**Network Intrusion Detection System**

**Project Description:**

**Implementation:**

1. Generate a Pcap File using Wireshark and Download
2. Convert Pcap file to CSV format using rdpcap from scapy library
3. Read this CSV file using Pandas

Procedure:

Step-by-Step Process

# Step 1: Data Collection and Preprocessing

1. Capture Network Traffic:

- Use network capture tools like Wireshark or tcpdump to collect network traffic data. Save this data in a `.pcap` file.

2. Convert `.pcap` to CSV:

- Use a Python script with Scapy to read the `.pcap` file and convert the captured packet data into a CSV format for easier manipulation and analysis.

3. Preprocess the CSV Data:

- Handle missing values, normalize numerical features (like time and length), and encode categorical features (like protocol types) to prepare the data for machine learning models.

# Step 2: Model Training

1. Load and Prepare Data:

- Load the preprocessed data from the CSV file into a pandas DataFrame. Split this data into training and testing sets.

2. Train a Neural Network Model:

- Use TensorFlow or Keras to build and train a neural network model on the training data. Save the trained model to a file.

# Step 3: Building the Flask Application

1. Set Up the Flask Application:

- Create a Flask application to serve the trained model. This application will handle HTTP POST requests, preprocess incoming data, and use the model to make predictions.

2. Run the Flask Application:

- Start the Flask application on a local server. This application will be ready to receive and process requests for intrusion detection.

# Step 4: Testing the Flask Application

1. Create a Test Script:

- Write a Python script to send a test request to the Flask application. This script will send sample data (such as packet information) to the server and receive the model's prediction as a response.

2. Run the Test Script:

- Execute the test script to verify that the Flask application is working correctly. Check the output to ensure that the server receives the data, processes it, and returns a prediction.

Expected Outcome

- Flask Server Logs:

- Logs will show incoming data, any preprocessing steps taken, and the prediction made by the model.

- Test Script Output:

- The script will print the server's prediction, confirming that the system is functioning as expected.

By following these steps, you will develop a functional Network Intrusion Detection System that captures network traffic, preprocesses the data, uses a machine learning model to detect anomalies, and provides real-time detection through a web-based interface.