BL(u)E CRAB:

RSSI Detection Pattern Analysis for Flagging System Development

Zhi Qu

Bluetooth Low Energy (BLE)

- Low power devices
- Application
  - Smart Home
  - Fitness devices
  - Trackers

Bluetooth Low Energy (BLE)

- Low power devices
- Application
  - Smart Home
  - Fitness devices
  - Trackers

- AirTag
- Tile
- Chipolo
- SmartTag

What is the Threat?

Stalking

- Making unwanted and persistent phone calls
- Approaching or showing up in places uninvited
- Following and watching the person
- Sending unwanted texts, emails, and social media messages
- Delivering unwanted gifts
- Utilizing technology for monitoring and tracking

Facing the Shadows: A Comprehensive Look at Stalking. (n.d.). https://safepass.org/2025/01/10/ facing-the-shadows-a-comprehensive-look-at-stalking

What is the Threat?

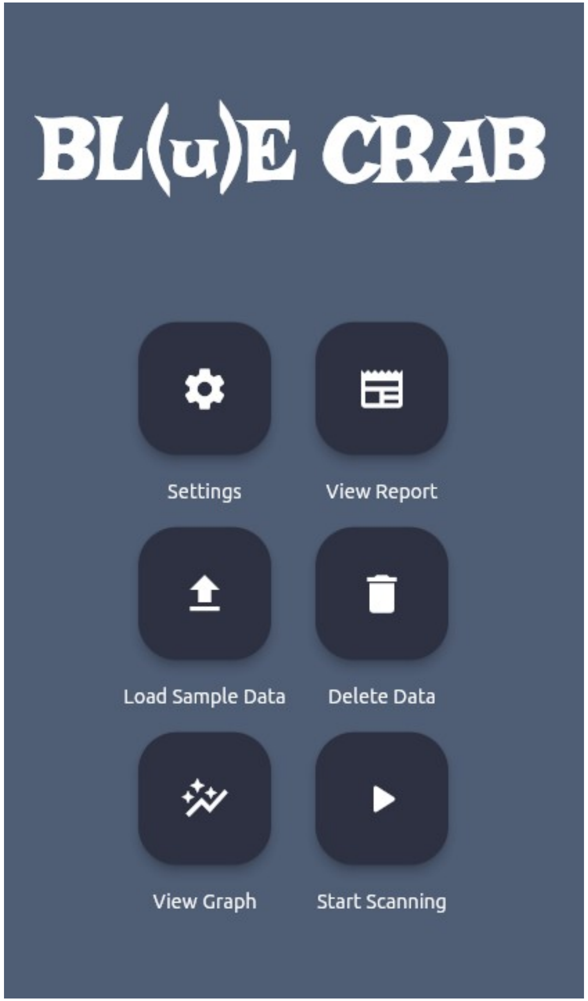
Stalking

- Making unwanted and persistent phone calls
- Approaching or showing up in places uninvited
- Following and watching the person
- Sending unwanted texts, emails, and social media messages
- Delivering unwanted gifts
- Utilizing technology for monitoring and tracking

Facing the Shadows: A Comprehensive Look at Stalking. (n.d.). https://safepass.org/2025/01/10/ facing-the-shadows-a-comprehensive-look-at-stalking

BL(u)E CRAB

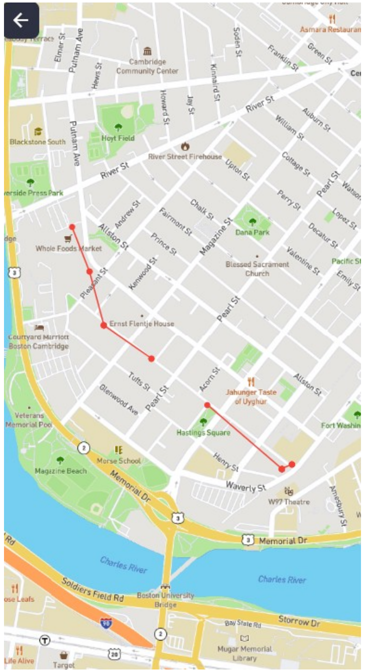
- Flutter app
- Scans for BLE devices nearby
- Assess risk
- Flags device
- Logs device info



D. Conklin, P. Pappachan, and R. Yus, “Bl (u) e crab: A user-centric framework for identifying suspicious bluetooth trackers,” in 2025 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). IEEE Computer Society, 2025, pp. 570–572.

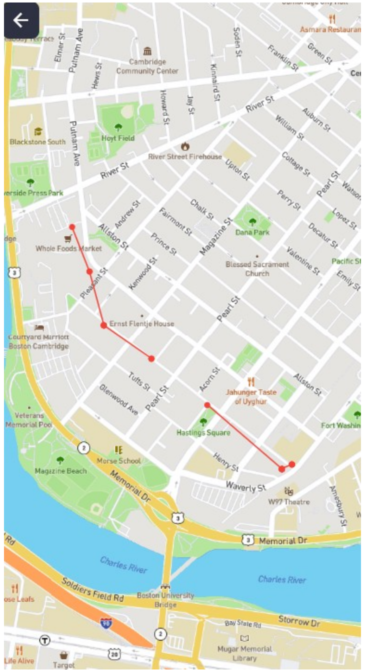
Information Log

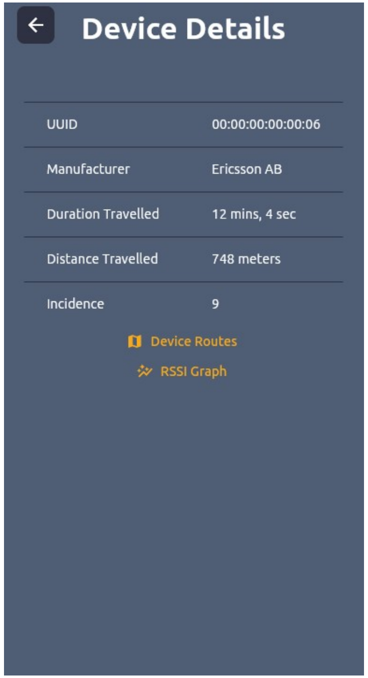
- location
- time
- RSSI



Information Log

- location
- time
- RSSI



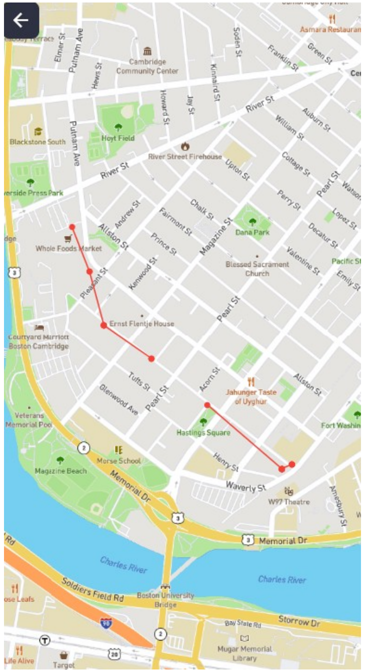


Device Details

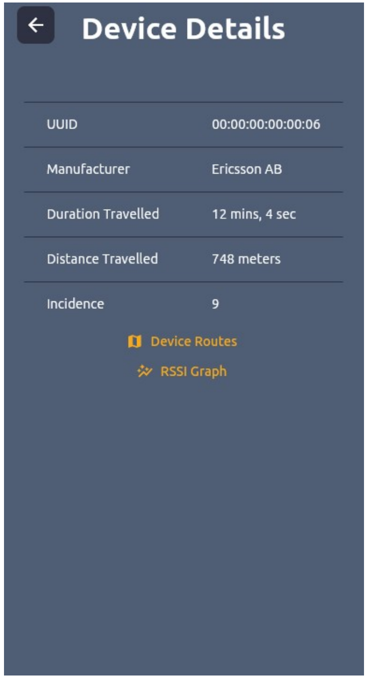
UUID	00:00:00:00:00:06
Manufacturer	Ericsson AB
Duration Travelled	12 mins, 4 sec
Distance Travelled	748 meters
Incidence	9

Information Log

- location



- time



Device Details

UUID	00:00:00:00:00:06
Manufacturer	Ericsson AB
Duration Travelled	12 mins, 4 sec
Distance Travelled	748 meters
Incidence	9

Device Routes

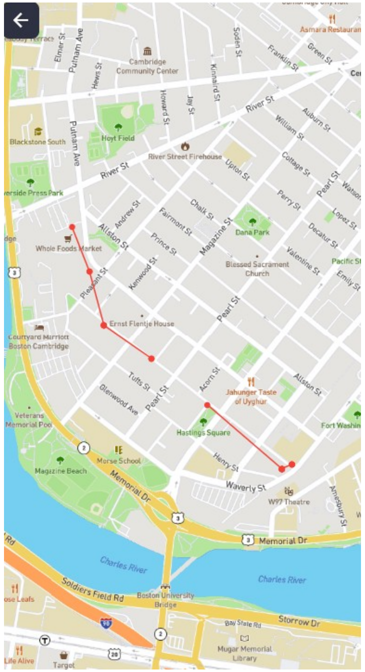
RSSI Graph

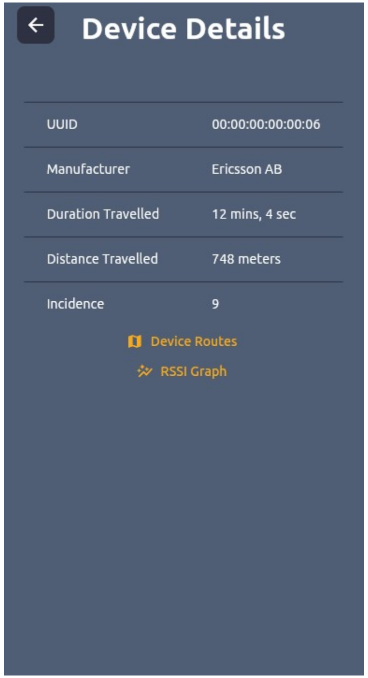
- RSSI

What is this?

Information Log

- location
- time
- RSSI





Device Details

UUID	00:00:00:00:00:06
Manufacturer	Ericsson AB
Duration Travelled	12 mins, 4 sec
Distance Travelled	748 meters
Incidence	9

Device Routes

RSSI Graph

Received Signal Strength Indicator

Question

How does the RSSI values differ for suspicious and non-suspicious devices?

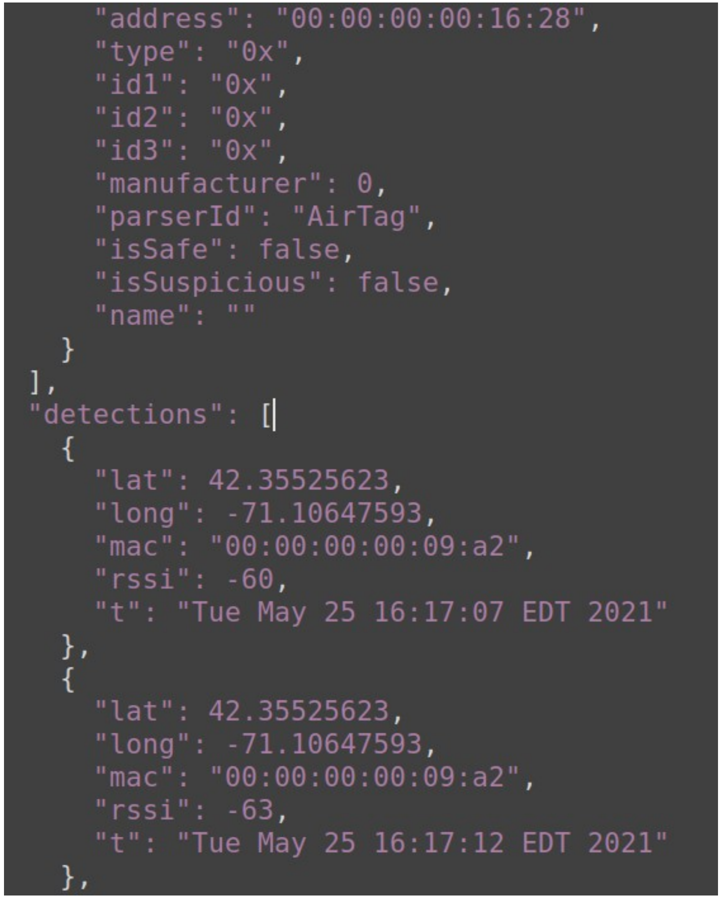
Reading the data

- The data is from BLE-Doubt
- JSON
- Various travel and planted methods
- Ground truth

	Movement	Location
A	Walking	Backpack
B	Walking	Backpack
C	Walking	Pockets
D	Walking	Pockets
E	Car	Car
F	Jogging	Backpack
G	Walking	Backpack
H	Walking	Backpack
I	Train	Backpack
J	Train	Backpack
K	Walking	Pockets
L	Walking	Pockets
M	Train	
N	Car	Car

J. Briggs and C. Geeng, “Ble-doubt: Smartphone-based detection of malicious bluetooth trackers,” in 2022 IEEE Security and Privacy Workshops (SPW). IEEE, 2022, pp. 208–214.

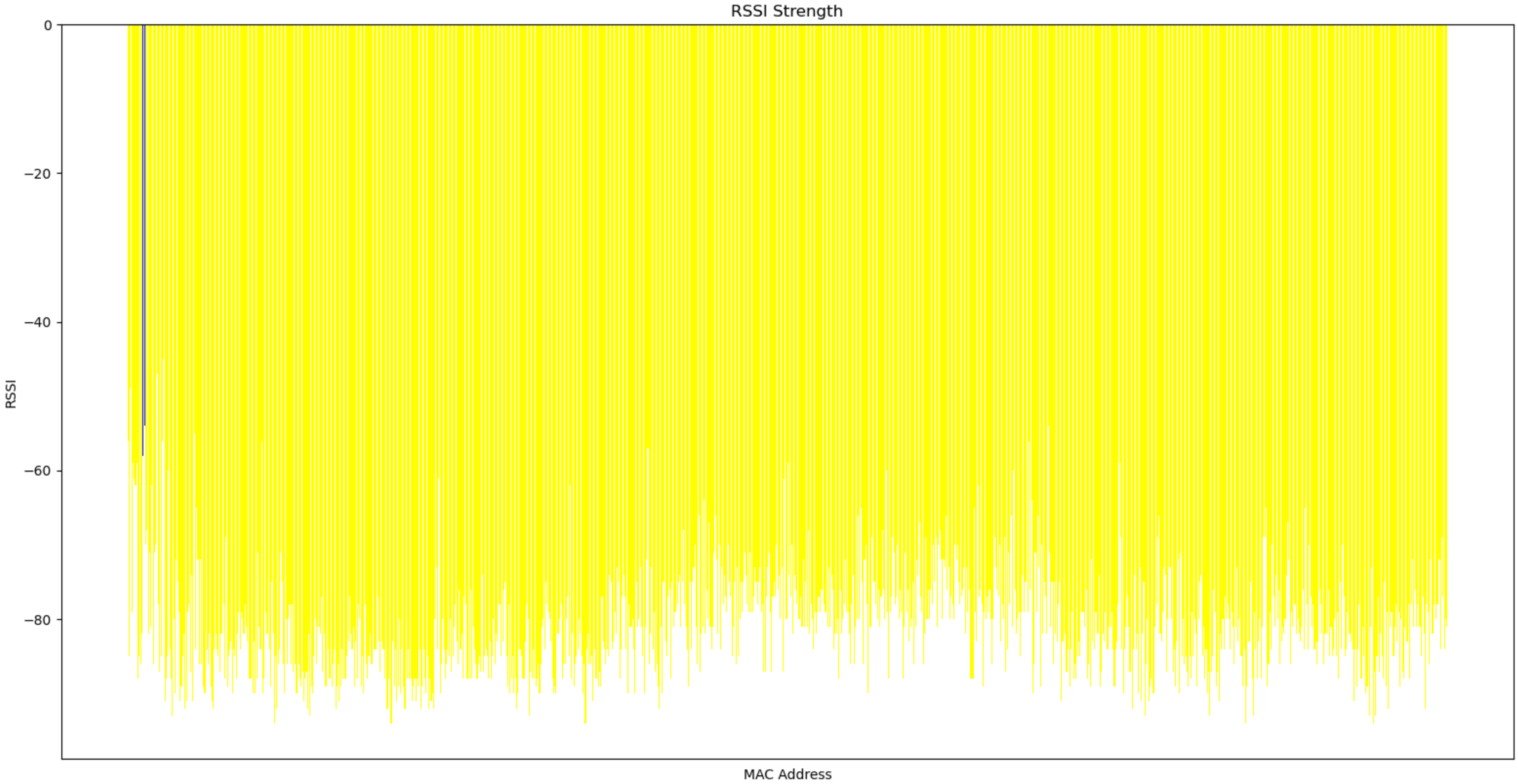
Sample of the dataset



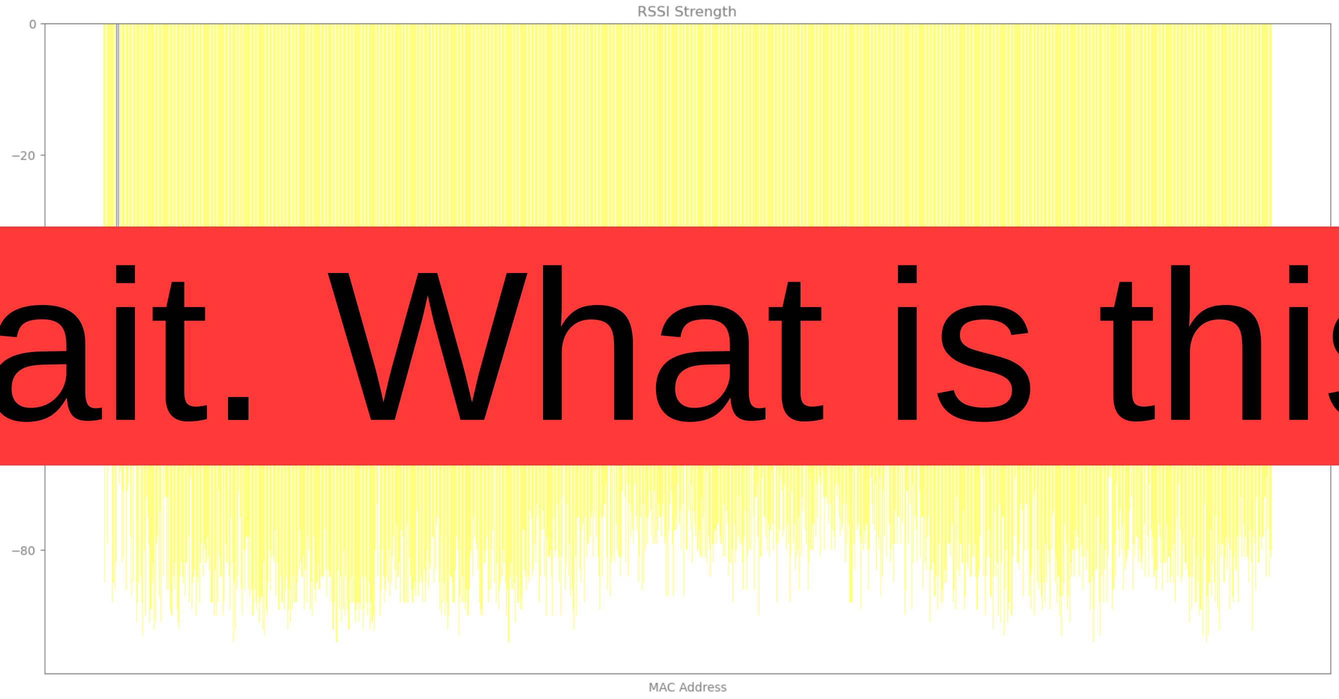
```
    "address": "00:00:00:00:16:28",
    "type": "0x",
    "id1": "0x",
    "id2": "0x",
    "id3": "0x",
    "manufacturer": 0,
    "parserId": "AirTag",
    "isSafe": false,
    "isSuspicious": false,
    "name": ""
  }
],
"detections": [
  {
    "lat": 42.35525623,
    "long": -71.10647593,
    "mac": "00:00:00:00:09:a2",
    "rssi": -60,
    "t": "Tue May 25 16:17:07 EDT 2021"
  },
  {
    "lat": 42.35525623,
    "long": -71.10647593,
    "mac": "00:00:00:00:09:a2",
    "rssi": -63,
    "t": "Tue May 25 16:17:12 EDT 2021"
  },
```

- Dictionary
- Detections key
- Same MAC?

Graph

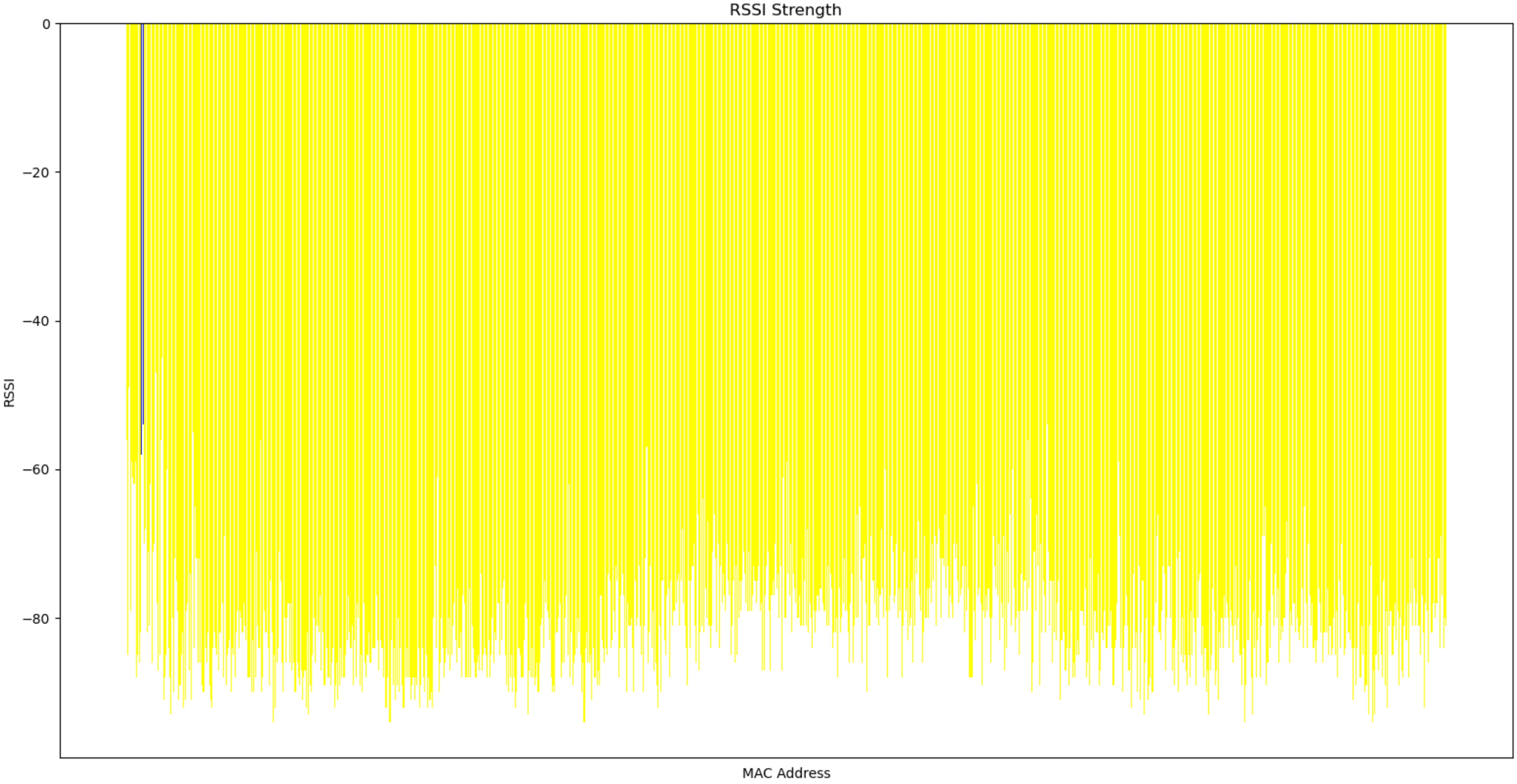


Graph

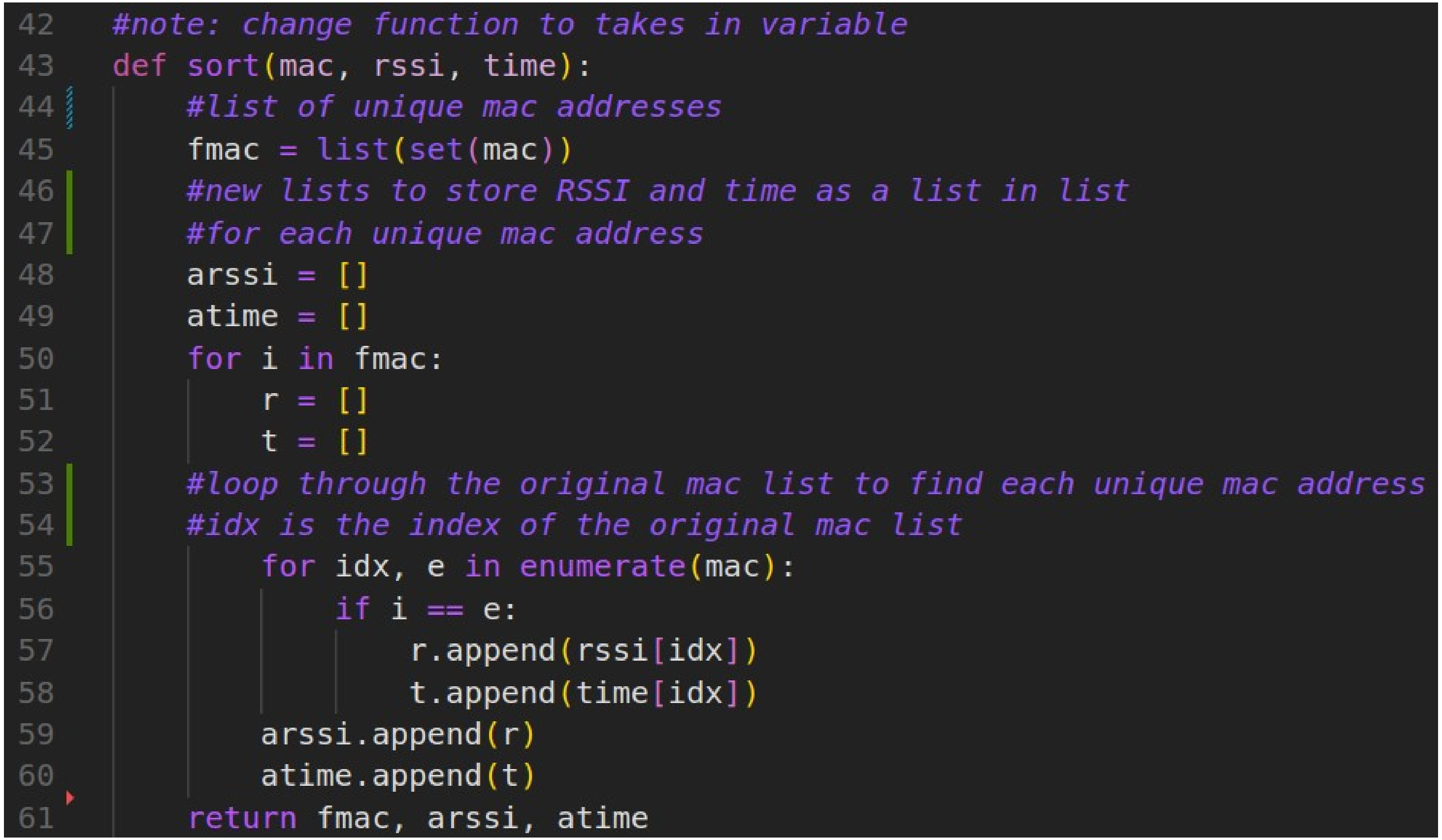


Wait. What is this?

Graph

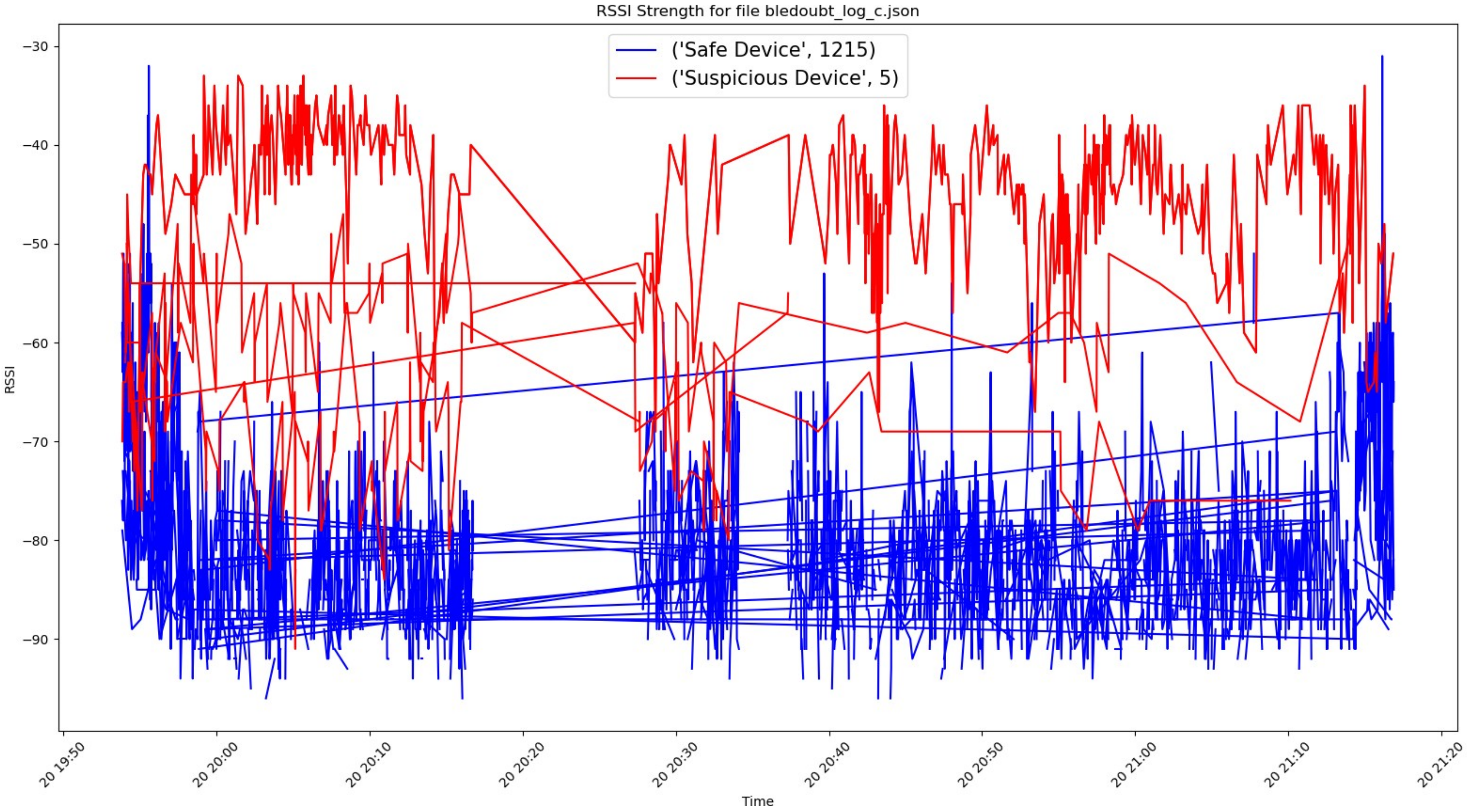


Identifying each mac id

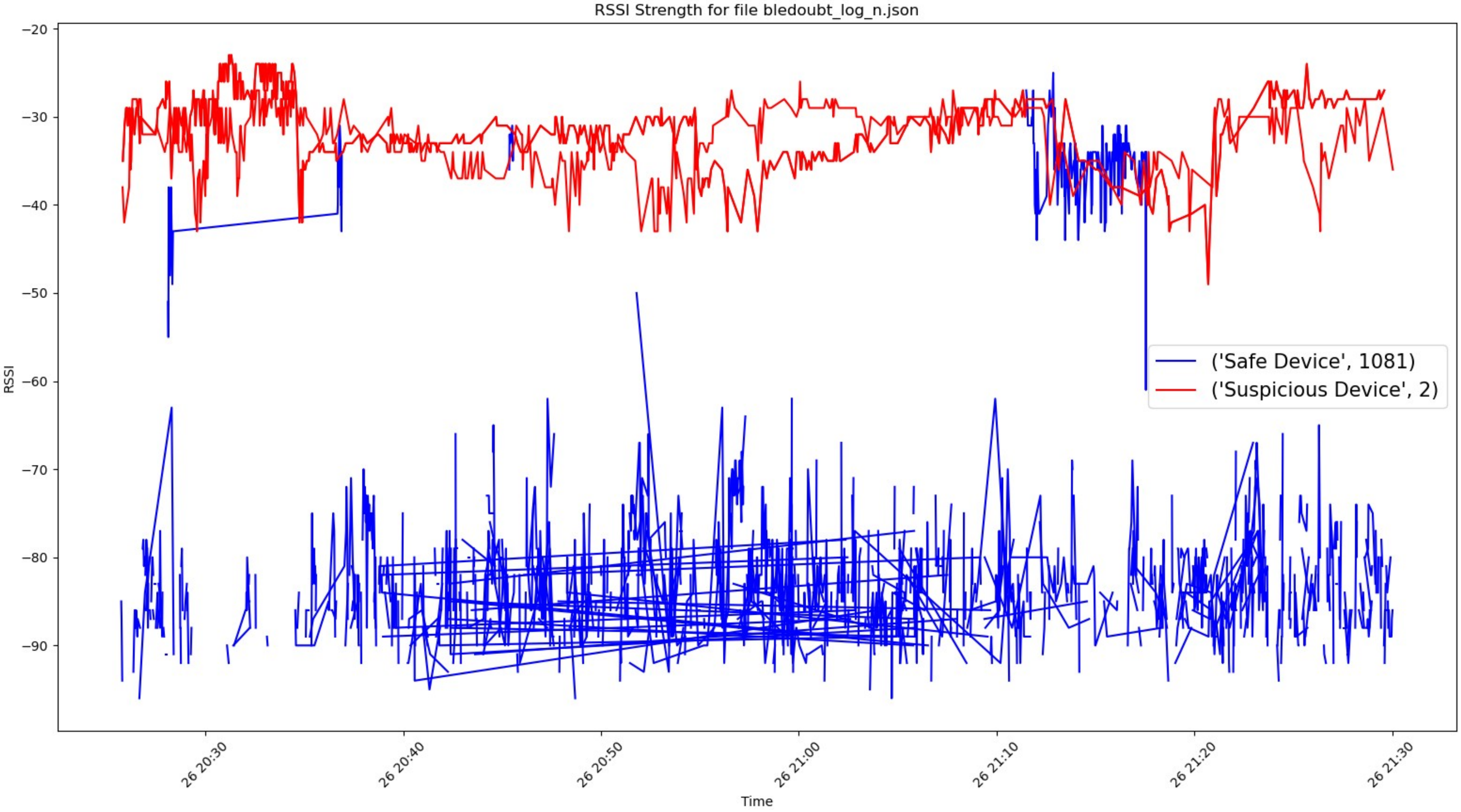


```python
#note: change function to takes in variable
def sort(mac, rssi, time):
    #list of unique mac addresses
    fmac = list(set(mac))
    #new lists to store RSSI and time as a list in list
    #for each unique mac address
    arssi = []
    atime = []
    for i in fmac:
        r = []
        t = []
    #loop through the original mac list to find each unique mac address
    #idx is the index of the original mac list
        for idx, e in enumerate(mac):
            if i == e:
                r.append(rssi[idx])
                t.append(time[idx])
        arssi.append(r)
        atime.append(t)
    return fmac, arssi, atime
```

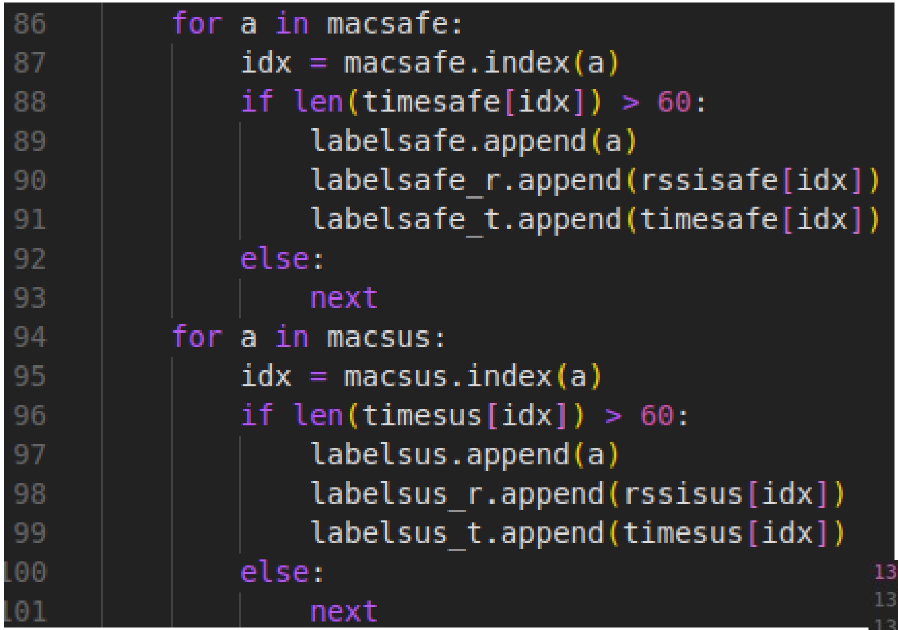
RSSI value over time – walking, pocket



RSSI value over time – Car



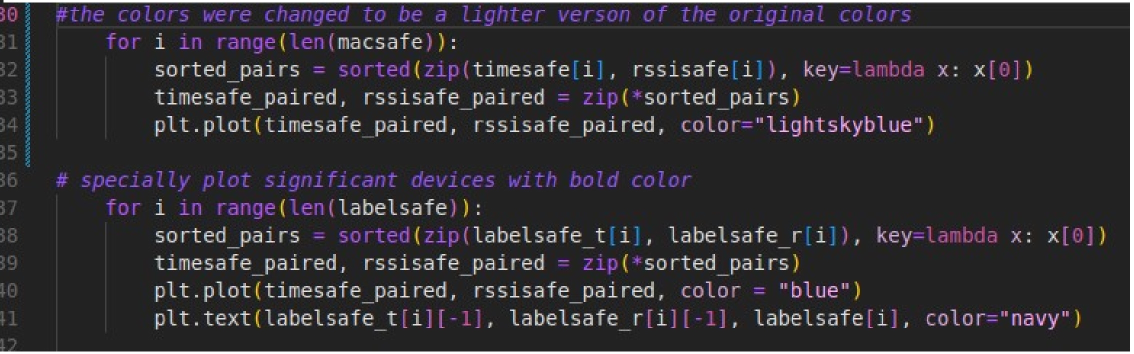
Significant time Detected



```python
    for a in macsafe:
        idx = macsafe.index(a)
        if len(timesafe[idx]) > 60:
            labelsafe.append(a)
            labelsafe_r.append(rssisafe[idx])
            labelsafe_t.append(timesafe[idx])
        else:
            next
    for a in macsus:
        idx = macsus.index(a)
        if len(timesus[idx]) > 60:
            labelsus.append(a)
            labelsus_r.append(rssisus[idx])
            labelsus_t.append(timesus[idx])
        else:
            next
```

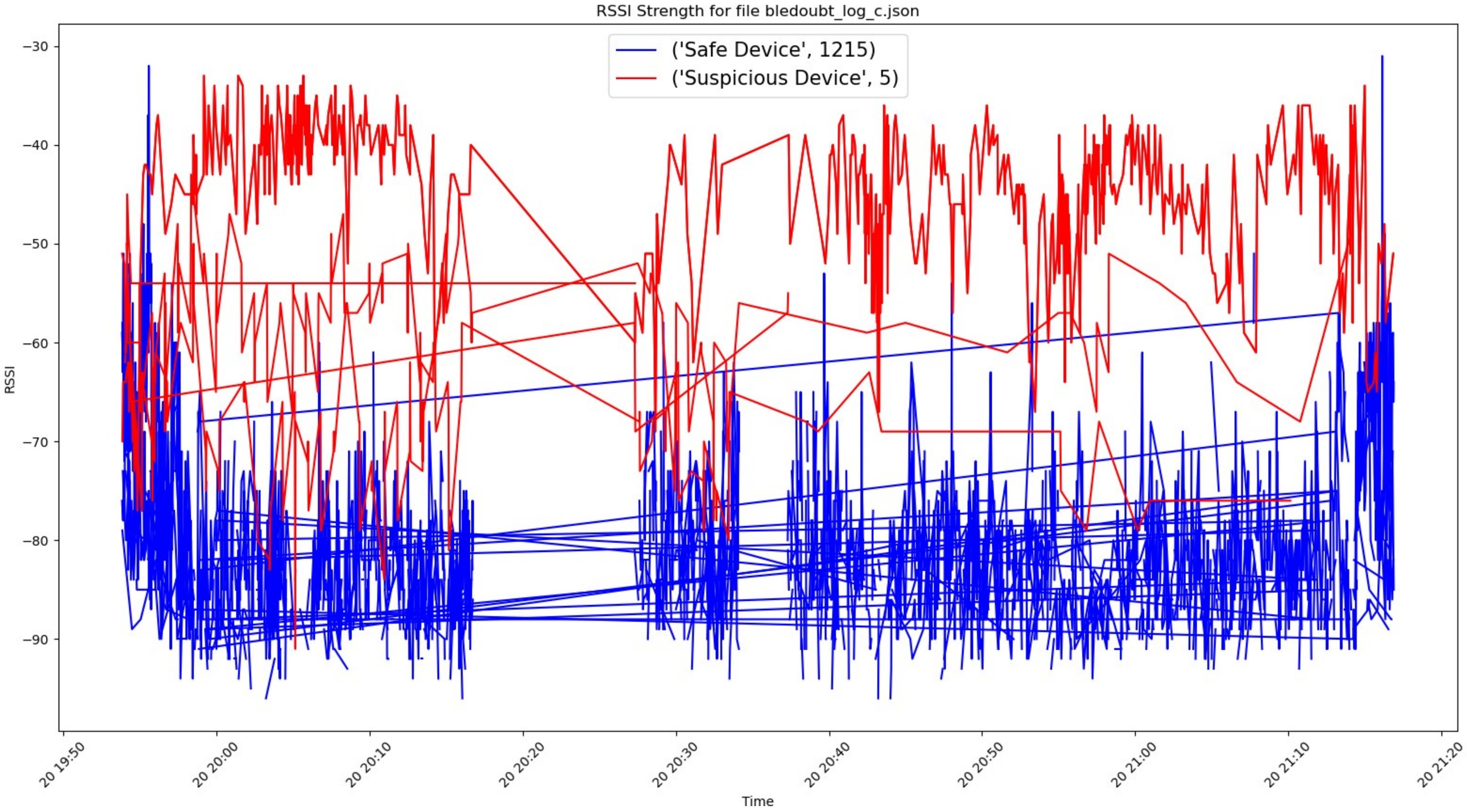
Setting threshold to 60 seconds

Plotting with different colors

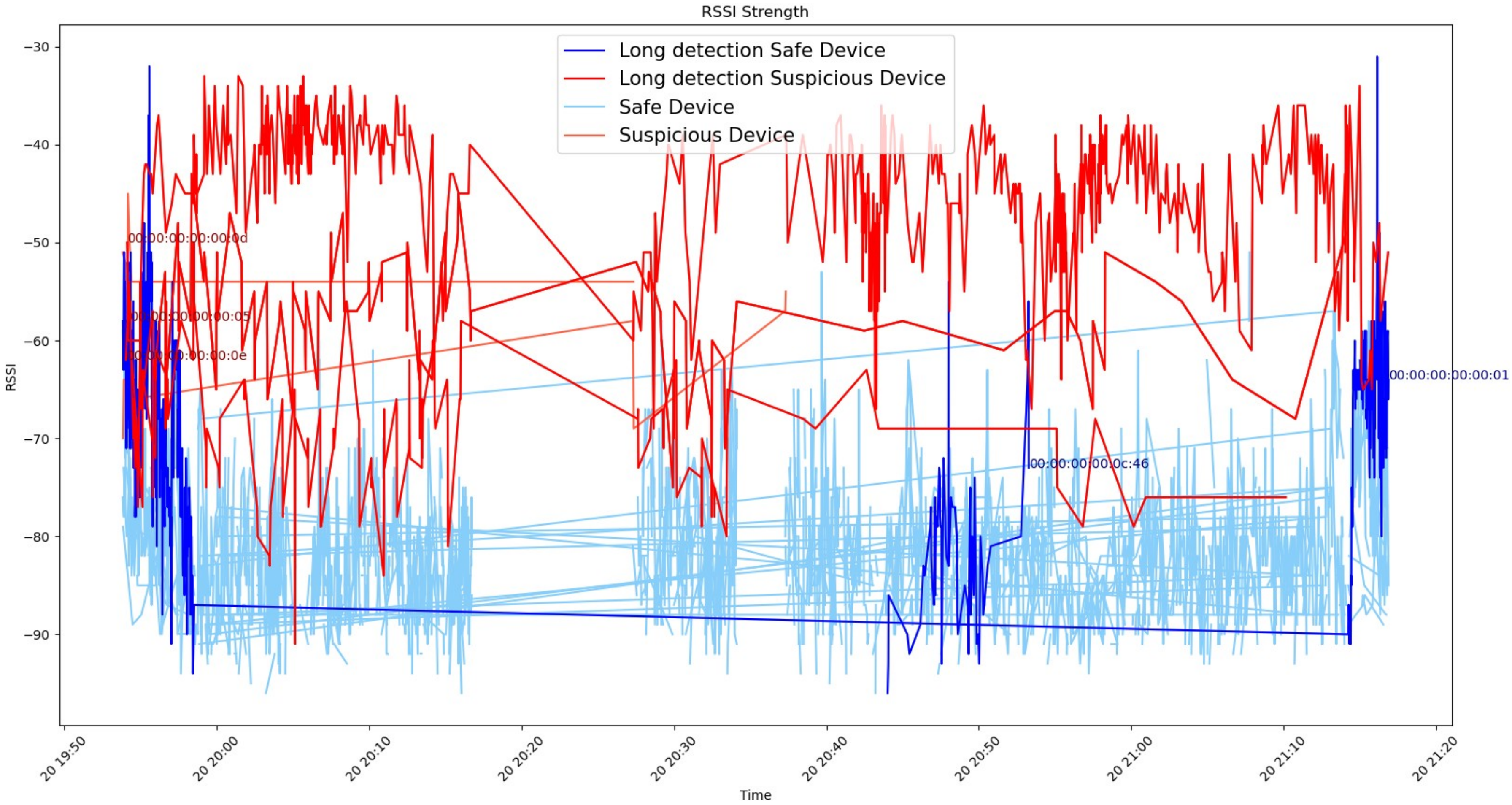


```python
#the colors were changed to be a lighter verson of the original colors
    for i in range(len(macsafe)):
        sorted_pairs = sorted(zip(timesafe[i], rssisafe[i]), key=lambda x: x[0])
        timesafe_paired, rssisafe_paired = zip(*sorted_pairs)
        plt.plot(timesafe_paired, rssisafe_paired, color="lightskyblue")

# specially plot significant devices with bold color
    for i in range(len(labelsafe)):
        sorted_pairs = sorted(zip(labelsafe_t[i], labelsafe_r[i]), key=lambda x: x[0])
        timesafe_paired, rssisafe_paired = zip(*sorted_pairs)
        plt.plot(timesafe_paired, rssisafe_paired, color = "blue")
        plt.text(labelsafe_t[i][-1], labelsafe_r[i][-1], labelsafe[i], color="navy")
```

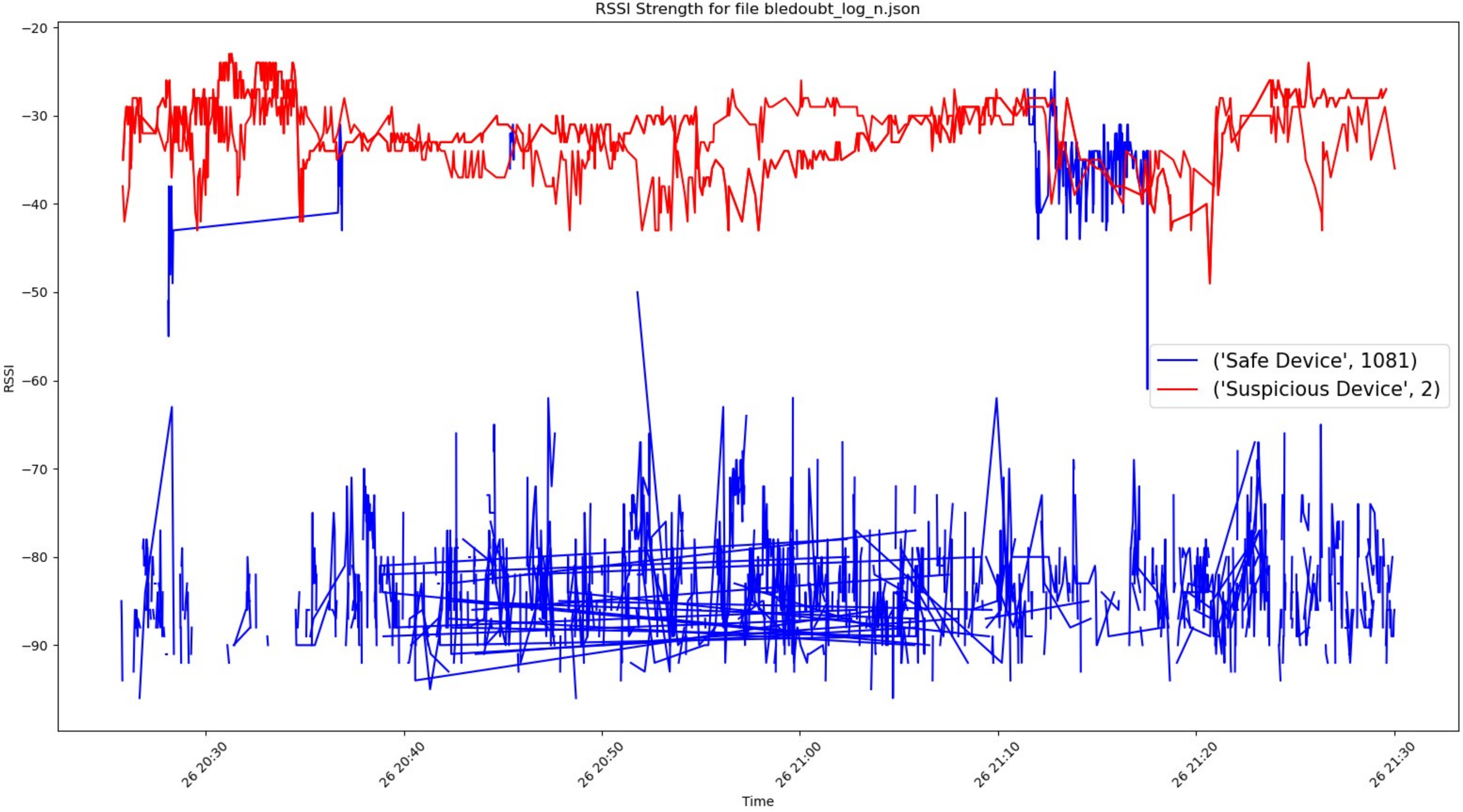
RSSI value over time – walking, pocket



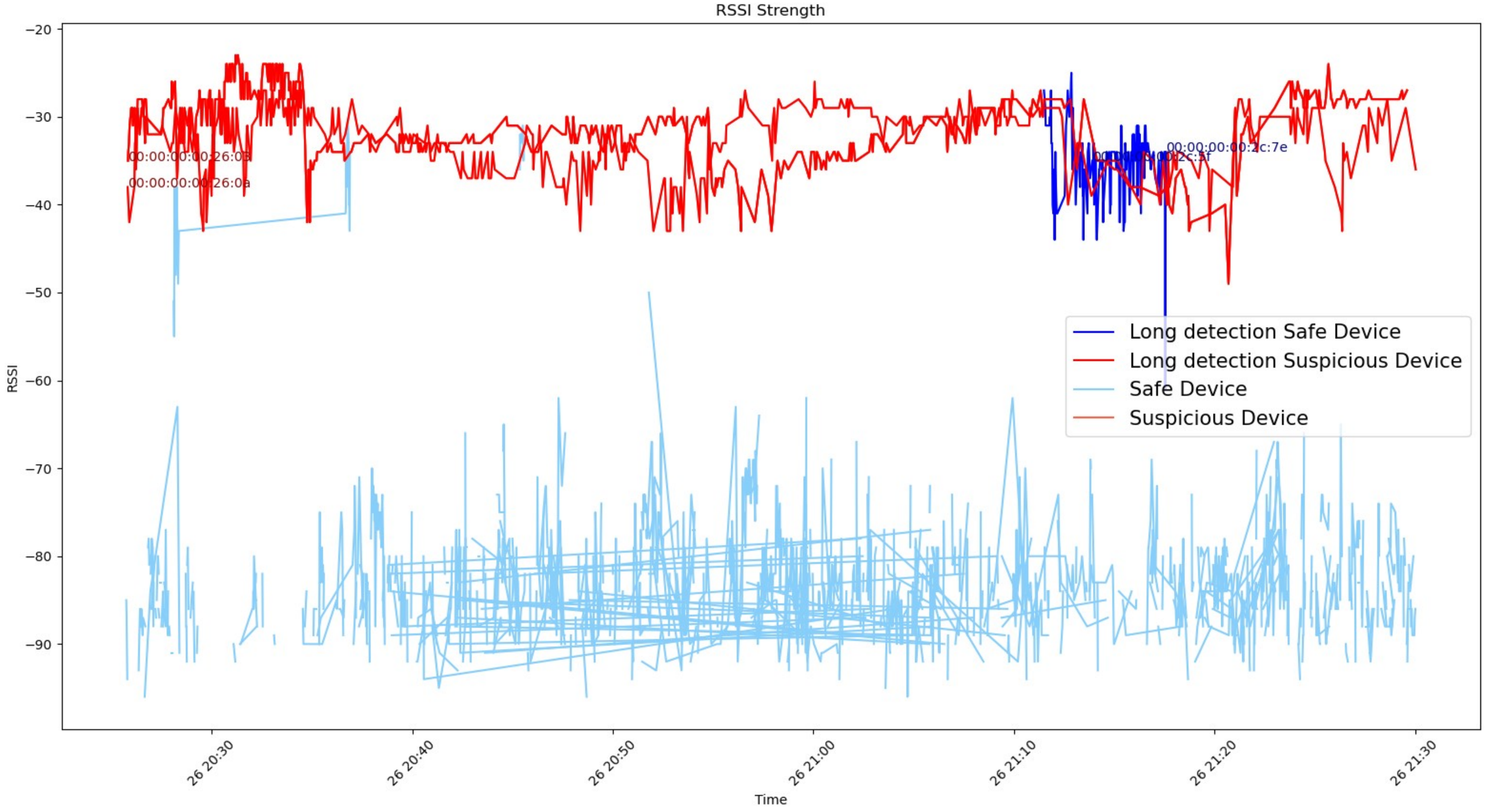
RSSI over time – walking, pocket- long detection



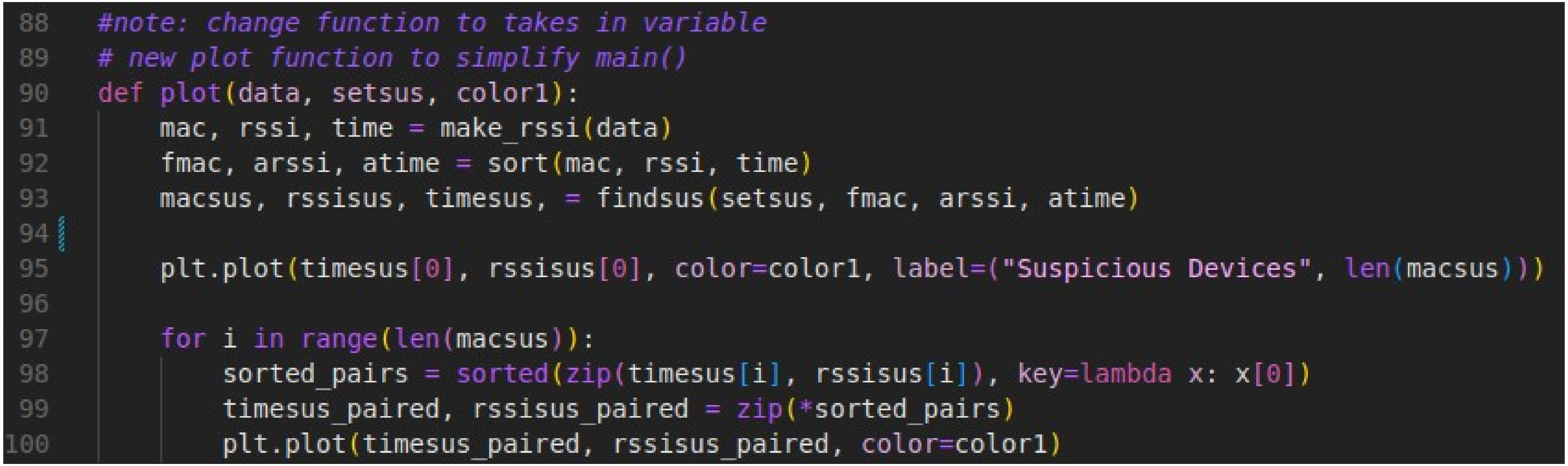
RSSI over time – Car



RSSI over time - c - long detection

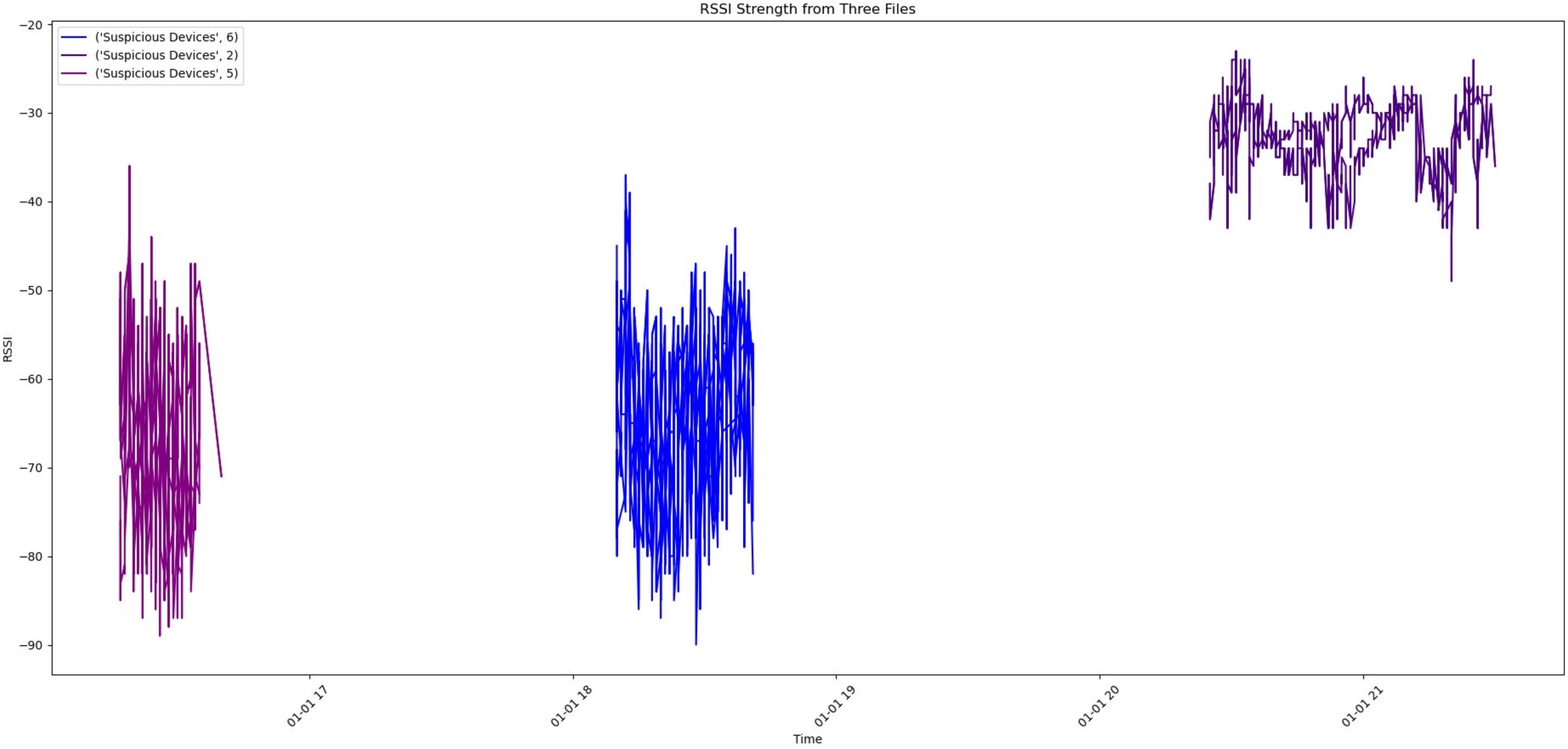


Time of Day

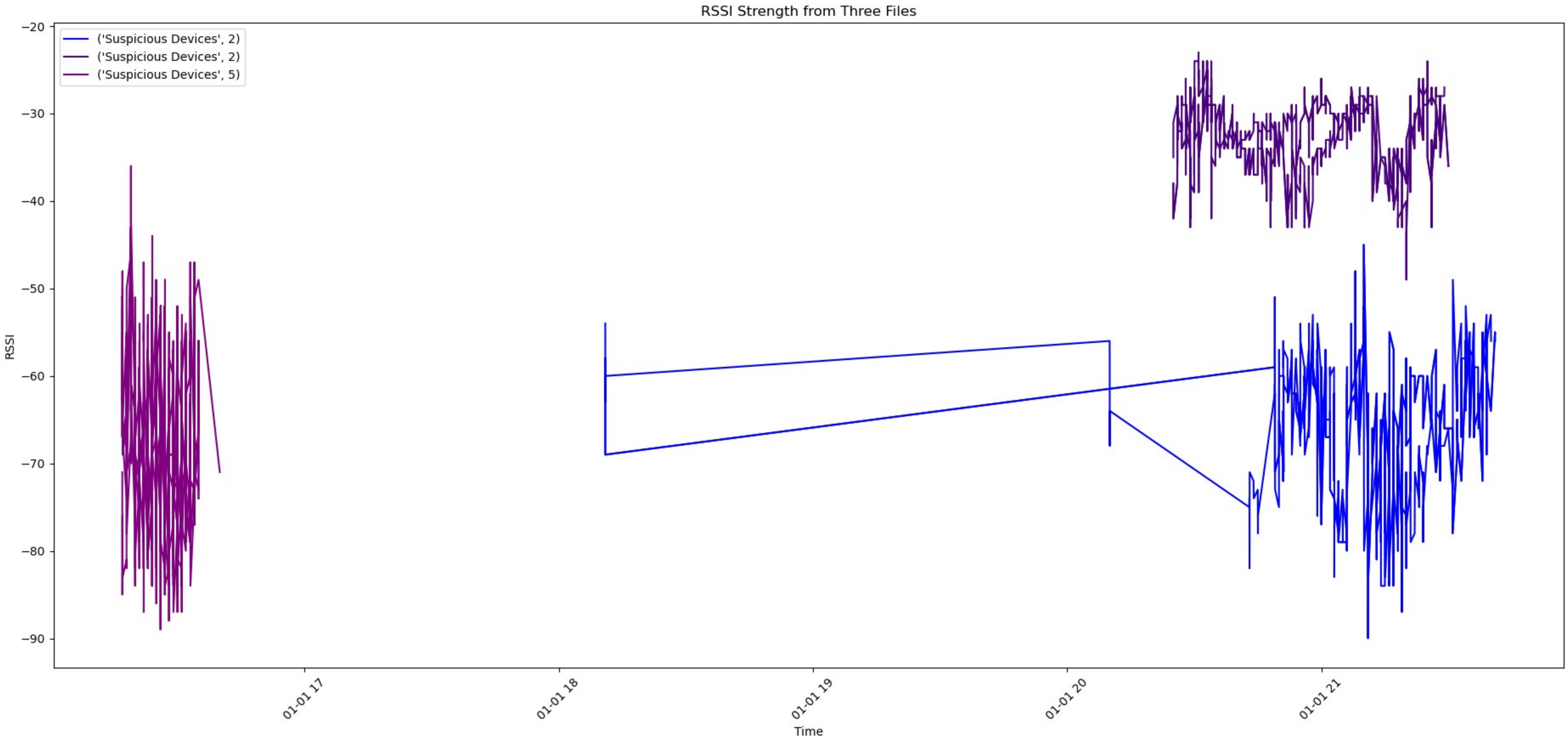


```python
#note: change function to takes in variable
# new plot function to simplify main()
def plot(data, setsus, color1):
    mac, rssi, time = make_rssi(data)
    fmac, arssi, atime = sort(mac, rssi, time)
    macsus, rssisus, timesus, = findsus(setsus, fmac, arssi, atime)

    plt.plot(timesus[0], rssisus[0], color=color1, label=("Suspicious Devices", len(macsus)))

    for i in range(len(macsus)):
        sorted_pairs = sorted(zip(timesus[i], rssisus[i]), key=lambda x: x[0])
        timesus_paired, rssisus_paired = zip(*sorted_pairs)
        plt.plot(timesus_paired, rssisus_paired, color=color1)
```

Time of day – G(walking, backpack),N(Car),J(train, Backpack)



Time of day – G$_{\text{(walking, backpack)}}$,N$_{\text{(Car)}}$,A$_{\text{(walking, backpack)}}$



Conclusion

- Suspicious devices have higher RSSI values
- Suspicious devices have long detection time
- Time of day has no statistically significant difference

Discussion

- Surrounding environment
- Time significance
- Time of day vs. location

Acknowledgments

- DIPr Lab
  - Primal Pappachan
  - Dylan Conklin
  - Orobosa Ekhator
- Institute for Computing in Research
  - Aaron Grinberg
  - Mark Galassi