implementation.

Resource	Quantity	Cost	Basis
New direct expenditure	—	$0 expected	Assumes sufficient existing compute/storage resources.

G. Final Project Deliverables

The deliverables at the end of the process will take into account the actual implementation, configuration, security measures employed, and test results. Wazuh, Sysmon, Ollama, and the AI model selected will remain as third party software applications; this is what the deliverables of this project entail.

Technical Deliverables

- Wazuh Server, Indexer, Dashboard configuration (Bastion)
- Enroll Wazuh agents configured with Sysmon enhanced Windows and Linux telemetry (Pawn)
- Endpoint asset inventory document and agent health monitoring.
- Detection rules/queries for the chosen scenarios along with MITRE ATT&CK mappings.
- Alert and Investigation dashboards based on Wazuh features.
- Local AI workflow configuration for Wazuh alerts (Rook).
- Traceability from Rook output back to Wazuh alerts and evidence.
- Role-based access and administrative workflow.

Security and Process Deliverables

- Procedure for handling alerts raised by the project.
- Procedure for use of AI and human review process ensuring that Rook is purely advisory.
- Risk register for pre- and post-deployment risks.
- Procedure for log retention and management within the laboratory.
- Evidence of rule validation and tuning.
- Maintenance/updating schedule for Wazuh, Sysmon, the AI and detection rules.

Documentation and Evaluation Deliverables

- Network/Security Architecture Diagram Before Implementation.
- Network/Security Architecture Diagram After Implementation.
- Wazuh Installation & Configuration Documentation.
- Endpoint Onboarding & Sysmon Configuration Guide.
- Configuration of Rook & Analyst User Guide.
- Baseline Test Dataset & Measurement Worksheet.
- Formative Test Results.
- Summative Test Results.
- Comparison & Analysis of KPIs.
- Training Documentation & Training Validation Checklist.
- Capstone Final Implementation Report.

G1. Projected Timeline

The new schedule will cover a period starting from October 19, 2026, to January 10, 2027. There may be changes in the schedule dates depending on whether there are any changes in the course schedule.

Milestone	Duration	Start	End	Assigned Resources
Sprint 1 – Requirements and Baseline	2 weeks	Oct. 19, 2026	Nov. 1, 2026	Project Security Engineer; Windows/Linux lab systems; Wireshark; Nmap; native logs
Sprint 2 – Wazuh Central Deployment	2 weeks	Nov. 2, 2026	Nov. 15, 2026	Project Security Engineer/System Administrator role; Ubuntu Server; Wazuh central components
Sprint 3 – Endpoint Telemetry Rollout	2 weeks	Nov. 16, 2026	Nov. 29, 2026	Project Security Engineer; Wazuh Agent; Sysmon; Windows/Linux endpoints
Sprint 4 – Detection Engineering and Dashboards	2 weeks	Nov. 30, 2026	Dec. 13, 2026	Security Analyst role; Wazuh rules/dashboard; MITRE ATT&CK
Sprint 5 – Rook Integration	2 weeks	Dec. 14, 2026	Dec. 27, 2026	Project Security Engineer; Ollama; approved local model; Wazuh alert evidence
Sprint 6 – Summative Testing, Training, Closure	2 weeks	Dec. 28, 2026	Jan. 10, 2027	All lab systems and tools; analyst/admin roles; documentation resources

H. Project Evaluation Approach

The evaluation will be done by comparing the initial environment to the post-deployment environment after Wazuh, Sysmon, and Rook have been installed. Rule-based detection by Wazuh and AI-powered interpretation by Rook will each be graded separately. This is crucial since an accurate Wazuh alert does not necessarily mean that the AI interpretation is accurate.

H1. Formative and Summative Test Plans

Formative Testing

Test	Procedure	Tools	Expected Result
Wazuh Central Health	Check for the server, indexer, and dashboard to be healthy, then log into the dashboard; make sure indices and health details are available.	Wazuh dashboard and system services	Central components operate without critical errors.
Agent Enrollment and Connectivity	Register one pilot endpoint and ensure that it has Active state and disconnect/reconnect; verify state change and telemetries resumes.	Wazuh Agent and dashboard	Endpoint remains correctly identified and resumes telemetry after reconnection.
Sysmon Ingestion	Create known good process and a network event; compare local Sysmon event and Wazuh event.	Sysmon, Event Viewer, Wazuh	Required fields appear in Wazuh and timestamps/host identity match.
Linux Log Ingestion	Generate successful/failed login events and confirm ingestion and searchability.	Linux auth/system logs, Wazuh	Events are searchable with correct host/user/time context.
Detection Rule Validation	Check rules by applying known positive, known negative, and benign samples. Always use Wazuh rule tester wherever possible.	Wazuh ruleset test/dashboard	Rules trigger only on the intended conditions or are tuned before summative testing.
RBAC Test	Login dashboard operates as authenticated admin, restricted user, and unauthorized user.	Wazuh RBAC/dashboard	Administrative functions are unavailable to unauthorized users.
Rook Evidence Test	Send a predefined alert containing known evidence to the local AI workflow; score factual consistency and unsupported claims.	Ollama/local model, Rook workflow	Output references supplied facts, avoids invented evidence, and remains advisory.

Test	Procedure	Tools	Expected Result
Endpoint Performance Test	Measure CPU/memory before and after Wazuh Agent/Sysmon under comparable benign workload.	OS performance tools	Average additional CPU overhead remains below the project threshold.

Summative Testing

Scenario	Controlled Activity	Industry Mapping / Purpose	Expected Result
1. Repeated Authentication Failures	Generate a controlled sequence of invalid logins on a lab account.	ATT&CK T1110	Wazuh logs the failure events and generates the alert or investigation; Rook summarizes the presented evidence.
2. Test Account Creation	Create a temporary authorized test account specifically for detection validation.	ATT&CK T1136	Wazuh detects the creation of an account and the host/user context around it.
3. Benign PowerShell Detection Scenario	Run a benign PowerShell command that is tailored for process/command line telemetry test.	ATT&CK T1059.001	Sysmon/Wazuh will collect all the data regarding the execution of PowerShell but will not use any harmful scripts.
4. Network Service Discovery	Run Nmap only against authorized lab hosts.	ATT&CK T1046	Wazuh and/or endpoint telemetry provide evidence sufficient to identify the discovery activity according to the documented rule/query.
5. EICAR Security Event	Utilize the common EICAR virus detection file if supported by the endpoint security system configuration. Safe malware detection verification	Safe malware-detection validation	Wazuh ingests the resulting security event; no operational malware is introduced.
6. Normal User Activity	Perform standard logins, browsing, application launches, file creation, and approved administrative activity.	Benign control set	The environment does not produce a false-positive rate that is unacceptable for the benign set defined.

H2 Minimum Acceptance Criteria and Key Performance Indicators

Each acceptance requirement listed below is clearly connected to the tests in H1. Formative tests check the setup. Prepare the system before the evaluation. Summative tests do the checks in the environment and give the results that are used for approval. Just passing a test, by itself does not meet a requirement.

Before starting the official testing the test register will set the endpoint inventory the Rook alert types that are allowed how many times each scenario is run what the rule or investigation query is expected to find the level of severity the evidence fields that are needed the benign control activities and the start and end times, for the test window.

Host clocks will be synchronized. Scenarios 1 through 4 will be included; scenario 5 will be included only if the endpoint security configuration supports EICAR. Its applicability will be decided and documented during formative testing, before the final scenario set is frozen. Scenario 6 is the benign control and is scored separately from detection coverage. Planned cases that cannot be executed will remain in the acceptance denominator and will not count as successes.

Operational and detection acceptance

KPI and minimum criterion	H1 test alignment and acceptance evidence
Endpoint enrollment 100% of in-scope endpoints.	Formative: Agent Enrollment and Connectivity. Summative: verify the frozen inventory before scenarios 1 through 6. Compare active, correctly identified Wazuh agents against the endpoint inventory; retain the agent status export. Enrollment = enrolled in-scope endpoints / all in-scope endpoints x 100.
Telemetry availability At least 95% during the formal test window.	Formative: Agent Enrollment and Connectivity, Sysmon Ingestion, and Linux Log Ingestion. Summative: monitor collection throughout scenarios 1 through 6. Generate a scheduled benign verification event on each endpoint every minute and retain source and indexed timestamps. Availability = endpoint-minute checks with the expected event searchable within 60 seconds / all scheduled endpoint-minute checks x 100.
Required evidence field completeness At least 90% across applicable scenarios.	Formative: Sysmon Ingestion and Linux Log Ingestion. Summative: scenarios 1 through 5, where applicable. Use the frozen field checklist: host and timestamp for each case; user for authentication/account tests; process and command line for PowerShell; applicable source/destination context for discovery; security product result for EICAR. Completeness = populated, accurate required fields / all applicable required fields x 100. Retain source and Wazuh exports.
Defined scenario detection or investigation success At least 90% of planned applicable positive test cases.	Formative: Detection Rule Validation and ingestion tests. Summative: scenarios 1 through 5, where applicable. Score each planned repetition against its predefined alert or investigation query and evidence requirements. Success = cases satisfying those requirements / all planned applicable positive cases x 100. Missing evidence or an unexecuted planned case is not a success. Benign scenario 6 is excluded.
High priority alert availability No more than 60 seconds for every planned high priority case requiring an alert.	Formative: Detection Rule Validation with source and alert timestamps. Summative: the frozen high priority, alert-expected subset of scenarios 1 through 5. Measure source event time to the time the corresponding alert becomes searchable in Wazuh. For a threshold rule use the source event that completes the triggering condition. Retain event ID rule ID, timestamps and elapsed seconds.
False positive rate for the defined benign set No more than 10%.	Formative: Detection Rule Validation with known-negative and benign samples. Summative: scenario 6. Define one case as one scripted benign activity with a fixed observation window. False positive rate = benign cases incorrectly flagged as suspicious / all executed benign cases x 100. Retain the activity log and associated alerts; multiple alerts from one activity count once for this rate.

AI performance and access acceptance

KPI and minimum criterion	H1 test alignment and acceptance evidence
Rook processing success At least 90% for supported alert types.	Formative: Rook Evidence Test. Summative: supported Wazuh alerts generated by applicable scenarios 1 through 5. Success requires a returned, readable analysis with the predefined summary, suggested severity, and investigation steps. Success rate = completed valid outputs / all planned supported submissions x 100. Record submission ID, source alert ID, output, and failures.

KPI and minimum criterion	H1 test alignment and acceptance evidence
Rook factual agreement At least 90% of scored factual fields.	Formative: Rook Evidence Test with predefined facts. Summative: the supported alert cases from scenarios 1 through 5. Compare output fields against supplied Wazuh and endpoint evidence using a fixed scoring rubric. Agreement = correct factual fields / all scored factual fields x 100. Invented claims and missing required factual fields are errors; analyst-approved suggestions are scored separately from factual claims.
Rook traceability to source evidence 100% of supported test cases.	Formative: Rook Evidence Test. Summative: every supported Rook case associated with scenarios 1 through 5. Require the retained output record to identify the input Wazuh alert ID and the source evidence used for its factual statements. Traceability = fully linked cases / all planned supported cases x 100. Missing output or a broken source link fails the case.
Average additional endpoint CPU overhead Less than 10 percentage points per endpoint.	Formative: Endpoint Performance Test. Summative: repeat the same measurement during scenario 6 on every in-scope endpoint. Use the harmless work load and the same 30-minute baseline and after installation periods checked every 5 seconds. Extra load is the CPU use after installation minus the baseline CPU use. Keep the data points and the calculation, for each device. Do not hide a device that is not working by using the average for all devices.
Wazuh availability At least 95% during the formal test window.	Formative: Wazuh Central Health. Summative: continue health checks throughout scenarios 1 through 6. Check the server, indexer, and authenticated dashboard search every minute. Availability = scheduled checks in which all three function correctly / all scheduled checks x 100. Retain health and search records; failed or missing checks count as unavailable.
Unauthorized administrative access blocked 100% of defined unauthorized access attempts.	Formative: RBAC Test. Summative: repeat the RBAC matrix with final roles and configuration during the formal test window. Try each set action using a restricted account and also as an unauthenticated user. Check that access is denied and nothing changes in the system state. Calculate the rate by dividing the number of blocked attempts, by the number of unauthorized attempts defined then multiply the result by 100. Keep all audit records and results. Also test the administrator separately to confirm that the administrator can perform the actions. This acts as a control.

Comparative outcomes and final acceptance

MTTD = detection time minus event start time. MTTA = preliminary classification time minus detection or alert time. MTTD improvement (%) = (baseline MTTD minus post-implementation MTTD) / baseline MTTD x 100. The detection success and false-positive calculations are defined in the acceptance matrix above.

MTTD and MTTA will also be compared across matched baseline and post-implementation cases using the calculations in H2 and records in H4. These are comparative outcomes; this proposal does not set an additional numerical improvement threshold. Undetected cases will be reported separately rather than assigned a zero detection time. No detection or analysis improvement will be claimed when the corresponding baseline is missing or not comparable. A zero baseline makes a percentage improvement calculation undefined; absolute times will be reported instead.

Each final result will identify the KPI, H1 test or scenario, case IDs, numerator and denominator or timing calculation, evidence location, and pass/fail outcome. Formative failures will be corrected and retested before the configuration is frozen. Summative failures will be shown as not meeting the requirements. If fixing the problem changes the setup that was tested the tests that are affected will be run again and both results will be kept. Final approval needs every required rule to be met; Wazuh detection success can't make up for Rook output that is wrong or can't be followed. Rook output that is wrong or can't be followed can't be fixed by Wazuh detection

success. Every required basic rule must be met. Wazuh detection success can't make up for Rook output that's wrong or can't be followed. Final approval needs every required basic rule to be met.

H3. Justification of Test Cases and Scenarios

The scenarios were selected in order to cover multiple categories of endpoint evidence while still remaining safe and reproducible in the lab environment. Failed logon attempts test authentication logging and ATT&CK's Brute Force techniques. Test account creation tests account modification monitoring. PowerShell is selected as a scenario in order to test process and command line telemetry using a legitimate administration utility which is covered by ATT&CK as well. Network service scanning is used as a scenario to test network and endpoint visibility using Nmap against legitimate lab assets. The use of EICAR can safely simulate an antivirus attack without any viruses involved. Normal user activity serves as a benign control for false positives detection.

These scenarios also coincide with the functionality of the selected security products. Sysmon is capable of collecting Windows process and optional network telemetry, and Wazuh collects endpoint data, performs rules processing, centralized alerting, dashboards and MITRE ATT&CK mappings. This test plan does not test the brand new security product, but verifies the documented features of the product.

H4. Results Analysis

Each scenario will be assigned an ID number and tested in a controlled environment. The results form will record the following information: endpoint, event time, Wazuh alert time, analyst time, preliminary classification time, expected evidence, actual evidence gathered, Wazuh rule and level of severity, Rook output, analyst classification, Rook processing time, false positive, CPU/Mem impact where applicable, and any additional notes from the test.

Quantitative Analysis

- Compare MTTD before and after implementation.
- Compare manual analysis time with Wazuh/Rook MTTA.
- Determine scenario detection success using the planned applicable positive-case denominator defined in H2; report executed-case coverage separately if used.
- Determine completeness of evidence field.
- Determine endpoint telemetry availability.
- Determine the false-positive rate as benign test cases incorrectly flagged as suspicious divided by all executed benign test cases, multiplied by 100.
- Determine Rook factual agreement and supported alert processing rate.
- Determine endpoint performance overhead.

Qualitative Analysis

- Ease of finding endpoint evidence through Wazuh compared to manually going through source logs.
- Value and clarity of dashboards and event search within Wazuh.
- Value of using MITRE ATT&CK framework in investigation.
- Accuracy and value of Rook summary and recommendations.
- Administrative challenges and limitations faced during implementation.
- Gaps that would need further telemetry, tuning, or enterprise capabilities beyond the scope of the capstone project.

AI Analysis Validation

Scoring of Rook will be separate from that of the findings of Wazuh's detection. The output of Rook will be evaluated based on the test scenario, original endpoint information, Wazuh's alert, intended classification, and analyst's determination. Where Rook claims something that isn't supported by facts, it is classified as an error, regardless of the correctness of the overall determination made by the AI. This prevents good writing skills from being taken as accurate information.

Final Acceptance Determination

The success of the implementation will be judged based on whether the acceptance criteria are satisfied and whether the measurements indicate improved central visibility and/or faster detection/analysis without unacceptably high endpoint overhead or access control issues. If any criterion is not satisfied, that finding will be reported along with an explanation and a fix for the problem. In the final report, the real measurements will be used, not estimated numbers.

References

Autio, C., Schwartz, R., Dunietz, J., Jain, S., Stanley, M., Tabassi, E., Hall, P., & Roberts, K. (2024). Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile (NIST AI 600-1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.600-1

Canonical. (2026). Download Ubuntu. https://ubuntu.com/download

Cybersecurity and Infrastructure Security Agency. (n.d.). Use logging on business systems. https://www.cisa.gov/audiences/small-and-medium-businesses/secure-your-business/use-logging-on-business-systems

Kent, K., & Souppaya, M. (2006). Guide to Computer Security Log Management (NIST SP 800-92). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-92

Microsoft. (2026). Sysmon v15.22. Microsoft Sysinternals. https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon

Microsoft Sysinternals. (2022). Sysinternals Licensing FAQ. https://learn.microsoft.com/en-us/sysinternals/license-faq

MITRE. (2026). MITRE ATT&CK Enterprise techniques: T1110, T1136, T1059.001, and T1046. https://attack.mitre.org/

National Institute of Standards and Technology. (2024). The NIST Cybersecurity Framework (CSF) 2.0 (NIST CSWP 29). https://doi.org/10.6028/NIST.CSWP.29

Nelson, A., Rekhi, S., Souppaya, M., & Scarfone, K. (2025). Incident Response Recommendations and Considerations for Cybersecurity Risk Management: A CSF 2.0 Community Profile (NIST SP 800-61 Rev. 3). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-61r3

Nmap Project. (2026). Nmap Public Source License and Nmap documentation. https://nmap.org/npsl/

Ollama. (2026). Ollama source repository and MIT License. https://github.com/ollama/ollama/blob/main/LICENSE

Scarfone, K., Souppaya, M., & Dodson, D. (2022). Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities (NIST SP 800-218). https://doi.org/10.6028/NIST.SP.800-218

Wazuh. (2026a). Installation guide and getting started documentation. https://documentation.wazuh.com/current/installation-guide/index.html

Wazuh. (2026b). Wazuh dashboard components and capabilities. https://documentation.wazuh.com/current/getting-started/components/wazuh-dashboard.html

Wazuh. (2026c). MITRE ATT&CK framework integration. https://documentation.wazuh.com/current/user-manual/ruleset/mitre.html

Wireshark Foundation. (2026). About Wireshark. https://www.wireshark.org/about