ExploreCSR@OSU Autumn 2020

Hacking Bluetooth for Fun and Profit

1 Key Objectives

- Reinforce your concept about Bluetooth workflow.
- Get familiar with Bluetooth security and privacy .
- Learn how to use tools to analyze Bluetooth traffic.

2 Experiment Setup

2.1 Installing the Analysis Platform

1. Please install VMware (free for educational/personal uses):
   - If you are a MacOS user, please download and install VMware Fusion. You can download a free trial version (at `http://www.vmware.com/go/try-fusion-en`). For more details, please refer to `https://kb.vmware.com/s/article/2014097`.
   - If you are either a Windows or Linux user, please download and install VMWare workstation at `https://my.vmware.com/en/web/vmware/downloads/details?downloadGroup=PLAYER-1600&productId=1039&rPId=51984`.
2. Please download the System Image we have prepared at `https://drive.google.com/file/d/14YBBeqaPM7UCCf-DXLAdOKg0yNoERaRG/view?usp=sharing`.
3. Unzip and import the downloaded System Image into VMware (Check the link to see how to import the system image: `https://docs.vmware.com/en/VMware-Workstation-Player-for-Linux/14.0/com.vmware.player.linux.using.doc/GUID-DDCBE9C0-0EC9-4D09-8042-18436DA62F7A.html`).
4. Login onto the system using the following credentials[1]:

Username: bluetoothlab Password: 123456

[1] Root user's password is 123456 as well

2.2 Installing nRF Connect

- Please open your App Store (or Google Play for Android users) and search a mobile app named nRF Connect (See Figure 1).
- Install the app onto your iPhone or Android Phone. nRF Connect is a tool that enables Bluetooth Low energy traffic analysis.

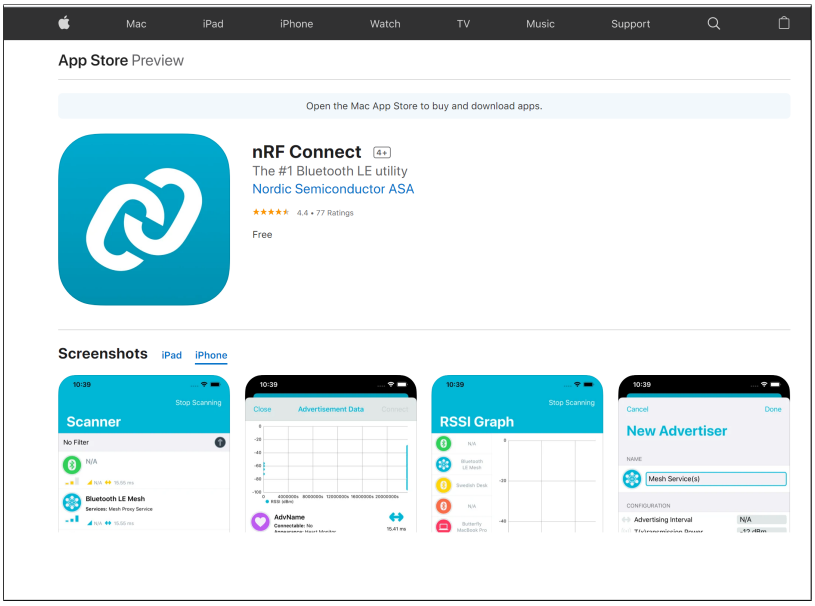

Figure 1: nRF Connect

3 Task 1: Using mobile sniffer to observe the nearby BLE devices

1. Open nRF Connect. nRF Connect will list the nearby BLE devices.
2. Check the manufacture data and MAC addresses of nearby BLE devices. The manufacture data will reveal the manufacturers of the BLE devices. Please note that iOS do not support to view the MAC addresses of other BLE devices.

   Warning: Do not attempt to connect other BLE devices, since it may be the violation of ethics.

4 Task 2: Analyzing the captured packets

1. Please log in your analysis platform using the provided username and password.
2. Run Wireshark.

```
$ wireshark
```

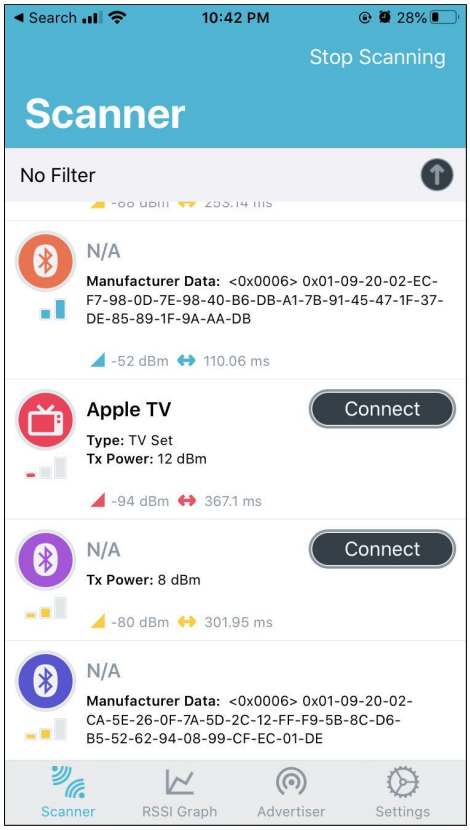

Figure 2: Nearby Bluetooth devices discovered by nRF Connect

3. Configure your Wireshark to view Bluetooth packets.
   - Edit ⟶ Preferences ⟶ Protocol ⟶ DLT_USER (See Figure 3)
   - Configure the encapsulation Table. Particularly, "DLT" should set to "147" and the payload protocol should set to "btle".

4. Download "blecap.pcap" file at `https://drive.google.com/file/d/18iQ2MfIQYmQs8WOcRztuHce1W6YQSubp/view?usp=sharing`, and use wireshark to open it.

5. Identify the types of the Bluetooth packets. In the example, there are 5 types of Bluetooth packets, including advertising packets (i.e., ADV_IND), scan request packets (i.e., SCAN_REQ), scan response (i.e., SCAN_RSP), connection request (i.e., CONNECT_REQ) packets, and data exchanging packets.

6. Check the format of advertising packets (i.e., ADV_IND).
   - Check the MAC address of the broadcasting device.
   - Check the manufacture data of the broadcasting device.
   - Check the UUID of the broadcasting devices.

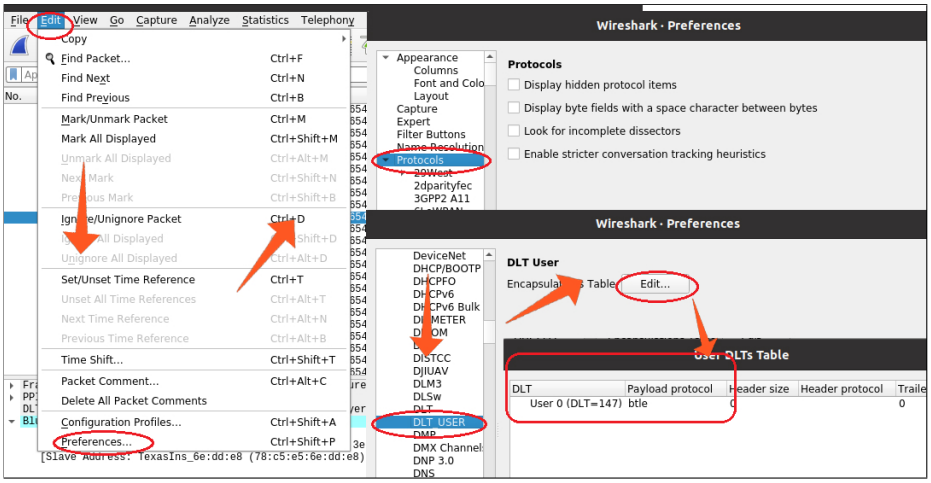

Figure 3: Configure Wireshark

7. Check the format of scan request packets (i.e., SCAN_REQ) and scan response (i.e., SCAN_RSP).
   - Check the MAC address of the initiating device in SCAN_REQ.
   - Check the MAC address of the device that the initiating device attempts to scan in SCAN_REQ.
   - Check whether the SCAN_REQ has a corresponding SCAN_RSP.
   - Check the response data in SCAN_RSP.
8. Check the security requirements of the broadcasting device. This can be done by checking the Pairing Request packets, which is a special type of data exchanging packet (See Figure 4).

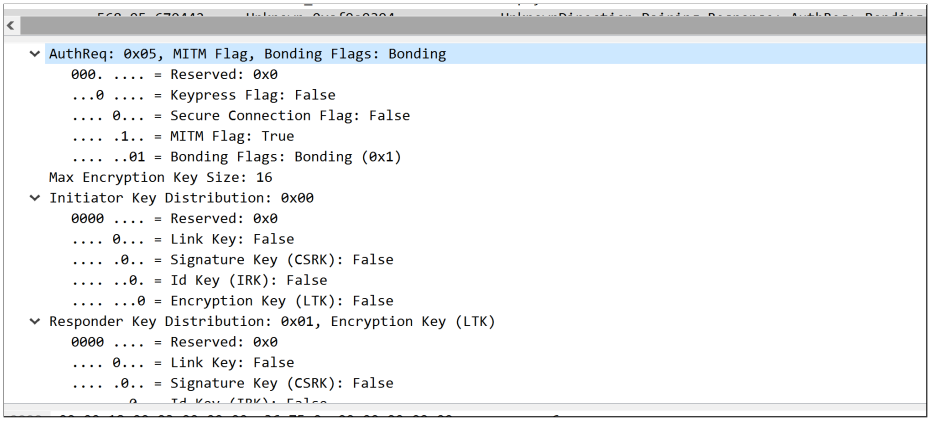

Figure 4: Checking security requirements of a device

5 Task 3: Using Crackle to decrypt encrypted Bluetooth packets

Crackle is a Bluetooth hacking tool which works against Bluetooth legacy (i.e., Bluetooth 4.0 and Bluetooth 4.1). It allows an attacker to brute force the Temporary Key (TK) and decrypt the encrypted Bluetooth packets.

1. Download and unzip the testing samples at `http://lacklustre.net/bluetooth/crackle-sample.tgz`.
2. The tool can be found in the folder `/home/mylab/crackle-0.1/`.
3. Identify the Just Works Pairing:

```
$ crackle -i ltk_exchange.pcap
```

4. View encrypted Bluetooth packets in file `encrypted_known_ltk.pcap` using Wireshark. It can be observed that once connected, the data exchanging packets are unreadable (See Figure 6).
5. Decrypt the Bluetooth packets using a given Long Term Key:

```
$ crackle -i encrypted_known_ltk.pcap -o decrypt.pcap -l 7f62c053f104a5bbe68b1d896a2ed49c
```

We now explain how the LTK can be obtained.

Determining Temporary Key (TK): TK is a 128-bit key that is used to generate the Short Term Key (STK). In Bluetooth Legacy, there are types of pairing methods are supported to share the TK, known as Just works, Passkey Entry and OOB. In Just works, TK is always set to 0. In Passkey Entry, the validated portion of TK is a random number between 000000 and 999999, while other bits can be padded with 0. Only in OOB, the 128 bit TK is randomly generated.

Determining Short-Term Key (STK): During the pairing process, few values exchanged in plain text. These values are confirm value, devices information, pair request and response, Mrand (random value from the master device) and Srand (random value from the salve device) (See Equation 1). Figure 5 displays the a random value sent from one device. After checking the correctness of the confirm values, devices exchange the generated random numbers (Srand and Mrand). Then the Short-Term Key (STK) is generated using the TK and random numbers (as shown in equation (2)).

$$\begin{aligned} Mconfirm &= c1(TK, Mrand, Pairinginfo, deviceinfo) \\ Sconfirm &= c1(TK, Srand, Pairinginfo, deviceinfo) \end{aligned} \tag{1}$$

$$STK = AES128(TK, Srand||Mrand) \tag{2}$$

Determining Long-Term Key (LTK): After that, a random chosen long term key will be generated and delivered from the salve to the master. The traffic will be encrypted by STK.

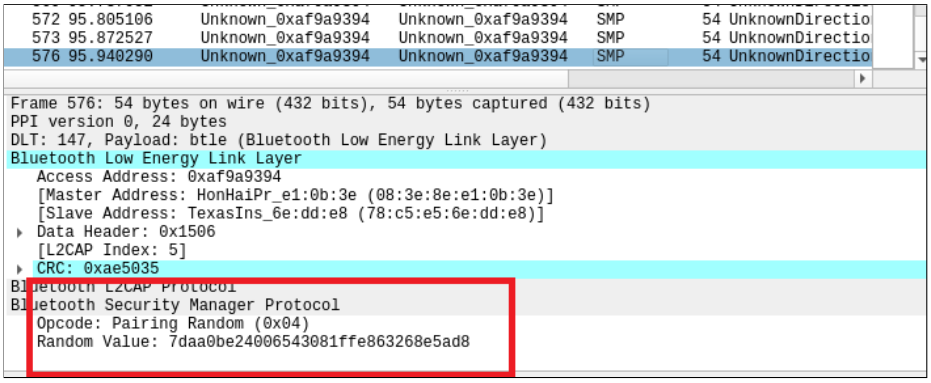

Figure 5: Random value exchanged in BLE packets

By knowing the *Confirm value generation function* c1 that is defined in 4.0 (as shown in equation (1)), attackers could use the captured values to brute force the TK (Only 1000000 possibilities). After obtain the TK, the attacker can then derive STK and obtain LTK.

6. View the decrypted Bluetooth packets in file `encrypted_known_ltk.pcap` using Wireshark. We can see the GATT read request now.

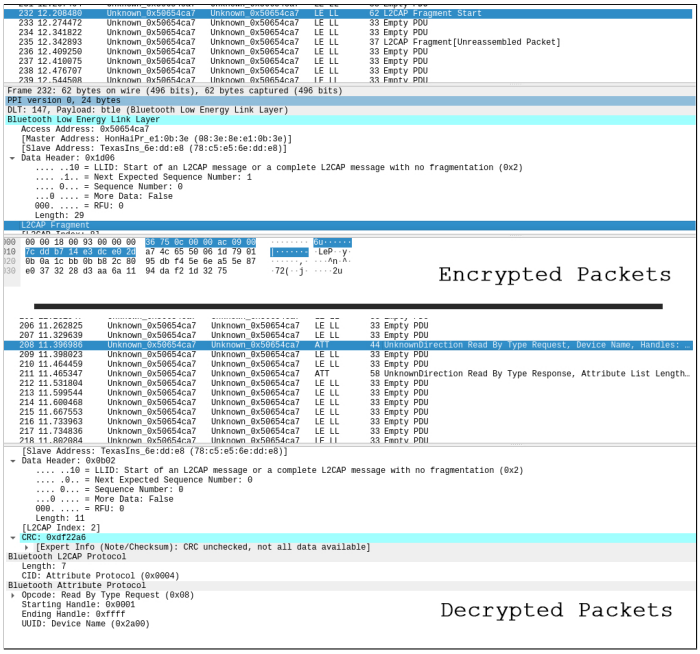

Figure 6: Encrypted packets v.s. decrypted packets