Sniffer Readme

Hardware requirements:

- Raspberry Pi 4 Model B 4GB Ram [2 units]
    - Operating system used: kali-linux-2022.1-raspberry-pi-arm64
- Linux compatible TP-Link external antenna [2 units, optional]
- 15W Power Bank [optional for short field deployments]

External Python libraries:

- scapy [pre-installed with kali linux]
- firebase-admin
- netifaces
- ntplib

Dependencies:

- airmon-ng package [pre-installed with kali linux]

Installation:

- Install the kali linux operating system on the raspberry pi
- Set up no password root log in [for headless use in the field]
- Download the sniffer package from github and place in the desktop
- The sniffer folder should have a file called ServiceAccountKey.json, containing a token for a firestore account, this should be added by the user after creating a firestore database. the sniffer uploads data to the database and downloads configuration settings autonomously so this step is mandatory!
- Add execution privileges to the sniffer folder and it's files
- Run the script install_headless.py
    - install python dependencies
    - set up system boot service to run the sniffer automatically
    - set up the datetime zone to Jerusalem
- Reboot the device or run sniffer_main.py

No password root log in:

on the kali linux file system, make the following changes:

/etc/lightdm/lightdm.conf: change the following lines:
before:

- autologin-user=
- autologin-user-timeout=

after:

- autologin-user=root
- autologin-user-timeout=0

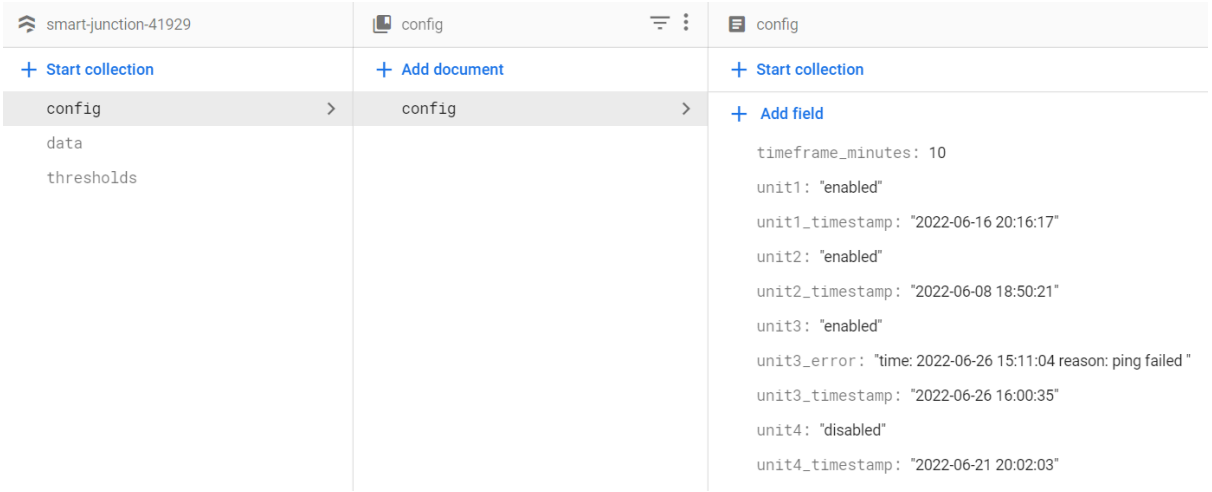
/etc/pam.d/lightdm-autologin: comment or delete the following line:

- auth required pam_succeed_if.so user != root quiet_success

after this step, the raspberry pi should boot as root user without entering a password.

Database set up:

The database should have a collection called config, and inside a document called config, containing values used to control the sniffers.
these values could be configured in the desktop GUI for ease of use



Configurable fields: [these values must exist in order to operate properly!]

- **timeframe_minutes:** intervals in which the sniffer stops capturing to upload the data to the database
- **unit1:** either “enabled” or “disabled” to control auto start after boot when in headless operation [same for other unit numbers]

Read only fields:

- **unit1_timestamp:** last known time in which the device sent a query to the database
- **unit1_error:** last known error reported by the device [hard reset due to ping failures, watchdog failures etc..]

Deployment:

- The 2 raspberry pi devices should be placed in either side of a crosswalk, with a power source and internet connection in order to operate.
- if set up correctly, the devices should operate autonomously at the field, and upload the data to the database at the configured intervals.
- once the data is uploaded to the database, it could be downloaded in the desktop GUI to analyze it.