**Real-Time Network Intrusion Detection Using Wireshark and Advanced Ensemble Learning Techniques**

**Software Requirements Specification**

Version 1.0



**Group Id: <Mention your group id>**

**Supervisor Name: Laraib Sana**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| Current date | 1.0 | Introduction of the project | Write student(s) id |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. [Scope (of the project)](#scope)
2. [Functional Requirements Non Functional requirements](#FRNFR)
3. [Use Case Diagram](#UCD)
4. [Usage Scenarios](#UCS)
5. [Adopted Methodology](#Adopted)
6. [Work Plan (Use MS Project to create Schedule/Work Plan)](#Gantt)

**SRS Document**

**Scope of Project:**

The project aims to develop an intrusion detection system (IDS) that can analyze real-time network traffic to identify potential cyber threats. It involves using Wireshark to capture network traffic, preprocessing the data for machine learning analysis, and leveraging advanced ensemble models like TabNet, CatBoost, and LightGBM for intrusion detection. A user-friendly web application will also be created for uploading and analyzing network data in real-time.  
**Key Deliverables**:

* Real-time network traffic analysis.
* Preprocessing and feature extraction pipeline.
* Implementation of machine learning models.
* Web application for user interaction and analysis visualization.

**Functional and NonFunctional Requirements:**

**Functional Requirements**

Traffic Capture:

The system must use Wireshark to capture real-time network traffic data.

Captured data must be saved as a CSV file for preprocessing and analysis.

Data Preprocessing:

Clean raw traffic data by handling missing or inconsistent values.

Encode categorical features such as protocol types and normalize numerical features like packet size and time intervals.

Label the dataset with normal and malicious traffic for supervised machine learning.

Model Training and Evaluation:

Train ensemble models, including TabNet, CatBoost, and LightGBM, using the preprocessed dataset.

Evaluate models using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

Intrusion Detection:

Analyze uploaded network traffic data using trained machine learning models to detect potential intrusions.

Classify network traffic as normal or malicious and generate detailed results.

Web Application:

Provide an intuitive interface for users to upload CSV files of network traffic data.

Display real-time analysis results, including key metrics and influencing factors.

Allow users to save or export the analysis results for further review.

Non-Functional Requirements

Performance:

The system must process and analyze data within 2 seconds for real-time functionality.

Scalability:

The system must handle datasets of up to 1 GB without performance degradation.

Usability:

The web interface must be user-friendly and accessible to technical and non-technical users.

Reliability:

The system should provide consistent results under various traffic loads.

Security:

Ensure secure upload and processing of data to prevent unauthorized access or manipulation.

Validate input files to protect against malicious or corrupt data uploads.

Portability:

The application must be compatible with multiple operating systems, including Windows and Linux.

Maintainability:

Use a modular design to facilitate updates and additions, such as integrating new machine learning models.

Data Integrity:

Ensure the integrity of processed and analyzed data, avoiding loss or corruption.

These requirements provide a clear foundation for developing a robust, efficient, and user-friendly intrusion detection system.

Use Case Diagram(s):

<Provide here the use case diagram of your system>

Usage Scenarios:

<Provide here the usage scenarios of all use cases in table format explaining Use Case title, Use Case Id, Actions, Description, Alternative Paths, Pre and Post Conditions, Author, Exceptions. You are supposed to provide a usage scenario for each of use case shown in use case diagram>

**Adopted Methodology**

For this project, the **VU Process Model** is adopted, combining the structured approach of the **Waterfall Model** with the iterative flexibility of the **Spiral Model**. This ensures a systematic development process while allowing refinements as needed.

**Phases of the Methodology**

1. **Requirement Analysis (Waterfall Phase)**:
   * Gather and document project requirements, including functional and non-functional specifications.
   * Define the scope of work and deliverables.
2. **Design and Planning (Waterfall Phase)**:
   * Design the system architecture, including data capture, preprocessing pipeline, machine learning models, and web application.
   * Create a detailed project timeline using a Gantt chart.
3. **Development (Spiral Phase)**:
   * **First Iteration**: Implement Wireshark integration for traffic capture and CSV export.
   * **Second Iteration**: Develop and test preprocessing and machine learning modules.
   * **Third Iteration**: Build and refine the web application for real-time analysis.
   * Adjust based on iterative feedback after each phase.
4. **Testing (Spiral Phase)**:
   * Conduct unit testing for individual modules (e.g., data preprocessing, machine learning models).
   * Perform integration and system testing to ensure overall functionality.
5. **Deployment (Waterfall Phase)**:
   * Deploy the finalized system with a user-friendly interface and document instructions for usage.

**Advantages of this Approach**

* **Structure**: The waterfall phase ensures clear requirements and structured planning.
* **Flexibility**: The spiral phase allows iterative development and adjustments based on testing feedback.
* **Risk Mitigation**: Continuous testing and refinement reduce the risk of major issues.

This methodology ensures the project is delivered on time, meets quality standards, and achieves the desired objectives.

Work Plan (Use MS Project to create Schedule/Work Plan)

<Provide Gantt chart of your final project>