

# COMP 8047 User Manual

[COMP 8037 – Major Project User Manual]

[Eric Wu – A00961904]

[2021/04/08]

# Table of contents

<b>Table of contents</b>	<b>1</b>
<b>Description</b>	<b>2</b>
<b>Prerequisite</b>	<b>2</b>
<b>How to run</b>	<b>2</b>
Launching the program	2
Adding Bluetooth identity	3
Spoof Bluetooth identity	3
Pair with Bluetooth victim device	5
DoS attack against Bluetooth victim device	7

# Description

This tool is a computer security project. The tool provides few attack vectors for exploiting the CTKD (Cross transport Key Deviation) vulnerability that resides in Bluetooth 5.0 and the prior versions. The vulnerability allows attackers to overwrite the authentication key or reduce the key strength used by the target blue device. The attacker can then connect to the device unauthorizedly, and steal the Bluetooth session between the Bluetooth client and host.

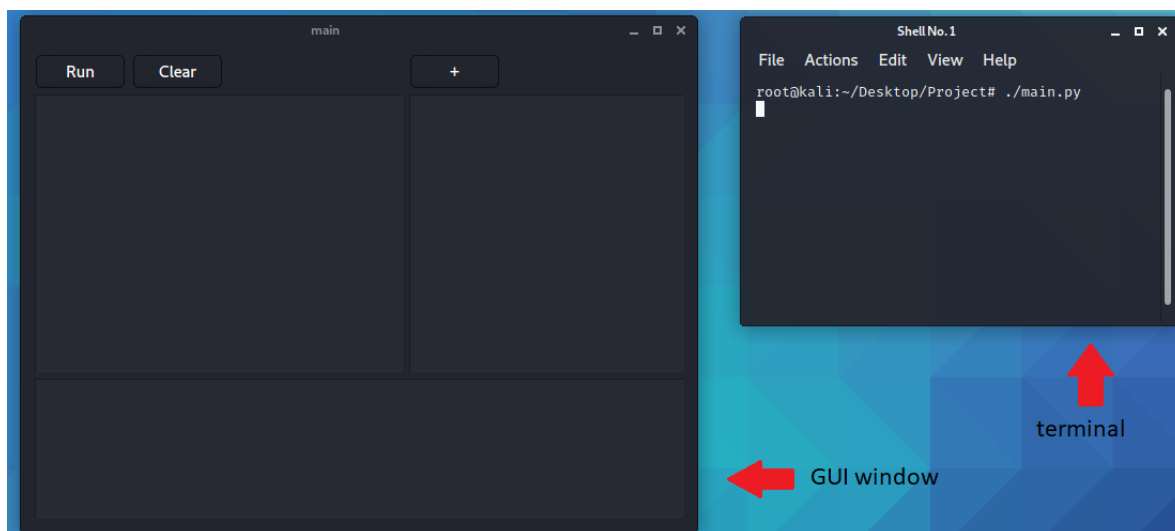
## Prerequisite

- Python 3.0
- Kali Linux machine (with Bluetooth support)
- btmgmt
- bluetoothctl

## How to run

### Launching the program

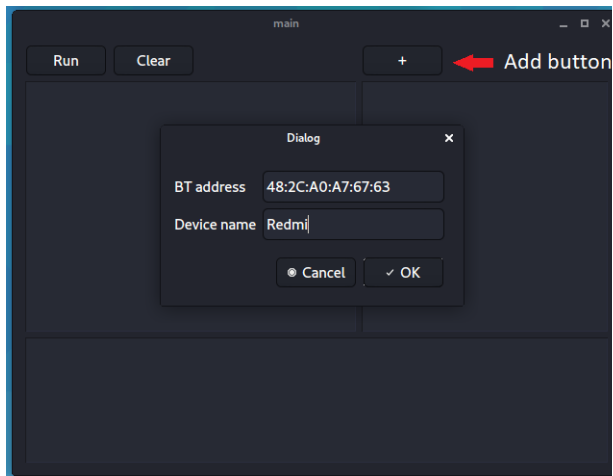
Before launching the program, we need to first cd into the project directory and make sure the execution bit on the “main.py” file is on. This can be done by running this command - “**chmod 744 main.py**”. Next, simply run the python file by running “**./main.py**”. This will launch the program in GUI. As shown in the figure below.



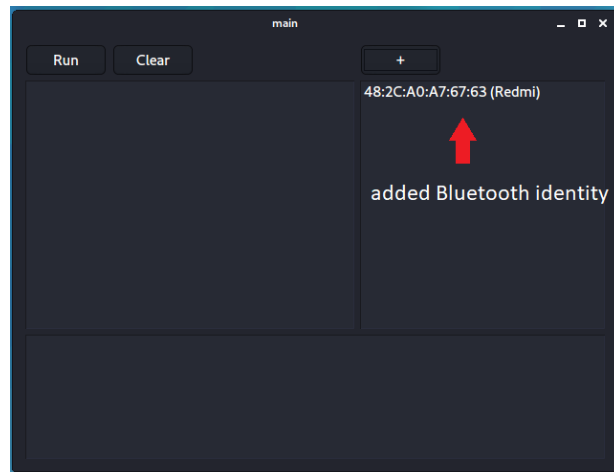
## Adding Bluetooth identity

Before we perform any kind of attack, we need to first add some Bluetooth addresses to the list. To do that, click on the “+” sign button, which will prompt you to enter a Bluetooth address and a device name. Fill out the two fields and press “OK”. The panel on the right should show the Bluetooth address you just added. The figure below illustrates this process.

Before add



After add



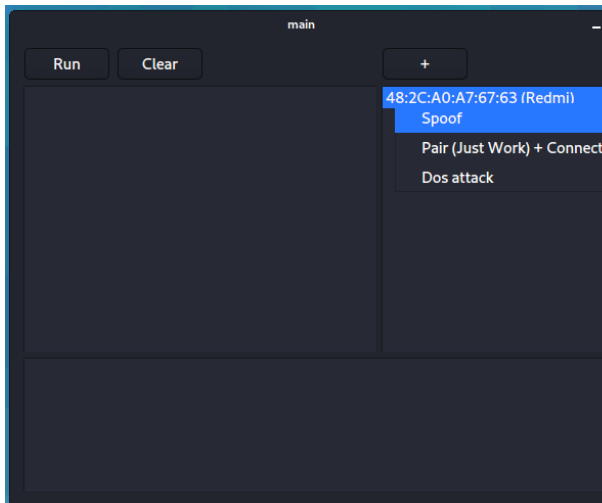
## Spoof Bluetooth identity

To spoof the Bluetooth identity we just added, we first need to know which HCI we are using. Open up a terminal and run “**hciconfig**” to see the list of the HCI available. The figure below shows 2 results - hci1 and hci0. (Assuming the one we will use later is “hci0”)

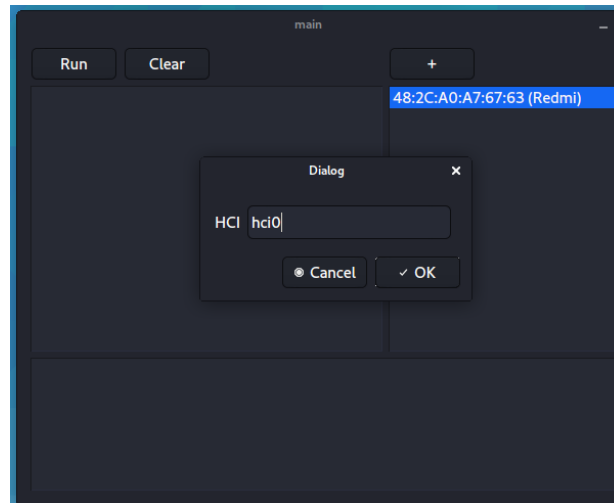
```
Shell No.1
File Actions Edit View Help
root@kali:~# hciconfig
hci1: Type: Primary Bus: USB
      BD Address: 6C:DD:BC:3D:4D:B4 ACL MTU: 310:10 SCO MTU: 64:8
      UP RUNNING PSCAN
      RX bytes:8527 acl:0 sco:0 events:96 errors:0
      TX bytes:4934 acl:0 sco:0 commands:62 errors:0
hci0: Type: Primary Bus: USB
      BD Address: 5C:C5:D4:BB:C6:7C ACL MTU: 1021:5 SCO MTU: 96:5
      UP RUNNING PSCAN
      RX bytes:2162 acl:0 sco:0 events:225 errors:0
      TX bytes:32469 acl:0 sco:0 commands:224 errors:0
```

Now that we have acquired the HIC name to use, we can begin the spoofing process. Bring up the tool window, and right-click on the added address. A list of attack vectors will show up. Click **“Spoof”**, enter the name of the HCI to use (In this case, we are using hci0), and click **“OK”**. The panel on the right should show the attack vector we just registered. Click **“Run”** to launch the spoofing attack. After a while, the lower panel should show messages that indicate the result of the spoofing operation. The figures below illustrate the process.

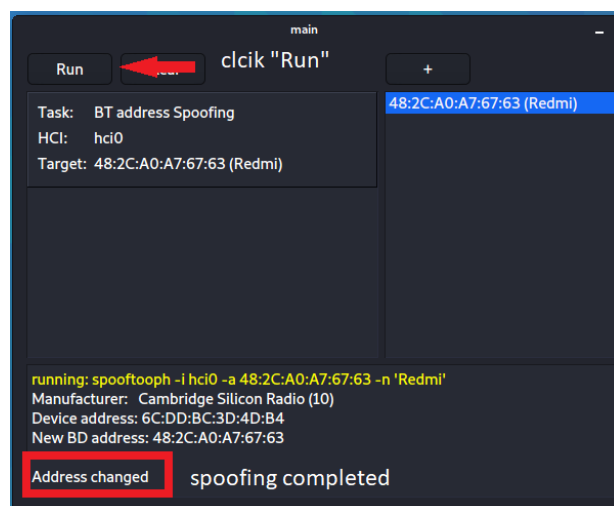
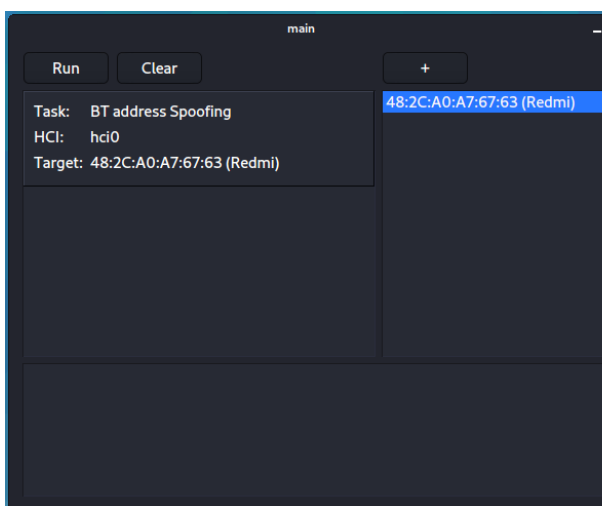
step 1.



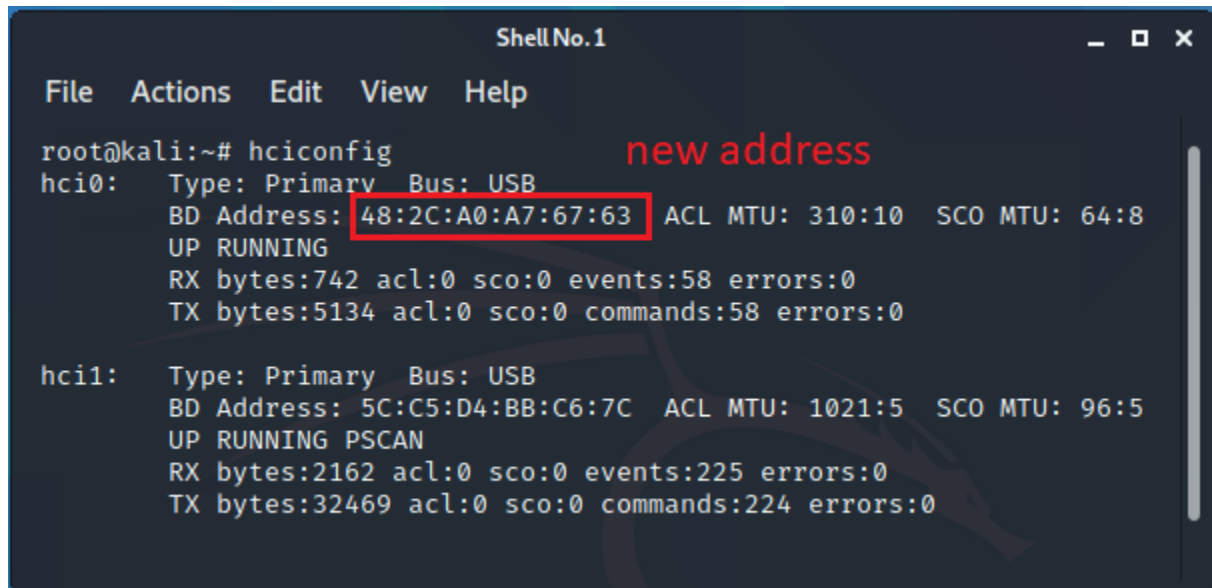
step 2.



step 3.



To verify that the spoofing operation is successful, we can check the address of the target HCI with the “**hciconfig**” command. The figure below shows that the HCI has a new Bluetooth address after we ran the program spoofing.

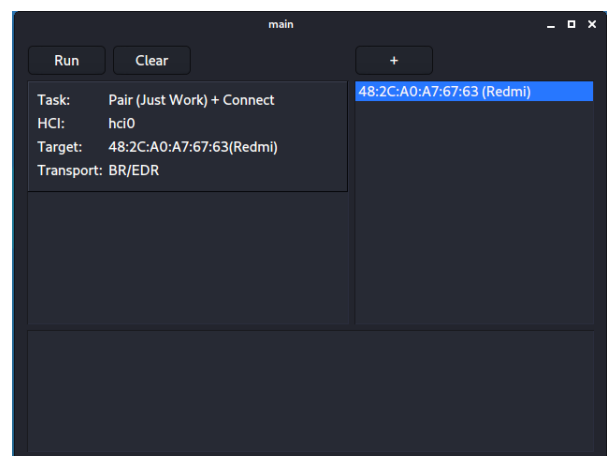
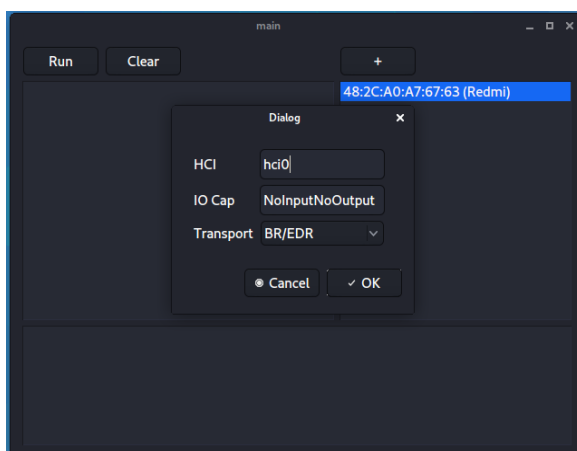


```
Shell No. 1
File Actions Edit View Help
root@kali:~# hciconfig
hci0:  Type: Primary  Bus: USB
      BD Address: 48:2C:A0:A7:67:63  ACL MTU: 310:10  SCO MTU: 64:8
      UP RUNNING
      RX bytes:742 acl:0 sco:0 events:58 errors:0
      TX bytes:5134 acl:0 sco:0 commands:58 errors:0

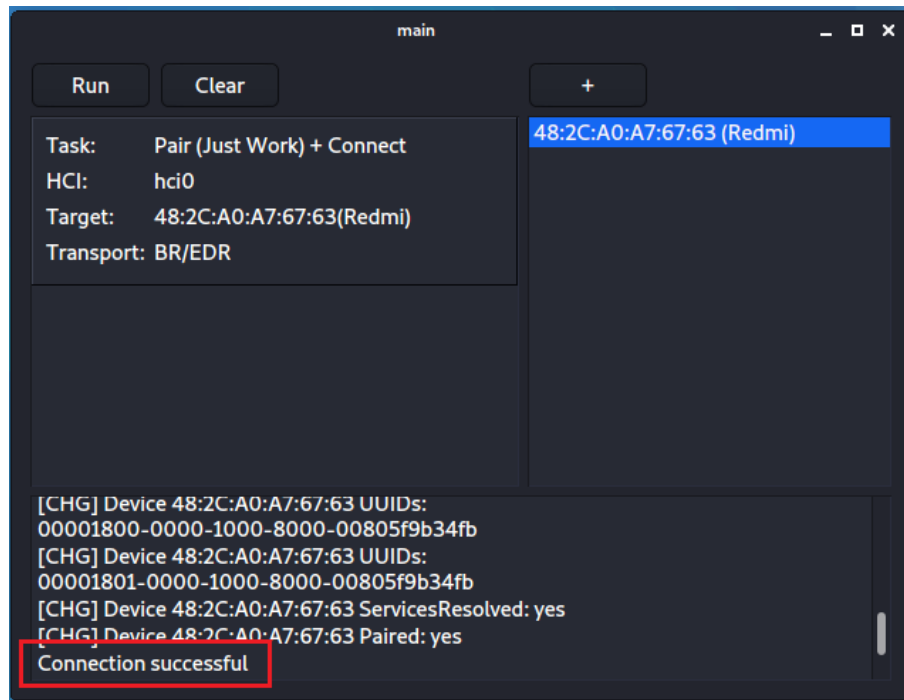
hci1:  Type: Primary  Bus: USB
      BD Address: 5C:C5:D4:BB:C6:7C  ACL MTU: 1021:5  SCO MTU: 96:5
      UP RUNNING PSCAN
      RX bytes:2162 acl:0 sco:0 events:225 errors:0
      TX bytes:32469 acl:0 sco:0 commands:224 errors:0
```

### Pair with Bluetooth victim device

In a typical classic Bluetooth (BR/EDR) connection, master and slave roles are not fixed. The attacker can take advantage of this role asymmetry to impersonate a slave device that is already trusted by a master device and send a pairing request to the master device over BR/EDR transport. Right-click on the added address, and click “**Par(Just Work) + Connect**”, enter the HCI (in this case, we are using hci0), keep the rest of the value unchanged, and click “**OK**”. The figure below illustrates the process.

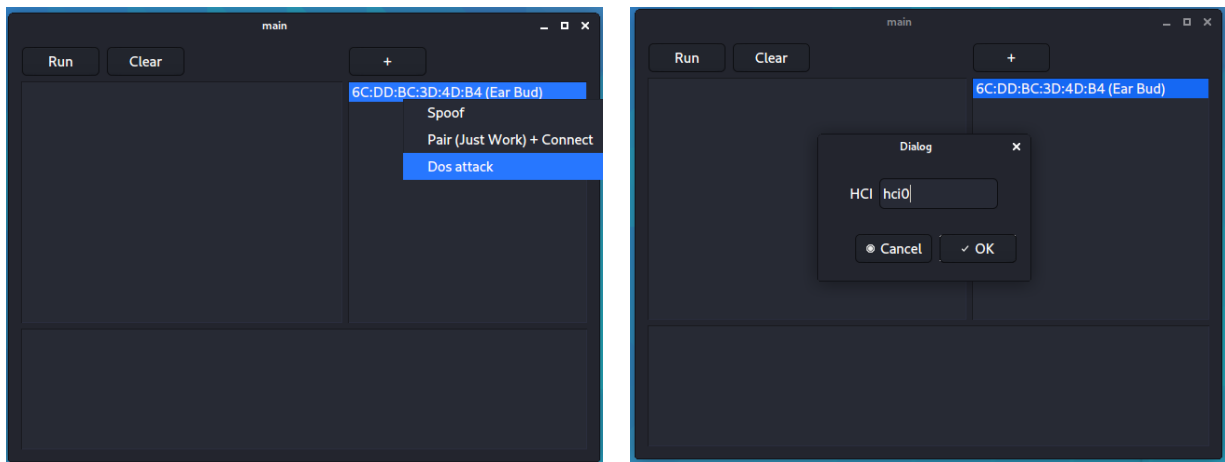


Click **“Run”** to launch the attack. After a while, the lower panel should show messages that indicate the result of the operation. As shown in the figure below, the attacking machine was able to pair and connect to the victim device.



## DoS attack against Bluetooth victim device

In some instances, the victim device might refuse the pairing request if it is connected to another device. This behaviour would render the previous attack (Just Work Pairing) useless. To bypass this security mechanism, we can launch a DoS attack against the victim first, which would bring down the established connection. After that's done, we can then attempt to pair with the victim. Right-click on the added address, and click "**Dos attack**", enter the HCI (in this case, we are using hci0), and click "**OK**"



Verify that the left panel shows the Dos attack we just added, and click "**Run**". After the Bluetooth connection is down, the lower panel should show messages that indicate the result of the operation. The message "**Recv failed: Connection reset by peer**" suggests that the Bluetooth was brought down. The figure below shows the result of the operation.

