

# PROJECT SENTINEL PLATFORM ARCHITECTURE

Document: Sentinel\_Platform\_Architecture\_v2.2.pdf  
Status: Approved | Version 2.2 | Effective Date: November 1, 2023  
Owner: Security Engineering & Cloud Infrastructure  
Audience: Security Engineers, DevOps, SREs, System Architects

## 1.0 EXECUTIVE SUMMARY

This document defines the complete technical architecture, data flows, integration points, and operational model for the Sentinel Threat Intelligence Platform. It is the master blueprint for building, deploying, and maintaining the system.

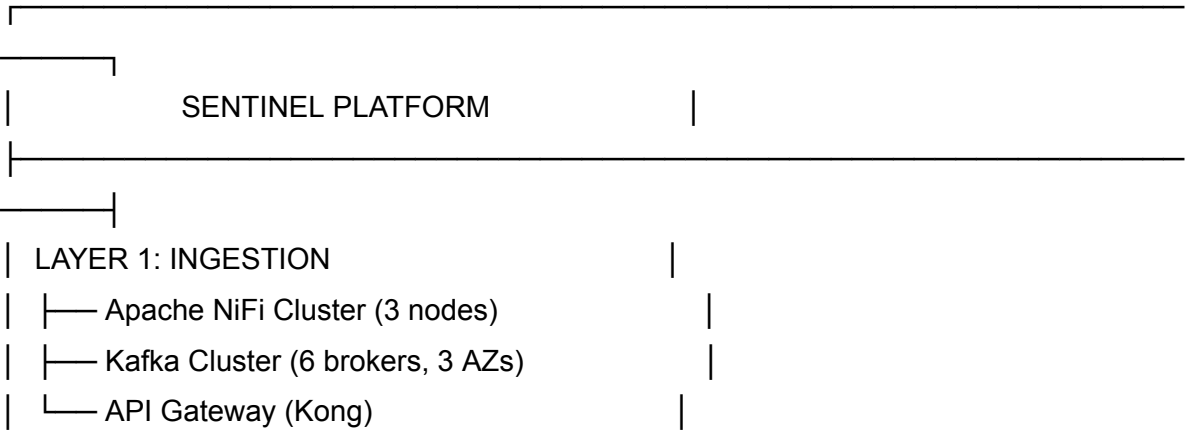
## 2.0 ARCHITECTURAL PRINCIPLES

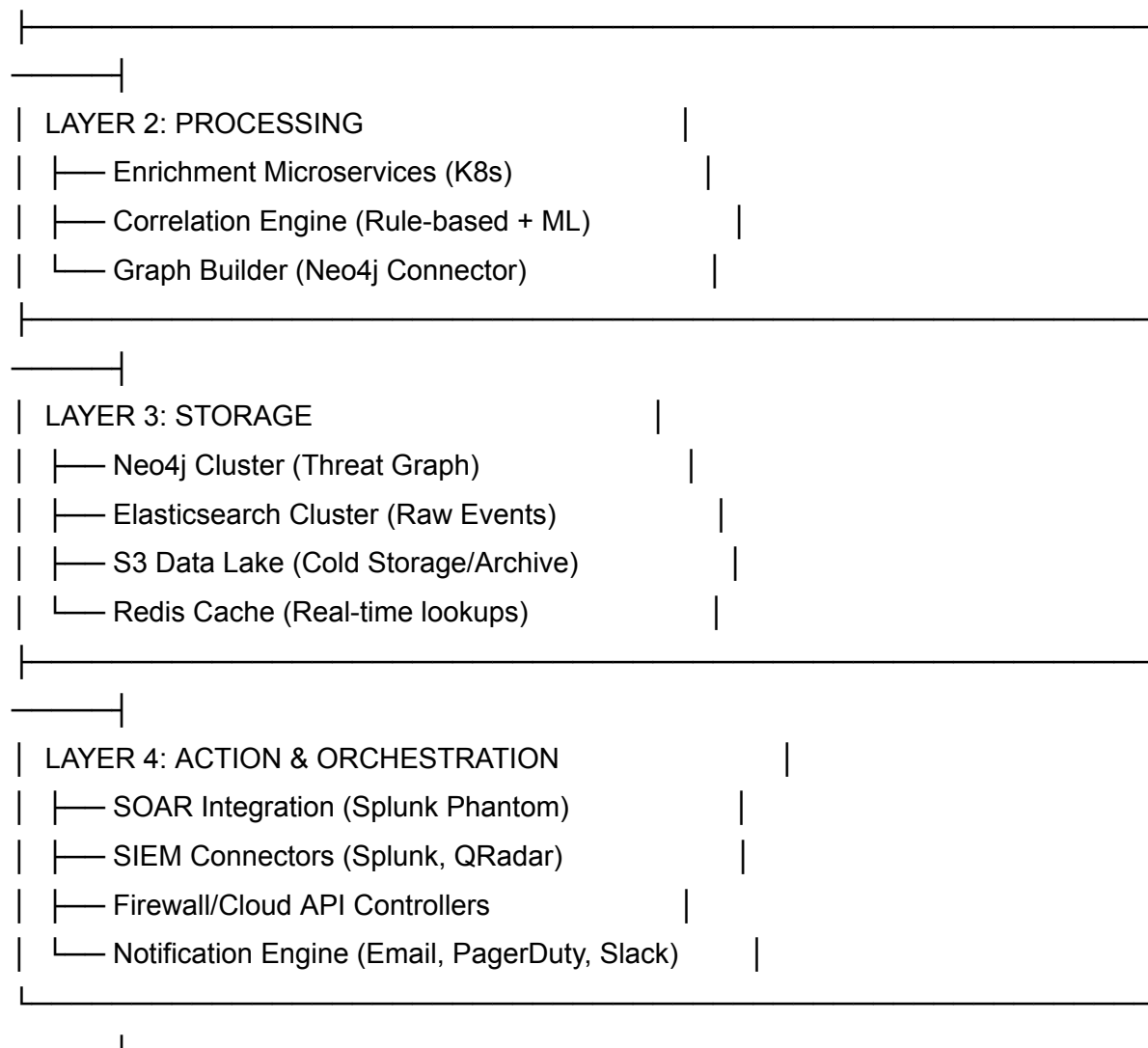
- Tier-0 Asset: Platform treated as most critical infrastructure
- Zero Trust: No implicit trust between components
- Defense in Depth: Multiple security layers at each boundary
- Immutable Infrastructure: No SSH/RDP access to production instances
- Declarative Configuration: All infra defined as code (Terraform, Ansible)

## 3.0 HIGH-LEVEL ARCHITECTURE

### 3.1 Component Diagram

text





## 4.0 DATA INGESTION LAYER

### 4.1 Source Systems & Protocols

Source Category	Protocol	Volume/Day	Retention	Owner
Internal Telemetry	Syslog/S3/Kafka	1.8 TB	90 days hot, 7 years cold	SOC
Commercial Intel Feeds	HTTPS/API/SFTP	150 GB	30 days	Threat Intel
ISAC/Community Feeds	STIX/TAXII	50 GB	90 days	Threat Intel
Unstructured Intel	Custom Scrapers	100 GB	30 days	Research Team
Third-Party Integrations	REST API/Webhook	Varies	30 days	Engineering

### 4.2 NiFi Flow Design

Flow Registry: Git-backed for version control

Processors: Custom processors for each feed type

Error Handling: Dead Letter Queues with 7-day retention

Performance: Target throughput of 50K events/sec per NiFi node

#### 4.3 Kafka Topics Configuration

Topic	Partitions	Replication	Retention	Purpose
raw.internal	24	3	7 days	Raw internal logs
raw.external	12	3	3 days	External intel feeds
enriched.events	36	3	2 days	Post-enrichment events
correlation.alerts	6	3	1 day	High-confidence alerts
actions.commands	3	3	1 day	Platform action commands

#### 5.0 PROCESSING LAYER

##### 5.1 Enrichment Microservices

Deployment: Kubernetes (EKS), 10+ microservices

Pattern: Event-driven, stateless services

Key Services:

Geo-IP Enricher: MaxMind database + custom mapping

Reputation Service: Internal threat scoring engine

VT/Hybrid-Analysis: VirusTotal and sandbox integrations

MITRE Mapper: ATT&CK technique mapping service

Asset Context: CMDB/Asset database enrichment

##### 5.2 Correlation Engine

Technology: Combination of:

Drools Rule Engine: For deterministic, rules-based correlation

Custom ML Models: For anomaly detection and pattern recognition

Time-Series Analysis: For behavioral baselining

Execution Flow:

Event enters correlation queue

Rules engine evaluates against active rule set

ML models score for anomalous patterns

Results merged with confidence scoring

High-confidence matches trigger alert creation

## 6.0 DATA STORAGE LAYER

### 6.1 Neo4j Threat Graph Schema

text

```
(:Indicator {type, value, first_seen, last_seen, score})
(:Actor {name, aliases, suspected_origin, techniques})
(:Campaign {name, start_date, active, objectives})
(:Technique {mitre_id, name, description})
(:Event {timestamp, source, raw_data, processed})
(:Asset {hostname, ip, owner, criticality})
```

```
[OBSERVED_IN] // Indicator → Event
[USED_BY] // Technique → Actor
[PART_OF] // Event → Campaign
[TARGETS] // Campaign → Asset
[RELATED_TO] // Indicator → Indicator
```

### 6.2 Elasticsearch Index Strategy

Index Pattern	Shards	Replicas	Retention	Use Case
logs-raw-*	5	2	30 days	Raw event storage
alerts-*3	2		90 days	Generated alerts
enriched-*	10	2	7 days	Post-enrichment data
metrics-*	1	1	365 days	Platform metrics

### 6.3 Data Retention & Archiving

Hot Storage (Elasticsearch/Neo4j): 90 days

Warm Storage (S3 Standard): 1 year

Cold Storage (S3 Glacier): 7 years (for compliance)

Archival Process: Automated daily via Lambda functions

## 7.0 ACTION & ORCHESTRATION LAYER

### 7.1 Automated Response Actions

Action Type	Trigger Threshold	Target Systems	Approval Required
Auto-Block IP (Automated)	Confidence > 0.9, Severity = Critical	Palo Alto FW, AWS SG	No
Auto-Quarantine Host (1-click)	Confidence > 0.85, Severity = High	CrowdStrike EDR	Yes
User Session Kill (SOC Lead)	Confidence > 0.8, Active Compromise	Okta, Azure AD	Yes
DNS Sinkhole	Malicious Domain, High Confidence	Infoblox DNS	No (Automated)
Ticket Creation	All confirmed alerts	ServiceNow	No (Automated)

### 7.2 Integration Specifications

Splunk SIEM:

HEC endpoint with mutual TLS

Custom CIM-compliant data model

5-second ingestion SLA

Splunk Phantom SOAR:

REST API integration

Pre-built playbooks for 15+ use cases

Bidirectional sync for alert status

Firewall Controllers:

API-based configuration pushes

Change audit logging

Rollback capability on false positive

## 8.0 SECURITY & COMPLIANCE CONTROLS

### 8.1 Network Architecture

VPC Design: Isolated, no inbound internet access

Peering: Private VPC peering to security tools only

Proxy: All outbound traffic via Squid proxy with filtering

Monitoring: VPC Flow Logs to dedicated security account

### 8.2 Access Control

Platform Access: JIT via BeyondCorp, hardware key required

Data Access: Role-based in Neo4j/Elasticsearch

API Access: OAuth 2.0 with scoped permissions

Audit Logging: All access attempts logged to secure SIEM

### 8.3 Encryption Standards

At Rest: AES-256 (AWS KMS managed keys)

In Transit: TLS 1.3 only

Key Rotation: Automatic 90-day rotation

Key Storage: AWS KMS + HashiCorp Vault

## 9.0 DEPLOYMENT & OPERATIONS

### 9.1 Infrastructure as Code

Terraform: All AWS resources (VPC, EC2, RDS, etc.)

Ansible: Configuration management for VMs

Helm Charts: Kubernetes application deployment

GitOps: ArgoCD for continuous deployment

## 9.2 CI/CD Pipeline

text

Code Commit → Security Scan → Build → Test →

Deploy to Staging → Validation → Approval →

Deploy to Production → Smoke Tests → Monitoring

Security Gates: SAST/DAST, dependency scanning, secret detection

## 9.3 Disaster Recovery

RPO: 15 minutes (data loss)

RTO: 30 minutes (full platform restore)

Backup Strategy:

Neo4j: Hourly snapshots to S3

Elasticsearch: Daily snapshots

Configuration: Version-controlled in Git

DR Site: AWS us-west-2 region (warm standby)

## 10.0 MONITORING & ALERTING

### 10.1 Platform Health Metrics

Metric	Collection Method	Alert Threshold
--------	-------------------	-----------------

NiFi Queue Backlog	Prometheus	> 10,000 items
--------------------	------------	----------------

Kafka Consumer Lag	Burrow	> 5 minutes
--------------------	--------	-------------

Neo4j Query Latency	Custom exporter	P95 > 1 second
---------------------	-----------------	----------------

Elasticsearch JVM Heap	Elastic metricbeat	> 85%
------------------------	--------------------	-------

Enrichment Service Error Rate	Prometheus	> 5% for 5 minutes
-------------------------------	------------	--------------------

### 10.2 Dashboards

Platform Health: Grafana - Node resources, service status

Data Pipeline: Grafana - Throughput, latency, errors

Threat Processing: Kibana - Events processed, alerts generated

Business Value: Custom - MTTR/MTTD, incidents prevented

## 11.0 APPENDICES

### Appendix A: API Specifications

OpenAPI 3.0 specs for all platform APIs

### Appendix B: Terraform Module Reference

Reusable modules for platform deployment

### Appendix C: Performance Benchmarks

Load testing results and scaling guidelines

### Appendix D: Change Log

Version history of this architecture document

## APPROVAL SIGNATURES

Director of Security Engineering: \_\_\_\_\_ Date: \_\_\_\_\_

Cloud Infrastructure Lead: \_\_\_\_\_ Date: \_\_\_\_\_

Chief Information Security Officer: \_\_\_\_\_ Date: \_\_\_\_\_