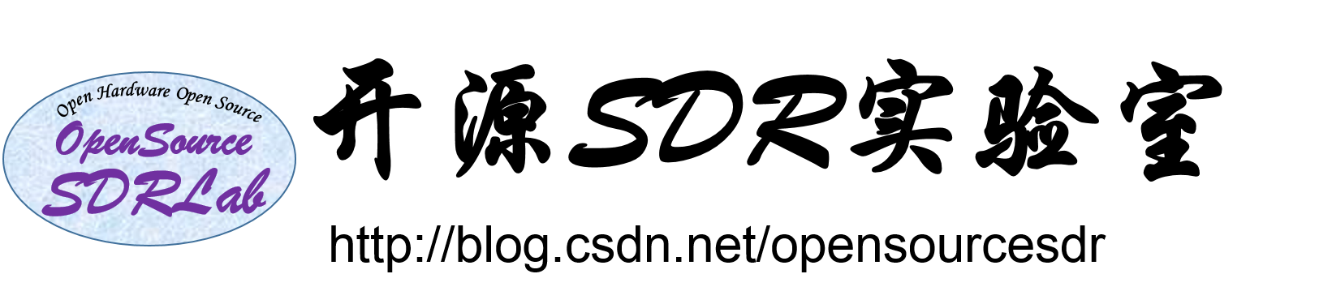
|  |  |  |  |
| --- | --- | --- | --- |
| **Ubertooth基本使用方法** | **文 档 编 号** | **版本** | **页数** |
| **ubertooth-01** | **1.0** | **9** |

| 版本 | 作者 | 版本描述 | 完成日期 |
| --- | --- | --- | --- |
| 1.0 | 开源sdr | 初始版本 | 2018. 12.14 |
| 2.0 | 开源sdr | 修改了kismet编译遇到的问题 | 2019.8.7 |
|  |  |  |  |
|  |  |  |  |



**目录**

[1 安装依赖包 1](#_Toc16141187)

[2 安装 libbtbb 1](#_Toc16141188)

[3 安装Ubertooth tools 2](#_Toc16141189)

[4 安装wireshark 2](#_Toc16141190)

[5 安装kismet 2](#_Toc16141191)

[5.1 安装kismet需要的依赖包 2](#_Toc16141192)

[5.2 安装kismet 3](#_Toc16141193)

[6 安装 BLE 解密工具 crackle 3](#_Toc16141194)

[7 ubertooth-btle 抓包蓝牙数据包并保存成pcap 文件 3](#_Toc16141195)

[8 修改 wireshark 使其正常显示 4](#_Toc16141196)

[9 一些截图 6](#_Toc16141197)

[10 参考文献 8](#_Toc16141198)

[11 常见错误及其解决方法 9](#_Toc16141199)

Ubertooth基本使用方法

# 安装依赖包

sudo apt-get install cmake libusb-1.0-0-dev make gcc g++ libbluetooth-dev pkg-config libpcap-dev python-numpy python-pyside python-qt4 python-software-properties

# 安装 libbtbb

wget https://github.com/greatscottgadgets/libbtbb/archive/2015-09-R2.tar.gz -O libbtbb-2015-09-R2.tar.gz

tar xf libbtbb-2015-09-R2.tar.gz

cd libbtbb-2015-09-R2

mkdir build

cd build

cmake ..

make

sudo make install

sudo ldconfig

备注：如果你想下载最新版本的，则可以按以下步骤执行：

wget https://github.com/greatscottgadgets/libbtbb/archive/2018-08-R1.tar.gz -O

libbtbb-2018-08-R1.tar.gz

tar -xf libbtbb-2018-08-R1.tar.gz

cd libbtbb-2018-08-R1

mkdir build

cd build

cmake ..

make

sudo make install

sudo ldconfig

# 安装Ubertooth tools

wget https://github.com/greatscottgadgets/ubertooth/releases/download/2015-09-R2/ubertooth-2015-09-R2.tar.xz -O ubertooth-2015-09-R2.tar.xz

tar xf ubertooth-2015-09-R2.tar.xz

cd ubertooth-2015-09-R2/host

mkdir build

cd build

cmake ..

make

sudo make install

sudo ldconfig

备注：如果你想下载最新版本的，则可以按以下步骤执行：

wget https://github.com/greatscottgadgets/ubertooth/releases/download/2018-08-

R1/ubertooth-2018-08-R1.tar.xz

tar xf ubertooth-2018-08-R1.tar.xz

cd ubertooth-2018-08-R1/host

mkdir build

cd build

cmake ..

make

sudo make install

sudo ldconfig

# 安装wireshark

sudo apt-get install wireshark

# 安装kismet

## 安装kismet需要的依赖包

sudo apt-get install build-essential git libmicrohttpd-dev pkg-config zlib1g-dev libnl-3-dev libnl-genl-3-dev libcap-dev libpcap-dev libncurses5-dev libnm-dev libdw-dev libsqlite3-dev libprotobuf-dev libprotobuf-c-dev protobuf-compiler protobuf-c-compiler libsensors4-dev

sudo apt-get install python-setuptools python3-setuptools python-protobuf python-requests

备注：官网教程上还需要安装 python2，但是我运行 sudo apt-get install python2 操作，安装不成功，所以这里我没有安装，实际后边使用验证也不影响。

## 安装kismet

git clone https://www.kismetwireless.net/git/kismet.git

cd kismet

./configure

make

sudo make suidinstall

# 安装 BLE 解密工具 crackle

git clone https://github.com/mikeryan/crackle.git

cd crackle

make

make install

# ubertooth-btle 抓包蓝牙数据包并保存成pcap 文件

ubertooth-btle -f -c test.pcap

上述语句含义是运行 ubertooth-btle 工具来进行蓝牙抓包并保存成 pcap 文件。

ubertooth-btle - passive Bluetooth Low Energy monitoring

Usage:

-h this help

Major modes:

-f follow connections

-p promiscuous: sniff active connections

-a[address] get/set access address (example: -a8e89bed6)

-s<address> faux slave mode, using MAC addr (example: -s22:44:66:88:aa:cc)

-t<address> set connection following target (example: -t22:44:66:88:aa:cc)

Interference (use with -f or -p):

-i interfere with one connection and return to idle

-I interfere continuously

Data source:

-U<0-7> set ubertooth device to use

Misc:

-r<filename> capture packets to PCAPNG file

-q<filename> capture packets to PCAP file

(DLT\_BLUETOOTH\_LE\_LL\_WITH\_PHDR)

-c<filename> capture packets to PCAP file (DLT\_PPI)

-A<index> advertising channel index (default 37)

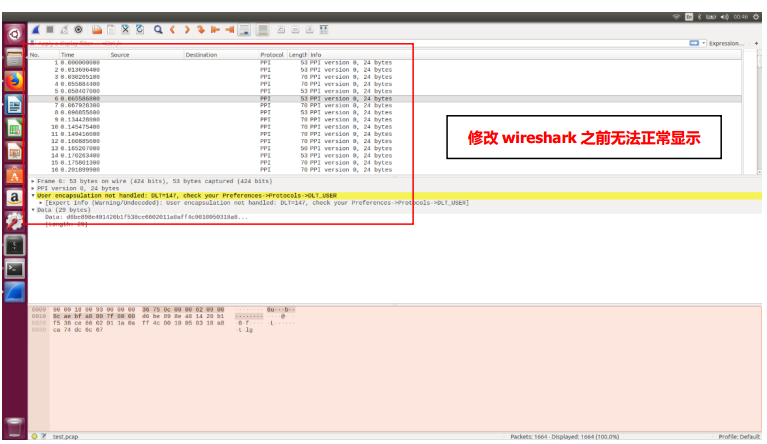
-v[01] verify CRC mode, get status or enable/disable

-x<n> allow n access address offenses (default 32)

If an input file is not specified, an Ubertooth device is used for live capture.

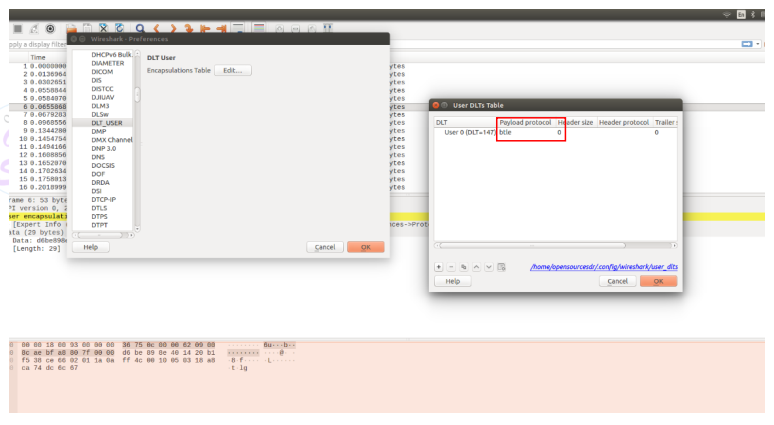
In get/set mode no capture occurs.

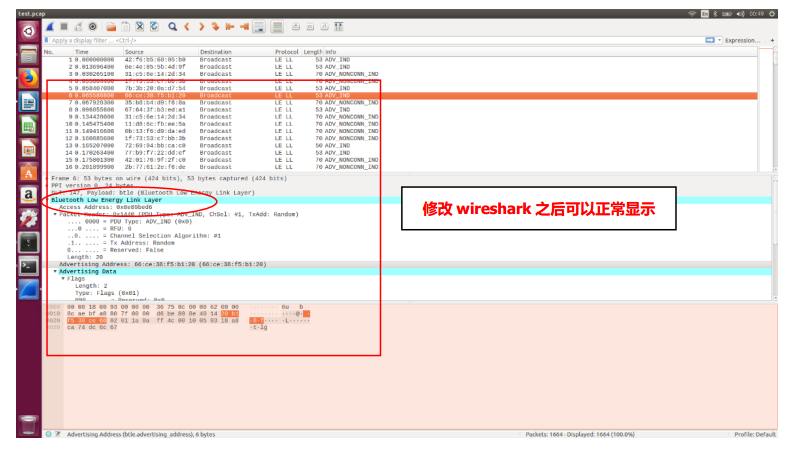
# 修改 wireshark 使其正常显示



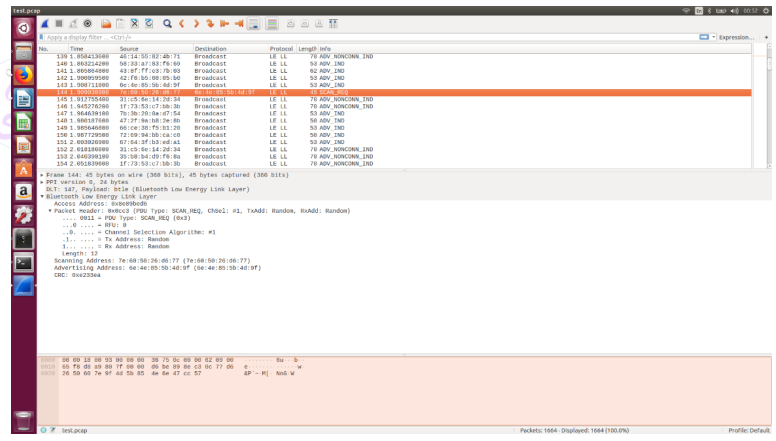
在 wireshark 中，依次找到 Edit → Preferences → Protocols → DLT\_USER → Edit →

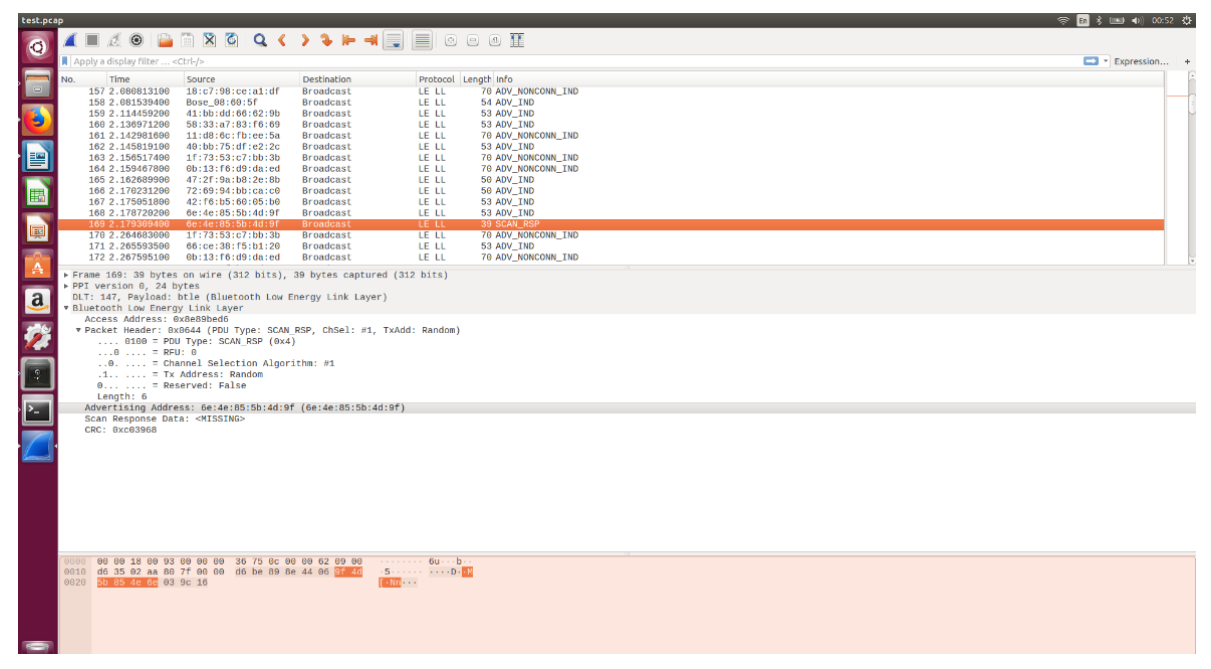
New（加号+），在 payload protocol 中输入 btle，确定即可。

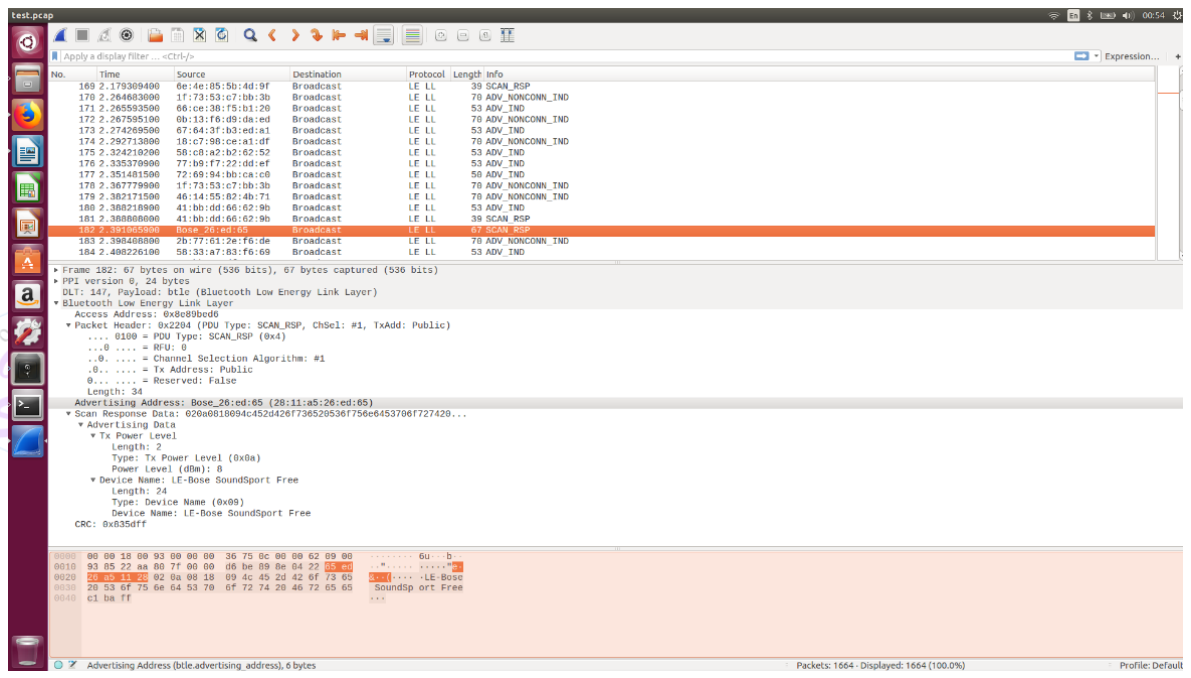


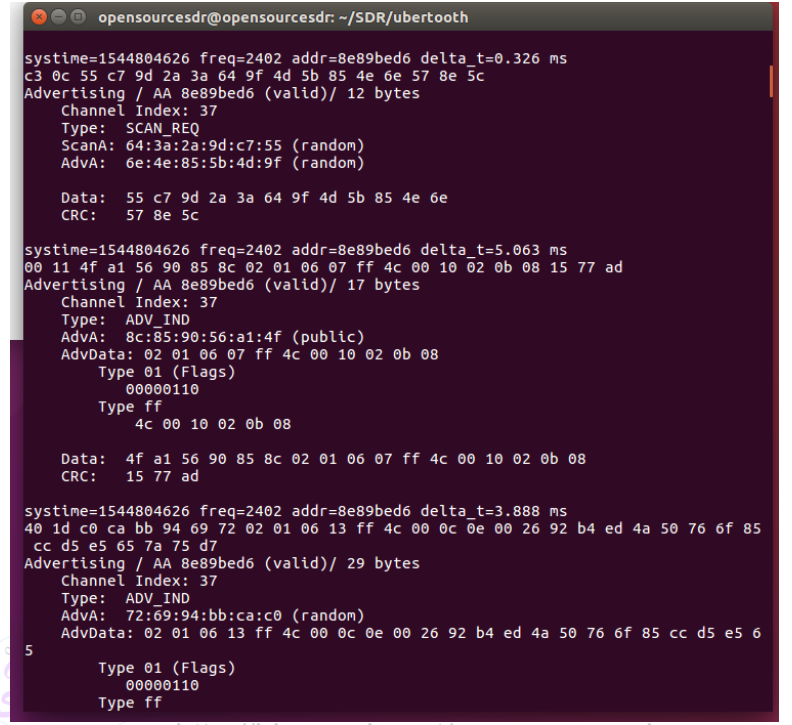


# 一些截图









# 参考文献

BLE Hacking ： 使用Ubertooth one 扫描嗅探低功耗蓝牙

https://www.freebuf.com/articles/wireless/106298.html

https://www.greatscottgadgets.com/ubertoothone/

https://www.kismetwireless.net/docs/readme/quickstart/

# 常见错误及其解决方法

kismet安装过程中configure出错：

opensourcesdr@opensourcesdr:~/SDR/kismet$ ./configure

checking build system type... x86\_64-unknown-linux-gnu

checking host system type... x86\_64-unknown-linux-gnu

checking for gcc... gcc

checking whether the C compiler works... yes

checking for C compiler default output file name... a.out

checking for suffix of executables...

checking whether we are cross compiling... no

checking for suffix of object files... o

checking whether we are using the GNU C compiler... yes

checking whether gcc accepts -g... yes

checking for gcc option to accept ISO C89... none needed

checking for gcc option to accept ISO C99... none needed

checking for gcc option to accept ISO Standard C... (cached) none needed

checking for g++... g++

checking whether we are using the GNU C++ compiler... yes

checking whether g++ accepts -g... yes

checking for a BSD-compatible install... /usr/bin/install -c

checking whether make sets $(MAKE)... yes

checking how to run the C preprocessor... gcc -E

checking for platform-specific compiler flags... none needed

checking gcc version... 5.4.0

checking whether g++ supports C++17 features with -std=gnu++17... no

checking whether g++ supports C++17 features with -std=gnu++1z... no

checking whether g++ supports C++17 features with -std=c++17... no

checking whether g++ supports C++17 features with +std=c++17... no

checking whether g++ supports C++17 features with -h std=c++17... no

checking whether g++ supports C++17 features with -std=c++1z... no

checking whether g++ supports C++17 features with +std=c++1z... no

checking whether g++ supports C++17 features with -h std=c++1z... no

configure: No compiler with C++17 support was found

checking whether g++ supports C++14 features with -std=gnu++14... yes

checking how to run the C++ preprocessor... g++ -std=gnu++14 -E

checking for grep that handles long lines and -e... /bin/grep

checking for egrep... /bin/grep -E

checking for ANSI C header files... yes

checking for sys/types.h... yes

checking for sys/stat.h... yes

checking for stdlib.h... yes

checking for string.h... yes

checking for memory.h... yes

checking for strings.h... yes

checking for inttypes.h... yes

checking for stdint.h... yes

checking for unistd.h... yes

checking whether byte ordering is bigendian... no

checking python3 module: setuptools... no

configure: error: Missing python setuptools, if you would like to build without python entirely, use --disable-python-tools, otherwise install python setuptools for your python version

原因是缺少python3-setuptools，解决方法是安装python3-setuptools。

sudo apt-get install python3-setuptools