# Software Requirements Specification (SRS)

## Intelligent Cyber Threat Intelligence System

Group: Security Insights
Version: 4.0

November 29, 2024

# Contents

## Revision History

| Version | Date | Author(s) | Description |
|---|---|---|---|
| 1.0 | 2024-11-28 | Security Insights | Initial draft. Monolithic design. |
| 2.0 | 2024-11-30 | Security Insights | Microservices architecture and multi-database integration. |
| 3.0 | 2024-12-01 | Security Insights | Enhanced Azure deployment, CTI-specific pipelines, and dashboards. |
| 3.1 | 2024-12-02 | Security Insights | Improved RBAC, integration with SIEM/SOAR, and compliance-focused revisions. |
| 4.0 | November 29, 2024 | Security Insights | Comprehensive improvements, including risk analysis, testing traceability, and scalability enhancements. |

# 1 Introduction

## 1.1 Purpose

This document specifies requirements for developing a robust Cyber Threat Intelligence (CTI) system for a Fortune 10 company. The system will:

- Integrate structured, unstructured, and relationship-based threat data across relational, document, and graph databases.
- Provide real-time threat ingestion, enrichment, and scoring pipelines.
- Enable scalability and resilience via microservices on Azure Kubernetes Service (AKS).
- Offer actionable insights to C-suite executives and SOC analysts via interactive dashboards.

## 1.2 Scope

The application addresses the needs of enterprise-level CTI by:

- Automating ingestion from threat feeds (OSINT, commercial, proprietary).
- Supporting real-time processing and enrichment pipelines.
- Aligning with frameworks such as MITRE ATTCK.
- Integrating with SIEM/SOAR platforms for actionable intelligence.
- Ensuring security compliance with GDPR and ISO 27001.

## 1.3 Document Overview

The SRS is structured into the following sections:

1. Introduction: Goals, scope, and definitions.
2. Overall Description: High-level product perspective and operating environment.
3. Specific Requirements: Detailed functional and non-functional requirements.
4. System Architecture: Design details for deployment and microservices.
5. Security: Authentication, authorization, and data protection measures.
6. Appendices: Supporting diagrams and documentation.

## 1.4 Definitions, Acronyms, and Abbreviations

- CTI: Cyber Threat Intelligence
- SIEM: Security Information and Event Management
- SOAR: Security Orchestration, Automation, and Response
- RBAC: Role-Based Access Control
- AKS: Azure Kubernetes Service

### 1.5 References

1. Spring Boot Documentation
2. Azure Services Documentation
3. MITRE ATTCK Framework
4. OWASP Security Standards

## 2 Overall Description

### 2.1 Product Perspective

This system integrates backend APIs and CTI pipelines into a scalable, modular architecture:

- **Databases**: MySQL for structured data, MongoDB for nested datasets, Neo4j for relationship traversal.
- **Deployment**: Hosted on Azure AKS with global redundancy.
- **Integration**: Real-time SIEM/SOAR integration for enriched threat data dissemination.

### 2.2 Product Features

- Ingests and enriches threat data from multiple sources.
- Supports multi-database CRUD operations via REST APIs.
- Provides role-based dashboards with KPIs and trend analysis.
- Ensures compliance with regulatory standards.

### 2.3 Assumptions and Dependencies

- Azure cloud services will remain operational and available globally.
- Threat feed providers will deliver data in agreed formats.
- End-users will have high-speed internet connections.

### 2.4 Operating Environment

- Backend: Java Spring Boot.
- Databases: MySQL, MongoDB, Neo4j.
- Deployment: Azure Kubernetes Service (AKS).
- Analytics: Azure Machine Learning and Power BI.

# 3 Specific Requirements

## 3.1 Functional Requirements

- FR-1: Provide REST APIs for CRUD operations on all databases.
- FR-2: Implement real-time ingestion pipelines via Azure Event Hubs.
- FR-3: Support role-based access for data retrieval and processing.
- FR-4: Enrich threats using MITRE ATTCK mapping.
- FR-5: Deliver data to SIEM/SOAR systems in enriched format.

## 3.2 Non-functional Requirements

- NFR-1: Ensure 99.99% uptime with automated failover.
- NFR-2: Handle 50k+ IOCs/day with scalable microservices.
- NFR-3: API response time must not exceed 200ms under load.
- NFR-4: Encrypt all sensitive data at rest and in transit.

# 4 System Architecture

## 4.1 Microservices Design

Each microservice is independent, focusing on a specific database or task:

- Relational DB Microservice: '/mysql/api/v1/...'
- Document DB Microservice: '/mongodb/api/v1/...'
- Graph DB Microservice: '/neo4j/api/v1/...'
- Enrichment Service: Integrates MITRE ATTCK and predictive analytics.

## 4.2 Azure Deployment Architecture

- AKS for containerized deployment.
- Azure SQL Database for relational storage.
- Azure Cosmos DB for document storage.
- Azure Monitor and Sentinel for observability.

# 5 Risk Assessment

## 5.1 Risk Table

| **Risk** | **Impact** | **Likelihood** | ¿Mitigation¿ |
|---|---|---|---|
| Service Outage | High | Moderate | Implement AKS redundancy and failover systems. |
| Data Breach | Critical | Low | Encrypt data and use Azure Key Vault. |
| API Rate Limits | Medium | High | Throttle API requests and implement caching. |

# 6 Testing Traceability

## 6.1 Traceability Matrix

| **Requirement** | **Test Case ID** | **Testing Tool** |
|---|---|---|
| FR-1 | TC-01 | Postman API tests |
| FR-2 | TC-02 | JMeter load test |