

**TOBB ETU**

**Economy & Technology University**

**BIL 495 / YAP 495**

**Software Requirements Specification (SRS)**

***Reference:*** *IEEE 830-1998 / ISO/IEC/IEEE 29148:2018*

## Document Control Table

| Document Title | Project Closure Report |
| --- | --- |
| Document ID | (To be assigned) |
| Prepared By (Author) | Mehmet Eski |
| Reviewed By |  |
| Approved By |  |
| Preparation Date | 27.10.2025 |
| Approval Date |  |
| Version / Revision | 1.0 |
| Confidentiality Level | Internal / Restricted / Public |
| Reference Standards | IEEE 15288, IEEE 12207, INCOSE SE Handbook v5 |

## Change Record (Revision History Table)

| Revision | Date | Prepared By | Reviewed/Approved By | Description of Change |
| --- | --- | --- | --- | --- |
| 1.0 | (Date) | (Author) | (Approver) | Initial Release |
| 1.1 |  |  |  | Minor editorial updates |
| 1.2 |  |  |  | Added stakeholder feedback section |

[1. Introduction](#_heading=)

[1.1 Purpose](#_heading=)

[1.2 Scope](#_heading=)

[1.3 Definitions, Acronyms, Abbreviations](#_heading=)

[1.4 References](#_heading=)

[1.5 Overview](#_heading=)

[2. Overall Description](#_heading=)

[2.1 Product Perspective](#_heading=)

[2.2 Product Functions](#_heading=)

[2.3 User Characteristics](#_heading=)

[2.4 Constraints](#_heading=)

[2.5 Assumptions and Dependencies](#_heading=)

[3. Specific Requirements](#_heading=)

[3.1 Functional Requirements](#_heading=)

[3.2 External Interface Requirements](#_heading=)

[3.3 Performance Requirements](#_heading=)

[3.4 Design Constraints](#_heading=)

[3.5 Software System Attributes](#_heading=)

[3.6 Other Requirements](#_heading=)

[4. Appendices](#_heading=)

[4. References](#_heading=h.a6qvajnlk889)

## 1. Introduction

## 1.1 Purpose

This Software Requirements Specification (SRS) document describes the functional and non-functional requirements for the SOC Analyst Security Management Web Application. The purpose of this system is to provide cybersecurity analysts with a centralized platform for monitoring, detecting, and analyzing potential threats against their domain infrastructure. This document will serve as a reference for developers and testers.

## 1.2 Scope

The system is a web-based SOC (Security Operations Center) tool that enables users to perform domain-based attack surface analysis, track CVE/CVSS vulnerabilities, manage blacklists, detect phishing and malware, integrate with IDS systems, simulate attacks using Caldera, and perform cryptographic operations. The platform will include real-time dashboards, AI-powered anomaly detection, and secure user authentication mechanisms such as 2FA.

## 1.3 Definitions, Acronyms, Abbreviations

SOC – Security Operations Center  
 CVE – Common Vulnerabilities and Exposures  
 CVSS – Common Vulnerability Scoring System  
 IDS – Intrusion Detection System  
 2FA – Two-Factor Authentication  
 NVD – National Vulnerability Database  
 API – Application Programming Interface

## 1.4 References

1. IEEE 830-1998 Software Requirements Specification Standard  
 2. ISO/IEC/IEEE 29148:2018 Systems and Software Engineering Requirements  
 3. ISO/IEC 25010:2011 Quality Model  
 4. NVD (National Vulnerability Database)  
 5. ExploitDB

## 1.5 Overview

The document is structured to describe the overall purpose and context of the system, followed by detailed functional, interface, performance, and quality requirements. It concludes with appendices that include requirement IDs and traceability.

## 2. Overall Description

## 2.1 Product Perspective

The system functions as an independent monolithic web application built using Django as the backend framework and MySQL as the primary database. It integrates external APIs (NVD, ExploitDB, Caldera) for data retrieval and threat simulation. The system fits within the cybersecurity environment of an organization, serving as a centralized monitoring hub.

## 2.2 Product Functions

• Display and manage blacklisted IPs  
 • Fetch and visualize recent CVEs and CVSS scores  
 • Perform attack surface analysis on domains and subdomains  
 • Integrate with IDS for traffic anomaly detection  
 • Conduct phishing detection on connected mailboxes  
 • Detect malware via AI models  
 • Test password strength via a password security model  
 • Perform encryption/decryption using multiple cryptographic algorithms  
 • Export system reports in PDF format  
 • Provide real-time dashboard for network traffic and threat visualization

## 2.3 User Characteristics

Primary users are SOC analysts or cybersecurity professionals with basic to advanced knowledge of network security, threat intelligence, and vulnerability management. Users must own at least one domain to analyze.

**2.4 Constraints**

**Software Constraints:** The system operates as a web-based platform built on a stable backend framework with RESTful communication.   
 A structured relational database is required to manage user, domain, and activity data with integrity and consistency.   
 All modules must remain compatible within a unified monolithic architecture, supporting modular expansion and integration with external APIs.   
 Secure communication (HTTPS) and data encryption are mandatory for all client–server interactions.

***Hardware, Software, Legal, or Policy Limitations***

| **ID** | **Constraint Description** | **Category** |
| --- | --- | --- |
| **C-01** | The hosting environment must provide adequate CPU, memory, and network resources for real-time analysis and dashboard updates. | Hardware |
| **C-02** | The backend shall operate on a web-application framework that supports session handling and authentication. | Software |
| **C-03** | The database shall ensure data consistency and concurrency for multiple active users. | Software |
| **C-04** | All data transmission must occur over secure HTTPS channels. | Security / Policy |
| **C-05** | Sensitive data, including passwords, must be stored only in encrypted or hashed form. | Legal / Security |
| **C-06** | The system shall comply with applicable data-protection laws and institutional privacy policies. | Legal |
| **C-07** | Access to user information is restricted to authenticated and authorized roles. | Policy |
| **C-08** | Deployment is limited to environments compatible with standard browsers and web protocols. | Hardware / Compatibility |

**Filled forms:**

[Cross-discipline constraint tracking form.docx](https://docs.google.com/document/u/0/d/1H46fEVdw59qUWkS254KJQ89aKlhDUVji/edit)

[Universal SW Constraint Tracking Form](https://docs.google.com/document/u/0/d/15WB_a-sgNOMlmKi_NWzdu59uLxoMO5Q-/edit)

## 3. Specific Requirements

## 3.1 Functional Requirements

FR1: The system shall allow users to view and manage blacklisted IP addresses.  
 FR2: The system shall automatically fetch CVE/CVSS vulnerability data from NVD.  
 FR3: The system shall perform attack surface analysis on given domains.  
 FR4: The system shall integrate with IDS to detect traffic anomalies.  
 FR5: The system shall allow users to simulate attacks via Caldera (DoS, Port Scan, etc.).  
 FR6: The system shall provide phishing detection for linked mailboxes.  
 FR7: The system shall provide malware detection through AI-based analysis.  
 FR8: The system shall include a cryptography module for encryption/decryption.  
 FR9: The system shall export user security reports in PDF format.  
 FR10: The system shall support role-based access (Admin, Analyst, ReadOnly).

## 3.2 External Interface Requirements

• User Interface: Web-based dashboard with navigation tabs (Blacklisted IPs, CVE Monitor, IDS Logs, Cryptography, etc.)  
 • Software Interfaces: APIs for NVD, ExploitDB, Caldera, email providers  
 • Hardware Interfaces: Server hosting environment  
 • Communication Interfaces: HTTPS, SMTP, REST APIs

## 3.3 Performance Requirements

• The system shall fetch vulnerability data updates at least once every 12 hours.  
 • Real-time dashboard updates shall have latency under 3 seconds.  
 • IDS alert detection and visualization delay shall be under 2 seconds.  
 • System uptime shall exceed 99% excluding scheduled maintenance.

## 3.4 Design Constraints

• Framework: Python Django  
 • Database: MySQL  
 • Frontend: HTML/CSS/JS (React optional)  
 • Architecture: Monolithic  
 • Hosting: Render or equivalent service  
 • Compliance: Follows ISO/IEC 29148:2018 and ISO/IEC 25010:2011

## 3.5 Software System Attributes

The system follows **ISO/IEC 25010** and **IEEE 1061** quality standards.  
 All non-functional requirements are defined as measurable **software quality attributes** to ensure a secure, reliable, and maintainable SOC analysis platform.

* **Performance Efficiency:**  
   The system shall respond within **1–3 seconds** under standard load.  
   *Metric:* Average response time.  
   *(ISO/IEC 25010)*
* **Reliability:**  
   The system shall operate continuously with **MTBF ≥ 100 hours**.  
   *Metric:* Mean Time Between Failures.  
   *(ISO/IEC 25010)*
* **Security:**  
   Data integrity errors shall not exceed **0.5% per 10,000 transactions**. Passwords must be hashed and all traffic encrypted via HTTPS.  
   *Metric:* Data Error Rate.  
   *(ISO/IEC 27001)*
* **Maintainability:**  
   System modules shall maintain **Coupling Ratio ≤ 0.25** for easy updates and testing.  
   *Metric:* Coupling Ratio.  
   *(IEEE 982.1)*
* **Availability:**  
   Monthly uptime shall remain **≥ 99%**.  
   *Metric:* Availability Ratio.  
   *(ISO/IEC 25010)*
* **Usability:**  
   Analysts should complete key tasks with **≥ 80% success** after 30 minutes of training.  
   *Metric:* Task Success Rate.  
   *(ISO/IEC 9241-210)*
* **Portability:**  
   The application shall run on any Linux-based environment with **≥ 90% compatibility**.  
   *Metric:* Compatibility Ratio.  
   *(ISO/IEC 25010)*
* **Extensibility:**  
   New modules or AI models shall be integrated within **24 hours**.  
   *Metric:* Integration Time.  
   *(IEEE 1061)*

## 3.6 Other Requirements

• Anomaly alerts may trigger SMS or network isolation actions.  
 • Admins can monitor all analysts’ systems.  
 • Users’ actions and system logs must be timestamped for auditability.

## 4. Appendices

Appendix A: Requirement Traceability Matrix  
 FR1–FR10 correspond to system modules as described in Section 2.2.

## 5. References

* **IEEE 830-1998**, *IEEE Recommended Practice for Software Requirements Specifications*, Institute of Electrical and Electronics Engineers (IEEE), 1998.
* **ISO/IEC/IEEE 29148:2018**, *Systems and Software Engineering — Life Cycle Processes — Requirements Engineering*, International Organization for Standardization, 2018.
* **ISO/IEC 25010:2011**, *Systems and Software Quality Requirements and Evaluation (SQuaRE) — System and Software Quality Models*, International Organization for Standardization, 2011.
* **IEEE 1061-1998**, *IEEE Standard for a Software Quality Metrics Methodology*, Institute of Electrical and Electronics Engineers, 1998.
* **IEEE 982.1-2005**, *IEEE Standard Dictionary of Measures to Produce Reliable Software*, Institute of Electrical and Electronics Engineers, 2005.
* **ISO/IEC 27001:2022**, *Information Security, Cybersecurity and Privacy Protection — Information Security Management Systems — Requirements*, International Organization for Standardization, 2022.
* **ISO/IEC 9241-210:2019**, *Ergonomics of Human-System Interaction — Human-Centered Design for Interactive Systems*, International Organization for Standardization, 2019.
* **NVD (National Vulnerability Database)**, U.S. National Institute of Standards and Technology (NIST), <https://nvd.nist.gov>.
* **Exploit Database (ExploitDB)**, Offensive Security, <https://www.exploit-db.com>.
* **Caldera Platform**, MITRE Corporation, <https://caldera.mitre.org>.