

AI-Based Network Intrusion Detection System

Technical Implementation Manual

1. Introduction

This document serves as the primary implementation guide for the **AI-Powered Network Intrusion Detection System (NIDS)**. It outlines the necessary environment prerequisites, installation procedures, and execution steps required to deploy the Machine Learning model and the real-time monitoring dashboard.

2. Environment Setup & Prerequisites

To ensure successful deployment, the host machine must meet the following software requirements:

2.1 Core Software

1. **Python Interpreter (v3.8 or higher)**
 - **Source:** [Official Python Website](#)
 - **Critical Configuration:** During installation, ensure the checkbox "**Add Python to PATH**" is selected. This allows Python commands to be executed from the terminal.
2. **Integrated Development Environment (IDE)**
 - **Recommended:** Visual Studio Code (VS Code)
 - **Alternative:** PyCharm or Jupyter Notebook

2.2 System Tools

- **Command Line Interface:** Windows Command Prompt (cmd), PowerShell, or macOS/Linux Terminal.

3. Installation & Configuration

3.1 Workspace Initialization

1. Navigate to your Desktop or preferred directory.
2. Create a new directory named **AI_NIDS_Project**.
3. Launch **VS Code** and select **File > Open Folder**, choosing the directory created in the previous step.

3.2 Dependency Management

The project relies on specific Python libraries for data processing, machine learning, and

visualization.

1. Open the integrated terminal in VS Code (Ctrl + `).
2. Execute the following command to install all dependencies:
`pip install pandas numpy scikit-learn streamlit seaborn matplotlib`

Library Overview:

- pandas & numpy: Data manipulation and numerical operations.
- scikit-learn: Implementation of the Random Forest algorithm.
- streamlit: Framework for the web-based dashboard UI.
- seaborn & matplotlib: Data visualization and plotting.

4. Source Code Integration

4.1 Script Creation

1. Inside the `AI_NIDS_Project` directory, create a new file named **nids_main.py**.
2. Populate this file with the Python source code provided in the project artifacts (refer to the `nids_main.py` block).
3. Save the file (Ctrl + S).

5. Data Management Strategy

5.1 Simulation Mode (Default)

The application includes a built-in data simulator using numpy. This allows the system to function immediately for demonstration purposes without external dependencies. No action is required to enable this mode.

5.2 Production Mode (Real-World Data)

For advanced demonstrations using the **CIC-IDS2017** benchmark dataset:

1. **Source:** Download the CSV files from the [UNB CIC Dataset Repository](#).
2. **Integration:** Move the downloaded .csv file into the `AI_NIDS_Project` root directory.
3. **Code Adaptation:** In `nids_main.py`, locate the `load_data()` function and replace the simulation logic with:

```
# Example for real data loading
df = pd.read_csv('Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv')
```

6. Execution & Operation

To launch the NIDS Dashboard, perform the following steps:

1. Ensure your terminal path is set to the project directory: `AI_NIDS_Project`.
2. Execute the startup command:
`streamlit run nids_main.py`
3. **Access:** The system will initialize a local web server and automatically open the default web browser to `http://localhost:8501`.

Operational Workflow:

1. **Dashboard Overview:** Review the project description.
2. **Model Training:** Click the **"Train Model Now"** button in the sidebar to initialize the Random Forest Classifier.
3. **Live Simulation:** Use the "Live Traffic Simulator" section to input packet parameters and test the model's detection capabilities.

7. References & Technical Resources

- **Language Documentation:** [Python.org Docs](https://docs.python.org/3/)
- **Editor:** [Visual Studio Code](https://code.visualstudio.com/)
- **Dataset:** [CIC-IDS2017 - Canadian Institute for Cybersecurity](https://www.cicids.org/)
- **UI Framework:** [Streamlit Documentation](https://streamlit.io/)
- **ML Algorithm:** [Scikit-Learn Random Forest](https://scikit-learn.org/stable/modules/random_forest.html)

8. Troubleshooting

- **'streamlit' is not recognized:** This usually means Python was not added to your PATH. Reinstall Python and check the "Add to PATH" box, or try running `python -m streamlit run nids_main.py`.
- **ModuleNotFoundError:** Ensure you ran the pip install command in Section 3.2.
- **Browser doesn't open:** Manually open your browser and type `http://localhost:8501`.

9. Conclusion

This implementation guide provides the necessary steps to build a functional AI-driven security tool. By following these instructions, you will have a working prototype capable of training a model and detecting simulated network anomalies, suitable for academic demonstration and further research.