**Use Case Scenarios**

1. **Phishing & Malware Detection**

This scenario allows users to connect their email accounts to the system for phishing analysis. Also users can upload files to detect potential malware threats using AI-based models. The system analyzes the uploaded file and identifies malicious content, gives the information to users with a detailed security status or warning message after the analyzing process. This helps users to protect themselves against phishing attacks and infected files.

Required Resources:

* Application server for hosting the phishing analysis and file upload modules.
* AI processing unit for running AI-based malware detection models.
* Email integration API’s for connecting and scanning user mails.

Requirements:

* The system will allow users to connect their email for phishing analysis.
* The system will enable file uploads for malware detection using AI-based models.
* The user will receive a security status or warning message after analysis.

1. **IDS & Attack Simulation**

In this scenario, the system continuously monitors network traffic to detect unusual or suspicious activities such as anomalies, functioning as an Intrusion Detection System (IDS). Users can also perform controlled cyberattack simulations on their own domains to test the system’s security metrics. The results, alerts, and related statistics are displayed on a visual dashboard, helping users better understand vulnerabilities and strengthen their network defenses.

Required Resources:

* Network monitoring tools with entegred with backend system.
* High-Performance CPU & RAM for real time analysis.
* Isolated sandbox or virtual machine for safe testing of simulated attacks.
* Strong network bandwidth for continuous traffic capture.

Requirements:

* The system will monitor incoming and outgoing traffic to detect anomalies.
* The system will allow users to simulate cyberattacks on their own domain for testing purposes.
* The system will present results, alerts, and statistics in a visual dashboard.

1. **Blacklist & Threat Management**

This scenario enables users to manage potentially harmful IP addresses through a blacklist system. Users can view the existing list, add new IP addresses, or remove unnecessary ones. This functionality serves as a preventive measure against malicious traffic, better control over network access and improving the overall security posture of the system.

Required Resources:

* Database server for storing and managing of blacklisted IP’s.(e.g PostgreSQL)
* Backend server for CRUD(Create, Read, Update, Delete) operations.
* Authentication control ensures only authorized users can modify the list.

Requirements:

* The user will be able to view, add, and remove IP addresses from the blacklist.

1. **CVE & Vulnerability Management**

In this scenario, the system automatically retrieves and displays recent CVE (Common Vulnerabilities and Exposures) data from trusted security sources. Each vulnerability is presented with its CVSS (Common Vulnerability Scoring System) score and detailed information in a structured format. The CVE news will be updated periodically without user intervention, allowing users to stay informed about the latest threats and take timely action to mitigate vulnerabilities.

Required Resources:

* Backend service to fetch and update CVE data periodically.
* External CVE API’s such as NIST NVD API, MITRE CVE Feed
* Scheduler for refreshing CVE data regularly
* Frontend framework to visualize the vulnerability details.
* Moderate network connectivity to pull the latest CVE data.

Requirements:

* The system will automatically fetch and display recent CVE vulnerabilities from trusted sources.
* The user will be able to view each vulnerability’s CVSS score and details in a structured format.
* The system will update vulnerability data periodically without user intervention.

A small number of actual users are the target audience for this project, which was created as an academic prototype. In order to validate the system's functionality and essential security features in a small scaled settings, the design and evaluation were conducted assuming that there would be like 50 registered users and 5–10 concurrent users. This project’s primary goals are to show how the features that have been implemented work correctly and modeling fundamental threat detection scenarios. Resource planning and scalability considerations would be recreated if the system were ever to be implemented in a production setting with many other users.