Build

```
# move to build folder
cd Cybersecurity_Lesson
mkdir build
cd build

# configure, build, and install app
cmake ..
make
make install
```

Project Layout

This section will denote the overall structure and layout of the cybersecurity lesson in qt.

Libs

This folder contains third-party libraries we integrated into our project.

- ***NetworkManagerQt*** is a stripped down version of the library included as part of the KDE Framework.
    - NetworkManagerQt can be built with Qt6, but the framework still depends on Qt5.
    - This variation of NetworkManagerQt does not have any KDE Framwork specific cmake rules.
- ***Radiotap*** is a library for parsing the radiotap header in network packets.
    - This library helps with packet filtering in the deauther attack.
    - The radiotap header often varies in size and causes issues when trying to parse nearby network traffic.
    - The Radiotap library is normally available in the scope of the linux kernel. This version has been adjusted to operate stand alone.

Models

This folder contains models based on the Qt concept of Model/View Programming. These models help manage and collect information on networks.

- ***Iface_Model*** is a model for collecting information on network interfaces connected to the device.
  - The model handles adding and removeing interfaces as they appear.
  - The model uses the QNetworkInterface library to find network adapters that are currently connected.
- ***Network_Model*** uses NetworkManagerQt to collect information on different Access Points near by.
  - The model currently collects the connection state, device name, device path, name, essid, uuid, security type, and specific path.
- ***Station_Model*** finds the mac address of nearby devices also referred to as stations.
  - The model uses libpcap to collect packets being sent between devices and access points.
  - The model filters this information to identify the devices mac address and the access point it wants to connect to.

Deauther

The deauther attack forcibly disconnects a device from a nearby access point

- The first combobox uses the network interface model to allow the user to select the interface they will use to carry out the attack. The interface must support monitoring mode.
- The second combobox sets the reason we are telling the access point that the device wants to disconnect. In theory the reasons has no effect on the success of a attack, but in the real world, the access points programming might react differently depending on the disconnect reason.
- The Start/Stop Monitoring button uses the libpcap library to create a monitoring interface from the selected network interface. The interface is then used to start data collection from the station model.
- The table lists devices that have been discovered with the station model.
- The Deauther Attack button starts sending deauth packets to disconnect the currently selected device in the table.

Rubber_Ducky

The rubber ducky attack provides the users with the necessary buttons to create a script to change the background of a windows computer.

- The save icon, on the top right of the view window, saves the created script as a text file. The save path is based on the current home directory.

War Driving

The wardriving attack uses the Network Model to collect information on nearby access points and display the information to the user.

- The view window displays the access point name and security type.
- Press the Start/Stop button to start network scanning.
- The contents of the view window may also be exported to a csv.

Styles

This folder contains color settings for the app's ui elements. This includes reimplementing some of the default ui elements.

Fonts

Contains fonts to embed in the app. This ensures the fonts are available on the Raspberry Pi.

Icons

Contains icons to embed in the app. Many of the icons were based on icons from the breeze theme's icon pack.