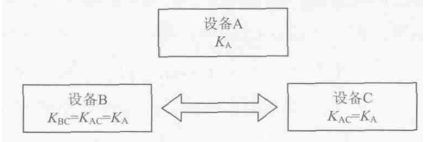
# 材料说明

现在已知的蓝牙安全漏洞包括如下几个：跳频时钟、PIN码、链路密钥欺骗、加密密钥流重复。针对这些漏洞，产生了一些针对蓝牙设备的攻击。

## 蓝牙中间人攻击原理

在两个设备之间的攻击者截获数据一方发送的数据后再转发给另一方，可在不影响双方通信的情况下获得双方通信的内容，是一种广泛应用于无线网络的攻击方式。蓝牙 4.0 版本的低功耗蓝牙技术（Bluetooth Low Energy，BLE）在设计初始时有防范中间人攻击的安全措施，但是在产品阶段考虑到产品功耗成本等因素，这方面并没有得到足够的重视，依然容易受到攻击。最常见的是用软硬件结合的蓝牙攻击设备伪造BLE 通信进行中间人攻击。中间人攻击示意图下图所示。



## 蓝牙数据包简析

在蓝牙协议中，数据首先有一个交互的过程，在这个过程中，会发送一些SCAN\_REQ，SCAN\_RSP这样的扫描数据包来进行连接，我们就可以通过抓包的软件或是其他方式查看蓝牙设备发送的这些包。

## 针对BLE设备中间人攻击实例

### GATT服务初探

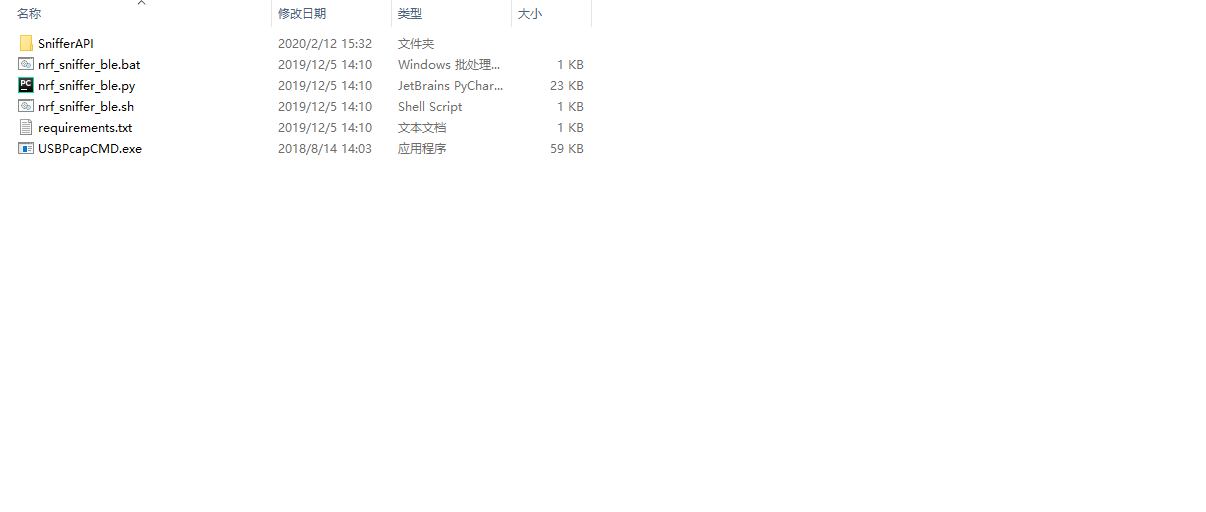
我使用了IOS上的一款名为LightBlue的蓝牙调试软件进行初步测试，这是测试图：



从这张图我们可以看到我们的测温枪的设备名和使用的服务名以及设备的一些信息以及使用该服务发送的一些值。

### WireShrak抓包

我们继续使用WireShark软件对测温枪的发送数据进行探究，要注意的是，在进行这些活动时，测温枪不能连接其专用的APP，否则将会无法抓包。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：

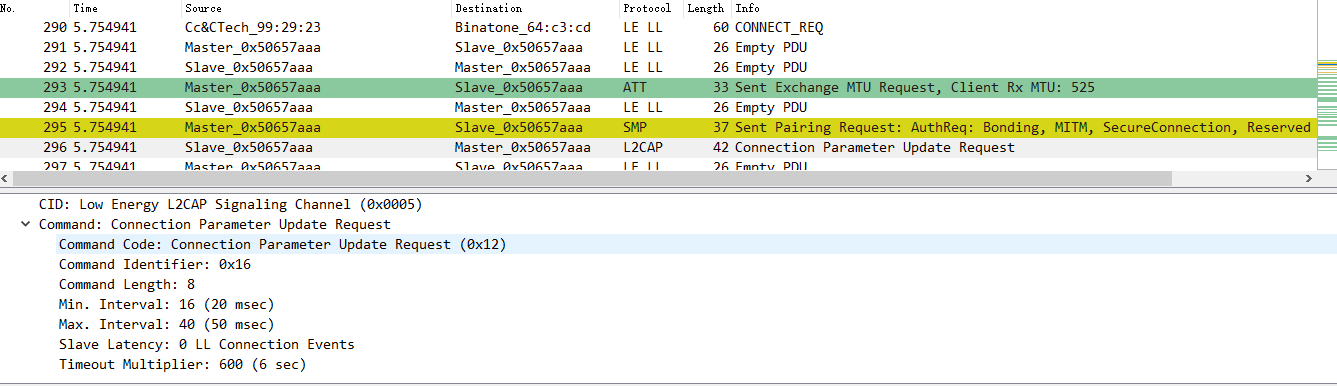


复制后的文件夹



插件示意

下面是WireShark抓包的图：



### 进行MITM攻击

### 接收

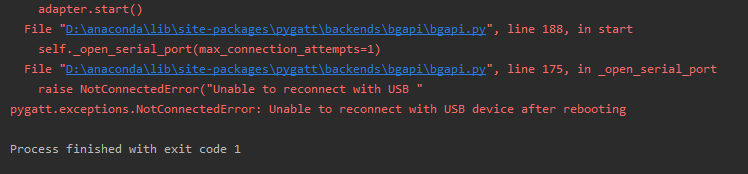
从上面的分析可知，我们这里可以实施较为初级的攻击，即首先将测温枪连接到自己的电脑，同时读取数据，得到测温枪发送的数据格式，但是这一步并不是最重要的。

因为在我们的项目中，测温枪发送温度这一最重要的步骤我们是无法探知的，这不同于其他一些蓝牙设备的交互逻辑，在我们的项目中，从设备（即测温枪）是数据的发送方，所以他发送的数据必须要连接上主设备后才开始发送，所以我们不能达到他发送温度数据的格式。但是如果是一个使用手机蓝牙控制小灯这样的应用中，就算没有连接上小灯，我们仍然可以通过手机发送命令，这就给了我们探究该命令数据格式的机会。但是所幸我们的项目发送数据必须使用规定的服务方式，所以理论上数据格式是固定的。

然后这一步也是没有成功，因为这一步需要使用蓝牙接收器来连接上测温枪接收数据并分析，但是我的电脑当蓝牙接收器连接上的时候没办法识别COM口，所以就没办法搜索设备。：



蓝牙接收器adapter

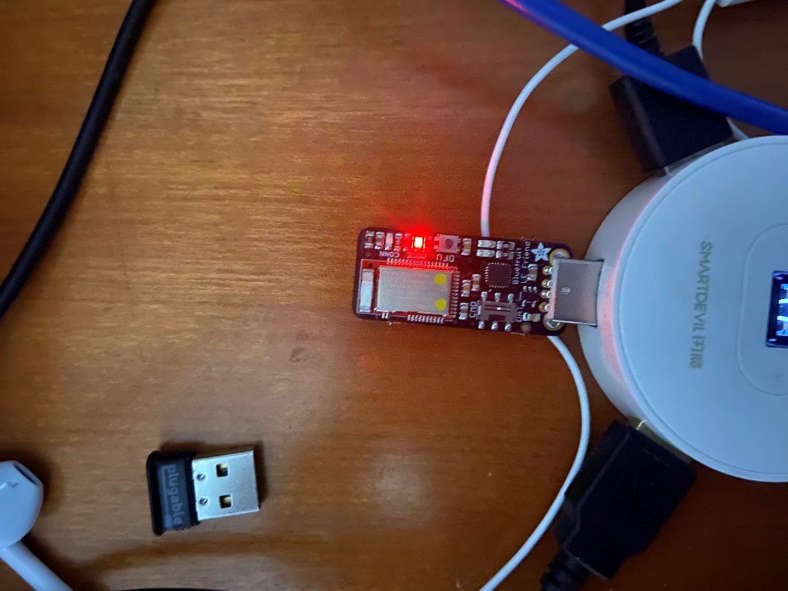


代码运行失败截图

这个部分属于硬件问题，可能出问题的地方是这个蓝牙接收器、或者是windows下这个代码的问题，因为我现在没有Linux系统的环境，所以这个问题暂时是无解的。

### 发送

接收到了数据，接下来我们分析一下这个数据就将它发送出去，发送的时候使用LE Friend：



LE friend

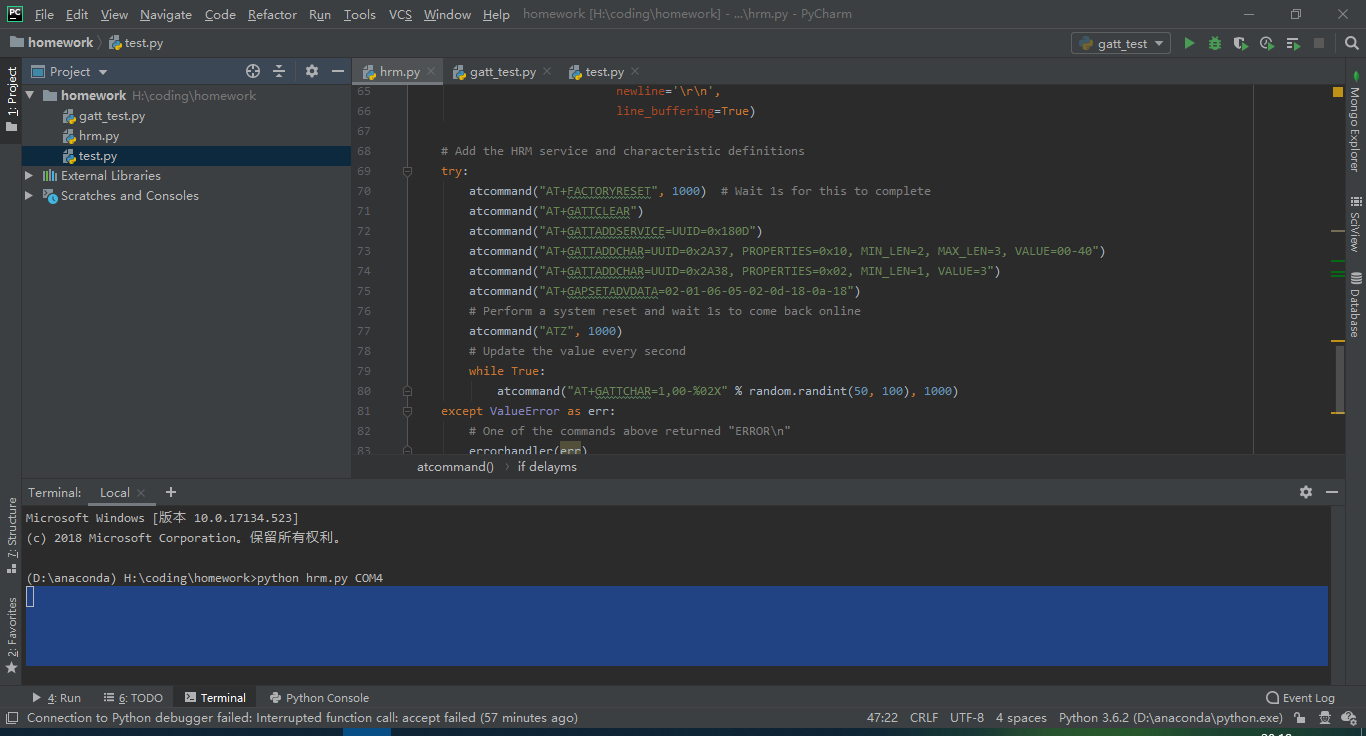
同样的，这个LE friend也是亮红灯，无法工作，但是你给的那个sniffer和这个LE friend是一模一样的东西，然后它可以使用，所以我用了你给的sniffer：



Sniffer

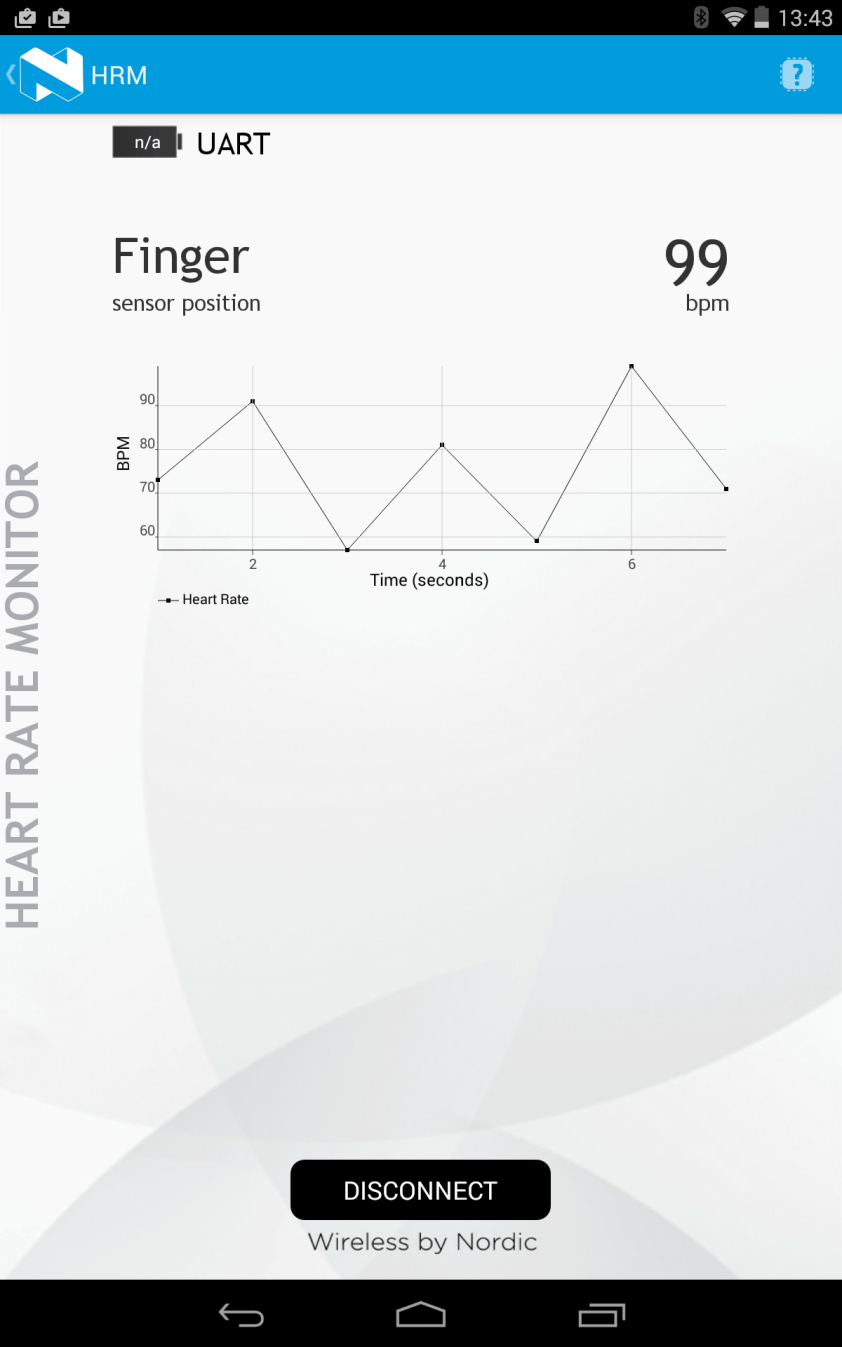
可以看到，sniffer蓝灯的部分是在进行嗅探，而黄灯的地方是在向外发送数据。

程序的运行时如下：



程序运行截图

同时，在手机端我们使用一款名叫nRF Toolbox的软件，可以看到我们自己修改的值被发送出来：



这就是在发端我们做的工作，到这里，一个完整的中间人攻击就结束了。

代码我已经上传了在<https://github.com/HanjuWang/BLE-man-in-the-middle-attack-test>

其中有三个文件thermometer.py是发送端代码

GATT.py是接收端代码

文件夹是wireshark插件

使用linux系统

我们在这里想要使用bluez平台，接下来我们将纤细介绍这个工具，它包含了许多实用的蓝牙工具并且便于使用。不幸的是，没有跨Windows，Mac，Linux等平台的跨平台蓝牙协议栈或API。因此，如果想使用其他平台，则需要查看该平台的蓝牙低能耗协议栈和API。

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：

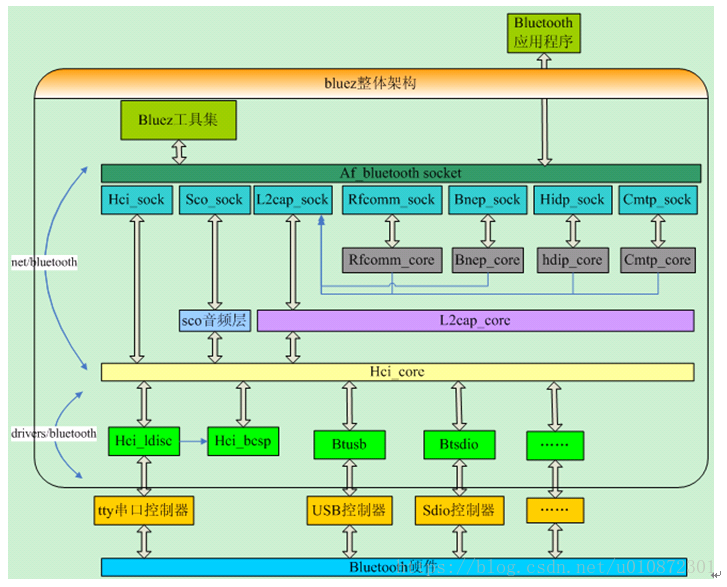
* 完整的模块化实现
* 均衡的多处理安全
* 支持多线程数据处理
* 支持多个蓝牙设备
* 硬件抽象
* 提供所有层的标准socket接口
* 支持设备级和服务级安全保证

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

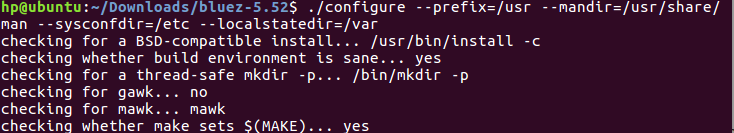
tar xvf bluez-5.52.tar.xz

进入解压后的目录：

cd bluez-5.52

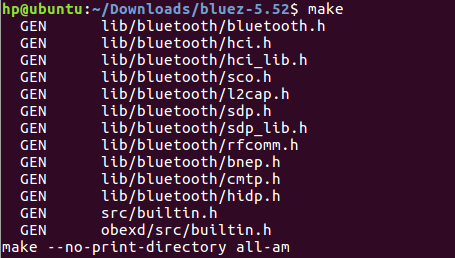
3、执行配置文件

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

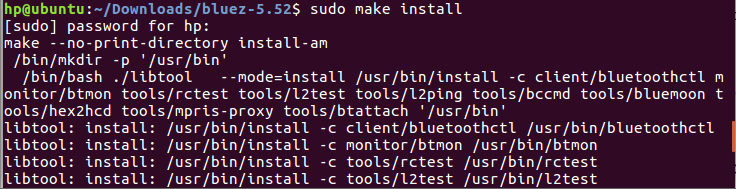


4、执行make和install

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

