Designing a Lightweight Proactive Threat Detection Framework Using Open-Source SOC Tools for Mid-Sized Enterprises

Olusegun Victor Adeola
Independent Researcher
San Antonio, TX, USA
segunvic.adeola@gmail.com

***Abstract*— Mid-sized enterprises increasingly face sophisticated cyber threats while operating under constrained budgets and limited Security Operations Center resources. Many organizations rely on fragmented security tooling that emphasizes reactive alerting rather than early detection of adversarial activity. This paper proposes a lightweight, proactive threat detection framework built on open-source SOC tools to improve visibility and detection across network, endpoint, identity, and cloud environments. The framework emphasizes architectural design, data source integration, and analyst-oriented workflows rather than novel detection algorithms. By correlating telemetry from multiple security layers, the proposed approach enables earlier identification of common attack behaviors such as credential misuse, anomalous authentication patterns, and lateral movement. The framework is designed to be incrementally adoptable and suitable for mid-sized enterprises seeking to strengthen cybersecurity resilience without dependence on costly commercial platforms. This work provides a practical reference architecture that bridges existing visibility gaps and supports proactive security operations using accessible open-source technologies.**

Keywords — ***Proactive threat detection, Security Operations Center (SOC), open-source security tools, telemetry correlation, cybersecurity monitoring, mid-sized enterprises***

I. INTRODUCTION

Cybersecurity threats continue to grow in both scale and sophistication, forcing organizations to improve how quickly they can detect and respond to malicious activity. Large enterprises typically address this through mature Security Operations Centers backed by comprehensive commercial platforms. Mid-sized enterprises face a different reality. They operate under tighter budgets, smaller teams, and limited operational capacity. Security teams in these environments often piece together fragmented tools and spend most of their time responding to alerts rather than hunting for threats. This reactive posture makes it difficult to catch attackers early.

Mid-sized enterprises are a critical part of economic and digital infrastructure, but they are frequently targeted precisely because of gaps in visibility, identity governance, and network monitoring. Attackers know this. Industry reports and government guidance show that credential misuse, cloud service abuse, and lateral movement techniques are increasingly common because they bypass traditional perimeter defenses [17]. In organizations with limited monitoring coverage, this activity can go unnoticed for weeks or months, giving attackers time to entrench themselves and cause serious damage.

Modern cybersecurity frameworks stress the need for continuous monitoring, telemetry correlation, and outcome-driven security operations [1]. The challenge is turning these principles into something practical for organizations that cannot afford to deploy and maintain large-scale SIEM platforms. Many security teams adopt open-source monitoring tools but deploy them in isolation. The result is a collection of disconnected data sources that provide little contextual awareness for analysts trying to piece together what is actually happening.

Open-source security technologies have matured significantly over the past few years. Tools for network traffic analysis, endpoint monitoring, log aggregation, and event correlation are now robust enough to support serious detection work. When integrated properly, these tools can enable proactive detection strategies that align with modern frameworks and zero trust principles [11]. The challenge is designing an architecture and operational workflow that lets small SOC teams extract useful signals from telemetry.

This paper proposes a lightweight proactive threat detection framework designed for the operational realities of mid-sized enterprises. The framework focuses on architectural clarity, incremental adoption, and analyst-friendly workflows rather than novel detection algorithms or performance metrics. By correlating telemetry across network, endpoint, identity, and cloud layers, the approach reduces visibility gaps and helps teams identify common attack behaviors earlier using accessible open-source technologies.

II. BACKGROUND AND RELATED WORK

Security Operations Center architectures have traditionally been built around centralized log collection and correlation platforms. These platforms aggregate events from endpoints, network devices, identity providers, and applications so that analysts can investigate alerts in one place. Security Information and Event Management systems are the standard approach for large enterprises that can afford the substantial financial investment, specialized infrastructure, and dedicated staff they require.

Mid-sized enterprises often cannot meet those requirements. Many end up deploying security controls and monitoring tools in a piecemeal fashion, focusing on compliance-driven log retention or perimeter detection rather than building holistic visibility. Government and industry guidance has pointed out that this reactive, siloed approach makes it harder to catch modern attack techniques. Credential misuse, cloud service abuse, and lateral

movement across internal networks are particularly difficult to spot without broader correlation.

Recent cybersecurity frameworks have shifted toward continuous monitoring and outcome-driven security operations [6]. Instead of relying on individual alerts, these frameworks encourage correlating identity, endpoint, and network telemetry to identify behavioral patterns that suggest adversarial activity. Zero trust architectures take this further by treating identity and access events as primary detection signals, especially in hybrid and cloud environments where traditional network boundaries no longer apply.

At the same time, the open-source security ecosystem has gotten much stronger. Network monitoring solutions now provide detailed protocol-level and connection metadata. Intrusion detection systems generate structured alerts based on known attack patterns. Host-based agents capture system activity and configuration changes. Log aggregation and search platforms support centralized analysis and investigation workflows. These tools can handle many of the SOC functions that used to require commercial platforms. What is still missing is practical guidance on how to integrate them into a coherent detection strategy for organizations with limited resources.

Most academic literature and industry reports focus on SOC maturity models, threat intelligence integration, or detection effectiveness at an enterprise scale [16]. There is much less emphasis on lightweight, modular architectures designed for incremental adoption and built with mid-sized enterprises in mind. This paper builds on established security monitoring principles and widely accepted frameworks but addresses the practical gap between high-level guidance and operational implementation.

III. PROBLEM STATEMENT

Mid-sized enterprises face a combination of technical and operational challenges that make effective threat detection difficult. One of the most persistent problems is limited and uneven visibility. Network activity, endpoint behavior, identity and access events, and cloud service usage are often monitored separately, if at all. Security teams collect these data sources independently and review them in isolation, which makes it nearly impossible to spot correlated patterns that indicate an attacker is present.

Security monitoring in these organizations tends to be reactive. Alerts fire based on individual events or signature matches rather than behavioral indicators that span multiple telemetry sources. This generates a lot of false positives and wears analysts down. Early signs of compromise get buried in noise, and by the time something obvious happens, adversaries have already gained persistence, escalated privileges, and moved laterally.

Operational constraints make things worse. Small SOC teams have to juggle continuous monitoring with incident response, system administration, and compliance tasks. Managing a collection of disparate security tools takes time and effort, which discourages teams from pursuing deeper integration even when it would improve detection. Inconsistent workflows and limited automation mean that investigation quality varies depending on who is working the case.

The growing reliance on cloud services and remote access has also reduced the effectiveness of perimeter-focused monitoring. Identity-based attacks, including credential misuse and service account abuse, have become a primary way attackers get in. When authentication events, endpoint activity, and network telemetry are not correlated, these attacks look harmless when viewed individually.

The core problem this work addresses is the lack of a practical, resource-efficient framework for proactive threat detection in mid-sized enterprises using open-source technologies. What is needed is an approach that emphasizes architectural clarity, telemetry correlation, and analyst-friendly workflows while allowing for incremental adoption. The proposed framework provides a structured method for integrating telemetry from multiple layers and identifying common attack behaviors earlier without depending on expensive commercial platforms.

IV. PROPOSED PROACTIVE THREAT DETECTION FRAMEWORK

This section presents a lightweight proactive threat detection framework designed to address the visibility and operational challenges that mid-sized enterprises face. The framework prioritizes architectural clarity, modular integration, and analyst usability rather than optimizing individual tools or chasing detection performance metrics. It can be adopted incrementally and adapted to different organizational environments.

A. Design Principles

The framework is built around three core principles. First, multi-layer telemetry correlation takes priority over isolated alerting. Detection improves when signals from network, endpoint, identity, and cloud domains are correlated rather than treated as independent indicators. Second, operational simplicity matters. Small SOC teams have limited resources, so tool selection and data flows should minimize administrative overhead while giving analysts the context they need. Third, the framework maintains tool agnosticism and modularity. Organizations can substitute equivalent open-source or commercial components without changing the underlying detection logic.

B. Framework Architecture Overview

The framework has four functional layers: telemetry collection, data normalization and aggregation, detection and correlation, and analyst workflows.

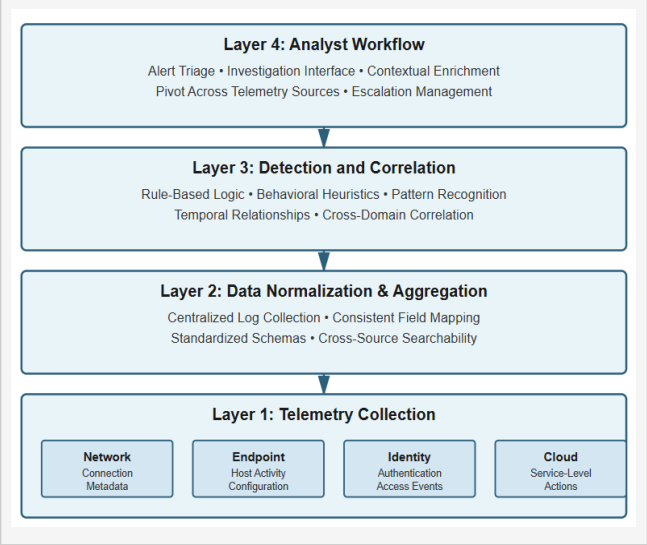

Fig 1: Modular framework architecture with functional layers

The **telemetry collection layer** ingests security-relevant data from multiple sources. Network monitoring systems capture connection metadata and protocol behavior. Endpoint agents provide host-level activity and configuration changes. Identity and authentication services record login and authorization events. Cloud platforms contribute service-level actions. Together, these sources provide the raw signals needed for behavioral detection.

The **data normalization and aggregation layer** centralizes the collected telemetry and applies consistent field mappings so that events from different sources can be correlated. Standardized schemas and common event fields reduce fragmentation and

improve searchability. This layer supports both real-time alerting and retrospective analysis by making sure events from different domains can be examined together.

The **detection and correlation layer** applies rule-based logic and behavioral heuristics to identify patterns that suggest adversarial activity. Instead of relying only on signature matches, detection logic looks for relationships between events. An anomalous authentication followed by unusual network connections or endpoint activity is more telling than either event alone. This approach aligns with widely recognized adversary behavior models, including those described in the MITRE ATT&CK framework [13], without requiring direct technique attribution.

The **analyst workflow layer** provides interfaces for alert triage, investigation, and escalation. Analysts see correlated events and contextual enrichment instead of isolated alerts, which reduces cognitive load and improves decision-making. The framework supports iterative investigation, letting analysts pivot across telemetry sources as their hypotheses develop.

C. Telemetry Sources and Correlation Strategy

Proactive detection depends on correlating telemetry across multiple security domains. The framework treats identity signals as first-class detection inputs because credential misuse and anomalous access patterns are now dominant initial access vectors. Authentication anomalies are evaluated alongside endpoint behavior and network activity to separate benign deviations from malicious actions.

Network telemetry adds behavioral context by revealing communication patterns that might indicate lateral movement, command and control activity, or data exfiltration. Endpoint telemetry confirms process execution, privilege changes, and persistence mechanisms that often follow successful access. Cloud and application logs enrich investigations by capturing service-level actions that may not be visible at the network layer.

Correlation logic prioritizes temporal relationships and behavioral sequences over single-event thresholds. This enables earlier detection of attack chains and reduces reliance on static signatures that adaptive adversaries can evade.

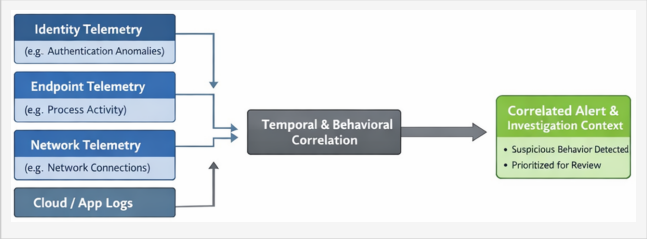

Fig 2: Correlation of identity, endpoint, network, and cloud telemetry to identify behavioral attack patterns

D. Incremental Adoption and Deployment Considerations

A key objective of the framework is supporting incremental adoption. Organizations can start with a subset of telemetry sources and expand coverage as resources allow. For example, they might implement network monitoring and identity log ingestion first, then add endpoint telemetry and cloud service integration later.

The framework does not mandate a specific implementation stack. Modular open-source components can be combined to realize each functional layer. Integrated SOC distributions can serve as validation examples rather than required platforms. This flexibility lets organizations tailor deployments based on existing infrastructure, skill sets, and risk priorities.

V. TOOL MAPPING AND OPEN-SOURCE INTEGRATION

The proposed framework is intentionally tool-agnostic and focuses on functional capabilities rather than prescriptive implementations. This section maps the framework's functional layers to representative open-source technologies that show how the architecture can work in practice. The tools discussed are illustrative examples chosen for their maturity, community adoption, and alignment with SOC workflows. They are not exclusive recommendations.

A. Network Telemetry and Intrusion Detection

Network visibility comes from open-source network monitoring and intrusion detection systems. Network monitoring tools generate detailed connection metadata, protocol-level logs, and behavioral indicators that help analysts spot lateral movement, anomalous outbound connections, and unusual communication patterns. Intrusion detection systems add structured alerts based on known attack signatures and threat intelligence. Their outputs are machine-readable and integrate easily into centralized log aggregation pipelines, supporting both real-time detection and retrospective investigation.

B. Endpoint and Host-Level Telemetry

Endpoint visibility comes from host-based security agents that monitor system activity, configuration changes, and security-relevant events. These agents capture process execution, privilege escalation attempts, persistence mechanisms, and policy violations that often follow initial access. In the framework, endpoint data serves as a corroborating signal rather than a standalone detection source. Correlating it with identity and network telemetry reduces false positives and helps analysts prioritize alerts that represent real risk.

C. Identity and Authentication Signals

Identity and authentication logs are first-class inputs in the framework because credential-based attacks are so common. Logs from directory services, identity providers, and access management systems provide visibility into login attempts, authentication anomalies, service account usage, and privilege assignments. When correlated with endpoint and network telemetry, these signals enable early detection of compromised credentials, anomalous access patterns, and abuse of non-interactive accounts. This identity-centric perspective aligns the framework with modern security architectures that emphasize access verification and continuous trust evaluation.

D. Log Aggregation, Normalization, and Correlation

Centralized log aggregation is the foundation for telemetry correlation. Open-source log collection and search platforms provide scalable ingestion, indexing, and querying capabilities that let analysts investigate events across multiple security domains in one interface. Normalization is critical for effective correlation. Applying consistent field mappings and event schemas reduces fragmentation and enables cross-source searches without extensive custom parsing. This supports both rule-based detection logic and exploratory investigations driven by analyst hypotheses.

E. Analyst Workflow and Investigation Support

The final integration layer focuses on analyst workflows rather than raw detection output. Dashboards, alert queues, and investigation views present correlated events with contextual enrichment, which reduces the cognitive burden of reviewing isolated alerts. Analysts can pivot between identity, endpoint, and network data as investigations progress, supporting iterative hypothesis testing and efficient escalation. Automation in this layer is intentionally limited to enrichment and routing tasks. This preserves analyst judgment while improving consistency and reflects the operational realities of small SOC teams.

F. Representative Open-Source Implementations

While the proposed framework is intentionally tool-agnostic, several mature and widely adopted open-source technologies illustrate how its functional components can be implemented in practice within mid-sized enterprise environments.

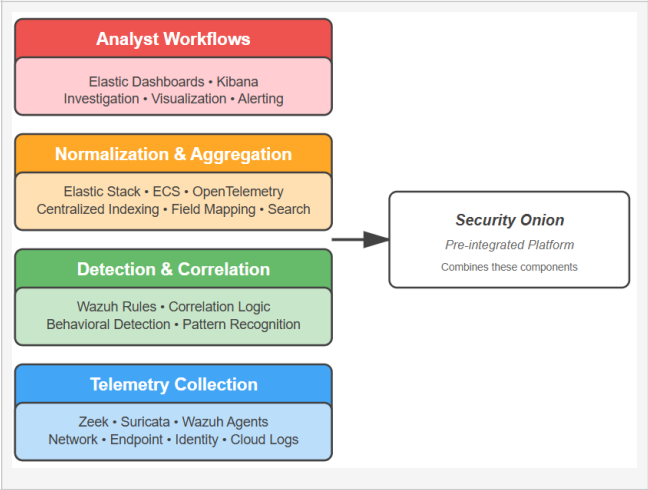

Fig 3: Example mapping of representative open-source SOC tools to the proposed framework layers

Network telemetry and protocol-level visibility can be provided by tools like Zeek [23], which generates rich connection metadata and behavioral logs that support detection of lateral movement, anomalous communication patterns, and policy violations. Complementary intrusion detection capabilities can come from Suricata [24], which produces structured alerts based on known attack signatures and threat intelligence feeds. This enables correlation between behavioral and indicator-based detection signals.

Endpoint and host-level telemetry can be collected using platforms like Wazuh [22], which integrates host monitoring, log collection, file integrity monitoring, and rule-based alerting into a centralized open-source security platform. Tools like this work well in resource-constrained environments because of their modular design and active community support.

Centralized log aggregation, search, and visualization can be supported through open-source components of the Elastic Stack, which lets analysts investigate events across network, endpoint, and identity domains in a unified interface. Event normalization within this layer can be handled using the Elastic Common Schema [21], which promotes consistent field mappings and reduces fragmentation across heterogeneous data sources.

Telemetry transport and standardization concepts within the framework align with emerging observability standards like OpenTelemetry [20], which provide a vendor-neutral approach to collecting and exporting log data. Although not required for implementation, these standards enhance extensibility and interoperability as monitoring requirements evolve.

Integrated SOC distributions like Security Onion demonstrate how several of these components can be pre-integrated into a cohesive platform. While such distributions may accelerate deployment, the proposed framework remains modular to support incremental adoption and selective integration based on organizational needs and existing infrastructure.

VI. EXAMPLE DETECTION SCENARIOS

This section presents representative detection scenarios that show how the framework supports proactive threat identification through telemetry correlation and analyst-driven investigation. The scenarios are illustrative and focus on detection logic and workflow progression rather than quantitative evaluation or alert accuracy.

A. Anomalous Authentication and Credential Misuse

In this scenario, the framework identifies suspicious authentication behavior that deviates from established access patterns. Identity logs show a successful login from an unfamiliar geographic region or device, followed by additional authentication attempts within a short time window. Viewed alone, these events might look like benign user travel or routine access anomalies.

Correlation with endpoint telemetry reveals that the affected account initiates process execution and access patterns inconsistent with its history. Concurrent network telemetry shows outbound connections to previously unseen external endpoints. The temporal alignment of these signals elevates the event from a low-confidence authentication anomaly to a credible indicator of credential misuse.

Within the analyst workflow layer, correlated events appear as a single investigative context rather than multiple isolated alerts. Analysts can quickly assess the sequence of events, validate whether the access is legitimate, and initiate containment actions like session termination or credential reset if needed.

B. Suspicious Service Account Activity

Service accounts often operate with elevated privileges and minimal interactive oversight, which makes them attractive targets. In this scenario, the framework detects abnormal behavior associated with a non-interactive account that typically performs limited, predictable actions. Identity telemetry shows authentication activity occurring outside expected execution windows or from unexpected hosts. Network telemetry indicates lateral connections originating from systems not previously associated with the service account's function. Endpoint telemetry confirms the execution of administrative commands or scripts inconsistent with the account's baseline usage. Correlating these signals enables early detection of service account abuse that might otherwise evade traditional alerting mechanisms.

C. Indicators of Lateral Movement

Lateral movement often manifests as subtle, low-noise activity distributed across multiple systems. In this scenario, network telemetry identifies a series of internal connection attempts that deviate from normal communication patterns. These connections may involve administrative protocols or access attempts across multiple hosts within a short period.

Endpoint telemetry from affected systems reveals authentication failures followed by successful access attempts, along with process execution associated with remote administration tools. Individually, these events may not exceed alert thresholds. When correlated, they form a behavioral pattern consistent with lateral movement techniques. The framework's correlation logic highlights the sequence and timing of these events, which helps analysts investigate the scope of potential compromise and identify affected assets. This proactive detection reduces dwell time and limits the opportunity for attackers to escalate privileges or establish persistence.

D. Unusual Outbound Network Activity

Outbound network activity can provide early indicators of command and control communication or data exfiltration. In this scenario, network telemetry reveals sustained or periodic connections to external endpoints that are atypical for the originating host or user context. Endpoint telemetry indicates background processes initiating network connections without

corresponding user activity, while identity logs confirm that no interactive session is active during the communication period. Correlation across these telemetry sources strengthens confidence that the activity is not user-initiated. Analysts use the unified investigation view to examine connection patterns, timing, and associated processes, enabling informed decisions about traffic containment or endpoint isolation often before data loss occurs.

VII. METHODOLOGY

To complement the architectural contributions of this framework, a controlled proof-of-concept experiment was conducted to validate the feasibility and behavioral characteristics of a lightweight, modular anomaly detection architecture. The experiment was not designed to introduce a novel detection algorithm or to generate comparative performance benchmarks. Rather, it was intended to demonstrate that the framework's core design principles of telemetry normalization, modular detection processing, and analyst-facing alert workflows can be realized in a reproducible, resource-efficient implementation. This validation supports the framework's applicability to mid-sized enterprise environments that operate under real infrastructure constraints.

A. Experimental Objectives

The experiment aimed to achieve four specific objectives: to demonstrate structured telemetry generation in a containerized environment consistent with the framework's normalization layer; to validate deterministic rate-based anomaly detection using a sliding time window reflecting the behavioral detection logic described in the framework's detection and correlation layer; to evaluate alert amplification control confirming that detection logic produces single, actionable alerts rather than redundant noise; and to measure false positive behavior under benign traffic conditions to establish baseline discrimination performance.

B. System Architecture

A three-container architecture was implemented using Docker Compose to reflect the framework's modular, layered design. The first container hosted a Flask web application backed by a Redis instance, which generated structured JSON telemetry and implemented the sliding-window rate anomaly detection logic.

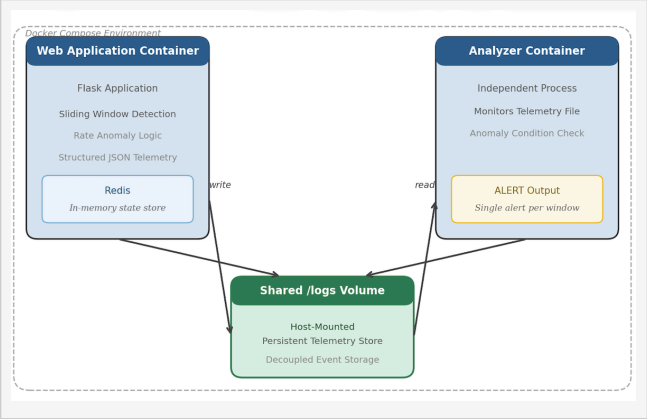

Fig 4: Minimal three-layer experimental architecture showing web container, Redis backend, shared /logs volume, and analyzer container

A persistent, host-mounted volume served as a shared telemetry store, enabling decoupled event storage without direct inter-container dependencies. The third container operated as an independent analyzer, monitoring the telemetry volume and emitting alerts when anomaly conditions were met. This separation of telemetry generation, storage, and detection processing directly mirrors the framework's layered architecture and ensures that each functional component can be modified or replaced independently.

C. Detection Logic

The anomaly detection model used a sliding time window of ten seconds with a threshold of thirty HTTP requests per source IP address. The detection trigger was implemented as an equality condition at threshold crossing, ensuring that a single anomaly event is produced per window regardless of how many subsequent requests are observed. This design choice directly addresses the alert amplification problem that burdens small SOC teams by preventing redundant alert generation while preserving detection determinism.

All events were logged in structured JSON format capturing timestamp, event type, HTTP method, request path, source IP address, user agent, and threshold metadata. This normalized schema is consistent with the framework's data normalization layer and supports downstream correlation and integration with larger SIEM or host and network sensor pipelines.

D. Experimental Procedure

Two controlled traffic scenarios were executed to evaluate detection and false positive behavior. The first scenario introduced a burst of eighty rapid HTTP GET requests from a single IP address to simulate volumetric abuse behavior within a concentrated time window. The second scenario distributed thirty HTTP GET requests over approximately fifteen seconds to simulate normal browsing activity. Telemetry logs from both scenarios were parsed using a structured evaluation script to compute event distributions and alert ratios, enabling direct comparison of detection outcomes under abusive and benign conditions.

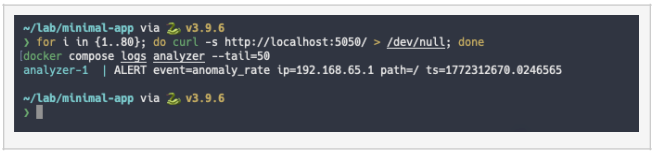

Fig 5: Terminal output showing controlled burst execution and resulting single ALERT output

VIII. EVALUATION

A. Burst Traffic Results

Under the controlled burst of eighty HTTP requests within a ten-second window, the system produced eighty-one request_ok events and a single anomaly_rate event, yielding an alert-to-request ratio of 0.0123. The anomaly event was triggered precisely at threshold crossing and no additional alerts were generated for subsequent requests within the same window.

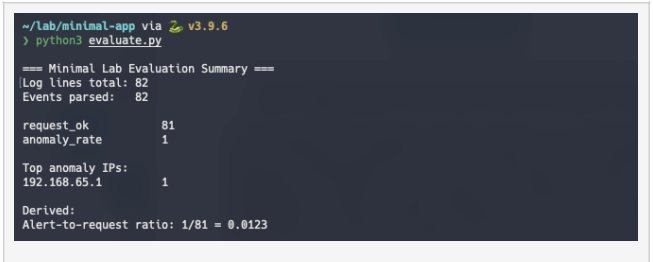

Fig 6: Evaluation summary output for burst condition showing request_ok = 81, anomaly_rate = 1

These results demonstrate deterministic detection behavior, effective suppression of redundant alert generation, and correct enforcement of the sliding-window boundary condition. The single-alert outcome is particularly relevant for resource-constrained SOC environments, where alert fatigue is a significant operational challenge.

B. Benign Traffic Baseline

Under distributed normal traffic conditions consisting of thirty requests spread over approximately fifteen seconds, the system produced thirty request_ok events and zero anomaly events, yielding an alert-to-request ratio of 0.0000. This result confirms that the detection logic correctly discriminates between burst and non-burst traffic patterns and does not produce false positive alerts under baseline conditions. The clean separation between abusive and benign outcomes validates the temporal discrimination properties of the sliding-window model.

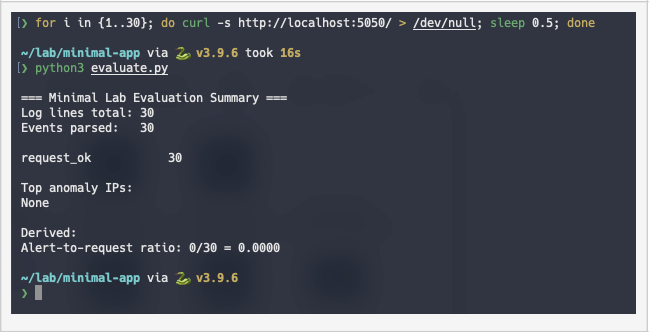

Fig 7: Evaluation summary output for benign condition showing zero anomaly events

C. Detection Behavior Analysis

Taken together, the experimental results demonstrate several properties that directly support the framework's design claims. The detection model is threshold-deterministic, producing exactly one alert per anomalous window. Alert amplification is suppressed through equality-based threshold logic, avoiding the redundant notification patterns that degrade analyst effectiveness. The architecture cleanly separates telemetry generation from detection processing, confirming the modularity principle central to the proposed framework. False positive rates are zero under baseline traffic conditions, suggesting that the rate-based model provides sufficient discrimination for practical deployment without requiring extensive tuning.

These findings validate the framework's core claim that lightweight containerized anomaly detection mechanisms can provide early-stage abuse detection without reliance on full-scale SIEM infrastructure. While the experimental scope is intentionally modest, the behavioral properties observed, namely determinism, amplification control, and false positive suppression, reflect the operational qualities that mid-sized enterprise SOC teams need most.

D. Operational Implications

Although the experimental system represents a simplified instantiation of the broader framework, it validates core architectural principles at the implementation level. The three-container deployment demonstrates telemetry normalization through structured JSON logging, modular detection processing through container-level separation, an independent alerting workflow decoupled from traffic generation, and reproducible deployment via Docker Compose. These properties confirm that the framework's layered design can be operationalized with minimal infrastructure overhead, supporting the paper's central argument that proactive detection capabilities can be incrementally deployed in mid-sized enterprise environments with limited resources.

IX. DISCUSSION AND LIMITATIONS

The proposed framework provides a practical approach to proactive threat detection in mid-sized enterprises, but several limitations should be acknowledged. These limitations stem primarily from operational realities, resource constraints, and the inherent tradeoffs of lightweight security architectures.

One key consideration is the potential for false positives when correlating diverse telemetry sources. Behavioral detection relies on contextual interpretation rather than static signatures, which introduces ambiguity, particularly in dynamic or rapidly changing environments. Organizations with limited baseline data may initially see elevated alert volumes until normal activity patterns are established and detection logic is refined.

The framework's effectiveness depends on the quality and completeness of telemetry sources. Inconsistent log generation, limited retention periods, or gaps in endpoint or identity coverage can reduce correlation fidelity. Smaller organizations may struggle to deploy comprehensive telemetry collection across all assets, especially in hybrid or cloud environments where visibility depends on proper configuration and access controls.

Operational maturity is another constraint. Although the framework emphasizes simplicity and incremental adoption, successful implementation still requires foundational security expertise. Analysts must be able to interpret correlated events, validate hypotheses, and distinguish malicious behavior from benign anomalies. Organizations without dedicated security personnel may need to invest in training or external support to fully realize the framework's benefits.

Scalability is also a consideration. Open-source components can handle moderate data volumes, but increased event ingestion rates may introduce performance and storage challenges if infrastructure resources are limited. While the framework does not prescribe specific technologies, organizations must evaluate capacity planning and maintenance requirements as telemetry coverage expands.

Finally, the framework focuses on detection and investigation rather than automated response. This design choice reflects the operational preferences of many small SOC teams, where excessive automation may obscure detection logic or introduce unintended disruptions. Response actions remain analyst-driven, which may limit speed in certain high-impact scenarios. Despite these limitations, the framework offers a balanced and achievable approach to improving threat detection capabilities in resource-constrained environments.

X. CONCLUSION AND FUTURE WORK

This paper presented a lightweight proactive threat detection framework designed to address the operational and visibility challenges faced by mid-sized enterprises. By emphasizing architectural clarity, telemetry correlation, and analyst-oriented workflows, the framework demonstrates how open-source technologies can be integrated to support proactive security operations without relying on expensive commercial platforms. Rather than focusing on detection algorithms or performance metrics, the framework prioritizes practical design principles that align with contemporary security guidance and real-world SOC constraints.

The proof-of-concept experiment described in Sections VII and VIII provides concrete validation of the framework's architectural principles. The containerized implementation demonstrated deterministic anomaly detection, effective alert amplification control, and zero false positives under benign conditions, all within a modular, reproducible deployment environment. These results reinforce the framework's central claim

that proactive detection capabilities are achievable in resource-constrained settings without dependence on large-scale commercial infrastructure.

The framework highlights the value of correlating identity, endpoint, network, and cloud telemetry to surface behavioral patterns that indicate adversarial activity. Through representative detection scenarios, the paper illustrates how early indicators of compromise can be identified and investigated more effectively when diverse data sources are analyzed together. This approach supports reduced dwell time, improved analyst decision-making, and more consistent response outcomes in resource-constrained environments.

Future work could extend this framework in several directions. Potential enhancements include automated enrichment mechanisms, integration with threat intelligence feeds, and risk-based prioritization to further reduce analyst workload. Additional research might also explore scalable deployment patterns, long-term operational metrics, and controlled evaluations of detection effectiveness across different organizational contexts.

Overall, this work contributes a practical reference architecture that bridges the gap between high-level security frameworks and actionable implementation strategies. By leveraging accessible open-source technologies and emphasizing incremental adoption, the framework offers a viable path for mid-sized enterprises seeking to strengthen proactive threat detection capabilities and improve cybersecurity resilience.

REFERENCES

[1] NIST, Cybersecurity Framework (CSF) 2.0, National Institute of Standards and Technology, 2024.

[2] NIST, CSF 2.0 Resource and Overview Guide, 2024.

[3] NIST, CSF 2.0 Small Business Quick Start Guide, 2024.

[4] NIST, CSF 2.0 Enterprise Risk Management Quick Start Guide, 2024.

[5] NIST, SP 800-92: Guide to Computer Security Log Management, 2006.

[6] NIST, SP 800-137: Information Security Continuous Monitoring, 2011.

[7] NIST, SP 800-137A: ISCM Program Assessment, 2022.

[8] NIST, SP 800-61 Rev. 3: Computer Security Incident Handling Guide, 2024.

[9] NIST, SP 800-53 Rev. 5: Security and Privacy Controls, 2020.

[10] NIST, SP 800-53A Rev. 5: Assessing Security Controls, 2022.

[11] NIST, SP 800-207: Zero Trust Architecture, 2020.

[12] NIST, SP 800-207A: Zero Trust Access Model, 2023.

[13] MITRE, ATT&CK Enterprise Framework, 2024.

[14] CISA, Continuous Diagnostics and Mitigation (CDM) Program, 2023.

[15] CISA, CDM Program Overview, 2023.

[16] ENISA, How to Set Up CSIRT and SOC Capabilities, 2023.

[17] Verizon, 2025 Data Breach Investigations Report, 2025.

[18] Mandiant, M-Trends 2025, 2025.

[19] Microsoft, Digital Defense Report 2024, 2024.

[20] OpenTelemetry, Logs Specification, 2024.

[21] Elastic, Elastic Common Schema Reference, 2024.

[22] Wazuh, Official Documentation, 2024.

[23] Zeek Project, Zeek Documentation, 2024.

[24] Suricata, EVE JSON Output Documentation, 2024.