Chapter 5
BT labs

Marco Mellia

Lab requirements - Linux

- A Linux with BT interface

- Be sure to have the `bluez` stack installed and the `Bluetooth` service enabled
  - `sudo systemctl start bluetooth.service` [use `enable` to make the change permanent]
  - This will start some deamons that implement the upper part of the BT software stack

```
$ ps ax | grep blue
   1563 ?        Sl     0:01 /usr/bin/python3 /usr/bin/blueman-applet
   1769 ?        Ss     0:00 /usr/libexec/bluetooth/obexd
   2530 ?        Ss     0:00 /usr/libexec/bluetooth/bluetoothd
   2533 ?        Sl     0:03 /usr/bin/python3 /usr/bin/blueman-tray
  10109 pts/4    S+     0:00 grep --color=auto blue
```

- Use Wireshark to analyse packets and filter packets - https://www.wireshark.org
  - You can use the the `bluetooth0` interface to capture traffic

Using the BT linux stack

- bluetoothctl: - interactive Bluetooth control tool
  - This is an interactive command line interface to interact with the BT stack
  - There is a online help, accessed via the `help` command

- Useful commands to manage your device status:

`list`	List available controllers
`show [ctrl]`	Controller information
`devices [Paired/Bonded/Trusted/Connected]`	List available devices, with an optional property as the filter
`power <on/off>`	Set controller power
`pairable <on/off>`	Set controller pairable mode
`discoverable <on/off>`	Set controller discoverable mode
`advertise <on/off/type>`	Enable/disable advertising with the given type

Using the BT linux stack

- bluetoothctl: - interactive Bluetooth control tool
  - This is an interactive command line interface to interact with the BT stack
- Useful commands:

`scan <on/off/bredr/le>`	Scan for devices
`info [dev/set]`	Device/Set information
`pair [dev]`	Pair with device
`cancel-pairing [dev]`	Cancel pairing with device
`block [dev]`	Block device
`unblock [dev]`	Unblock device
`remove <dev>`	Remove device
`connect <dev>`	Connect device
`disconnect [dev]`	Disconnect device

Configuring the BT interface

- Via bluetoothctl, enter the mgmt menu

```
menu mgmt
```

- Use the info command to show the status of your device

```
[bluetooth]# info
[bluetooth]# Index list with 1 item
[bluetooth]# hci0:      Primary controller
[bluetooth]#    addr 04:7F:0E:66:3F:8B version 10 manufacturer 2279 class 0x6c0000
[bluetooth]#    supported settings: powered connectable fast-connectable discoverable
bondable link-security ssp br/edr le advertising secure-conn debug-keys privacy static-addr
phy-configuration
[bluetooth]#    current settings: powered bondable br/edr le secure-conn
[bluetooth]#    name kali
[bluetooth]#    short name
```

Configuration parameters

- Useful configuration commands in the `menu -> mgmt` section

`ssp <on/off>`	`Toggle SSP mode`
`sc <on/off/only>`	`Toggle SC support`
`hs <on/off>`	`Toggle HS support`
`le <on/off>`	`Toggle LE support`
`advertising <on/off>`	`Toggle LE advertising`
`bredr <on/off>`	`Toggle BR/EDR support`
`privacy <on/off> [irk]`	`Toggle privacy support`

- Note: the support for these features may change based on the hardware and driver version

Possible actions

- **Step 1: Get into Bluetoothctl's Interactive Mode**
  `~$ bluetoothctl`
  - Notice, how the prompt is changing from `~$` to `[bluetooth]#`

- Now, you can execute any command like help after the # symbol
  - Use `help` to find all existing commands and their one-liner explanations

- **Step 2: Turn on Bluetooth in Linux**
  `[bluetooth]# power on`
  - You get an output like:
    ```
    [CHG] Controller F8:89:D2:C8:2E:54 Class: 0x006c010c
    Changing power on succeeded
    ```
  - You can use the `power off` command to turn it off

Possible actions

- **Step 3: Scan for Available Bluetooth Devices**
  `[bluetooth]# scan on`
  - As you turn on your Bluetooth device, `bluetoothctl` will list it in the output
  - After you have found your device, copy its address for future use
- If you want your Linux computer to be discoverable by other devices, set it as discoverable
  `[bluetooth]# discoverable on`
  - This is not necessary
- **Step 4: List devices**
  `[bluetooth]# devices`
  - This will list all discovered devices
  - You can list `Paired`, `Bonded`, and `Trusted` devices, too

Possible actions

- **Step 5: Connect to a Bluetooth Device**
  - Select one of the devices by finding its MAC address
    - Eventually make it discoverable (e.g. your mouse)
  - Start the pairing on your Linux computer using

```
[bluetooth]# pair 90:78:B2:C7:8F:A8
```

  - Depending on the type of device, you might have to accept the connection with the PIN eventually

- You can then connect with the same device

```
[bluetooth]# connect 90:78:B2:C7:8F:A8
```

- Step 6: List device capabilities

```
bluetooth]# info 90:78:B2:C7:8F:A8
```

Wireshark

- Capture packets with Wireshark
  - Note: Wireshark captures the messages on the HCI interface – not on the physical interface
  - There is no equivalent of “monitor mode” with standard BT interfaces

- Use filters to select the packets
  - By address: `bluetooth.addr == 20:f4:78:5c:96:d9`
  - By protocol: `_ws.col.protocol == "SMP”`
  - Removing advertising reports: `!(bthci_evt.code == 0x3e)`
  - … use the right-click -> apply as a filter -> options

LAB TODO

- Choose one device to explore the BT capabilities and protocol
- Use Wireshark to capture packets
- Use the client to perform
  - Scan operations
  - Pairing operations
  - Connecting
- Observe on Wireshark the
  - Pairing procedures
  - Capability listing
  - …
- Verify that messages follow the expected format
  - For privacy, the system may not let you/Wireshark see some messages
  - Can you see the Secure Connections messages? Why?

Lab requirements - MacOS

- A Macbook with BT interface

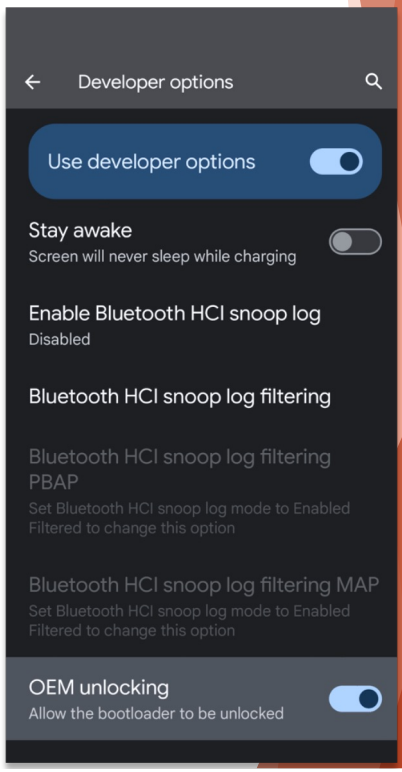
- Install **Xcode 11**
  - Visit the App Store or this link to install Xcode 11 on your Mac
- Install the **packetLogger** tool
  - Download the **Additional Tools for Xcode 11** via this link
    - When downloaded successfully, open the Additional_Tools_for_Xcode_11.dmg and access the Hardware folder. The packetLogger is inside the Hardware folder
  - Download the **logging** profile and install it
    - Activate it in the Preferences -> General -> Device Management

- It can also be used to capture BT packets from an iOS device remotely
  - See https://www.bluetooth.com/blog/a-new-way-to-debug-iosbluetooth-applications/ for details

- Use Wireshark to analyse packets and filter packets - https://www.wireshark.org

Lab requirement - Android

- On Android devices, it is possible to capture Bluetooth traffic as follows:
  - Go to Settings
  - Enable Developer mode (tap 7 times on the “Build number” in the “Software information” menu
  - Go into developer options
  - Enable the option Enable Bluetooth HCI snoop log
  - Enable the Bluetooth option and Connect to the device
- Transfer the file to the PC
  - Make sure the device is connected to the PC
  - The files might be shown in a PC's file browser in 'Internal Storage’
  - If not, use the Android Debug Bridge: adb pull /sdcard
- The files of interest are **btsnoop_hci.**log and all files with the extension **.cfa**
  - These are binary files, which can be opened with Wireshark
- Remember to disable the logging at the end!
  - Turn off the Bluetooth on the device
  - Disable the option Enable Bluetooth HCI snoop log



On windows [untested]

- You can get the Microsoft Bluetooth Test Platform software package
  - Download and install it from
  https://learn.microsoft.com/en-us/windows-hardware/drivers/bluetooth/testing-btp-setup-package
- Use the Bluetooth Virtual Sniffer (btvs.exe) to capture HCI traces
  https://learn.microsoft.com/en-us/windows-hardware/drivers/bluetooth/testing-btp-tools-btvs

Wireshark operation

- Usage for Wireshark on the same machine
  - Run btvs.exe using the command prompt\PowerShell console:

```
btvs.exe -Mode Wireshark
```

  - If Wireshark is installed, Wireshark opens automatically
- Otherwise, manually start Wireshark and provide the default TCP pipe as the interface:

```
wireshark -k -i TCP@127.0.0.1:24352
```

Trace collection

- **Goal:** collect many traces related to the discovery and pairing process with different devices

Process:

- Set up your PC to collect traces using Wireshark
- Choose a Bluetooth device
  - Avoid using common devices like your smartphone or tablet
  - E.g., a speaker, a fridge, a headset, a keyfob, a tracker, a toothbrush, a BT beacon, a remote controller, a lightbulb, …
- Identify the MAC address of the target device
  - You may need to make it discoverable. Use Wireshark to list BT devices

Trace collection

- Collect traces related to
  - Advertisement
  - Scan
  - Pairing
  - Possible data exchange
- Use Wireshark filters to show only packets related to each phase
  - You may include different traces (e.g., enabling Secure Simple Pairing or Secure Connection)
- Save a separate trace, naming it "`DeviceName_PhaseName.pcap`"
  - Save packets after the filter – include only the strictly related packets
- Add a `Setup.txt` (get it from the teaching portal) file with the description of the setup, including the
  - Device Name
  - Device Vendor
  - Operating System used to capture the trace
  - BT adapter model [e.g., use the `lsusb` command on Linux to get it]
  - Other useful information
- Create a zip file containing all traces and the Setup.txt file, name it `s1234567_DeviceName.zip`, and upload it to https://www.dropbox.com/request/SfqfFLG8LGaspDlQGr2f
- Deadline: 20/5/2025 -23:59 CET