# **Cybersecurity Risk Assessment System Documentation**

## **Student Information**

**Anastasya Selena Anandita Adnan** (001202300148)  
**Indrasworo Suryo Prayogo** (001202300147)

## **Project Overview**

The **Cybersecurity Risk Assessment System** is designed to help organizations assess, identify, and manage cybersecurity risks. The system integrates with the **Malware Information Sharing Platform (MISP)** to fetch real-time threat intelligence, enhancing risk assessment capabilities. It leverages **OCTAVE Allegro** methodology for conducting risk assessments and visualizes the data through an interactive dashboard.

### **Key Features**

* 📡 **Real-time Threat Intelligence Integration**: Fetches threat data from MISP to aid risk identification.
* 📊 **Risk Assessment Dashboard**: Visualizes risk data using a bar chart for likelihood and impact.
* 🗄️ **Database Integration**: Stores risks and associated data in a PostgreSQL database.
* 📑 **Automated Reporting**: Generates and exports PDF reports summarizing risk information.
* 🌐 **Interactive Web Interface**: Provides a user-friendly dashboard for easy risk management.

## **System Architecture**

The system follows a **client-server architecture**, comprising the following components:

### **1. Frontend (Client Side)**

* **Web UI**: Developed using HTML, CSS, and JavaScript, featuring **Chart.js** for risk visualization.
* **Static Assets**: Includes style.css for UI styling and script.js for API interactions and rendering visualizations.
* **User Interaction**: Users can fetch threat data from MISP and analyze risk assessment results through the interface.

### **2. Backend (Server Side)**

* **Flask Web Framework**: Handles HTTP requests and serves the frontend templates.
* **MISP Integration**: Utilizes **PyMISP** to fetch real-time threat intelligence.
* **Database Management**: Stores risk data in a PostgreSQL database using **psycopg2**.
* **PDF Report Generation**: Uses **FPDF** for exporting risk assessment summaries.

### **3. Database**

* PostgreSQL stores risk-related information, enabling efficient retrieval and analysis.

## **Database Schema**

### **Database: cyber\_risk**

The database contains a table named **risks**, structured as follows:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| id | SERIAL | Primary key, auto-incremented integer |
| risk\_name | TEXT | Name of the risk (e.g., "Data Breach") |
| likelihood | INTEGER | Likelihood score (1-5 scale) |
| impact | INTEGER | Impact score (1-5 scale) |
| threat\_data | JSONB | Threat intelligence data fetched from MISP |

### **SQL Table Creation**

CREATE DATABASE IF NOT EXISTS cyber\_risk;

\c cyber\_risk;

CREATE TABLE IF NOT EXISTS risks (

id SERIAL PRIMARY KEY,

risk\_name TEXT NOT NULL,

likelihood INTEGER NOT NULL,

impact INTEGER NOT NULL,

threat\_data JSONB

);

## **API Usage**

### **Endpoints**

#### **1. GET /**

* **Description**: Serves the homepage (index.html).
* **Response**: Interactive dashboard with risk assessment tools.

#### **2. GET /test-misp**

* **Description**: Tests the MISP connection.
* **Response**: { "message": "MISP connection successful" } or { "error": "MISP connection failed" }

#### **3. GET /fetch-threats**

* **Description**: Fetches threat data from MISP.
* **Response**: JSON containing real-time threat intelligence.

#### **4. POST /add-risk**

* **Description**: Adds a new risk to the database.

**Request Payload:** {

"risk\_name": "Phishing",

"likelihood": 4,

"impact": 5,

"threat\_data": { "event\_id": 123, "threat\_type": "Phishing" }

}

* **Response:** { "message": "Risk added successfully" }

#### **5. GET /get-risks**

* **Description**: Retrieves all stored risks.

**Response:** [

[1, "Phishing", 4, 5, {"event\_id": 123, "threat\_type": "Phishing"}],

[2, "Data Breach", 3, 4, {"event\_id": 124, "threat\_type": "Data Breach"}]

]

#### **6. GET /export-report**

* **Description**: Generates and exports a PDF report containing all risks.
* **Response**: A downloadable PDF file (cyber\_risk\_report.pdf).

## **Running the Project**

### **1. Clone the Repository**

git clone <repository\_url>

cd cybersecurity-risk-assessment

### **2. Install Dependencies**

pip install -r requirements.txt

### **3. Setup the Database**

Run the SQL script to initialize the PostgreSQL database:

psql -U <username> -d cyber\_risk -f database.sql

### **4. Configure MISP**

Modify app.py to include your **MISP\_URL** and **MISP\_API\_KEY**.

### **5. Run the Flask Application**

python app.py

* The app will run on http://localhost:5000/

### **6. Access the Web Interface**

* Open a browser and navigate to <http://localhost:5000/>

## **Conclusion**

The **Cybersecurity Risk Assessment System** offers a comprehensive framework for evaluating and mitigating cybersecurity risks. By integrating **real-time threat intelligence**, **data visualization**, and **automated reporting**, it enhances the risk assessment process and provides organizations with actionable insights.