Software Requirements Specification

for

Malware Incident Response System

Version 1.0 approved

Prepared by Osman Furkan Önal

7.11.2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

*This Software Requirements Specification (SRS) document describes the software requirements for the **Malware Incident Response System (MIRS)**.*
The purpose of MIRS is to provide an integrated platform for detecting, analyzing, and responding to malware-related security incidents within an organization.
This document defines the functional and non-functional requirements of the system, including its features, interfaces, user interactions, and operational environment.
The SRS serves as a reference for developers, analysts, testers, and stakeholders throughout the software development lifecycle to ensure consistent understanding and implementation of the system.

1.2 Document Conventions

*This document follows the **IEEE SRS (IEEE 830-1998)** standard structure.*
The following conventions are used throughout the document:

- *Requirements are labeled using identifiers such as **FR-x** for functional requirements and **NFR-x** for non-functional requirements.*
- *Priorities are indicated as **High (H)**, **Medium (M)**, or **Low (L)**.*
- *Acronyms and specialized terms are defined in **Appendix A: Glossary**.*
- *Figures and diagrams, where applicable, are numbered sequentially (e.g., Figure 2.1).*
- *All requirements stated in this document are **verifiable**, **traceable**, and **testable**.*

1.3 Intended Audience and Reading Suggestions

This SRS is intended for:

- ***Software Developers** – to design and implement system features according to the stated requirements.*
- ***Security Analysts** – to understand the functional expectations of incident detection, analysis, and reporting modules.*
- ***System Administrators** – to configure and maintain system operations and integrations.*
- ***Project Managers** – to plan resources, schedules, and milestones in accordance with system capabilities.*
- ***Quality Assurance (QA) Teams** – to verify system compliance with requirements and validate performance.*
- ***End Users (Security Teams, IT Staff)** – to gain a high-level understanding of system purpose and workflow.*

*It is recommended that all readers begin with **Section 2 (Overall Description)** for a conceptual overview before reviewing **Section 4 (System Features)** and **Section 5 (Nonfunctional Requirements)** for implementation and performance details.*

1.4 Product Scope

*The **Malware Incident Response System (MIRS)** is a centralized software solution designed to support organizations in managing malware incidents efficiently.*
*It enables **automatic detection**, **manual reporting**, **threat analysis**, **response coordination**, **incident tracking**, and **report generation**.*
By integrating detection tools, log analysis, and notification systems, MIRS improves response time and reduces the impact of malware infections.
Key objectives include:

- *Reducing response time to detected threats.*
- *Enhancing collaboration among analysts and administrators.*
- *Providing comprehensive incident visibility and reporting.*
- *Maintaining data integrity and compliance with security standards.*

MIRS aligns with organizational cybersecurity strategies to strengthen incident management processes and protect critical assets.

1.5 References

Not applicable

2. Overall Description

2.1 Product Perspective

*The **Malware Incident Response System (MIRS)** is a **standalone, web-based platform** that integrates with existing organizational security tools such as antivirus systems, intrusion detection systems (IDS), and Security Information and Event Management (SIEM) platforms.*
*MIRS operates as the central hub for **incident detection, analysis, coordination, and reporting** within an organization's cybersecurity infrastructure.*

The system collects data from multiple sources, correlates alerts, and assists analysts in investigating and mitigating threats.
*It is a **new, self-contained application**, but it relies on secure connections and APIs to communicate with other enterprise systems such as:*

- *Endpoint protection platforms*
- *System log repositories*
- *Network monitoring tools*
- *Email or SMS notification services*

2.2 Product Functions

The major functions of MIRS include:

1. ***Incident Detection***
    - *Automated monitoring of incoming alerts from connected security systems.*
    - *Manual submission of incident reports by users.*
2. ***Incident Analysis***
    - *Classification of malware type and threat level.*
    - *Correlation of events to identify affected assets and systems.*
3. ***Response Coordination***
    - *Assigning response tasks to analysts.*
    - *Providing recommended response actions (e.g., isolate host, block IP, remove file).*
4. ***Reporting and Notification***
    - *Generating detailed and summary reports of incidents.*
    - *Sending real-time notifications to relevant personnel.*
5. ***Data Storage and Logging***
    - *Maintaining a secure database of incidents, logs, and user actions.*
    - *Ensuring traceability for audit and compliance.*

2.3 User Classes and Characteristics

User Class	Description	Key Responsibilities	Technical Skill Level
Security Analyst	Primary user responsible for detecting, analyzing, and responding to malware incidents.	Investigate incidents, classify malware, initiate responses.	Advanced cybersecurity knowledge.
System Administrator	Manages system configurations, integrations, and access control.	Maintain system performance and security.	Intermediate IT and system management.
Security Manager	Oversees incident handling processes and reviews reports.	Approve response actions, review metrics, generate compliance reports.	High-level analytical and managerial skills.
Auditor / Compliance Officer	Reviews system logs and reports for regulatory compliance.	Verify incident documentation and process adherence.	Basic system operation and policy knowledge.

2.4 Operating Environment

*The MIRS platform is designed to operate in **enterprise IT environments** with the following configuration:*

- ***Operating Systems:** Windows Server, Linux (Ubuntu/CentOS)*
- ***Client Access:** Web browsers (Chrome, Edge, Firefox) via HTTPS*
- ***Server Requirements:***
    - *Minimum 8 GB RAM, Quad-core CPU, 200 GB storage*
    - *Supports containerized deployment (Docker/Kubernetes)*
- ***Database:** PostgreSQL or MySQL*
- ***Programming Stack:** Python backend (Flask/Django), JavaScript frontend (React or Vue)*
- ***Network Environment:** Secure intranet or VPN connection with TLS encryption*

2.5 Design and Implementation Constraints

- *The system must comply with **ISO/IEC 27001** information security management standards.*
- *Must integrate with existing **SIEM and endpoint detection** solutions via RESTful APIs.*
- *All stored data must be **encrypted at rest and in transit**.*
- *Development language must be compatible with the organization's infrastructure (Python preferred).*
- *System design must allow for **role-based access control (RBAC)**.*
- *Implementation should support **scalability** to handle a growing number of incidents.*

2.6 User Documentation

The following documentation will be delivered with the system:

- ***User Manual:** Step-by-step guidance for analysts and administrators.*
- ***System Configuration Guide:** Installation and setup procedures.*
- ***API Documentation:** Details for integrating third-party tools.*
- ***Training Materials:** Tutorials, sample workflows, and best practices.*
- ***Online Help System:** Built-in context-sensitive help and tooltips.*

2.7 Assumptions and Dependencies

- *The organization has an existing **security infrastructure** (antivirus, IDS/IPS, log servers).*
- *The system will be hosted on **secure servers** managed by the organization's IT team.*
- *Users have valid credentials and sufficient permissions to access the system.*
- *Internet or internal network connectivity is stable and reliable.*
- *Third-party libraries and APIs used by MIRS are actively maintained and compatible with system requirements.*

3. External Interface Requirements

3.1 User Interfaces

3.1 User Interfaces

*The **Malware Incident Response System (MIRS)** will provide a **web-based graphical user interface (GUI)** accessible via standard web browsers. The interface is designed for clarity, usability, and efficiency in incident management tasks.*

3.1.1 3.1.1 General Interface Characteristics

- *The GUI follows a **dashboard-based layout**, displaying key metrics such as active incidents, system health, and alert summaries.*
- *The interface uses a **role-based design**, showing different menus and permissions for Analysts, Administrators, and Managers.*
- ***Color coding** and icons are used to indicate incident severity (e.g., red for critical, orange for high, yellow for medium, green for resolved).*
- *Navigation is provided through a **sidebar menu**, with modules such as:*
    - *Dashboard*
    - *Incident List*
    - *Analysis Tools*
    - *Reports*
    - *Settings & User Management*

3.1.2 Typical Screens

- ***Login Screen:*** *Secure authentication with username, password, and optional multi-factor verification.*
- ***Incident Dashboard:*** *Displays real-time alerts, summary statistics, and visual charts.*
- ***Incident Details Page:*** *Shows full information about a selected incident (type, source, impact, timeline, and recommended actions).*
- ***Response Coordination Panel:*** *Allows assigning and tracking tasks among analysts.*
- ***Reports Page:*** *Generates and exports summaries in PDF or CSV format.*

3.1.3 Accessibility & Usability

- *Supports both **light and dark modes** for user comfort.*
- *Designed for **responsive display**, compatible with desktops, laptops, and tablets.*
- *Provides **tooltips, search bars, and filtering options** for ease of use.*
- *Includes **inline help** and a quick tutorial for new users.*

3.2 Hardware Interfaces

MIRS is primarily software-driven and does not directly control or depend on specialized hardware. However, it interacts indirectly with network devices through data collection and communication interfaces.

- ***Supported Server Hardware:***
  - *x86_64 architecture servers (physical or virtualized).*
  - *Network Interface Cards supporting standard TCP/IP.*
- ***Peripheral Requirements:***
  - *Optional support for hardware security modules (HSMs) to enhance encryption key management.*
  - *Compatibility with standard data storage devices for backup and log archiving.*
- ***External Sensors / Agents:***
  - *May receive data from endpoint security agents installed on user machines.*

3.3 Software Interfaces

*The MIRS system interacts with several external software components and internal modules through **Application Programming Interfaces (APIs)** and secure data connections.*

Interface Type	External System	Data Exchanged / Purpose	Communication Protocol
Security Information and Event Management (SIEM)	Splunk, QRadar, etc.	Log and alert ingestion for correlation.	REST API / JSON
Antivirus / EDR Platforms	Microsoft Defender, CrowdStrike, etc.	Malware detection reports and status updates.	REST API / XML
Email & Notification Services	SMTP servers, SMS gateways	Sending notifications and alerts.	SMTP, HTTPS
Database	PostgreSQL / MySQL	Store and retrieve incident data and logs.	SQL
Authentication Services	LDAP / Active Directory	Manage user authentication and access control.	LDAP / SAML
Web Interface	Frontend (React/Vue)	Interaction between user browser and backend server.	HTTPS / JSON

3.4 Communications Interfaces

The communication layer of MIRS ensures secure and reliable data exchange between system components and external systems.

3.4.1 Network Protocols

- *All communication occurs over **HTTPS** with **TLS 1.3** encryption.*
- *Internal microservices (if applicable) communicate over **secured REST endpoints**.*
- *System updates and notifications use **secure webhooks** or **message queues (e.g., RabbitMQ)**.*

3.4.2 Data Transfer

- *Data transmission between client and server must be encrypted.*
- *Logging and telemetry data are compressed and securely transmitted to the central server.*
- *Large report exports are handled asynchronously to avoid network congestion.*

3.4.3 Communication Security

- *Implements **role-based message filtering** to ensure that only authorized components receive sensitive data.*
- *All API tokens, keys, and credentials are stored securely using encryption.*
- *Failed communication attempts are logged and retried automatically.*

4. System Features

*This section describes the main features and functions provided by the **Malware Incident Response System (MIRS)**.*
Each feature includes a description, priority, stimulus/response sequence, and detailed functional requirements.

4.1 System Feature 1

4.1.1 4.1.1 Description and Priority

This feature enables the system to automatically and manually detect malware-related security incidents from multiple data sources.
It collects and aggregates alerts from antivirus systems, intrusion detection systems, and endpoint monitoring tools.

Priority: *High*

4.1.2 4.1.2 Stimulus/Response Sequences

- ***Stimulus:*** *Security event or alert generated by an integrated security tool.*
- ***Response:*** *MIRS receives, validates, and logs the event. The system generates a new incident record and displays it on the dashboard.*

4.1.3 Functional Requirements

- ***FR-1.1****: The system shall collect real-time security alerts from integrated detection systems via API.*
- ***FR-1.2****: The system shall validate incoming alerts to avoid duplication.*
- ***FR-1.3****: The system shall categorize detected incidents by severity (Critical, High, Medium, Low).*
- ***FR-1.4****: The system shall allow users to manually create a new incident report.*
- ***FR-1.5****: The system shall timestamp and store all detected incidents in the central database.*

4.2 Feature 2 – Incident Analysis

4.2.1 Description and Priority

This feature allows analysts to investigate and classify incidents by reviewing details, logs, and related system data.
It supports both static and behavioral analysis of malware activity.

Priority: *High*

4.2.2 Stimulus/Response Sequences

- ***Stimulus:*** *Analyst selects an active incident for investigation.*
- ***Response:*** *MIRS retrieves incident data, displays related logs, and provides analysis tools (e.g., file hash lookup, timeline view).*

4.2.3 Functional Requirements

- ***FR-2.1****: The system shall display complete incident metadata (source, affected hosts, malware type, and timestamp).*
- ***FR-2.2****: The system shall retrieve and present related system logs and network activity.*
- ***FR-2.3****: The system shall allow analysts to classify incidents (e.g., malware infection, phishing, ransomware).*
- ***FR-2.4****: The system shall calculate and display a confidence score for threat classification.*
- ***FR-2.5****: The system shall store analyst comments and investigation notes for each incident.*

4.3 4.3 Feature 3 – Response Coordination

4.3.1 Description and Priority

This feature provides tools for planning and executing response actions for each detected incident. It ensures collaborative handling and tracks task progress.

Priority: *High*

4.3.2 Stimulus/Response Sequences

- ***Stimulus:*** *Analyst initiates a response plan for an incident.*
- ***Response:*** *The system creates a response workflow, assigns tasks to team members, and tracks their completion status.*

4.3.3 Functional Requirements

- ***FR-3.1****: The system shall enable analysts to assign response tasks to team members.*
- ***FR-3.2****: The system shall provide predefined response templates for common malware scenarios.*
- ***FR-3.3****: The system shall allow marking incidents as “In Progress,” “Resolved,” or “Escalated.”*
- ***FR-3.4****: The system shall record all actions taken during response for auditing purposes.*
- ***FR-3.5****: The system shall notify relevant personnel of assigned tasks and status changes.*

4.4 4.4 Feature 4 – Reporting and Notification

4.4.1 Description and Priority

This feature generates detailed and summary reports of incidents and sends notifications to authorized users when specific events occur.
Reports assist managers and auditors in evaluating performance and compliance.

Priority: *Medium–High*

4.4.2 4.4.2 Stimulus/Response Sequences

- ***Stimulus:*** *Manager or analyst requests a report or a critical incident occurs.*
- ***Response:*** *MIRS compiles data, generates a formatted report, and optionally sends alerts to users.*

4.4.3 Functional Requirements

- ***FR-4.1****: The system shall generate incident reports summarizing type, source, severity, and resolution time.*
- ***FR-4.2****: The system shall allow exporting reports in PDF, CSV, and HTML formats.*
- ***FR-4.3****: The system shall send real-time notifications for high-severity incidents via email or SMS.*
- ***FR-4.4****: The system shall support scheduled automated reports (daily, weekly, monthly).*
- ***FR-4.5****: The system shall log all generated reports for future access.*

4.5 Feature 5 – User Management and Access Control

4.5.1 4.5.1 Description and Priority

This feature defines how users are created, authenticated, and authorized within MIRS. It ensures that system operations comply with organizational access policies.

Priority: *High*

4.5.2 4.5.2 Stimulus/Response Sequences

- ***Stimulus:*** *A new user attempts to log in or an administrator updates access rights.*
- ***Response:*** *The system authenticates credentials, checks role permissions, and grants or denies access.*

4.5.3 Functional Requirements

- ***FR-5.1****: The system shall provide role-based access control (RBAC).*

- ***FR-5.2**: The system shall integrate with external authentication services (e.g., LDAP, SAML).*
- ***FR-5.3**: The system shall enforce password complexity and expiration policies.*
- ***FR-5.4**: The system shall log all user access and modification attempts.*
- ***FR-5.5**: The system shall allow administrators to deactivate or remove user accounts.*

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- ***NFR-1**: The system shall support a minimum of **100 concurrent user sessions** without degradation of performance.*
- ***NFR-2**: The system shall process and display new incident alerts within **3 seconds** of receipt.*
- ***NFR-3**: Report generation for up to **1,000 incidents** shall complete in under **10 seconds**.*
- ***NFR-4**: Database queries for incident searches shall return results within **2 seconds** under normal load.*
- ***NFR-5**: The system shall handle an incoming data rate of **up to 10,000 log entries per minute** from integrated sources.*
- ***NFR-6**: The system shall maintain a **99.5% uptime** during normal operations.*

5.2 Safety Requirements

- ***NFR-7**: The system shall ensure no modification or deletion of security evidence data without authorization.*
- ***NFR-8**: In the event of a system failure, MIRS shall automatically perform a **graceful shutdown** and preserve all unsaved incident data.*
- ***NFR-9**: Backup operations shall be performed **daily**, and recovery procedures shall be verified periodically.*
- ***NFR-10**: The system shall include **confirmation prompts** for all critical actions (e.g., deletion, user removal, report overwrite).*
- ***NFR-11**: The system shall comply with internal **IT safety policies** and **data retention regulations** to prevent data loss.*

5.3 Security Requirements

- ***NFR-12**: All communication between system components shall be **encrypted using TLS 1.3**.*
- ***NFR-13**: All sensitive data stored in the database shall be **encrypted using AES-256** encryption.*
- ***NFR-14**: The system shall require **multi-factor authentication (MFA)** for administrative and analyst roles.*
- ***NFR-15**: Passwords shall be hashed using **bcrypt or PBKDF2** with a unique salt.*
- ***NFR-16**: The system shall implement **role-based access control (RBAC)** to restrict user permissions.*
- ***NFR-17**: The system shall automatically **lock accounts** after 5 unsuccessful login attempts.*
- ***NFR-18**: Audit logs shall record every user login, configuration change, and incident modification.*
- ***NFR-19**: The system shall comply with **ISO/IEC 27001** and **NIST SP 800-61r2** standards for incident response.*
- ***NFR-20**: The system shall perform **input validation and sanitization** to prevent injection and cross-site scripting (XSS) attacks.*

5.4 Software Quality Attributes

Attribute	Description	Target Goal / Standard
Reliability	The system must function continuously and recover gracefully from unexpected errors.	≥ 99.5% uptime
Usability	The user interface should be intuitive and accessible, supporting easy navigation and task completion.	≤ 1 hour learning curve for new users
Maintainability	System codebase must be modular and documented for easy updates and debugging.	Modular design, 90% code documentation
Scalability	The system should support future expansion in data volume and users without redesign.	Scalable to 5× data load
Portability	System should run on Windows and Linux servers without modification.	Full OS compatibility
Availability	The system should ensure high availability with minimal downtime.	≥ 99.5%
Interoperability	Must integrate with third-party tools via open standards (REST APIs, JSON, XML).	100% API-based interoperability

5.5 Business Rules

- ***BR-1**: Only authorized analysts and administrators may access or modify incident data.*
- ***BR-2**: All malware incidents must be logged, analyzed, and closed within defined Service Level Agreements (SLAs).*
- ***BR-3**: Incident data must be retained for a **minimum of one year** before archival or deletion.*
- ***BR-4**: Report generation and access must comply with **organizational confidentiality policies**.*
- ***BR-5**: Only Security Managers can approve and finalize response actions marked as "Resolved."*
- ***BR-6**: The system shall enforce unique user accounts; shared credentials are strictly prohibited.*
- ***BR-7**: Any deviation from standard response procedures must be documented and approved by management.*

6. Other Requirements

This section includes additional requirements not covered in previous sections, such as database structure, legal constraints, and future considerations.

6.1 Database Requirements

- ***DB-1**: The system shall use a **relational database** (PostgreSQL or MySQL) to store incident, user, and log data.*
- ***DB-2**: The database schema shall include tables for:*
  - *Users (user_id, name, role, credentials)*
  - *Incidents (incident_id, type, severity, timestamp, status, assigned_to)*
  - *Logs (log_id, source, event_time, details, related_incident)*
  - *Reports (report_id, author, created_on, file_path)*
  - *Actions (action_id, incident_id, performer, description, date)*
- ***DB-3**: All primary and foreign key relationships shall be **referentially consistent**.*
- ***DB-4**: The database shall support **indexing** on high-frequency query fields such as incident_id, severity, and status.*
- ***DB-5**: Database access shall be restricted to authenticated system components using **encrypted credentials**.*

6.2 Legal and Compliance Requirements

- ***LCR-1**: The system shall comply with applicable **data protection regulations** (e.g., GDPR, ISO/IEC 27035).*
- ***LCR-2**: Personally Identifiable Information (PII) in incident reports shall be masked or anonymized where required.*
- ***LCR-3**: System logs shall be retained for **at least 12 months** for audit and compliance review.*
- ***LCR-4**: Any third-party libraries used shall be **open-source or properly licensed** according to organizational policy.*
- ***LCR-5**: System deployment must follow organizational **cybersecurity and data-handling policies**.*

6.3 Reuse and Extensibility Requirements

- ***RER-1**: MIRS shall be developed using a **modular architecture** to support future extension (e.g., new malware analysis engines).*
- ***RER-2**: The API layer shall allow new integrations without requiring core code modifications.*
- ***RER-3**: Reusable software components such as authentication, reporting, and notification modules should be implemented as **independent services**.*
- ***RER-4**: The codebase shall follow **PEP8 (Python)** and internal style guidelines for maintainability.*

6.4 Internationalization and Localization

- ***IL-1**: The system interface shall support **English** as the default language.*
- ***IL-2**: The design shall allow easy addition of other languages through localization files.*
- ***IL-3**: All date, time, and number formats shall follow **ISO 8601** standards for global compatibility.*

Appendix A: Glossary

Term / Acronym	Definition
MIRS	**Malware Incident Response System**
SIEM	**Security Information and Event Management**
EDR	**Endpoint Detection and Response**
RBAC	**Role-Based Access Control**
MFA	**Multi-Factor Authentication**
SLA	**Service Level Agreement**
API	**Application Programming Interface**
GDPR	**General Data Protection Regulation**
NIST	**National Institute of Standards and Technology**

Term / Acronym	Definition
ISO/IEC 27001	**International Standard for Information Security Management**
Incident	**A malware-related security event that requires investigation and response**
Analyst	**Security team member who investigates and mitigates incidents**
Response Plan	**A structured set of actions designed to contain and resolve an incident**

Appendix B: Analysis Models

This section provides optional models representing MIRS components and interactions.

6.4.1 B.1 Use Case Diagram (Conceptual Description)

- **Actors:** Security Analyst, System Administrator, Security Manager
- **Use Cases:**
    - Detect Incident
    - Analyze Malware
    - Assign Response Actions
    - Generate Reports
    - Manage Users
    - View Audit Logs

B.2 Class Diagram (Conceptual Description)

Classes:

- Incident, ResponsePlan, ResponsePhase, Action, Person, User, Report
  Relationships:
- One *Incident* is associated with one *ResponsePlan*.
- A *ResponsePlan* consists of multiple *ResponsePhases*.
- Each *ResponsePhase* includes multiple *Actions*.
- Each *Action* is performed by a *Person*.
- *Users* can view and manage *Reports* related to incidents.

B.3 State Transition (Simplified)

- **States:** Detected → Under Investigation → Response Initiated → Resolved → Closed

Appendix C: To Be Determined List

ID	Item Description	Status
TBD-1	Selection of cloud or on-premise hosting environment	Pending
TBD-2	Final integration tools (specific SIEM/EDR vendors)	Pending
TBD-3	Detailed user interface design mockups	In Progress
TBD-4	Final reporting format templates (PDF layout)	Pending
TBD-5	Final deployment strategy and CI/CD pipeline tools	Pending