Iot Lab User Manual

By: Jeremy Gluck, Alex Winstead, Todd St. Onge, Cole Clements, Xuchao Jiang (Steven)

Table of Contents:

Internet of Things Lab Mission

The Florida Teach Internet of Things lab, also known as the IoT Lab, was created for Florida Tech's faculty and students to conduct IoT cybersecurity research. As IoT devices become more prevalent in the consumer market, companies begin to cut corners and sometimes sacrifice the user's security and or privacy in the process. Our mission is to conduct research on the security and privacy of these devices in order to protect the end users and keep IoT device manufacturers accountable. In order to do this our lab is set up with a wide range of devices with the ability to capture any kind of signal a device may output. It is important for the IoT lab to be up to date with any trends. This will allow us to produce the most current, meaningful, and impactful research possible.

Iot Lab Collaboration Guidelines

Summary

It is important that this lab can be used effectively by multiple participants . This lab space should be able to have multiple projects all happening concurrently. In order to achieve this guidelines for collaboration must be established so that one party's research does not interfere with another's.

Section 1: General Lab Space

As this space will be used by many people at once it is important to have some basic ground rules for everybody with lab access to follow.

1. Only students and faculty with lab access may enter and or use the lab and its equipment.
2. Only research will be done in the lab, it is not a place to hang out or goof off in.
3. Nobody is able to remove equipment from the lab without approval from faculty
4. Please keep the lab space clean and free of trash
5. There is no designated computer for you work, all workstations are shared
6. If any device information (ie password or logins) needs to be changed for research, please inform all lab members and/or be sure to return to specified defaults when done.
7. If an unsupervised test must be done please inform all lab members or leave a note so the experiment is not interfered with
8. Please do not bring any food or drink into the lab
9. Please do not set up any devices on personal account
10. Please keep communal documents up to date
11. Any disputes or issues should be brought up with faculty

Section 2: Technical Difficulties

In the event that a device breaks or is not working as intended be sure to initially follow some simple troubleshooting steps before looking to replace this device. Troubleshooting should include following the troubleshooting guides for the device provided by the company of said device. If this does not work then the support number for that device may need to get called. Be sure to collect the proper information before this call is made as it may require a form of two-factor authentication. If the device must be factory reset the nbe sure to follow the steps outlined in the section below for the corresponding device. If all else fails and the device is broken or must get returned for a new one, please be sure to inform the other lab participants and discuss with a faculty member the next actions to take.

Hardware/Setup Instructions

Summary

In the Iot Lab, there are various lab equipment, devices, and other types of hardware that use or collect IoT signals such as Z-wave, Zigbee, and Wi-Fi. For the ease of the User, Hardware will be separated into various sections per device type.

Section 1: New Company Application Setup.

If you are using an IoT device that is from a company not found in the Iot Lab. Follow these instructions on how to add the company.

1. Operate the Lab phone.
2. Open the app store and search for the IoT device's Company's official device operating App, you will know it is the correct app if the creator of the app is the Company in question. If you still cannot find the app, visit the Company site or contact the Company for the app description and information.
3. Download and install the Company app.
4. Open the Company app.
5. In the Company app, create a new account and password using the standard Iot Lab login credentials, if unable to apply the standard Iot Lab login credentials. Attempt to use login credentials similar to the standard Iot Lab login credentials and add it as an addendum to the Iot Lab login credentials sheet.
6. Begin to pair and link the IoT Devices through the Company App.

Section 2: Setting Up the Hub Station.

Some IoT companies use a Hub station to link their devices to the router.

1. After assembling the Hub Station and powering it up. Operate the Lab phone to respective IoT company applications (I.E If it is a Ring brand Hub station, use the Ring Home App on the phone.).
    a. If the Hub Station you are pairing is using a brand that has not been applied to the phone, please refer to the instructions in Section 1: Company Application Setup.
2. Begin following the app instructions on how to properly pair and link the Hub Station (Instructions vary per app and brand).
3. After successfully pairing the Hub Station, test it to ensure everything is secured.

Section 3: Setting Up the Motion Sensors.

IoT Motion Sensors send IoT signals whenever they detect movement within their sensor range.

1. After assembling the Motion Sensor and powering it up. Operate the Lab phone to respective IoT company applications (I.E If it is a Ring brand Motion Sensor, use the Ring Home App on the phone.).
    a. If the Motion Sensor you are pairing is using a brand that has not been applied to the phone, please refer to the instructions in Section 1: Company Application Setup.
2. Begin following the app instructions on how to properly pair and link the Motion Sensor (Instructions vary per app and brand).
3. After successfully pairing the Motion Sensor, test it to ensure everything is secured.

Section 4: Setting Up the Doorbells.

IoT Doorbells send IoT signals whenever the door bell chime is activated. IoT Doorbells also include built in cameras and motion sensors.

1. After assembling the Doorbell and powering it up. Operate the Lab phone to respective IoT company applications (I.E If it is a Ring brand Doorbell, use the Ring Home App on the phone.).
    a. If the Doorbell you are pairing is using a brand that has not been applied to the phone, please refer to the instructions in Section 1: Company Application Setup.
    b. Some Doorbells are required to be hardwired to work properly. In this case, contact facilities to set up the device for you.
2. Begin following the app instructions on how to properly pair and link the Doorbell (Instructions vary per app and brand).
3. After successfully pairing the Doorbell, test it to ensure everything is secured.

Section 5: Setting Up the Cameras.

IoT Cameras send IoT signals whenever the Camera feed detects movement and disturbances.

1. After assembling the Camera and powering it up. Operate the Lab phone to respective IoT company applications (I.E If it is a Ring brand Camera, use the Ring Home App on the phone.).
    a. If the Camera you are pairing is using a brand that has not been applied to the phone, please refer to the instructions in Section 1: Company Application Setup.
    b. Some Cameras are required to be hardwired to work properly. In this case, contact facilities to set up the device for you.
2. Begin following the app instructions on how to properly pair and link the Camera (Instructions vary per app and brand).

3. After successfully pairing the Camera, test it to ensure everything is secured.

Section 6: Setting Up the Door Locks.

IoT Door locks send IoT signals whenever an unlock or locking action happens.

1. Begin setting up the Door Lock by setting up the device onto a door piece. This will require a screwdriver.
2. After assembling the Door Lock. Operate the Lab phone to respective IoT company applications (I.E If it is a Kwikset brand Door Lock, use the Kwikset Home App on the phone.).
    a. If the Door Lock you are pairing is using a brand that has not been applied to the phone, please refer to the instructions in Section 1: Company Application Setup.
3. Begin following the app instructions on how to properly pair and link the Door Lock (Instructions vary per app and brand).
4. After successfully pairing the Door Lock, test it to ensure everything is secured.

Types of Data to Collect

Summary

This section will outline what kinds of data should be collected for each type of device when performing general data capture. It is important to standardize this so other lab members, if allowed to, may use another's collected data for their own research. It is also important as companies may update or change their own protocols and or methods of data transmission. This means data capture should be performed in the same manner so it is easier to compare to past results.

Section 1: Door Locks

Door lock data to collect:

1. Locking the lock manually
2. Unlocking the lock manually
3. Locking the lock through manufacturer application
4. Unlocking the lock through manufacturer application
5. Long term data capture to check for passive signals
6. Blank captures for when device is inactive
7. Any other special features of device

This should be as many times as the research requires. If any additional features are present on the lock see if they are applicable to the sections below.

Section 2: Doorbells

Doorbell data to collect:

1. Ringing of doorbells
2. Motion activation of doorbell (if applicable)
3. Camera feed of doorbell (if applicable)
4. Screenshot transmission (if applicable)
5. Long term data capture to check for passive signals
6. Blank captures for when device is inactive
7. Any other special features of device

This should be as many times as the research requires. If any additional features are present on the lock see if they are applicable to the sections below.

Section 3: Motion Sensors

Motion sensor data to collect:

1. Motion alerts
2. Long term data capture to check for passive signals
3. Blank captures for when device is inactive
4. Any other special features of device

This should be as many times as the research requires. If any additional features are present on the lock see if they are applicable to the sections below.

Section 4: Cameras

Camera data to collect:

1. Transmission of camera feed
2. Long term data capture to check for passive signals
3. Blank captures for when device is inactive
4. Any other special features of device

This should be as many times as the research requires. If any additional features are present on the lock see if they are applicable to the sections below.

Section 5: Other devices

If a device does not fall into the general categories above, then first the function of the device must be determined. From this, guidelines on data capture for it can be developed. The most important thing to consider when capturing data of a new device is to have closure of device features. This means a method of capture should be developed for every use case of the device. The initial captures should also be kept atomic. This means only capture one action at a time. This will help to keep everything less complex when one goes to analyze it. Once the data is better understood, more complicated capture procedures may be developed.

Section 6: Protocols

The lab is set up with hardware to capture almost every transfer protocol. This includes the following:

1. 802.11
2. Zigbee and Zwave
3. Bluetooth
4. Radio frequencies

Being able to capture all protocols is important as devices may implement multiple protocols in their routine use. For example, the august locks are capable of working over both 802.11 and Bluetooth if the phone is close enough. Zigbee and Zwave are protocols becoming popular to IoT manufacturers as it offers more security than the 802.11 protocol. However the Z family of protocols can not communicate with a home router on their own. They must pass through either a

hub or similar device that will transmit the signal to a router either wirelessly or through a wire. It is also sometimes necessary to capture radio frequencies as some IoT hubs will use 433.42hz signals to communicate with their devices. Keep in mind these signals may be hard to decode as they are often proprietary.

Software in IoT Lab

Section 1: wireshark

https://www.wireshark.org/

Wireshark is a powerful software with GUI which is capable of capturing network data from internet devices. It also has built in features such as graphing to help analyze data. It can prove invaluable to giving a visual representation to data. Wireshark has a twin brother in the command line form called tshark. There is documentation on the official website of wireshark.

Section 2: tshark

https://www.wireshark.org/docs/man-pages/tshark.html

Tshark is the command line version of wireshark. It is also able to read and parse data from a pcap file accordingly similar to wireshark. It is also able to generate a csv file with filters directly from a pcap file, which can be useful for machine learning modeling. There is documentation for command line arguments on the official website of tshark.

Section 3: aircrack-ng package

https://www.aircrack-ng.org/

Aircrack-ng is an open source software which is commonly used for 802.11 attacks. It is also capable of sniffing WiFi data. With airmon-ng, a WiFi dongle can be set to monitor mode. Airodump-ng can be used to check the channel and connected devices of a hotspot. Aireplay-ng is used for attacks to the WiFi hotspot. Aircrack-ng is used for cracking WiFi passwords from pcap files after a certain data is collected. There is complete documentation on the official website of aircrack-ng.

Section 4: tcpdump

https://www.tcpdump.org/

Similar to airodump-ng, tcpdump is also a WiFi capturing software in the command line. With the ability to apply filters and save as a file, tcpdump can shrink down pcap files into specific

bits. There is complete documentation on the official website of tcpdump. There is also a great collection of tutorials and cheat sheets for tcpdump online.

Section 5: scapy

https://scapy.readthedocs.io/en/latest/
Scapy is a python library which is capable of dissecting wire and wireless data. It is able to start captures with filtering commands from tcpdump. Scapy has a massive list of compatible data types and there are multiple libraries on github which expands the library further. The information on scapy's documentation is very detailed and can be located at the above link..

Section 6: gqrx

https://gqrx.dk/
Without any Windows availability, gqrx is a software which detects radio waves with proper hardware. It is a type of SDR but just on the collector side. It highlights the frequency when data is transmitted. With a wide range of supported hardware, only Unix based systems are supported with proper drivers installed. Gqrx has a documentation page on the official website.

Section 7: Universal Radio Hacker (urh)

https://github.com/jopohl/urh
Universal Radio Hacker is an open source project on github which enables users to capture and save radio wave data. It supports a wide range of hardware. This softare can be easily used to capture 433 type signals.

Section 8: rfcat

https://github.com/atlas0fd00m/rfcat
Rfcat is a Python based radio data sniffing command line tool on github. It is compatible with Yard Stick One to capture and save Z-Wave data. Note rfcat needs initializations in order to capture data. Z-Wave plus is tested to be incompatible with this method of capturing.

Section 9: killerbee

https://github.com/riverloopsec/killerbee
Killerbee is a github open source software which is capable of sniffing zigbee data. ATZB-X-212B-USB is a compatible USB dongle. Command "zbdump" can be used to capture and save zigbee data similar to tcpdump.

Capture/Parsing Instructions

Summary

Data needs to be captured so it can be fed to the machine learning methods. Softwares used for WiFi captures: tcpdump, aircrack-ng, wireshark, tshark, and Python 3 or 2. Software for 433.92 Mhz capture: urh from github. Software for Z-Wave capture: rfcat.

Section 1: Capture WiFi Data

Applications airmon-ng and airodump-ng are used to capture wifi data. Command “sudo” needs to be added at the front for non-root users.

1. Use “airmon-ng wlan0 x” to start monitor mode on the wifi dongle while ‘x’ is the channel of the hotspot.
2. Use “airodump-ng wlan0mon” to figure out the channel and the bssid of the hotspot needed to be captured.
3. Use “airodump-ng wlan0mon -c x --bssid xx:xx:xx:xx:xx:xx” to sniff data from the specific hotspot. During this session, figure out if the device is showing up as a connected and active device.
4. Use “airodump-ng wlan0mon -c x --bssid xx:xx:xx:xx:xx:xx --output-format pcap -w rec” to save a capture session into a pcap file. This case, the file should be called “rec-xx.pcap” in the local directory.
5. Use Ctrl-C to stop the capture session. Repeat step 4 and 5 to generate multiple files for capture sessions.
6. Use “sudo chown YourUsernameHere rec-*” to change ownership of the capture files if not running on root.

Section 2: Parse WiFi Data

Captured pcap files need to be modified to be usable for machine learning models. Note, a unix operating system is needed. Code tested on Ubuntu desktop 19.10 and Ubuntu server 18.04.

1. For all data, obtain the Mac addresses of both the hotspot and the target iot device.
2. Put Mac addresses into the according Python script provided.
3. Modify the “capturefilename” variable to the recording filename.
4. Modify the loop in the Python script to match the total amount of files needed to be parsed. Make sure the odd labeled number is the locking action if parsing door lock data.

5. Move all captured files to the folder called “originals”
6. Run Python script with “python parse.py”

Section 3: Capture 433.92 Mhz Data

433.92 Mhz radio signal is used in multiple daily electronics, such as garage door remote and glass breaking detector. Software Universal Radio Hacker (urh) can be used to capture the signals. Tested with Ubuntu desktop 19.10 with HackRF SDR and RTL2832U SDR.

1. Install Universal Radio Hacker from github, follow instructions from the readme.md file.
2. Connect HackRF SDR or RTL2832U SDR and open urh.
3. Record when ready.

Please note when capturing these forms of radio signals that file sizes can get very large. A few seconds of capture can produce a file up to a gigabyte or greater. So take note if research requires captures of this type.

Section 4: Capture Z-Wave Data

Z-Wave is used in some home devices, which can be captured with rfcat and Yard Stick One. After identification of the frequency with gqrx and put under the correct settings, Z-Wave data can be captured into a list which can be stored in a binary file.

Section 5: Capture zigbee Data

Zigbee uses 16 channels from 2.4GHz spaced 5 MHz apart, with each channel using 2 MHz of bandwidth. It follows IEEE 802.15.4 wireless standard. It can be captured with killerzee from github. The command “zbdump” is used to capture data, the usage is very similar to tcpdump. Channel needs to be specified before capture.

Machine Learning

Summary

When trying to detect patterns systematically machine learning can be used to make great approximate models by using historical data. Generally there are two kinds of machine learning problems, regression (continuous dependent variable) and classification (discrete dependent variable). The typical example of regression is linear regression: given a set of cartesian points find the line through them that minimizes the distance of each point to the line. The typical example for classification is handwritten character identification: given a picture of a handwritten character determine which character is in the image.

Given the problem to determine the make and model of an IoT device based on past traffic from that device and others, how can we go about using machine learning to develop the model? First recognize that this is a classification problem: the wi-fi transmission and statistics of the transmission of the independent variables, and the make/model is the dependent class.

Section 1: Data Collection/Preprocessing

Follow the previous sections and create a labeled dataset. Some machine learning models may require the normalization (rescaling) of numerical variables, this is referred to as the preprocessing step. Whether or not variables must be normalized and to what range they should be normalized depends on the model architecture. An example of a model that requires normalization is a neural network. The model we chose for our pet problems, Random Forest Classifiers, does not require normalization.

Section 2: scikitlearn Usage Example

The following is an example of using scikitlearn with the dataset to determine if Locks are locking or unlocking based on traffic. In this example the dataset is split into a training set and a holdout or test set for validation.

```
# Imports
import pandas as pd
import sklearn as sk
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
```

```
data = pd.read_csv("Labled_Lock_Data.csv")

X = data[['Time', 'Num of Frames','Length','Median Length']] #
Features

y=data['Label']  # Labels

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3)

#Create a Gaussian Classifier
clf=RandomForestClassifier(n_estimators=100)

#Train the model using the training sets y_pred=clf.predict(X_test)
clf.fit(X_train,y_train)

y_pred=clf.predict(X_test)

# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

Lab Inventory

IoT Devices:

Below is a list of devices currently in the lab. For more information on each device and the most up to date list please reference the device list Google spreadsheet.

Ring Doorbell Chime
Simplisafe Dev (1)
Simplisafe Dev (2)
Nest Secure Alarm
Arlo Base Station
Nest Cam IQ
Nest Hello Doorbell
Amazon Alexa Show
Amazon Alexa Dot
Amazon Dot Kids
Geeni Camera (1)
Geeni Camera (2)
Geeni Camera (3)
Ring Base Station
Ring Outdoor Camera
Smartthings Hub
Phillips Hue Light Bridge
TP-Link Kasa Camera
Night Owl Doorbell Camera
Google Home Speaker
Ring Doorbell Pro (1)
Ring Doorbell (2)
Ring Doorbell (3)
Smartthings Camera
Roku TV (Left)
RokuTV (Right)
Geeni Doorbell Camera
Schlage Lock
August Lock (Connected to August Lock Hub)
August Lock Hub (August 3rd gen)
August Lock Hub (Yale hub)

Ultraloq Lock Bridge
Blink Camera (3)
Sifely Lock Hub
Lockly Lock Hub
Blink Camera (1)
Blink Camera (2)
Yale Lock (Connected to August Hub)
Yale Keyless Lock (Connected to Smartthings Hub)
KwickSet Door Lock (Connected to Smartthings Hub)
Smartthings Arrival Sensor
Smartthings Motion Sensor
Smartthings Multipurpose Sensor
Smartthings Outlet
Smartthings Water Leak Sensor
Smartthings Button

Other Lab Equipment:

Lab workstations (Ubuntu 19)
Raspberry pi’s
Oscilloscopes
Digital multimeters
Solder stations
HackRF SDR
Yard Stick One
Local multipurpose server
WiFi dongles
ATZB-X-212B-USB (used for zigbee captures)