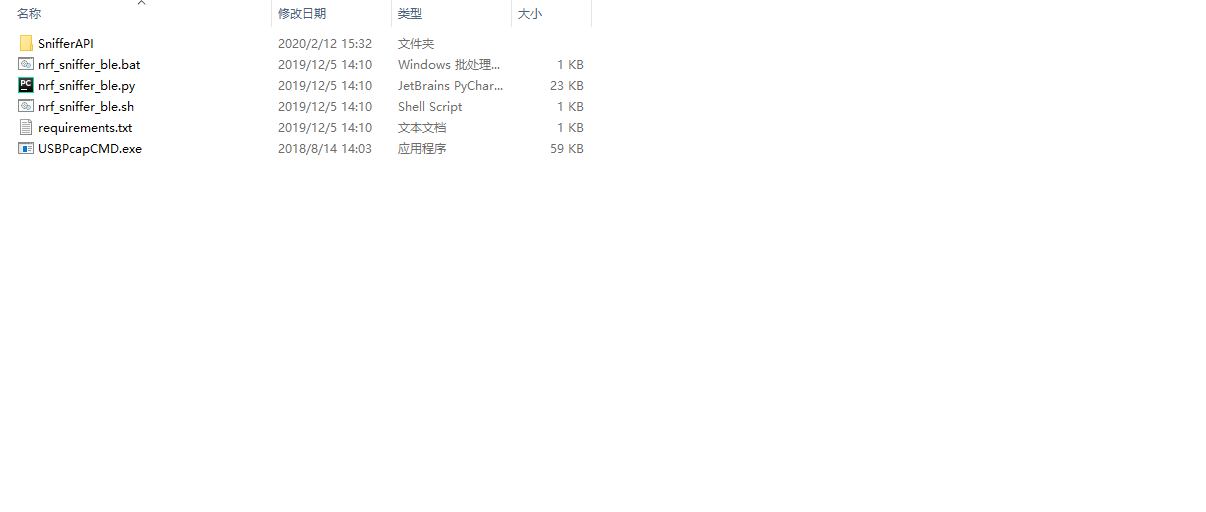
# 材料说明

WireShrak抓包

我们使用WireShark软件对测温枪的发送数据进行探究。同时，我在github上上传了名为nrf\_sniffer\_for\_bluetooth\_le\_3.0.0\_129d2b3的wireshark插件，将其中的extcap文件夹复制到wireshark对应的文件夹，即可在wireshark中使用低功耗蓝牙抓取的插件，使抓包过程更为便捷：



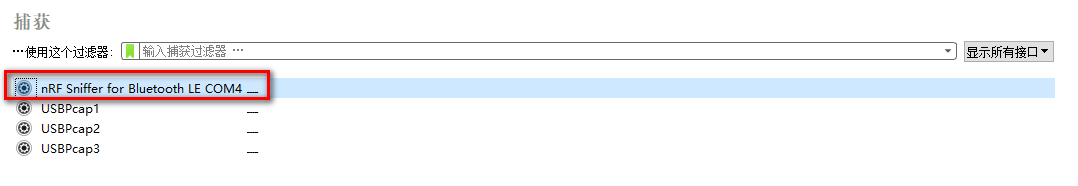
复制后的文件夹



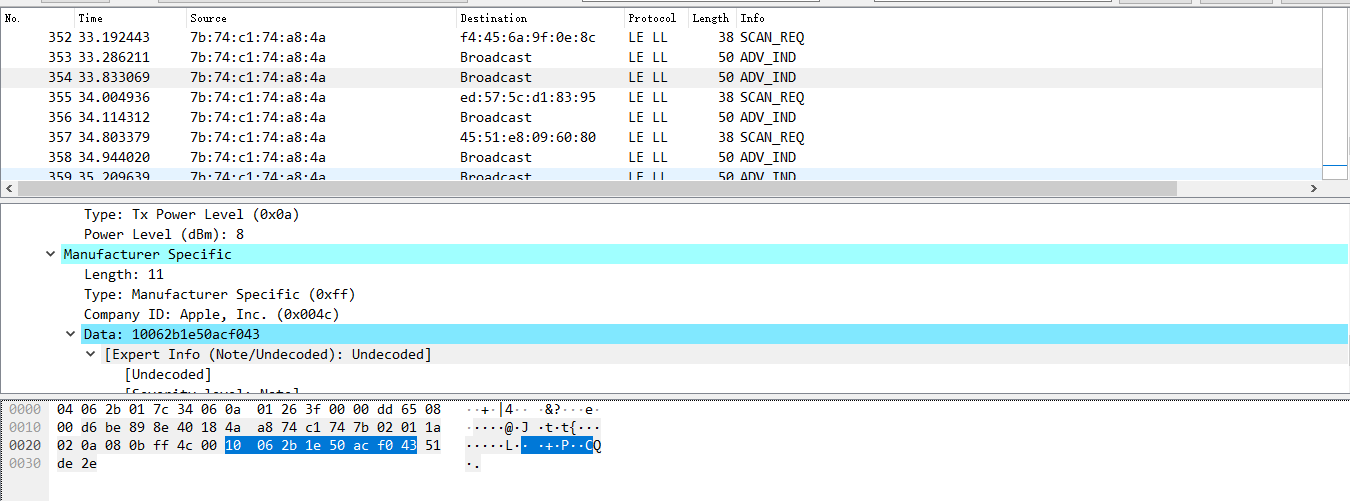
插件示意

下面我们正式开始使用sniffer抓取BLE包，在使用Bluefruit LE sniffer嗅探BLE流量之前，需要注意的一件事是，它可能对来自其他BLE设备的噪声敏感。尝试关闭附近的所有其他BLE设备，例如平板电脑，电话等。还要确保PC上正在运行的程序尽可能少，因为该工具需要尽快从Bluefruit嗅探器获取数据以防止数据包丢失。

启动wireshark后，选择sniffer双击进去进行抓包：



Wireshark加载后，很快就会看到大量广告包，例如：

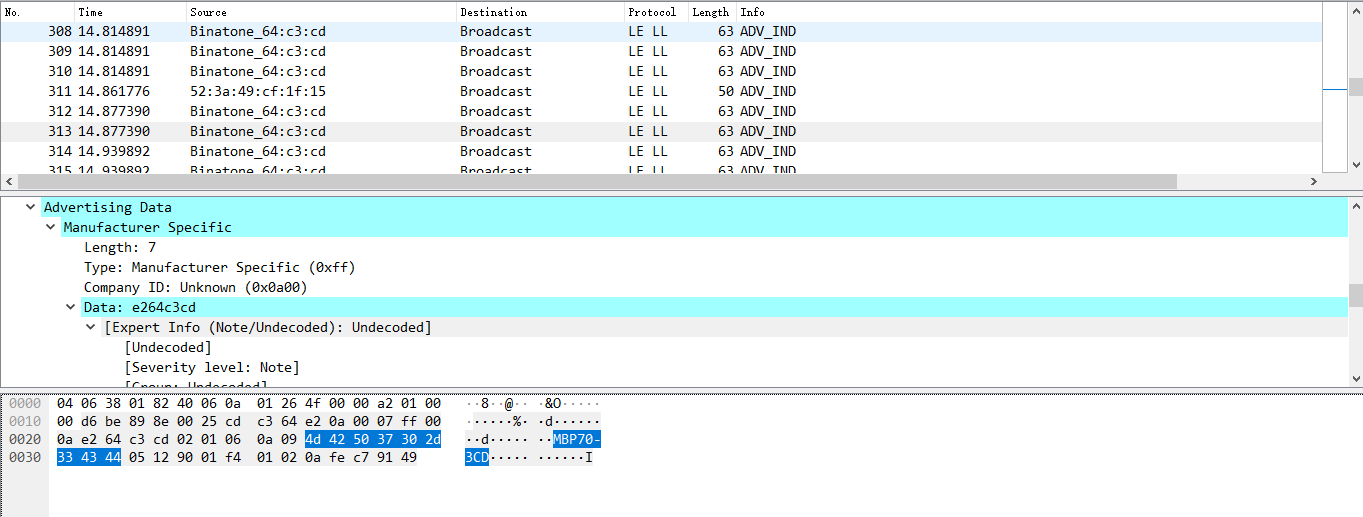


上图所示的窗口可以分为三个部分：

* 前三分之一是已捕获的数据包列表。当wireshark捕获数据包时，此列表将迅速增长。您可以上下滚动查看收到的数据包，也可以单击数据包以查看有关它的更多信息。
* 中间三分之一是已从数据包解码的信息。您可以深入查看数据包中的特定帧，以查看其作用。在这一层我们可以看到一些能够阅读的详细的数据包信息。
* 底部的三分之一是数据包数据的原始十六进制和ASCII表示形式。有趣的是，当您单击中间窗格中的信息时，您会在底部窗格中看到原始信息高亮表示。

在上图中，我选择了一个手机的广告包，并在中间窗格中突出了一些有趣的细节。您可以看到此数据包来自手机，他在不间断的向外发送广播包，同时广播包里面带上了一些自定义的数据。在上面的窗格中我们还可以看到，在app上点击搜索的时候，手机向一些固定的mac地址发送了SCAN\_REQ的广播包。

然后我们启动手机端的app，连接到测温枪，并在这个过程中一直启动sniffer，抓包结果如下图：



可以看到，测温枪一直在发送同一段广播包，里面包含了设备名，以及一段自身的mac地址，但是从协议端可以看到，我们只能观察到LE LL协议的内容，一直没有抓取到ATT类型的包，而只有那种的包才能识别出对我们有用的信息。到这里wireshark抓包就告一段落了。

使用gattacker

在正式开始之前，我们必须具备以下条件：

**硬件**

* 笔记本电脑
* BLE Dongles

**软件**

* Gattacker

****安装Gattacker并对其进行配置：****

**执行步骤**

**Step 1**：安装Gattacker，你需要最新版本的node和npm。可以使用以下命令安装：

sudo apt-**get** install bluetooth bluez libbluetooth-dev libudev-dev

**Step 2**：接下来，我们需要安装bleno，命令如下：

npm install bleno

npm install noble

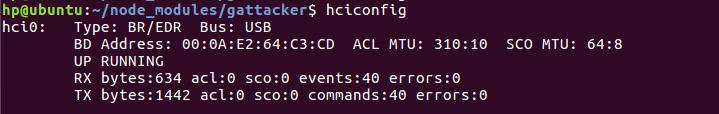
如果出现错误，请确保你之前已正确安装了node和npm软件包

**Step 3**：现在使用该命令安装Gattacker

npm install gattacker

**Step 4**：在另一台虚拟机（或系统）上重复相同的步骤，因为我们需要两台机器一台用于主机，另一台用于从机。（这个地方我是克隆了一台虚拟机，但是好像不是必须要有两台虚拟机，但是两台会好一些）

**Step 5**：完成后，插入ble适配器并使用sudo hciconfig确保其已插入。



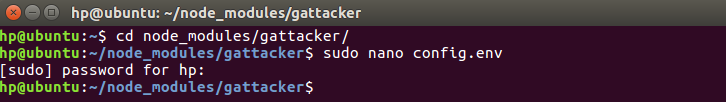
然后导航到gattacker文件夹

cd node\_modules/gattacker

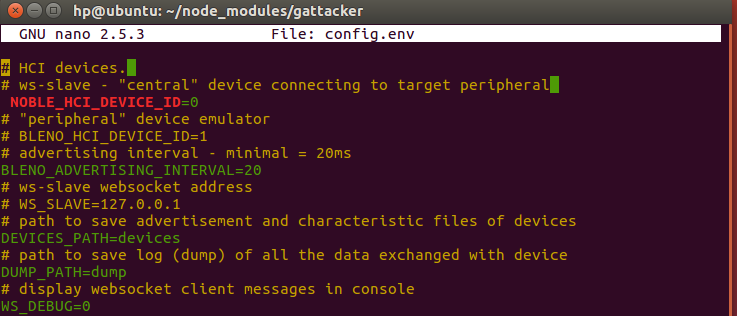
**Step 6**：

接下来，我们需要编辑config.env文件，配置gattacker。

**sudo** **nano** **config**.env



**Step 7**：取消**NOBLE\_HCI\_DEVICE\_ID**注释，然后将其替换为hciX，其中（X是我们之前通过hciconfig找到的值）并保存文件



**Step 8**：现在在主机中，插入BLE适配器并按照上述步骤操作。对于config.env，请按照以下步骤操作：

* 取消**BLENO\_HCI\_DEVICE\_ID**注释

将它们分配给hciX值。

完成后，在WS\_SLAVE中，将其中的IP地址替换成从机的地址：



使用Gattacker扫描和存储设备信息：

**执行步骤**

**Step 1**：打开从机VM并启动ws-slave.js，如下所示：

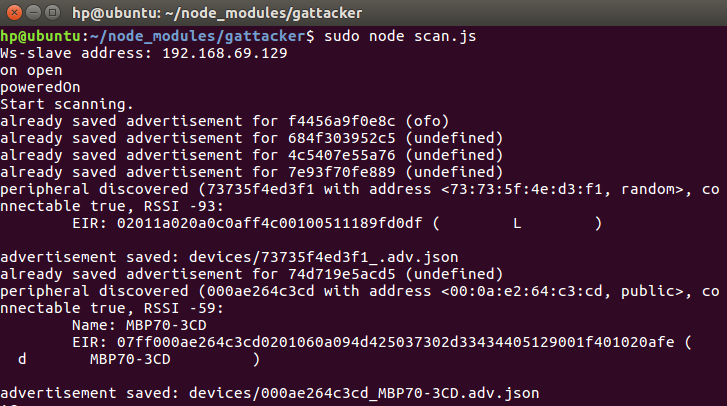
**sudo** **node** **ws-slave**.js



**Step 2**：现在在主机分支的gattacker文件夹下并启动扫描，如下所示

**sudo** **node** **scan**.js

现在打开测温枪，然后按Enter键以查看如下消息：



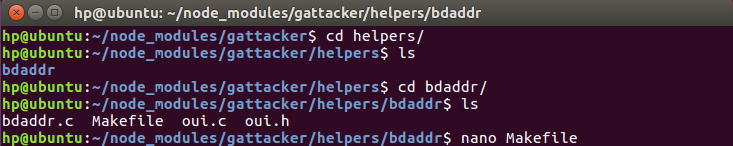
sudo node scan 00ae264c3cd其中00ae264c3cd是由gattacker保存的测温枪的peripheral名称。

**使用gattacker进行攻击：**

**Step 1**：按照前面的步骤存储advertisement和service数据包。 现在我们关闭测温枪。

**Step 2**：做克隆目标设备的MAC地址的准备，获取目前设备的蓝牙地址。

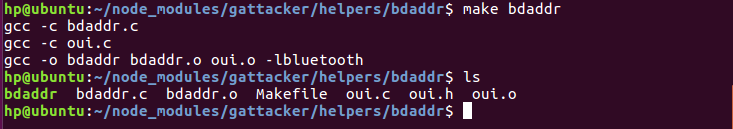
首先进入helpers下面的bdaddr



看看Makefile的内容：



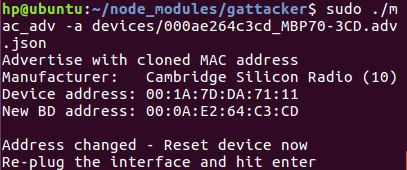
然后我们需要执行的命令是bdaddr：



**Step 3**：克隆目标设备的MAC地址。

sudo ./mac\_adv

-a devices/00ae264c3cd\_MBP70\_3CD.adv.json



**Step 2**：完成后，保险起见，重新插拔一下连接到主机的adapter在主机上结合advertisement和service数据包启动advertise.js。

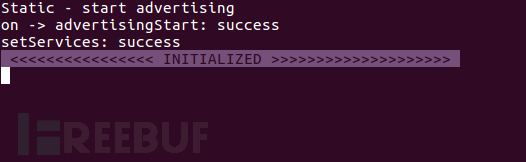
sudo node advertise.js

-a devices/00ae264c3cd\_MBP70\_3CD.adv.json

-s devices/00ae264c3cd.srv.json

现在按回车并等待消息：

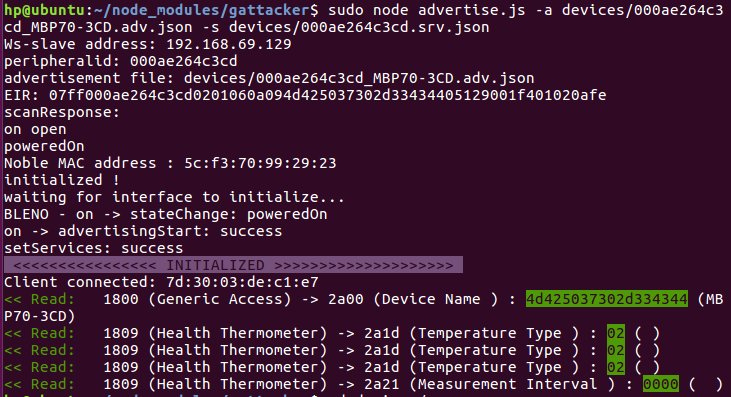
现在打开测温枪，你会看到如下信息：



**Step 3**：现在从手机中打开Thermometer for hubble应用程序并扫描设备：



**Step 4**：现在连接到设备，在主机中你可以看到如下内容:



Bluez

BlueZ是Linux官方蓝牙协议栈。它是一个基于GNU General Public License (GPL)发布的开源项目，从Linux2.4.6开始便成为Linux 内核的一部分。其基础代码均是由就职于Qualcomm(高通)的Maxim Krasnyansky完成的。包括：HCI，L2CAP，RFCOMM和基本socket的实现。Marcel Holtmann开发层的协议和应用，包括：BNEP, CMTP等。当然，这些中也有Maxim Krasnyansky的参预。有部分代码由Nokia提供的。

1. BlueZ支持蓝牙核心层和协议，它灵活、高效，以模块化方式实现，具有以下特点：

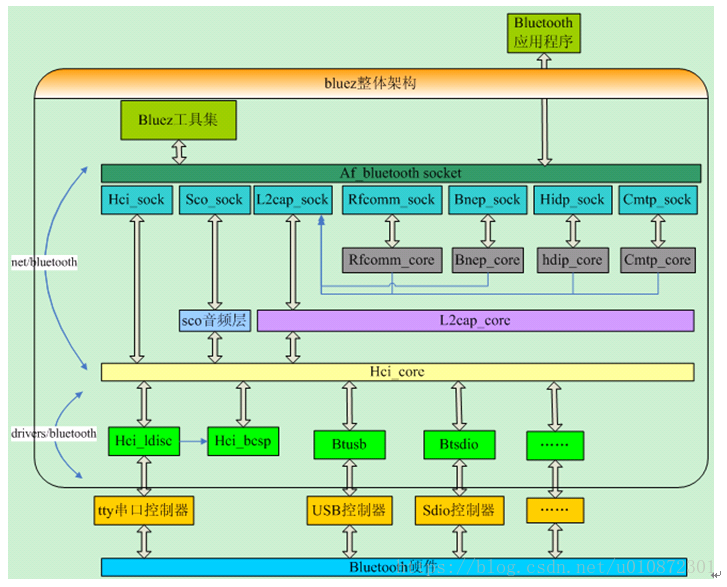
* 完整的模块化实现
* 均衡的多处理安全
* 支持多线程数据处理
* 支持多个蓝牙设备
* 硬件抽象
* 提供所有层的标准socket接口
* 支持设备级和服务级安全保证

1. BlueＺ包含多个相互独立的模块：

* Linux内核蓝牙子系统核心
* L2CAP 和 SCO 音频内核层
* RFCOMM, BNEP, CMTP 和 HIDP内核实现
* HCI UART, USB, PCMCIA 和虚拟设备驱动
* 通用的蓝牙和SDP库及守护进程
* 配置和测试小工具
* 协议解析和分析工具

我们主要使用了上面标红的这些模块

下图是bluez的一个架构示意图：



Bluez的安装

在ubuntu14.04 下编译安装Bluez-5.52

1、安装相关依赖库：

sudo apt-get install libusb-dev libdbus-1-dev libglib2.0-dev automake libudev-dev libical-dev libreadline-dev

2、下载Bluez-5.28压缩包并解压:

wget http://www.kernel.org/pub/linux/bluetooth/bluez-5.52.tar.xz

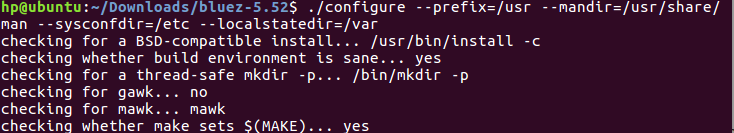
tar xvf bluez-5.52.tar.xz

进入解压后的目录：

cd bluez-5.52

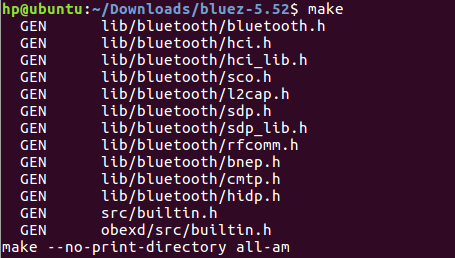
3、执行配置文件

./configure --prefix=/usr --mandir=/usr/share/man --sysconfdir=/etc --localstatedir=/var

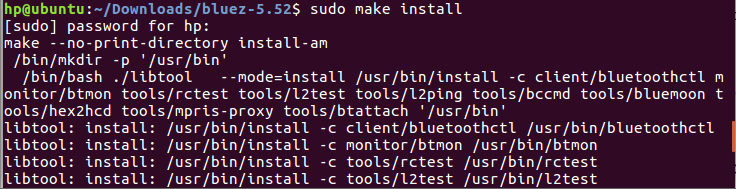


4、执行make和install

make



sudo make install



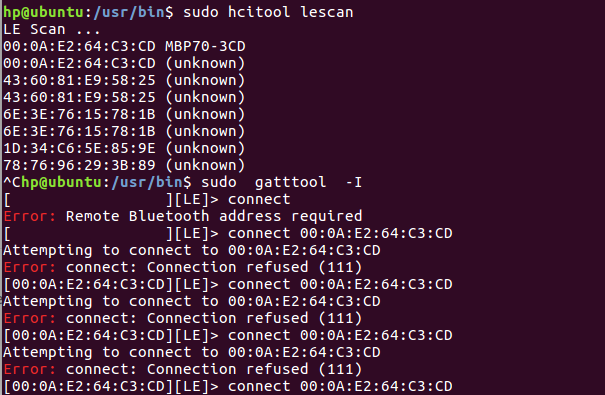
没有提示任何错误的话，安装完毕。

Bluez的使用

1. lescan扫描低功耗蓝牙设备

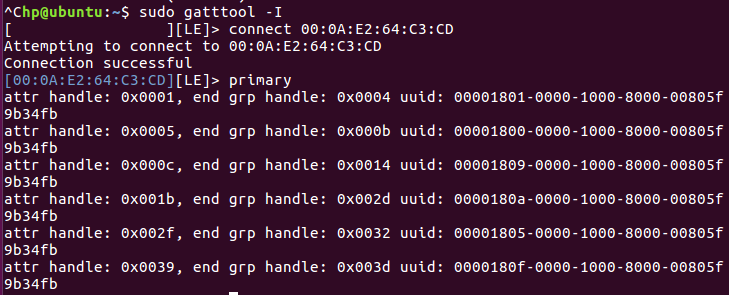


1. connect   //连接低功耗设备



1. primary

在连接成功后输入这个命令，可以看到该低功耗蓝牙设备能够支持的服务，和uuid



1. char-desc 0x000c 0x0014

可以运行的另一个命令是char-desc命令，以获取有关特定特征的详细信息。我运行以下命令来查询handle 0x000c到0014的特性：

