IT303 – Software Engineering

Implementation of Port Scanning Techniques

Installation Report:

This guide provides detailed instructions to set up and run the Port Scanning project on your local machine. This project consists of a Python/Flask backend and a ReactJS frontend.

1. System Requirements

Before you begin, ensure your system meets the following requirements:

- **Operating System:** A **Linux environment** is mandatory. The backend uses Scapy for raw socket operations, which is not supported on Windows outside of a Linux virtual environment.
    - *Windows Users:* You must use the Windows Subsystem for Linux (WSL-2).
- **Privileges:** You must have **sudo (root) access** to run the backend server, as raw socket creation requires elevated permissions.
- **Software:**
    - Python 3.8+
    - python3-venv (Install with sudo apt install python3-venv on Debian/Ubuntu)
    - Node.js v14+
    - npm

2. Backend Setup (Flask & Scapy)

The backend server handles all the packet-crafting and scanning logic.

1. **Navigate to the Backend Directory** From the project's root directory:
   >> cd backend
2. **Create and Activate Virtual Environment** It's highly recommended to use a virtual environment to manage Python dependencies.
   >> python3 -m venv venv
   >> source venv/bin/activate

Your terminal prompt should now be prefixed with (venv).

3. **Install Dependencies** Install the required Python packages using pip.
   >> pip install -r requirements.txt
4. Open the config.py file, and give correct credentials for google account and app-password for the account from which you would want to send email to users for verification.

3. **Frontend Setup (ReactJS)**

The frontend provides the web interface for interacting with the scanner.

1. **Navigate to the Frontend Directory** from the project's root directory
   >> cd frontend
2. **Install Dependencies** Install the required Node.js packages.
   >> npm install

4. Running the Application

You must have **two separate terminals** open to run both the backend and frontend servers simultaneously.

Terminal 1: Run the Backend

1. Navigate to the backend directory.
2. Make sure your Python virtual environment is activated
   >> source venv/bin/activate
3. Start the Flask server using **sudo** for Scapy's raw socket permissions.
   >> sudo venv/bin/python3 app.py
4. The backend server will start and listen on http://127.0.0.1:5000. Keep this terminal running.

Terminal 2: Run the Frontend

1. Navigate to the frontend directory.
2. Start the React development server.
   >> npm start
3. This command will automatically open the application in your default web browser.

5. Accessing the Application

Once both servers are running, you can access the tool at:
http://localhost:3000

6. Troubleshooting

- **Error: Permission denied (when starting backend)**
  - You forgot to use sudo. The backend *must* be run with sudo python3 app.py to access raw sockets.
- **Frontend UI loads, but scans fail (e.g., "Network Error")**
  - Your backend server is likely not running or is not accessible. Check the status of Terminal 1. Ensure it's running on http://127.0.0.1:5000 and didn't crash.
- **pip install fails**
  - Ensure your venv is activated. Check your internet connection and that python3 and pip are correctly installed.
- **npm install fails**
  - Ensure you have a compatible version of Node.js and npm installed. Try deleting the node_modules folder and package-lock.json file, then run npm install again.
- **Error: ImportError:** cannot import name '...' from '...' (e.g., url_quote from werkzeug.urls)
  - Install a known compatible version of the dependency. For the Werkzeug example:
    >> pip install werkzeug==2.3.8

Prepared by:
Anirudh S (231IT006)
Siddharth S Kolkar (231IT071)
T Srujan Swamy (231IT079)