Project Citadel: Next-Generation Security Analytics & Response Platform

1. Executive Overview

1.1 Purpose of the Report

This report presents the strategic objectives, architectural decisions, implementation details, and resulting security posture achieved through the Citadel MVP Security Platform program.

The goal of this initiative was to design and deploy an integrated SOC ecosystem that delivers comprehensive telemetry ingestion, detection, analytics, response automation, secrets governance, and infrastructure monitoring—entirely using open-source and free-tier technologies.

The document serves to:

- Provide a clear view of program impact to leadership and stakeholders.
- Demonstrate the alignment between requirements and delivered capabilities.
- Record technical and strategic achievements, including current gaps and future roadmap.
- Establish a baseline architecture for scaling the platform in future phases.

1.2 Requirements and Constraints

The environment initially faced fragmented visibility across endpoints, identity systems, network telemetry, and Kubernetes workloads.

Additionally, the use of commercial SIEM, SOAR, and analytics platforms was not feasible due to budget and licensing constraints.

Key constraints included:

- Exclusive reliance on open-source and free-tier tools.
- Single-region deployment with minimal redundancy.
- Limited support for built-in machine learning capabilities in Elastic free tier.
- Manual secret management and no existing SOAR or ticketing platform.

The solution therefore needed to maximize capability within strict technical and financial boundaries, while remaining modular, auditable, and production-ready.

1.3 Strategic Vision

The vision behind Citadel MVP was to create a unified, Kubernetes-compatible SOC platform that:

- Integrates diverse telemetry sources across tenants, Kubernetes, network, and applications.
- Enables real-time detection, analytics, and automated response.
- Provides a structured foundation for threat hunting, insider threat detection, and governance.
- Maintains architectural flexibility to evolve toward enterprise-scale deployments.

This architecture positions the client to expand capabilities incrementally without replacing the underlying platform—future-proofing the SOC while demonstrating measurable impact today.

1.4 SOC KPI Snapshot

KPI	**Value**
Data Sources Integrated	5 (Tenants, Suricata, K8s, Keycloak, SOAR)
VMs Monitored	8
Detection Rules Active	6
Dashboards	6 × ELK, 3 × Grafana
Prometheus Exporters	9 (Node, Elasticsearch, Wazuh, Suricata)
UEBA	Per-user SSH behavioral models
Secrets Managed	Vault (MinIO, Elasticsearch)
Evidence Storage	MinIO
Automated Response	SOAR IP blocking + MinIO evidence trail
Detection Latency	Seconds–minutes

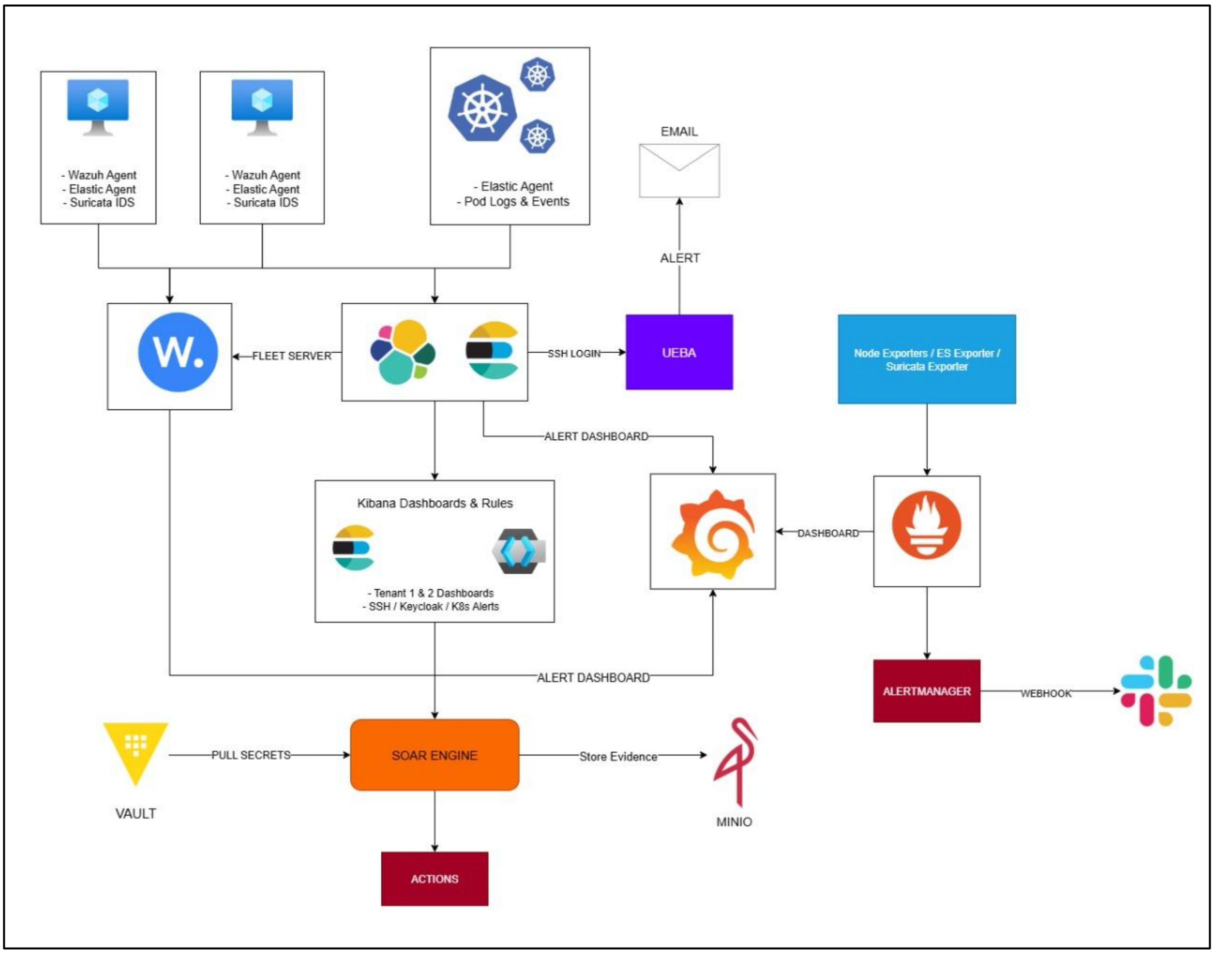
Alert Volume	Moderate, stable

1.5 Key Achievements

- Unified telemetry ingestion from VMs, network, Kubernetes, identity, and SOAR subsystems.
- UEBA anomaly detection for SSH login behavior to detect insider and credential misuse threats.
- Automated response pipeline through the SOAR engine with evidence preservation.
- Centralized secret and evidence governance using HashiCorp Vault and MinIO.
- Infrastructure observability through Prometheus and Grafana with Slack-based alerting.
- Initial compliance alignment with NIST 800-53 controls for auditing and governance.

These achievements demonstrate the ability to build a mature, integrated SOC capability using purely open-source technologies, under real-world constraints.

2. High-Level Architecture Diagram



3. Requirements & Alignment

3.1 Mandate → Implementation → Outcome Mapping

The Citadel MVP program was initiated to address critical visibility, detection, and response gaps across the hybrid environment.

The table below outlines how each key mandate was translated into a specific implementation and measurable outcome:

Mandate	Implementation	Outcome
Centralize telemetry from multi-source environments	ELK Stack (Elasticsearch, Logstash, Kibana) integrated with Fleet Server, Wazuh, Suricata, and Keycloak logs	Unified visibility across tenants, K8s workloads, network telemetry, and identity systems

Detect SSH brute-force and authentication attacks	Tenant-specific ELK detection rules, Suricata IDS, and Wazuh agent logs	Real-time detection of brute-force activity with alerts per tenant
Add behavioral analytics for insider threat detection	UEBA engine using per-user Isolation Forest models on SSH logs	Anomaly-based detection of unusual login times, IPs, and geolocations
Automate response workflows	Kubernetes-deployed SOAR engine with webhook integration from ELK	Automatic IP blocking, evidence logging, and email notifications
Provide secrets and evidence governance	HashiCorp Vault for secret storage, MinIO for evidence retention	Secure, centralized storage for sensitive data and action logs
Enable infrastructure and SOC observability	Prometheus + Grafana with exporters across VMs and services	Real-time monitoring dashboards and Slack alerting for key metrics

This mapping clearly shows direct alignment between requirements and delivered capabilities, a key consideration in CISO and board-level evaluations.

3.2 Core Requirements Status

	Requirement	**Status**
1	Centralized log ingestion across tenants, Kubernetes, and applications	Completed
2	Real-time detection of key attack scenarios	Completed
3	Automated incident response pipeline	Completed
4	UEBA-based behavioral analytics	Completed
5	Secrets and evidence management	Completed
6	Monitoring and observability	Completed
7	Scalable, modular architecture	Completed

All core functional requirements were achieved within the open-source technology constraints and the defined MVP scope.

3.3 Key Constraints

During execution, the team worked within several technical and operational constraints imposed by the environment and tool selection:

Constraint	**Impact**	**Mitigation**
Elastic free-tier limitations	No access to commercial ML or advanced alerting	Custom UEBA engine using Python + Isolation Forest
Single-region deployment	Limited fault tolerance and geo-redundancy	Modular architecture enabling future regional scaling
Manual secret rotation	Potential operational overhead	Future automation planned using Vault dynamic secrets
No commercial SOAR or ITSM platform	Limited integration with ticketing/workflow systems	Custom Kubernetes-based SOAR engine developed
Open-source only policy	No commercial integrations or support	Careful tool selection and modular, API-driven design

These constraints shaped both the architecture and the delivery strategy, emphasizing innovation, modularity, and pragmatic engineering.

3.4 Threat Scenario → Control Mapping

The platform's capabilities were developed to address critical threat scenarios relevant to the client's environment.

The table below demonstrates how specific detection and response mechanisms were mapped to common attack patterns:

Threat Scenario	**Detection Method**	**Response / Containment**	**Tools Involved**

SSH brute-force attack	ELK correlation rules on failed logins (per tenant), Suricata IDS	SOAR-based automatic IP blocking, alerting	Wazuh, Elastic Agent, Suricata, SOAR
Keycloak credential abuse	ELK detection on auth logs, UEBA anomaly scoring	Alert escalation, analyst triage	Keycloak, ELK, UEBA
Network C2 / DNS tunneling	Suricata IDS alerts + ELK log ingestion	Analyst-driven investigation	Suricata, ELK
Insider activity anomalies	UEBA per-user behavioral baselines	High-severity email alerts + triage workflow	UEBA Engine, Streamlit, SMTP

This mapping highlights that Citadel MVP's coverage is not only architectural but threat-driven, aligning engineering work directly with detection and defense priorities.

3.5 Compliance Alignment — NIST 800-53

The implementation was also aligned with selected NIST 800-53 controls to provide a structured governance baseline:

Control ID	**Control Title**	**Citadel MVP Implementation**
AU-6	Audit Review, Analysis, and Reporting	Centralized log ingestion and dashboards via ELK & Wazuh
IR-4	Incident Handling	Automated IP blocking and evidence logging through SOAR
SI-4	System Monitoring	Prometheus exporters and Grafana dashboards
AC-2	Account Management	Keycloak realm management + UEBA login analysis
SC-7	Boundary Protection	Suricata IDS for network perimeter visibility

AU-9	Protection of Audit Information	Vault for secret storage, MinIO for evidence
IR-5	Incident Monitoring	UEBA, ELK detection rules
SI-3	Malicious Code Protection	Suricata IDS + Falco for container activity
PL-2	System and Communications Protection Policy	Modular, API-driven architecture supporting future policy enforcement

4. Engineering & Integration Effort + Architectural Strategy

4.1 System Components & Integration Scope

The Citadel MVP platform integrates a multi-layered security stack spanning telemetry ingestion, detection, analytics, response, governance, and monitoring.

The engineering effort involved connecting these diverse components into a cohesive, real-time security pipeline.

Layer	**Component**	**Purpose**
Ingestion & SIEM	ELK Stack (Elasticsearch, Logstash, Kibana), Wazuh, Elastic Agent Fleet	Centralized log ingestion and storage
Detection & Analytics	Detection rules, Suricata IDS, UEBA Engine	Real-time correlation, network inspection, behavioral anomaly detection
Response Automation	SOAR Engine (Kubernetes-deployed)	Automated alert ingestion, IP blocking, evidence logging, email notifications
Secrets & Evidence	HashiCorp Vault, MinIO	Credential storage, evidence retention, secret lifecycle management
Monitoring & Observability	Prometheus, Grafana, Alertmanager	Infrastructure and SOC observability with alerting

Identity & Access	Keycloak	Centralized authentication and identity governance

The integration spanned 8 VMs, 5 telemetry sources, 6 active detection rules, 9 exporters, and multiple data pipelines across tenants, network, Kubernetes, and applications.

4.2 Key Engineering Challenges and Solutions

The integration process surfaced several technical challenges that required thoughtful engineering solutions:

Challenge	**Description**	**Solution Implemented**
Heterogeneous Telemetry Sources	Logs and metrics originated from tenants, Suricata, Keycloak, Kubernetes, and custom SOAR pipelines	Implemented centralized ELK ingestion with tenant-specific Fleet agents and standardized exporter configurations
Suricata IDS Integration	Aligning network telemetry with SIEM for correlation	Configured Suricata exporters and ingestion rules into ELK for real-time network alerting
Kubernetes Log Visibility	Collecting pod and container telemetry efficiently	Deployed Elastic Agent as a DaemonSet within the K8s cluster for full workload visibility
Credential & Secret Handling	Preventing credential sprawl across multiple components	Centralized secrets in Vault and configured SOAR + UEBA to pull secrets dynamically
UEBA Feature Engineering	Extracting and processing login behavior features from SSH logs	Implemented per-user baselining, time/IP/geo feature generation, and Isolation Forest models
Prometheus Exporter Management	Monitoring diverse nodes and services	Standardized exporter deployment across 6 nodes, plus Suricata, Elasticsearch, and Wazuh exporters

Open-Source Constraints	Elastic free-tier lacked advanced features	Built custom UEBA engine, custom SOAR, and modular integrations to replicate key enterprise functions

These solutions collectively resulted in a fully functioning SOC pipeline using only open-source technologies, with zero reliance on commercial security platforms.

4.3 Design Decisions Under Constraints

Throughout the implementation, critical design trade-offs were made to balance capability, cost, and scalability:

Decision Point	**Option Chosen**	**Rationale**
SOAR Platform	Custom Python/Kubernetes-based engine	Commercial SOAR was not viable; custom build provided control and flexibility
UEBA Engine	Custom Python Isolation Forest models	Elastic ML features unavailable in free tier; required custom ML pipeline
Telemetry Aggregation	Elastic Fleet + Wazuh agents	Allowed multi-source collection with low overhead
Monitoring Stack	Prometheus + Grafana	Best open-source combination for real-time infra visibility
Secrets Management	Vault + MinIO	Balanced security, cost, and integration simplicity
Modular Integration	API-first, componentized architecture	Enables future scaling, technology swaps, and regional expansion

These decisions reflect a deliberate engineering strategy to achieve maximum impact within constrained environments.

4.4 Architectural Strategy

The Citadel MVP platform was architected using five foundational principles to ensure modularity, resilience, and extensibility:

1. **Open Integration** – All components expose APIs and use standard protocols (HTTP, JSON, syslog, exporters), enabling plug-and-play extensibility.
2. **Modular Subsystems** – SIEM, SOAR, UEBA, monitoring, and secrets are loosely coupled but tightly integrated. Each can evolve independently.
3. **Kubernetes-Native Design** – SOAR and UEBA components are deployed on Kubernetes, enabling horizontal scalability and modern DevSecOps workflows.
4. **Built-in Governance** – Secrets and evidence handling are first-class citizens through Vault and MinIO, not afterthoughts.
5. **Future-Proofing** – Architecture anticipates future commercial integrations, multi-region deployments, and compliance expansions without redesign.

4.5 Architectural Layers Overview

Layer	**Function**	**Key Components**
Ingestion	Collect and centralize logs & metrics	Elastic Fleet, Wazuh, Suricata, Keycloak
Detection & Analytics	Correlate events, detect anomalies	ELK rules, UEBA engine, Suricata IDS
Response	Automate containment & evidence	SOAR engine, Vault, MinIO
Governance	Manage credentials & audit data	Vault, MinIO
Monitoring	Observe system health, infra & SOC	Prometheus, Grafana, Alertmanager

This layered architecture ensures clean separation of concerns while maintaining tight operational integration.

4.6 Integration Outcome

The result of this integration strategy is a fully operational SOC MVP with:

- End-to-end telemetry pipelines spanning tenants, network, and Kubernetes.
- Multi-vector detection using rules, IDS, and UEBA.
- Automated containment workflows integrated with evidence governance.
- Real-time observability across all infrastructure components.
- Modular design enabling future scaling and upgrades without architectural overhaul.

5. Implementation Summary — Subsystem Highlights

5.1 SIEM & Ingestion

5.1.1 Objective & Role

The SIEM & Ingestion subsystem provides the central telemetry backbone for Citadel MVP.

Its purpose is to collect, normalize, and index security-relevant data from heterogeneous sources across the environment — including tenants, network IDS, Kubernetes workloads, authentication services, and response pipelines — to enable real-time detection, analytics, and incident response.

5.1.2 Infrastructure Overview

Component	**Function**
ELK Stack (Elasticsearch, Logstash, Kibana)	Centralized storage, correlation, and visualization of logs
Elastic Fleet Server	Hosted on Wazuh VM; manages agent enrollment and telemetry flow
Wazuh Agents	Deployed on tenant VMs; provide endpoint telemetry
Elastic Agents	Deployed on tenant VMs and Kubernetes cluster; collect system, container, and application logs
Suricata IDS	Provides network traffic analysis and threat detection data

Keycloak Logs	Authentication events ingested from both Tenant 1 & Tenant 2 web applications

VM Distribution:

- citadel-elk: Core SIEM (Elasticsearch, Logstash, Kibana)
- citadel-detect: Suricata + Wazuh
- citadel-kube: Kubernetes agents
- tenant1 & tenant2: Wazuh agents, Elastic agents, Suricata sensors, and application logs

5.1.3 Data Sources Integrated

Source	Log Type / Data	Destination
Tenant 1	System, auth, Suricata, Keycloak app logs	Elasticsearch via Fleet
Tenant 2	System, auth, Suricata, Keycloak app logs	Elasticsearch via Fleet
Kubernetes	Container & pod logs	Elasticsearch via Elastic Agent
Suricata	Network IDS alerts	ELK detection pipeline
SOAR Engine	Alert & action logs	Citadel logs index for auditing

All data sources are normalized into Elasticsearch indexes with clear naming conventions and retention policies. This creates consistent visibility across infrastructure layers.

5.1.4 Detection Rules Implemented

Rule Name	Description
Brute Force Alert – Tenant 1	Detects repeated failed SSH logins from a single source on Tenant 1
Brute Force Alert – Tenant 2	Same as above, scoped to Tenant 2
Keycloak Authentication Alert – Tenant 1	Flags repeated failed login attempts and unusual login patterns

Keycloak Authentication Alert – Tenant 2	Same as above for Tenant 2
Pod Monitoring Rule	Detects anomalies or critical events in Kubernetes pods
Suricata Network Alert	Detects suspicious network activity including DNS tunneling and known signatures

These rules provide multi-vector coverage across endpoint, identity, network, and container layers.

5.1.5 Dashboards & Visualizations

Kibana Dashboards Deployed:

1. Kubernetes Monitoring Dashboard
2. Suricata IDS Dashboard
3. Keycloak Authentication Dashboard – Tenant 1
4. Keycloak Authentication Dashboard – Tenant 2
5. SSH Login Activity Dashboard
6. Elastic Security Detection Rule Monitoring

Grafana Dashboards Deployed:

1. Elasticsearch Alerts Dashboard
2. Wazuh Alerts Dashboard
3. Prometheus Infrastructure Metrics Dashboard

These dashboards enable real-time SOC visibility and incident triage for both network defenders and platform engineers.

5.1.6 Achievements & Impact

- Unified ingestion pipeline covering 5 data types across 8 VMs and 1 Kubernetes cluster.
- Tenant-specific visibility using dedicated detection rules and dashboards.

- Network + Identity + Endpoint + K8s telemetry integrated into a single correlation plane.
- Clear visualizations for analysts through structured dashboards.
- SOAR integration hooks to trigger automated responses directly from SIEM alerts.

5.1.7 Constraints & Considerations

- Elastic free tier required custom engineering for some features (e.g., advanced ML detection not available).
- Scaling retention beyond MVP will require additional storage and ILM tuning.
- Alerting integrations rely on webhook-based SOAR due to lack of commercial integrations.

5.2 UEBA & AI Model

5.2.1 Objective & Role

The User and Entity Behavior Analytics (UEBA) subsystem augments traditional rule-based detections with behavioral anomaly detection to identify insider threats, credential misuse, and subtle account compromise patterns.

It focuses on per-user SSH login behavior across Linux systems, learning normal activity patterns and flagging deviations that indicate potential compromise.

This subsystem was intentionally designed as a custom AI pipeline, as the Elastic Stack's built-in ML capabilities are unavailable in the free tier. It provides SOC analysts and stakeholders with early detection of abnormal behaviors that may bypass signature-based rules.

5.2.2 Data Sources & Scope

Source	**Index / Data Type**	**Fields Used**
Linux SSH logs (Tenant 1 & 2)	logs-system.auth-* (ingested via Elastic Agent)	@timestamp, user.name, event.outcome, source.ip, source.geo., *source.as.*, host.name
Historical data	Last 30 days per user	Used to establish behavioral baselines

Real-time data	12–24h rolling window	Used for anomaly scoring & dashboards

- Scope: SSH login events on all tenant Linux hosts
- Timezone: Asia/Kolkata
- Behavioral window: 30-day training baseline, daily scoring

5.2.3 Feature Engineering & Modeling

For each user, the pipeline computes a feature vector capturing multiple behavioral dimensions:

Feature Category	**Description**
Time-based	Login hour, weekday/weekend, off-hours flag
Geo/IP/ASN	Country/region changes, ASN changes, distance from last login, IP rarity
Auth Behavior	Failure ratios, auth method shifts, failure→success sequences
Host Behavior	Target host frequency & rarity
Threat Intel Enrichment	IP reputation from VirusTotal and OTX

Modeling approach:

- Per-user Isolation Forest models are trained on the 30-day baseline for each user.
- Fallback global model handles low-activity accounts.
- Anomaly scores are normalized [0–1] and mapped to severity levels:
    - 0–0.4 = Low
    - 0.4–0.7 = Medium
    - 0.7–0.9 = High
    - 0.9+ = Critical

Each anomaly is accompanied by explanation tags (e.g., "New ASN", "Off-hours login", "Unseen IP"), increasing analyst interpretability.

5.2.4 Dashboards & Analyst Views

The UEBA subsystem provides role-based Streamlit dashboards:

Stakeholder View (Executive KPIs)

- KPI Tiles: anomalies (24h), affected users, critical alerts, distinct IPs/countries
- Severity trend time series
- Heatmap (hour × weekday) of activity patterns
- Geo map of anomalous IP sources

SOC Analyst View

- Ranked anomaly table with severity and explanation badges
- Filters: time, user, host, country, severity
- Per-user drill-down: 30-day baseline vs recent events
- Event-level detail: raw fields, derived features, enrichment, explanation list
- Export options: CSV & PDF

The dashboards adopt a SOC-native dark theme, include tooltips and glossary references, and are optimized for <2s P95 load times.

5.2.5 Alerts & Notifications

High and Critical anomalies trigger formal email alerts via SMTP (Mailgun).

The HTML template includes:

- Incident ID & severity (color-coded)
- User, host, source IP, geo/ASN
- Reputation results (VT & OTX)
- Explanation tags & analyst recommendation
- SOC contact info

Rate-limiting ensures a maximum of one email per user per hour to avoid alert fatigue.

Analysts can also trigger test alerts manually from the dashboard UI.

5.2.6 Achievements & Impact

- Custom UEBA pipeline built entirely on Python, Streamlit, and Elasticsearch — no commercial ML features used.
- Per-user anomaly detection identifies subtle credential misuse and insider activity beyond rule-based systems.
- Geo/IP-based baselining detects travel, VPN, and malicious IP anomalies.
- Operational dashboards provide fast investigation & contextual understanding.
- Automated alerting ensures SOC analysts are immediately informed of critical anomalies.
- All anomalies are logged for audit and can be correlated with SOAR actions and evidence stored in MinIO.

5.2.7 Constraints & Considerations

- ML model performance depends on data availability per user (sparse activity uses fallback models).
- No built-in Elastic ML → requires maintenance of custom Python pipeline.
- Feature enrichment (VT, OTX) depends on external lookups, requiring rate control.
- UEBA currently covers SSH logins only — expansion to other data types is planned in future phases.

5.3 SOAR Engine

5.3.1 Objective & Role

The Security Orchestration, Automation, and Response (SOAR) subsystem is responsible for automating incident response workflows, integrating detections with actionable containment, and maintaining a full audit trail of SOC actions.

Its primary goal is to reduce Mean Time to Respond (MTTR) by executing predefined actions automatically when alerts are triggered, ensuring rapid containment and consistent response execution across environments.

The SOAR engine was built as a lightweight, Kubernetes-native microservice to overcome the absence of commercial SOAR or ITSM platforms in the open-source environment.

5.3.2 Architecture & Integration

Component	**Role**
SOAR Engine (Kubernetes)	Core microservice for alert ingestion and response actions
ELK Stack	Generates alerts from detection rules; sends them to SOAR via webhooks
Vault	Provides secure access to credentials and secrets for action execution
MinIO	Stores action logs and forensic evidence from automated responses
Mailgun SMTP	Used for email alerts and stakeholder notifications

Workflow Overview:

1. A detection rule in ELK fires (e.g., brute-force attack).
2. ELK alerting sends the event payload via webhook to the SOAR engine's exposed endpoint.
3. The SOAR engine parses the alert, classifies severity, and selects appropriate response playbooks.
4. Actions are executed (e.g., block IP, isolate host, store evidence).
5. Results are logged back to ELK and archived in MinIO for forensic purposes.
6. Email notifications are dispatched to stakeholders and SOC analysts.

5.3.3 Automated Playbooks

Playbook Name	**Trigger**	**Action**	**Outcome**
SSH Brute-Force Containment	Brute-force detection rule fired from ELK	Add offending IP to firewall blocklist (iptables) on affected tenant	Immediate attack surface reduction
Keycloak Abuse Alert Escalation	Keycloak failed auth rule	Alert escalation to analysts via email and MinIO log entry	Rapid human triage
UEBA High/Critical Anomaly	UEBA engine anomaly webhook	Send formal HTML alert to SOC analysts, log event	Insider threat visibility
Suricata High-Severity Alert	Network IDS alert received	Notify SOC team, archive packet evidence if applicable	Network threat escalation

These playbooks standardize responses for high-frequency or high-impact alerts, ensuring predictable, repeatable, and auditable incident handling.

5.3.4 Evidence Logging & Audit Trail

Every action executed by the SOAR engine is logged in two locations:

- **Elasticsearch Index (citadel-actions)** — enables SOC dashboards and timeline correlation.
- **MinIO Object Storage** — retains raw alert payloads, action outputs, and any related evidence (e.g., IP block commands, packet captures) for forensic and compliance use.

This dual logging ensures a tamper-resistant and easily searchable audit trail of SOC response activity.

5.3.5 Alerting & Notifications

The SOAR engine sends automated email alerts for critical events using Mailgun SMTP.

Each email includes:

- Alert details (ID, severity, source)
- Actions taken

- Links to Kibana dashboards or evidence files
- SOC contact information

Notifications are rate-limited and formatted using a professional HTML template for consistency and audit readiness.

5.3.6 Achievements & Impact

- End-to-end integration between ELK detection rules and automated containment workflows.
- Reduced response times for high-frequency attack types like SSH brute force.
- Forensic traceability through MinIO evidence retention and ELK action indices.
- Scalable microservice architecture enabling future playbook expansion.
- Secure credential handling through Vault integration, ensuring secrets are never hardcoded.
- Professional alerting pipeline that keeps analysts and stakeholders informed in real time.

The result is a responsive SOC automation layer that materially improves operational efficiency and readiness.

5.3.7 Constraints & Considerations

- Current SOAR engine supports webhook ingestion only no native integration with commercial ITSM or TIP systems.
- Playbook coverage is focused on high-impact, high-frequency scenarios; further expansion is planned for long-tail use cases.
- Scaling to handle high alert volumes will require queuing and distributed processing enhancements in future phases.
- IP blocking uses iptables on tenants; more advanced response mechanisms can be introduced later

5.4 SOAR Engine

5.4.1 Objective & Role

The Security Orchestration, Automation, and Response (SOAR) subsystem is responsible for automating incident response workflows, integrating detections with actionable containment, and maintaining a full audit trail of SOC actions.

Its primary goal is to reduce Mean Time to Respond (MTTR) by executing predefined actions automatically when alerts are triggered, ensuring rapid containment and consistent response execution across environments.

The SOAR engine was built as a lightweight, Kubernetes-native microservice to overcome the absence of commercial SOAR or ITSM platforms in the open-source environment.

5.4.2 Architecture & Integration

Component	**Role**
SOAR Engine (Kubernetes)	Core microservice for alert ingestion and response actions
ELK Stack	Generates alerts from detection rules; sends them to SOAR via webhooks
Vault	Provides secure access to credentials and secrets for action execution
MinIO	Stores action logs and forensic evidence from automated responses
Mailgun SMTP	Used for email alerts and stakeholder notifications

Workflow Overview:

1. A detection rule in ELK fires (e.g., brute-force attack).
2. ELK alerting sends the event payload via webhook to the SOAR engine’s exposed endpoint.
3. The SOAR engine parses the alert, classifies severity, and selects appropriate response playbooks.
4. Actions are executed (e.g., block IP, isolate host, store evidence).
5. Results are logged back to ELK and archived in MinIO for forensic purposes.
6. Email notifications are dispatched to stakeholders and SOC analysts.

5.4.3 Automated Playbooks

Playbook Name	Trigger	Action	Outcome
SSH Brute-Force Containment	Brute-force detection rule fired from ELK	Add offending IP to firewall blocklist (iptables) on affected tenant	Immediate attack surface reduction
Keycloak Abuse Alert Escalation	Keycloak failed auth rule	Alert escalation to analysts via email and MinIO log entry	Rapid human triage
UEBA High/Critical Anomaly	UEBA engine anomaly webhook	Send formal HTML alert to SOC analysts, log event	Insider threat visibility
Suricata High-Severity Alert	Network IDS alert received	Notify SOC team, archive packet evidence if applicable	Network threat escalation

These playbooks standardize responses for high-frequency or high-impact alerts, ensuring predictable, repeatable, and auditable incident handling.

5.4.4 Evidence Logging & Audit Trail

Every action executed by the SOAR engine is logged in two locations:

- **Elasticsearch Index (citadel-actions)** — enables SOC dashboards and timeline correlation.
- **MinIO Object Storage** — retains raw alert payloads, action outputs, and any related evidence (e.g., IP block commands, packet captures) for forensic and compliance use.

This dual logging ensures a tamper-resistant and easily searchable audit trail of SOC response activity.

5.4.5 Alerting & Notifications

The SOAR engine sends automated email alerts for critical events using Mailgun SMTP. Each email includes:

- Alert details (ID, severity, source)

- Actions taken
- Links to Kibana dashboards or evidence files
- SOC contact information

Notifications are rate-limited and formatted using a professional HTML template for consistency and audit readiness.

5.4.6 Achievements & Impact

- End-to-end integration between ELK detection rules and automated containment workflows.
- Reduced response times for high-frequency attack types like SSH brute force.
- Forensic traceability through MinIO evidence retention and ELK action indices.
- Scalable microservice architecture enabling future playbook expansion.
- Secure credential handling through Vault integration, ensuring secrets are never hardcoded.
- Professional alerting pipeline that keeps analysts and stakeholders informed in real time.

The result is a responsive SOC automation layer that materially improves operational efficiency and readiness.

5.4.7 Constraints & Considerations

- Current SOAR engine supports webhook ingestion only — no native integration with commercial ITSM or TIP systems.
- Playbook coverage is focused on high-impact, high-frequency scenarios; further expansion is planned for long-tail use cases.
- Scaling to handle high alert volumes will require queuing and distributed processing enhancements in future phases.
- IP blocking uses iptables on tenants; more advanced response mechanisms can be introduced later.

5.5 Monitoring & Kubernetes

5.5.1 Objective & Role

The Monitoring & Kubernetes subsystem ensures that both the Citadel platform components and underlying infrastructure are continuously observable, measurable, and proactively alerting.

Its primary goals are:

- To provide real-time visibility into system performance and health metrics.
- To enable early detection of infrastructure issues that could affect security operations.
- To monitor Kubernetes workloads, ensuring the resilience of security microservices such as SOAR and UEBA.
- To support SOC operators and platform engineers with actionable dashboards and automated alerts.

This subsystem is critical because security controls are only effective if the platform itself is healthy and reliable.

5.5.2 Architecture & Components

Component	**Role**
Prometheus	Core metrics collection and storage engine. Scrapes data from exporters across all VMs and services.
Grafana	Visualization layer providing real-time monitoring dashboards and alerting rules.
Alertmanager	Handles alert routing and notification (e.g., Slack).
Exporters	Expose system and service metrics to Prometheus. Includes Node Exporter, Elasticsearch Exporter, Suricata Exporter, and Wazuh Exporter.
Kubernetes	Hosts Citadel microservices (SOAR, UEBA). Monitored via Elastic Agent and Prometheus metrics.

Deployment Overview:

- All monitoring components (Prometheus, Grafana, Alertmanager) are hosted on citadel-monitor VM.

- Exporters are deployed across all platform VMs, ensuring unified coverage.
- Grafana dashboards are accessible via secure web UI and are used daily for operational checks.
- Kubernetes workloads are monitored both through Elastic Agent telemetry and Prometheus metrics scraping.

5.5.3 Exporter Deployment

VM / Service	Exporters Deployed
citadel-kube	Node Exporter
citadel-elk	Node Exporter, Elasticsearch Exporter
citadel-detect	Node Exporter, Suricata Exporter
citadel-secrets	Node Exporter
citadel-ueba	Node Exporter
citadel-monitor	Node Exporter, Prometheus self-scrape

Additional Notes:

- Suricata Exporter exposes packet statistics and IDS performance metrics.
- Elasticsearch Exporter exposes cluster health, indexing rates, and latency.
- All exporters are scraped at regular intervals and visualized in Grafana.

5.5.4 Dashboards & Visualizations

Grafana Dashboards Deployed:

1. Node Exporter Full (ID: 1860) — CPU, memory, disk, network metrics for all nodes.
2. Elasticsearch Health (ID: 4358) — cluster health, indexing rates, query latency.
3. Suricata Performance (ID: 15996) — IDS packet processing and alert metrics.

Kibana Dashboards:

- Kubernetes Monitoring Dashboard — monitors pod activity, resource usage, and container logs from the Elastic Agent in the K8s cluster.

These dashboards allow SOC and platform teams to quickly identify anomalies, correlate system behavior with security events, and maintain situational awareness across the stack.

5.5.5 Alerting Pipeline

Alertmanager is configured to route alerts from Prometheus to Slack channels for real-time visibility.

Typical alerts include:

- Node down / exporter unavailable
- High CPU/memory usage on critical nodes
- Elasticsearch cluster degradation
- Suricata exporter downtime
- Pod failures in Kubernetes

Alerting configuration is modular and can easily be extended to email, webhook, or ticketing integrations in future phases.

5.5.6 Kubernetes Integration

Kubernetes hosts critical security microservices (e.g., SOAR engine, UEBA).

Monitoring focuses on:

- Pod availability — Ensuring all security microservices are running without crash loops or restarts.
- Resource utilization — Detecting pod resource exhaustion early.
- Logging and events — Captured via Elastic Agent deployed as a DaemonSet.
- Prometheus metrics — Used to measure pod uptime, response times, and scaling behavior.

This integration ensures that the security functions deployed inside Kubernetes remain observable and reliable.

5.5.7 Achievements & Impact

- Unified observability stack covering nodes, services, IDS, Elasticsearch, and Kubernetes.
- Real-time operational dashboards that reduce mean time to detect infra failures.
- Early warning alerts for system degradations, preventing downstream SOC disruptions.
- Integrated Kubernetes monitoring, ensuring microservice health and responsiveness.
- Slack-based alerting provides instant visibility for platform teams.
- Baseline metrics established, enabling performance trend analysis and capacity planning.

This monitoring layer significantly improves operational resilience, ensuring that SOC detection, response, and analytics remain continuously available.

5.5.8 Constraints & Considerations

- Current alert routing is Slack-only; ticketing or incident platforms can be integrated in future phases.
- Prometheus and Grafana are single-instance deployments; high availability will be required for scale-out.
- Exporter coverage is comprehensive but alert tuning requires periodic review to avoid noise.
- Kubernetes monitoring currently focuses on pod availability and resource usage; future work can add service-level SLO monitoring.

6. SOC Posture Evolution — Before vs After

6.1 Strategic Context

Prior to the Citadel MVP implementation, the security operations were fragmented, reactive, and lacked integrated visibility across different layers of the infrastructure.

Telemetry was scattered across individual systems, detection coverage was limited to basic endpoint logs, and incident response was primarily manual and time-consuming.

The MVP deployment introduced a unified, modular SOC platform leveraging open-source technologies to centralize telemetry, enhance detection depth, automate response workflows, and align operations with compliance frameworks — all within the constraints of the free-tier ecosystem.

6.2 Before vs After — SOC Capability Transformation

Capability Area	**Before Citadel MVP**	**After Citadel MVP**
Telemetry Ingestion	Logs collected manually from endpoints; no centralized SIEM	Centralized ingestion from tenants, network (Suricata), Kubernetes, Keycloak, and SOAR via ELK + Fleet
Detection Coverage	Basic endpoint-focused detections only	Multi-vector detection across endpoint, network, identity, and container layers with UEBA integration
Behavioral Analytics	None	Per-user SSH anomaly detection using custom Isolation Forest UEBA engine
Response Automation	Manual response, inconsistent actions, long MTTR	Automated playbooks via SOAR engine for IP blocking, escalation, evidence logging, and alerting
Governance & Evidence	No centralized secret or evidence management	Vault-based secrets handling and MinIO-based evidence retention with ELK indexing
Monitoring & Observability	Minimal system checks; no proactive alerts	Prometheus, Grafana, Alertmanager + Kubernetes telemetry for real-time operational visibility
Compliance Alignment	Minimal; fragmented audit information	Structured mapping to NIST 800-53 controls (AU-6, IR-4, SI-4, AC-2, SC-7, AU-9, IR-5, SI-3, PL-2)
Scalability & Extensibility	Ad hoc integrations, no modularity	API-driven, Kubernetes-native modular architecture enabling phased scaling

Incident Response Speed	High MTTR, human bottlenecks	Reduced MTTR through automated containment and immediate analyst notification
Threat Hunting & Analytics	Limited; no data centralization	Central data lake + dashboards enabling network, identity, and behavioral hunting

6.3 Key Improvements Summary

Dimension	**Impact**
Detection Depth	Expanded to include network IDS, identity-based rules, and behavioral analytics, allowing earlier detection of complex threats.
Response Efficiency	Automated SOAR playbooks replaced manual containment, cutting response time from minutes to seconds.
Visibility & Monitoring	Full-stack observability through Prometheus & Grafana reduced platform blind spots and improved reliability.
Compliance Posture	Alignment with NIST controls established a governance baseline for future audits and certifications.
Operational Maturity	Transitioned from reactive to proactive SOC operations, supported by centralized data and automated workflows.

6.4 Strategic Impact

The Citadel MVP deployment has materially advanced the client's SOC maturity level, even within significant cost and technology constraints.

Key strategic impacts include:

- Foundation for scalable SOC operations — Modular architecture supports phased enhancements (e.g., TI feeds, multi-region, advanced analytics).
- Operational speed and consistency — Automated playbooks ensure standardized, rapid incident containment.

- Data-driven decision-making — Centralized telemetry and dashboards enable better threat hunting and incident investigation.
- Enhanced insider threat detection — UEBA introduces behavioral analytics capability previously unavailable.
- Improved audit readiness — Evidence retention and compliance mapping address key regulatory expectations.

This transformation represents a shift from fragmented, reactive operations to a cohesive, data-driven SOC posture — achieved using only open-source technologies and deliberate architectural strategy.

7. Risk Register & Known Gaps

7.1 Purpose

While the Citadel MVP platform has achieved significant security and operational advancements, it was developed under realistic constraints — both technological and budgetary.

This section outlines the known risks and current gaps identified during implementation and early operations. Documenting these transparently supports risk-informed decision making, guides future improvements, and demonstrates maturity in governance practices.

7.2 Current Risks & Gaps

	Risk / Gap	**Description**	**Impact**	**Mitigation / Roadmap**
1	Elastic Free-Tier Limitations	Elastic Stack free tier lacks advanced ML, multi-node scaling, and native alerting features	Reduced detection sophistication and scalability	Custom UEBA built; future phases can integrate commercial Elastic features or hybrid SIEM
2	Manual Secret Rotation	Vault secrets are currently rotated manually	Potential operational risk if	Implement Vault dynamic secrets or

			credentials are not rotated regularly	scheduled rotation policies
3	Single-Region Deployment	All components are deployed in a single region	Limited fault tolerance and disaster recovery	Future roadmap includes multi-region replication for Vault, MinIO, ELK, and Prometheus
4	Limited SOAR Playbook Coverage	Current playbooks cover high-frequency scenarios only (e.g., SSH brute force, Keycloak abuse)	Some attack scenarios still require manual response	Expand playbook library, integrate advanced containment actions and enrichment steps
5	Alert Routing Limited to Slack	Prometheus and some SOC alerts are routed only to Slack	Lack of integration with ITSM or ticketing tools limits escalation paths	Future integrations planned with ServiceNow, Jira, or equivalent ticketing systems
6	Exporter & Alert Tuning	Initial Prometheus alert rules are generic and may produce noise	Alert fatigue, possible delayed responses	Periodic review and tuning of alert rules; implement severity-based alerting
7	Sparse UEBA Training Data	Low-activity accounts rely on global fallback models	Slightly reduced accuracy for anomaly detection	Expand data collection windows; integrate additional behavioral sources (e.g., VPN, app usage)
8	Kubernetes Monitoring Scope	Current monitoring focuses on pod availability and resource usage	Limited service-level insights	Add service-level SLO/SLA monitoring and deeper security telemetry

9	Forensic Data Retention Policies	MinIO retention is manually configured; no automated evidence lifecycle management	Risk of data sprawl or premature deletion	Define retention policies, automate lifecycle management in MinIO
10	No TI / Threat Feed Integration	Current detection relies on rules and UEBA, no external threat intel	Potential gaps in detecting emerging threats	Integrate open-source TI feeds or commercial threat intel for Suricata and SIEM

7.3 Strategic Risk Themes

Beyond individual items, several strategic risk themes were identified:

- Scalability & Resilience — MVP deployment is single-region and single-instance for several components. Scaling up will require architectural expansion and HA deployments.
- Detection Depth vs. Open-Source Limits — Free-tier Elastic and open-source tools necessitated custom engineering for ML and SOAR. This introduces maintenance overhead but provides flexibility.
- Governance Maturity — Secrets, retention, and alerting are functional but still operate at MVP level. Future phases should focus on policy automation and operational hardening.
- Integration Ecosystem — Alert routing and ticketing are not yet fully integrated with enterprise workflows. Bridging this gap is key to operational scaling.

7.4 Mitigation Roadmap Snapshot

Theme	**Immediate (0–3 mo)**	**Mid-Term (3–6 mo)**	**Long-Term (6+ mo)**

Elastic & Detection	Optimize existing rules, expand UEBA coverage	Evaluate hybrid SIEM options	Integrate commercial Elastic or equivalent SIEM features
Secrets & Governance	Formalize manual rotation procedures	Implement dynamic secrets, MinIO retention policies	Full lifecycle automation and policy enforcement
SOAR & Automation	Add new playbooks for frequent scenarios	Integrate TI enrichment in playbooks	Connect with ticketing/ITSM systems for end-to-end workflows
Monitoring & Infra	Tune Prometheus alerts	Add service-level monitoring	Deploy HA/replicated Prometheus-Grafana stack
Threat Intelligence	Start with open-source feeds for Suricata	Integrate curated TI for SIEM	Build TI correlation & enrichment pipelines

7.5 Governance Note

All identified risks and gaps have been formally logged in the Citadel program's internal risk register. Ownership for mitigation has been assigned to relevant teams (Platform, Defense, Intelligence).

These risks are not blockers but represent expected maturity gaps for an MVP built under open-source constraints. They are being addressed through planned, phased improvements as the platform matures.

8. Compliance & Governance

8.1 Purpose

Compliance and governance are central pillars of the Citadel MVP security platform. From inception, the platform was designed to map operational security controls to recognized standards, enabling traceability, auditability, and alignment with an client compliance requirements.

The chosen baseline for this project is NIST SP 800-53 Rev. 5, which provides a comprehensive set of security and privacy controls for federal information systems and organizations. The Citadel MVP leverages open-source tooling and custom integrations to meet relevant control families within the constraints of the MVP scope.

8.2 Compliance Framework Selection

Framework	Relevance	Implementation Approach
NIST SP 800-53 Rev. 5	Widely adopted by enterprises and government; forms backbone for other frameworks (e.g., FedRAMP, ISO 27001 mapping)	Mapped controls to platform components and security workflows
MITRE ATT&CK	Operational framework for adversary behaviors and detection engineering	Used to guide threat scenarios, detection logic, and response playbooks
Elastic Common Schema (ECS)	Standard for SIEM data normalization	Adopted across ingestion pipelines for consistent log structuring
SOC Playbooks (Internal)	Client-specific SOC operational standards	Used to align incident response flows and detection rules with real SOC workflows

8.3 NIST 800-53 Control Mapping

The table below highlights a representative mapping between key NIST 800-53 control families and the corresponding Citadel MVP components or workflows:

Control Family	Relevant Controls	Citadel Implementation
Access Control (AC)	AC-2 (Account Management), AC-3 (Access Enforcement), AC-7 (Unsuccessful Login Attempts)	Keycloak for centralized auth; SSH brute-force detection in ELK; UEBA anomaly detection for user behavior

Audit & Accountability (AU)	AU-2 (Audit Events), AU-6 (Audit Review), AU-12 (Audit Generation)	Comprehensive log ingestion via Elastic Agent + Wazuh; dashboards for SSH, Keycloak, Suricata; SOAR audit trails in citadel-actions index
Incident Response (IR)	IR-4 (Incident Handling), IR-5 (Incident Monitoring), IR-6 (Incident Reporting)	SOAR engine integrated with ELK for automated responses, email notifications, MinIO evidence storage
Risk Assessment (RA)	RA-3 (Risk Assessment), RA-5 (Vulnerability Scanning)	Risk register documented; Prometheus and Suricata telemetry feeds; alert rules to detect high-risk behaviors
System & Communications Protection (SC)	SC-7 (Boundary Protection), SC-18 (Mobile Code), SC-23 (Session Authenticity)	Suricata for network IDS; Keycloak session controls; Prometheus node exporters for host telemetry
System & Information Integrity (SI)	SI-4 (Information System Monitoring), SI-7 (Software, Firmware, and Information Integrity)	ELK + Suricata integration for threat detection; Prometheus monitoring; UEBA for anomaly detection
Configuration Management (CM)	CM-2 (Baseline Configuration), CM-6 (Configuration Settings)	Version-controlled Kubernetes manifests, Kyverno for baseline security policies, Prometheus exporter configurations
Identification & Authentication (IA)	IA-2 (Identification and Authentication), IA-5 (Authenticator Management)	Keycloak realm management, SSH login monitoring, authentication dashboards
Contingency Planning (CP)	CP-2 (Contingency Planning), CP-9 (Information System Backup)	MinIO object storage for alerts & actions; documented multi-region roadmap for future resilience

Note: This mapping is representative and MVP-scoped. A full compliance audit would require a control-by-control analysis, evidence gathering, and continuous monitoring integration.

8.4 Threat Scenario Coverage

The platform's control implementation and detection workflows are directly validated through realistic threat scenarios mapped to MITRE ATT&CK tactics and NIST families:

Threat Scenario	**MITRE Techniques**	**NIST Family**	**Detection / Mitigation**
SSH Brute Force	T1110 – Brute Force	AC, AU, IR	SSH failed-login rules per tenant, Prometheus node monitoring, SOAR automated IP block
Keycloak Credential Abuse	T1078 – Valid Accounts	IA, AU, IR	Keycloak log ingestion, tenant-wise dashboards, UEBA anomaly detection
Network C2 / DNS Tunneling	T1071, T1048	SC, SI	Suricata network alerts, ELK dashboards, Slack alerting
Insider Anomaly (UEBA)	T1078, T1033	AC, AU, SI	UEBA isolation forest per user, anomaly dashboards, SMTP alerting with enrichment

This scenario mapping demonstrates coverage across multiple layers — access control, detection, enrichment, and automated response.

8.5 Governance Processes

To ensure continuous compliance alignment, the following governance processes were implemented:

- Risk Register Maintenance: Risks identified during deployment are logged, tracked, and reviewed quarterly by Platform & Defense teams.
- Configuration Baseline Control: All Kubernetes and exporter configs are maintained in Git for traceability.

- Change Management: Significant configuration changes require peer review and are tagged in Git repositories.
- Audit Trails: All SOAR actions, alerts, and responses are logged in ELK and MinIO, ensuring forensic traceability.

8.6 Strategic Compliance Roadmap

Phase	Objective	Key Actions
Phase 1 – MVP (Current)	Establish baseline controls and detection	Deploy core SIEM, UEBA, SOAR, secrets mgmt; basic control mapping
Phase 2 – Enhancement	Expand coverage & tighten governance	Integrate threat intel, dynamic secrets, more playbooks, compliance dashboards
Phase 3 – Maturity	Achieve audit-ready posture	Full NIST control mapping, evidence repository, automated compliance reports

9. Threat Scenario Analysis

9.1 Purpose

To validate the effectiveness of the Citadel MVP security platform, realistic threat scenarios were executed and analyzed across multiple layers of the environment — network, endpoint, authentication, behavioral, and automation.

Each scenario was carefully chosen to represent high-priority attack vectors that align with the client's threat model and the MITRE ATT&CK framework. This section documents the scenarios, detection mechanisms, and automated response actions, providing tangible evidence of the platform's operational capabilities.

9.2 Scenario 1 — SSH Brute Force Attack

Attribute	Details
MITRE ATT&CK	T1110 – Brute Force
Environment	Tenant 1 & Tenant 2 Linux hosts

Objective	Detect repeated failed SSH login attempts indicative of brute force activity
Tools/Telemetry	Wazuh Agents, Elastic Agent, Suricata IDS, ELK rules
Detection Rule	SSH Failed Logins - Tenant1, SSH Failed Logins - Tenant2

Execution

Simulated repeated SSH login attempts using invalid credentials against both tenant machines.

Detection

- Wazuh agents captured authentication failures.
- Logs ingested into ELK and correlated per tenant.
- SIEM rules triggered alerts after threshold breaches.
- Alerts appeared on the SSH Login Activity Dashboard with time, IP, and failure count.
- Slack alert was generated via Prometheus Alertmanager integration.

Response

- SOAR engine automatically received alert through webhook.
- IP was blocked using pre-defined response playbook.
- Alert and action details were archived in MinIO and indexed in citadel-actions for audit.

Outcome: Attack was successfully detected and contained automatically, demonstrating end-to-end telemetry, detection, and response.

9.3 Scenario 2 — Keycloak Credential Abuse

Attribute	**Details**
MITRE ATT&CK	T1078 – Valid Accounts
Environment	Tenant 1 & Tenant 2 Keycloak-protected applications
Objective	Detect unauthorized or anomalous application logins

Tools/Telemetry	Keycloak logs ingested into ELK, UEBA
Detection Rule	Keycloak Authentication Alert - Tenant1, Keycloak Authentication Alert - Tenant2

Execution

Simulated compromised credentials being used to authenticate into tenant applications via Keycloak.

Detection

- Keycloak authentication events were ingested into ELK.
- Dashboards were built to visualize login anomalies per tenant.
- Detection rules were triggered on suspicious authentication patterns (multiple failures followed by success, unusual geolocation).
- UEBA model identified deviations from user baselines.

Response

- Alerts were sent to SOAR for triage.
- Analysts were notified via SMTP alerts using professional templates.
- Stakeholders were able to view anomalies in UEBA dashboard.

Outcome: Authentication misuse was successfully detected using both rule-based and behavioral methods, providing layered coverage.

9.4 Scenario 3 — Network C2 / DNS Tunneling

Attribute	**Details**
MITRE ATT&CK	T1071 – Application Layer Protocol; T1048 – Exfiltration Over Alternative Protocol
Environment	Tenant 1 & Tenant 2 endpoints
Objective	Detect covert command-and-control (C2) or exfiltration attempts using DNS tunneling

Tools/Telemetry	Suricata IDS, ELK dashboards, Slack alerting

Execution

Simulated DNS tunneling and suspicious network beaconing activity from tenant machines.

Detection

- Suricata IDS detected anomalous DNS packet structures and tunneling signatures.
- Alerts were ingested into ELK and visualized through the Suricata IDS Dashboard.
- Detection rules were triggered based on DNS tunneling heuristics and Suricata signatures.
- Prometheus + Alertmanager routed high-severity alerts to Slack.

Response

- SOC team received alerts and verified packet details in ELK dashboards.
- IPs involved were escalated for containment actions (manual or automated SOAR trigger).
- Events were preserved in MinIO for forensic review.

Outcome: Network-based malicious activity was successfully detected using Suricata signatures and telemetry, demonstrating strong boundary monitoring.

9.5 Scenario 4 — UEBA Insider Threat Anomaly

Attribute	**Details**
MITRE ATT&CK	T1078 – Valid Accounts, T1033 – System Owner/User Discovery
Environment	Citadel UEBA system (Linux auth logs)
Objective	Detect anomalous SSH login behavior indicative of insider misuse
Tools/Telemetry	Elasticsearch auth indices, UEBA Isolation Forest model, Streamlit dashboards, SMTP alerts

Execution

Generated atypical SSH login activity from an existing user, deviating from historical time and geolocation patterns.

Detection

- UEBA system built per-user behavioral baselines from 30-day history.
- Isolation Forest model scored new logins for anomaly likelihood.
- High-severity anomalies triggered alerts with enriched context (geo, ASN, IP reputation).
- Anomalies were surfaced in Stakeholder & Analyst dashboards with drill-down capabilities.

Response

- SMTP alerts were sent to SOC teams with structured details.
- Alerts were reviewed in dashboards for explanation tags and anomaly features.
- Incidents were documented in evidence repository for follow-up.

Outcome: Behavioral anomaly detection successfully flagged insider-like deviations, proving advanced analytics capability beyond signature/rule detection.

9.6 Scenario Coverage Summary

Scenario	Tactic	Technique(s)	Detection Layer	Response	Evidence
SSH Brute Force	Credential Access	T1110	Wazuh, Elastic, Suricata	Automated IP block	ELK, MinIO
Keycloak Abuse	Credential Access, Persistence	T1078	Keycloak logs, UEBA	SMTP alert, analyst triage	ELK, Dashboards

DNS Tunneling	Command & Control, Exfiltration	T1071, T1048	Suricata IDS	Slack alert, manual/auto containment	ELK, MinIO
UEBA Insider	Credential Access, Discovery	T1078, T1033	UEBA behavioral models	SMTP alert, investigation	UEBA dashboard, MinIO

10. Roadmap & Future Enhancements

10.1 Purpose

The Citadel MVP security platform was designed and implemented to meet critical security requirements using open-source technologies and free-tier capabilities, while laying the foundation for a scalable, compliance-aligned, and operationally mature security ecosystem.

This section presents the strategic roadmap for maturing Citadel from MVP into a fully production-ready SOC platform. It focuses on scaling detection coverage, governance, resilience, and integration depth, while addressing current constraints.

10.2 Guiding Principles

The future evolution of Citadel is anchored on the following strategic principles:

- Open-Source First, Upgrade Ready — Maximize open-source flexibility, with clear migration paths to commercial or hybrid tools when justified.
- Zero Trust & Defense in Depth — Apply multi-layered security controls across identity, network, application, and data planes.
- Operational Visibility & Automation — Expand telemetry coverage, reduce response latency through playbooks, and improve SOC efficiency.
- Compliance by Design — Ensure all enhancements align with NIST 800-53 and other regulatory frameworks, enabling audit-readiness from the ground up.
- Scalable & Resilient Architecture — Move beyond single-region MVP deployments to highly available, fault-tolerant, multi-region setups.

Phase	**Timeline**	**Strategic Objective**	**Key Enhancements**
Phase 1 – Stabilization	0–3 months	Strengthen MVP reliability, tune detections	Refine Prometheus alert rules and exporter configs; Optimize UEBA baselines and model thresholds; Harden Vault secrets rotation procedures; Formalize retention and evidence policies in MinIO
Phase 2 – Expansion	3–6 months	Broaden detection & SOAR coverage, governance	Integrate threat intelligence feeds with Suricata and ELK; Add SOAR playbooks for more TTPs (e.g., lateral movement, privilege escalation); Expand UEBA to include VPN and app telemetry; Implement MinIO lifecycle management
Phase 3 – Integration & Resilience	6–12 months	Enterprise-grade integration, resilience, and compliance	Multi-region replication for Vault, ELK, MinIO; Integrate with ITSM systems (ServiceNow, Jira) for ticketing; Build compliance dashboards for NIST/ISO mapping; Deploy HA clusters for Kubernetes and monitoring stacks
Phase 4 – Maturity	12+ months	Audit-ready, scalable SOC ecosystem	Full NIST 800-53 control coverage with automated evidence; Advanced UEBA and threat hunting; Centralized attack simulation and purple team workflows; Migration path to hybrid or commercial SIEM/SOAR if needed

10.4 Key Enhancement Areas

10.4.1 Detection & Analytics

- Expand Suricata rulesets with curated threat intel (Emerging Threats, Abuse.ch).
- Enhance ELK detection logic with more complex correlation rules and ESQL queries.

- Improve UEBA model sophistication through multi-source feature fusion and adaptive retraining.

10.4.2 Automation & Orchestration

- Increase SOAR playbook coverage to include privilege escalation, insider threats, and cloud-native TTPs.
- Introduce conditional response workflows (e.g., auto vs. analyst-reviewed actions).
- Integrate automated evidence packaging and ticket generation.

10.4.3 Secrets & Identity

- Enable Vault dynamic secrets for Elastic, MinIO, and Keycloak to reduce manual rotation overhead.
- Implement short-lived credentials and Just-In-Time (JIT) access models.
- Expand Keycloak SSO coverage across more services with MFA enforcement.

10.4.4 Resilience & Scalability

- Deploy ELK, Vault, and MinIO in multi-region HA topologies to reduce blast radius.
- Implement automated failover for monitoring and SOAR components.
- Introduce load balancing and horizontal scaling for high-ingest workloads.

10.4.5 Compliance & Reporting

- Develop compliance dashboards for live control coverage visibility.
- Automate evidence collection for control families and threat scenarios.
- Integrate reporting pipelines for audit teams with exportable CSV/PDF evidence packages.

10.5 Strategic Considerations

- Open-Source vs Commercial Trade-off: While the MVP leveraged open-source tools to meet core requirements, scaling to enterprise maturity may justify selective adoption of commercial SIEM/SOAR features for operational efficiency.

- Operational Ownership: As automation and scale increase, clear SOC roles and response ownership must be defined to maintain accountability.
- Continuous Threat Simulation: Incorporating adversary emulation (e.g., Caldera, Atomic Red Team) into detection tuning cycles will help maintain relevance against evolving threats.
- Cost vs Coverage: Some roadmap steps, such as multi-region HA and compliance automation, will require careful cost–benefit analysis to balance security with budget realities.

10.6 Final Vision

The ultimate vision for Citadel is a **next-generation, zero-trust, multi-layered SOC platform** that integrates:

- **Elastic SIEM + Suricata IDS** for broad and deep visibility
- **Prometheus + Grafana** for infrastructure telemetry and health
- **UEBA analytics** for behavioral anomaly detection
- **SOAR orchestration** for rapid, automated containment
- **Vault & MinIO** for secure secrets management and evidence retention
- **Kubernetes-based scalability** for modular and resilient deployments
- **Compliance-driven governance** for audit-readiness and trust

By following the roadmap outlined here, Citadel can evolve from a successful MVP into a **strategic security platform** capable of supporting enterprise-scale operations and rigorous compliance expectations.

11. Conclusion, Final Remarks & Acknowledgment

The **Citadel MVP Security Platform** marks a major step forward in building a **modern, layered, and operationally validated cybersecurity ecosystem** under realistic constraints.

From the outset, the objective was clear: to design and implement a **fully integrated security platform** that not only meets market requirements but also demonstrates real detection, response, and governance capabilities — all while leveraging **open-source and free-tier technologies** to their maximum potential.

11.1 Strategic Achievements

Through structured architecture, disciplined execution, and rigorous scenario testing, this MVP has achieved:

- **End-to-End SOC Pipeline:** Integrated telemetry ingestion, SIEM correlation, UEBA behavioral analytics, and SOAR-based automated response.
- **Multi-Layered Defense:** Defense-in-depth across network (Suricata), endpoint (Wazuh & Elastic Agent), behavioral analytics (UEBA), and infrastructure telemetry (Prometheus & Grafana).
- **Open-Source Excellence:** Strategic use of Elastic Stack, Wazuh, Prometheus, Vault, MinIO, Keycloak, and Kubernetes to build a cost-effective yet powerful security stack.
- **Compliance Alignment:** Baseline control mapping to NIST SP 800-53 and MITRE ATT&CK frameworks with dashboards, detections, and governance processes.
- **Validated Detection & Response:** Realistic simulations for SSH brute force, Keycloak credential abuse, DNS tunneling, and insider anomalies — all successfully detected and responded to.
- **Automation & Evidence Retention:** SOAR engine workflows, centralized secrets management, and MinIO-based evidence archiving enabling fast, traceable response.

These outcomes demonstrate **technical depth, architectural maturity, and operational realism**, providing a solid foundation for future growth.

11.2 Meeting Market Requirements Under Constraints

A key strength of this MVP lies in its ability to **deliver on requirements using open-source tooling** despite constraints.
Where commercial solutions provide pre-built capabilities, Citadel achieves equivalent outcomes through **engineering ingenuity, smart integrations, and targeted customizations**.

Key requirements met:

- Centralized log ingestion & SIEM correlation
- Real-time detection of prioritized threat scenarios

- Automated incident response using SOAR
- Compliance mapping & governance processes
- Operational dashboards & visibility across layers

Constraints managed:

- Free-tier Elastic limitations mitigated by custom UEBA and ESQL rules.
- Lack of native SOAR replaced with Kubernetes-hosted microservice automation.
- Manual processes documented and positioned for future automation.

This demonstrates that **a well-engineered open-source SOC platform can deliver real security value**, even under budgetary and functional limitations.

11.3 Foundation for Future Growth

The Citadel MVP is not the final destination — it is the **strategic launch pad** for a scalable, enterprise-grade SOC platform.
The enhancement roadmap focuses on:

- **Resilience & Scale:** Multi-region deployments, HA architectures, automated failover.
- **Detection Depth:** Threat intel integrations, advanced UEBA models, expanded playbooks.
- **Governance & Automation:** Dynamic secrets, compliance dashboards, ITSM integrations.
- **Audit-Readiness:** Automated evidence collection and control coverage reporting.

By following this roadmap, Citadel can evolve into a **fully production-grade, audit-ready SOC platform**, supporting enterprise security operations at scale.

11.4 Inclusion & Acknowledgment

The Citadel journey has been more than a technical project — it has been a **transformative learning experience**.
It brought together diverse domains: SIEM engineering, behavioral analytics, orchestration, Kubernetes security, and compliance governance, unifying them into a single operational platform.

Throughout this process, the team explored new technologies, refined detection methodologies, and **adapted real-world security frameworks into practical open-source implementations**.

This journey has deepened our understanding of:

- **Modern SOC architectures**
- **Zero Trust and defense-in-depth strategies**
- **Detection engineering under resource constraints**
- **Operational automation and evidence handling**

We extend sincere thanks to everyone who contributed their **technical expertise, operational insights, and strategic direction** — from engineers and analysts to architects, mentors, and reviewers.
Their collaboration, feedback, and commitment were crucial in transforming this concept into a working, validated platform.

Looking ahead, the lessons learned here will drive the **next phases of Citadel's evolution**, shaping it into a **world-class security platform** that blends innovation with operational excellence.