

===== K9 Packet Sniffer

Description:

K9 packet sniffer is, as the name suggests, a network packet sniffer. A packet sniffer is a tool used to capture, analyze, and monitor network traffic. Its primary purpose is to help users troubleshoot network issues, analyze data packets for debugging or forensics, and improve network security by identifying potential vulnerabilities.

How to Download:

1. Download the necessary files from our [GitHub](#) (sniffer_api.zip and src.zip); if not already installed.

Requirements:

Ensure you have the following installed and configured before using the program:

- Any IDE
- [Python v3.10+](#)
- [NodeJS v22.11.0](#)
 - Then run the following commands on command prompt:
 - npm install -g create-react-app
 - npm install react-icons react-router-dom web-vitals
- [SQL Express](#) and [SQL Server Management Studio 19](#).
 - SQL latest version
 - SSMS v20.2
- Python Libraries (Install using pip): [bootstrap.pypa.io/get-pip.py](#)
 - [Scapy](#)
 - [Psutil](#)
 - [Pyodbc](#)
 - [Flask](#)
 - [flack-cors](#)

How to Set Up the Program:

SQL Setup:

1. After downloading SQL and SSMS, open SSMS and connect with the following information:
 - Server Name: localhost\SQLEXPRESS
 - Authentication: Windows Authentication
 - No server certificate required
2. Once done, open the SQLSetup file with SSMS and then run the queries given to setup the database.

NodeJS Setup:

To set up NodeJS, simply go to the official NodeJS website and download the version of NodeJS listed above. Upon installation, head into your terminal and execute `npm install -g create-react-app` (create-react-app can be changed to the name you desire).

For example:

- `create-react-app <FolderName>`

Upon doing this, move the `sniffer_api.py` and `src` files into the folder you just created.

Executing the program:

1. Open three node js terminals, and go to the directory where you downloaded the program files/where you created your nodejs app.
 - a. `cd sample/download/directory`
2. In the first terminal, execute the 'sniffer_api.py file':
 - a. `Python sniffer_api.py`
3. In the second terminal, execute the 'authentication_api.py' file:
 - a. `Python authentication_api.py`
4. In the third terminal, execute the following command:
 - a. `Npm start`
5. If everything was set up properly, the program should start on a new tab. From there you can follow the on-screen instructions to navigate through the program.

Possible Errors and Solutions:

- **Error 1:** [Description of the error, e.g., "ModuleNotFoundError"]
 - **Solution:** Run `pip install [missing module]`.
- **Error 2:** [Description of the error, e.g., "Database connection failed"]
 - **Solution:** Verify the database configuration in `[config file]` and ensure the database is accessible.
- **Error 3:** [Description of the error, e.g., "Permission denied"]

- **Solution:** Run the program with elevated permissions using `sudo` (Linux/Mac) or as an administrator (Windows).
- For any additional errors encountered, please contact any of the team members listed at the bottom of the page.

Disclaimers:

- This project is provided "as is" without any warranty. Use at your own risk.
- Ensure you comply with local laws and regulations when using this software.
- The developers are not responsible for any data loss, misuse, or issues caused by the use of this program.

How the Program Functions:

The program functions by combining the functionality of the `sniffer_api.py`, the `authentication_api.py`, and the front end. Enabling the user to have a smooth experience when capturing packets, and ease of access to previous capture sessions.

1. At program execution, 'authentication_api.py' works alongside the front end to ensure that the user logs in. If the user does not have an account, they can create one in the front menu. Once the user is created/entered, the SQL database is checked for matching sets of username and password, if there is a match, the user is granted access.
2. The user may select the settings option at the top of the page to be able to change their account's information (password, username, security questions, etc).
3. On the main screen, the user will see a screen with multiple layouts, one for session information, another one to hold the interface selected and action buttons, and lastly one for individual packet information.
4. The user must then select the appropriate interface in which they wish to capture packets from. Upon choosing the appropriate interface, they may then press 'start capture' to start gathering the information on the packets that are traversing around in their selected interface. Once the capture started, the user then gains the ability to click in a specific packet to take a look at the packet's more detailed information; furthermore, they can stop capture, save the capture or filter the captured packets by their desired element.
5. Saving the captured packets session results in the .pcap file for the captured packets being downloaded into the user's computer. Furthermore, it also adds the session information to the 'savedSessions' database and the individual packet information to the 'capturedPackets' database. These databases are used to store a user's session

information, so that they may be able to download their capture sessions at any time, and not only immediately after being done capturing the packets.

Contact Information:

If you encounter issues or have questions, contact:

- Carlos Imery
- Email: cimer001@fiu.edu

- Edward Medina
- Email: emedi074@fiu.edu

- Maritza Medina
- Email: mmedi185@fiu.edu

- Tiago Muller
- Email: tmull025@fiu.edu

- Sharek Rendon
- Email: srend011@fiu.edu

**===== End of
README**