**Software Requirement Specification (SRS)**

**Project Name :-**

**Detection of Cyber Attacks in Network**

**Using Machine Learning Techniques**

|  |  |
| --- | --- |
| **Introduction** | **2** |
| **Overall Description** | **3** |
| **Non functional requirements** | **4** |
| **External interface requirements** | **7** |
| **System features** | **8** |
| **Acceptance criteria** | **10** |
| **Deliverables** | **12** |

**# Introduction :-**

#### 1.1 Purpose :-

* The purpose of this document is to provide a comprehensive description of the software requirements for the development of a system designed to detect cyber attacks within a network using machine learning techniques. This document outlines the functionalities, user requirements, and constraints that the system must adhere to in order to effectively identify and mitigate cyber threats.

#### 1.2 Scope :-

* The system aims to enhance network security by leveraging machine learning algorithms to analyze network traffic and identify potential cyber attacks in real-time. It will provide network administrators and security analysts with tools to monitor network activity, receive alerts about suspicious behavior, and generate detailed reports on detected threats.

#### 1.3 Definitions, Acronyms, and Abbreviations :-

* **ML**: Machine Learning
* **IDS**: Intrusion Detection System
* **IPS**: Intrusion Prevention System
* **DDoS**: Distributed Denial of Service
* **SIEM**: Security Information and Event Management

#### 1.4 References :-

* "Machine Learning for Cybersecurity" by Xiaofeng Chen
* "Cybersecurity and Applied Machine Learning" by Richard Liu
* National Institute of Standards and Technology (NIST) Cybersecurity Framework
* ISO/IEC 27001 Information Security Management

#### 1.5 Overview :-

* This document is organized into several sections:

**Overall Description**: Provides an overview of the system's functionality, user characteristics, and operating environment.

* **Specific Requirements**: Details the functional and non-functional requirements of the system.
* **Interface Requirements**: Describes the user, hardware, and software interfaces.
* **Other Requirements**: Includes database, legal, and regulatory requirements.
* **Appendices**: Contains a glossary and a list of acronyms and abbreviations used in the document.
* The document serves as a guide for developers, testers, and stakeholders involved in the project, ensuring a clear understanding of the system’s objectives and requirements.

**# Overall Description :-**

#### 2.1 Product Perspective :-

* The system will be a standalone application that integrates with existing network infrastructure. It will capture network traffic data, apply machine learning models, and detect anomalies that may indicate cyber attacks.

#### 2.2 Product Functions :-

* Data Collection: Capture network traffic data.
* Data Preprocessing: Clean and prepare data for analysis.
* Feature Extraction: Identify relevant features from the data.
* Model Training: Train machine learning models on historical data.
* Real-Time Detection: Apply trained models to real-time data for attack detection.
* Alerting: Notify administrators of potential attacks.
* Reporting: Generate detailed reports on detected threats.

#### 2.3 User Classes and Characteristics :-

* **Network Administrators**: Responsible for monitoring and managing network security.
* **Security Analysts**: Analyze detected threats and respond to incidents.
* **System Administrators**: Maintain and update the detection system.

#### 2.4 Operating Environment :-

* The system will run on servers within the organization's network and should be compatible with various operating systems such as Windows, Linux, and macOS.

#### 2.5 Design and Implementation Constraints :-

* The system must handle high volumes of network traffic data.
* Machine learning models should be updated regularly to adapt to new threats.
* The system should have minimal impact on network performance.

**2.6 User Documentation :-**

* User manuals and online help guides will be provided for administrators and analysts.

#### 2.7 Assumptions and Dependencies :-

* The organization has existing network monitoring infrastructure.
* Sufficient historical data is available for model training.

**# Specific Requirements :-**

#### 3.1 Functional Requirements :-

##### 3.1.1 Data Collection :-

* The system shall capture network traffic data in real-time.
* The system shall store captured data securely for further analysis.

##### 3.1.2 Data Preprocessing :-

* The system shall clean and normalize the captured data.
* The system shall handle missing and inconsistent data appropriately.

##### 3.1.3 Feature Extraction :-

* The system shall extract relevant features from network traffic data.
* The system shall allow customization of feature extraction parameters.

##### 3.1.4 Model Training :-

* The system shall support various machine learning algorithms (e.g., SVM, Random Forest, Neural Networks).
* The system shall provide a mechanism for model validation and performance evaluation.

##### 3.1.5 Real-Time Detection :-

* The system shall apply trained models to real-time data for anomaly detection.
* The system shall detect and classify different types of cyber attacks (e.g., DDoS, phishing, malware).

##### 3.1.6 Alerting :-

* The system shall generate alerts for detected anomalies.
* The system shall provide configurable alert thresholds.

##### 3.1.7 Reporting :-

* The system shall generate detailed reports on detected threats.
* The system shall provide visualizations of attack patterns and trends.

#### 3.2 Non-Functional Requirements

##### 3.2.1 Performance

* The system shall process network data with minimal latency.
* The system shall handle large volumes of data efficiently.

##### 3.2.2 Reliability

* The system shall provide accurate and reliable detection of cyber attacks.
* The system shall have a high availability rate.

##### 3.2.3 Usability

* The system shall have an intuitive user interface.
* The system shall provide easy-to-understand alerts and reports.

##### 3.2.4 Security

* The system shall ensure the confidentiality and integrity of captured data.
* The system shall have access control mechanisms to restrict unauthorized use.

#### 3.3 Interface Requirements

##### 3.3.1 User Interfaces

* The system shall provide a dashboard for real-time monitoring.
* The system shall provide interfaces for configuring detection parameters and viewing reports.

##### 3.3.2 Hardware Interfaces

* The system shall integrate with existing network hardware for data capture.

##### Software Interfaces

* The system shall integrate with other security tools (e.g., SIEM systems).

**# Softwares Used :-**

* In developing and deploying a system for detecting cyber attacks in a network using machine learning techniques, several software tools and technologies are typically used. Here is a breakdown of the different categories of software that might be involved:

### 1. Data Collection and Preprocessing

* **Packet Capture Tools**:
  + Wireshark
  + tcpdump
  + TShark
* **Data Preprocessing Libraries**:
  + Pandas (Python)
  + NumPy (Python)
  + Scikit-learn (Python)

### 2. Feature Extraction

* **Feature Extraction Libraries**:
  + Scikit-learn (Python)
  + Featuretools (Python)
  + Tsfresh (Python)

### 3. Machine Learning Model Development

* **Machine Learning Libraries**:
  + Scikit-learn (Python)
  + TensorFlow (Python)
  + Keras (Python)
  + PyTorch (Python)
  + XGBoost (Python)
  + LightGBM (Python)

### 4. Model Training and Validation

* **Model Training Tools**:
  + Jupyter Notebook (for interactive model development)
  + Google Colab (for cloud-based model training)
  + AWS SageMaker (for scalable model training on the cloud)

### 5. Real-Time Detection

* **Real-Time Data Processing Frameworks**:
  + Apache Kafka (for real-time data streaming)
  + Apache Flink (for real-time data processing)
  + Apache Storm (for real-time data processing)

### 6. Alerting and Reporting

* **Alerting Tools**:
  + Prometheus (with Alertmanager)
  + Nagios
  + Zabbix
* **Reporting and Visualization**:
  + Grafana (for real-time dashboards)
  + Kibana (for data visualization and reporting)
  + Matplotlib (Python library for plotting)
  + Seaborn (Python library for statistical data visualization)

### 7. User Interface

* **Web Frameworks**:
  + Django (Python)
  + Flask (Python)
  + React (JavaScript)
  + Angular (JavaScript)
  + Vue.js (JavaScript)

### 8. Database Management

* **Relational Databases**:
  + MySQL
  + PostgreSQL
* **NoSQL Databases**:
  + MongoDB
  + Cassandra
  + Elasticsearch (for search and analytics)

### 9. Security and Compliance

* **Security Tools**:
  + OpenSSL (for encryption)
  + HashiCorp Vault (for secrets management)
  + SELinux (for enforcing security policies on Linux systems)

### 10. DevOps and Deployment

* **Containerization**:
  + Docker
* **Orchestration**:
  + Kubernetes
* **CI/CD Pipelines**:
  + Jenkins
  + GitLab CI/CD
  + Travis CI
* **Infrastructure as Code**:
  + Terraform
  + Ansible

### 11. Monitoring and Logging

* **Monitoring Tools**:
  + Prometheus
  + Grafana
* **Logging Tools**:
  + ELK Stack (Elasticsearch, Logstash, Kibana)
  + Splunk
  + Fluentd

### Integrated Development Environments (IDEs)

* **IDEs**:
  + PyCharm (Python)
  + Visual Studio Code
  + Jupyter Notebook
* These tools and technologies provide a robust ecosystem for building, training, deploying, and monitoring a machine learning-based cyber attack detection system. The choice of specific tools can depend on factors such as the team's expertise, project requirements, and existing infrastructure.

### # External Interface Requirements :-

#### User Interfaces

* **Dashboard Interface**: Web-based interface for real-time monitoring, displaying traffic visualization, alerts, and model performance.
* **Configuration Interface**: Web interface for setting detection parameters, thresholds, and feature extraction options.
* **Reporting Interface**: Web interface for generating and viewing reports on detected threats, with options to download in PDF or CSV.

#### Hardware Interfaces

* **Network Interface Cards (NICs)**: Capture network traffic data with support for various NICs and high-throughput capabilities.
* **Servers**: High-performance servers for processing and analyzing network traffic data with multi-core processors, high memory, and fast storage.

#### Software Interfaces

* **SIEM Systems Integration**: APIs for data exchange with SIEM platforms like Splunk, ArcSight, and QRadar.
* **Network Monitoring Tools Integration**: APIs for data exchange with tools like Nagios, Zabbix, and Prometheus.

#### Communication Interfaces

* **Email Notifications**: SMTP configuration for sending customizable email alerts to administrators and analysts.
* **SMS Alerts**: Integration with SMS gateway services for sending critical notifications.
* **Webhooks**: Configurable endpoints for real-time alerting and integration with other systems.

#### Data Interfaces

* **Database Interfaces**: Interaction with relational (MySQL, PostgreSQL) and NoSQL (MongoDB, Elasticsearch) databases for data storage and retrieval.
* **Data Import/Export**: Support for importing data from and exporting data to CSV, JSON, or XML files for historical analysis and backup.

### # System Features :-

#### 1. Real-Time Network Traffic Monitoring

* **Description**: Continuously monitor and capture network traffic data in real-time.
* **Benefits**: Enables immediate detection and analysis of potential cyber threats as they occur.

#### 2. Data Preprocessing

* **Description**: Clean, normalize, and prepare captured network traffic data for analysis.
* **Benefits**: Ensures data quality and consistency for accurate analysis and machine learning model training.

#### 3. Feature Extraction

* **Description**: Extract relevant features from network traffic data for use in machine learning models.
* **Benefits**: Enhances the ability of models to identify patterns and anomalies indicative of cyber attacks.

#### 4. Machine Learning Model Training

* **Description**: Train machine learning models using historical network traffic data to identify potential threats.
* **Benefits**: Leverages historical data to build predictive models capable of detecting various types of cyber attacks.

#### 5. Real-Time Anomaly Detection

* **Description**: Apply trained machine learning models to real-time network traffic data to detect anomalies and potential threats.
* **Benefits**: Provides timely identification of unusual or suspicious activity within the network.

#### 6. Attack Classification

* **Description**: Classify detected anomalies into specific types of cyber attacks, such as DDoS, phishing, or malware.
* **Benefits**: Enables targeted response and mitigation strategies based on the type of attack detected.

#### 7. Alerting System

* **Description**: Generate and send alerts to administrators and security analysts when potential threats are detected.
* **Benefits**: Ensures prompt notification and response to emerging threats, reducing the time to mitigate risks.

#### 8. Configurable Alert Thresholds

* **Description**: Allow users to set and adjust thresholds for generating alerts based on the severity and type of detected anomalies.
* **Benefits**: Provides flexibility in tuning the sensitivity of the alerting system to balance between false positives and missed threats.

#### 9. Detailed Reporting

* **Description**: Generate comprehensive reports on detected threats, including details on the nature and source of the anomalies.
* **Benefits**: Facilitates in-depth analysis and understanding of network security incidents for further investigation and improvement.

#### 10. Visualizations and Dashboards

* **Description**: Provide visual representations of network activity, detected anomalies, and trends over time through interactive dashboards.
* **Benefits**: Enhances situational awareness and makes complex data more accessible and interpretable for users.

#### 11. Integration with SIEM and Network Monitoring Tools

* **Description**: Integrate with existing Security Information and Event Management (SIEM) systems and network monitoring tools for comprehensive security management.
* **Benefits**: Allows for centralized monitoring and correlation of security events across different tools and platforms.

#### 12. Data Import/Export

* **Description**: Support importing historical data and exporting analysis results in various formats (CSV, JSON, XML) for backup and further analysis.
* **Benefits**: Ensures data portability and ease of use for further analysis, reporting, and archival purposes.

#### 13. Security and Compliance

* **Description**: Ensure the confidentiality, integrity, and availability of captured and processed data, complying with relevant data protection regulations.
* **Benefits**: Protects sensitive data and adheres to legal and regulatory requirements, enhancing the overall security posture of the organization.

#### 14. User Management and Access Control

* **Description**: Implement role-based access control to manage user permissions and ensure only authorized personnel can access specific system features and data.
* **Benefits**: Enhances security by restricting access to sensitive information and system functionalities based on user roles.

### # Acceptance Criteria :-

#### 1. Real-Time Network Traffic Monitoring

* **Criteria**: The system captures and displays network traffic data in real-time without significant delay.
* **Measurement**: Real-time data should be available within 1 second of capture.

#### 2. Data Preprocessing

* **Criteria**: The system successfully cleans and normalizes network traffic data, handling missing and inconsistent data appropriately.
* **Measurement**: Preprocessed data is free from errors and inconsistencies, verified through data quality checks.

#### 3. Feature Extraction

* **Criteria**: The system extracts relevant features from network traffic data accurately and consistently.
* **Measurement**: Extracted features match predefined criteria and are available for model training and real-time detection.

#### 4. Machine Learning Model Training

* **Criteria**: The system trains machine learning models using historical data and provides performance metrics.
* **Measurement**: Models achieve an accuracy, precision, and recall of at least 90% on validation datasets.

#### 5. Real-Time Anomaly Detection

* **Criteria**: The system detects anomalies in real-time network traffic data using trained models.
* **Measurement**: Anomalies are detected and flagged within 1 second of occurrence with a false positive rate below 5%.

#### 6. Attack Classification

* **Criteria**: The system accurately classifies detected anomalies into specific types of cyber attacks.
* **Measurement**: Attack classification accuracy is at least 85%, verified through cross-validation with labeled datasets.

#### 7. Alerting System

* **Criteria**: The system generates alerts for detected anomalies and sends notifications to administrators.
* **Measurement**: Alerts are generated and delivered within 5 seconds of anomaly detection, with customizable thresholds.

#### 8. Configurable Alert Thresholds

* **Criteria**: Users can set and adjust alert thresholds through the configuration interface.
* **Measurement**: Changes to alert thresholds take effect immediately and are correctly applied to new data.

9. Detailed Reporting

* **Criteria**: The system generates comprehensive and accurate reports on detected threats, available in multiple formats.
* **Measurement**: Reports are generated within 10 seconds of request and include all relevant details and visualizations.

#### 10. Visualizations and Dashboards

* **Criteria**: The system provides interactive dashboards with real-time visualizations of network activity and detected anomalies.
* **Measurement**: Dashboards update in real-time (within 1 second) and are user-friendly and informative.

#### 11. Integration with SIEM and Network Monitoring Tools

* **Criteria**: The system integrates seamlessly with existing SIEM and network monitoring tools, enabling data exchange.
* **Measurement**: Data is successfully exchanged and integrated with SIEM tools, with no data loss or corruption.

#### 12. Data Import/Export

* **Criteria**: The system supports importing and exporting data in CSV, JSON, and XML formats.
* **Measurement**: Data import/export operations complete successfully within 5 seconds and maintain data integrity.

#### 13. Security and Compliance

* **Criteria**: The system ensures data confidentiality, integrity, and availability, complying with relevant regulations.
* **Measurement**: Security measures are in place and verified through security audits, with no compliance violations.

#### 14. User Management and Access Control

* **Criteria**: The system implements role-based access control, restricting access to sensitive features and data.
* **Measurement**: Access control rules are enforced correctly, with only authorized users able to access restricted areas.

### # Deliverables :-

#### 1. Project Plan

* **Description**: Detailed project plan outlining the scope, timeline, milestones, and resources required for the project.
* **Format**: Document (PDF or Word)

#### 2. System Design Document

* **Description**: Comprehensive design document detailing the architecture, components, data flow, and technical specifications.
* **Format**: Document (PDF or Word)

#### 3. Source Code

* **Description**: Complete source code for all modules, including data collection, preprocessing, feature extraction, model training, real-time detection, alerting, and reporting.
* **Format**: Source files (organized in repositories)

4. Trained Machine Learning Models

* **Description**: Pre-trained machine learning models ready for deployment.
* **Format**: Model files (e.g., .h5, .pkl)

#### 5. Configuration Files

* **Description**: Configuration files for setting up the system, including parameters for feature extraction, alert thresholds, and integration settings.
* **Format**: Text files (e.g., .json, .yaml)

#### 6. User Interfaces

* **Description**: Fully functional web-based user interfaces for monitoring, configuration, and reporting.
* **Format**: Deployed web applications or source files

#### 7. Documentation

* **User Manual**: Instructions for using the system, including the dashboard, configuration interface, and reporting tools.
* **Technical Documentation**: Detailed technical documentation for developers and system administrators.
* **API Documentation**: Documentation for any APIs provided by the system for integration with other tools.
* **Format**: Documents (PDF or Word)

#### 8. Test Plans and Reports

* **Description**: Detailed test plans, test cases, and test reports for unit testing, integration testing, system testing, and user acceptance testing.
* **Format**: Documents (PDF or Word)

#### 9. Deployment Scripts

* **Description**: Scripts and instructions for deploying the system in a production environment.
* **Format**: Script files (e.g., .sh, .bat) and documents (PDF or Word)

#### 10. Training Materials

* **Description**: Training materials and sessions for end-users and administrators, including presentations and hands-on training sessions.
* **Format**: Documents (PDF or Word), presentations (e.g., PowerPoint), videos

11. Support and Maintenance Plan

* **Description**: Plan for ongoing support and maintenance, including issue tracking, updates, and upgrades.
* **Format**: Document (PDF or Word)

#### 12. Final Report

* **Description**: Summary of the project, including objectives, outcomes, challenges, and lessons learned.
* **Format**: Document (PDF or Word)

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form