巅峰极客（MISC)

转载

weixin_33810006 于 2018-07-24 11:49:00 发布 256 收藏
文章标签： 网络
原文链接：https://yq.aliyun.com/articles/649057

关于巅峰极客写一下wp吧

毕竟自己那么菜（留给以后的自己看吧QAQ)

这一篇主要是关于杂项的（不简单呀！自己太渣了）

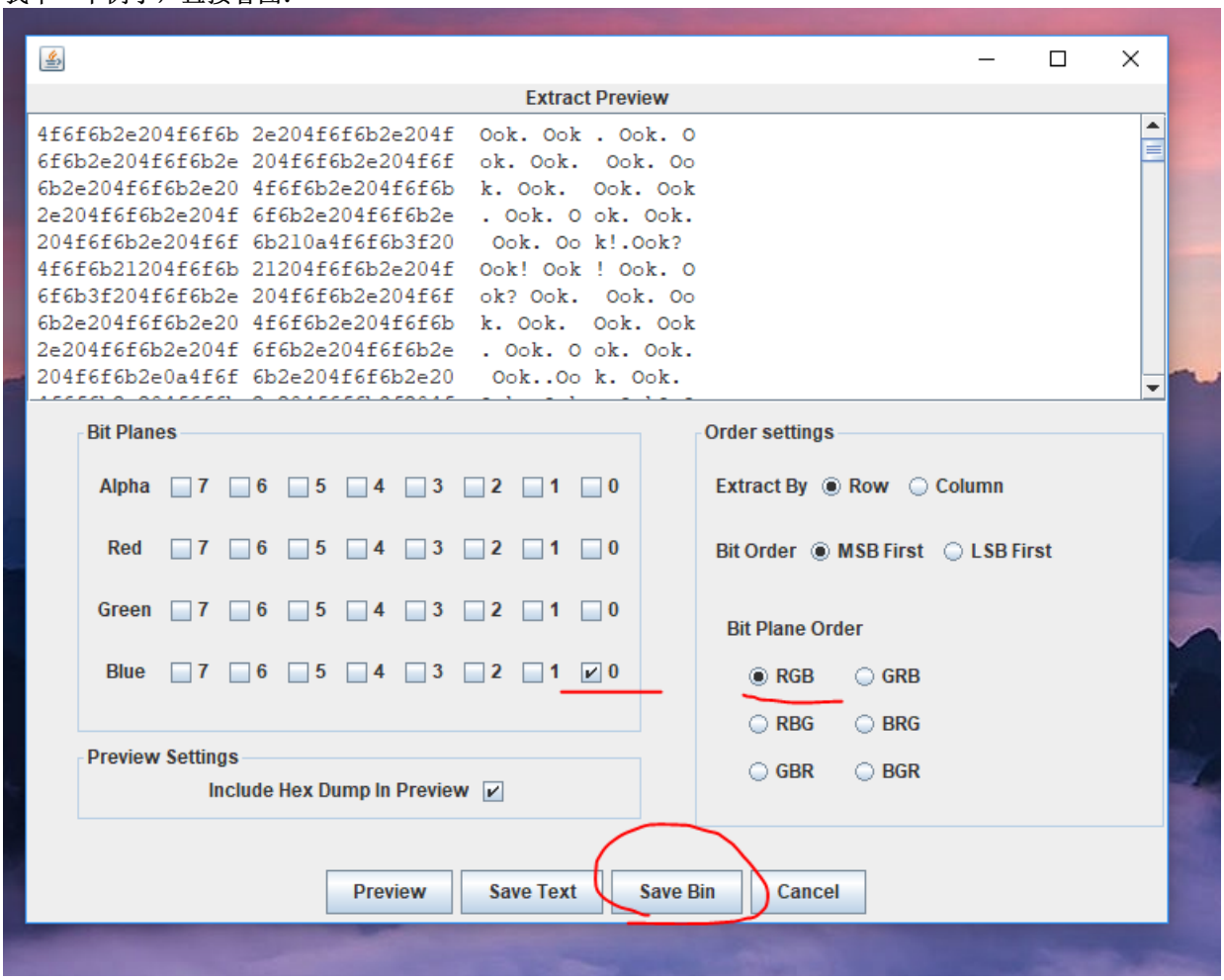
第一题：签到题：

直接打开就可以看到答案：flag{the_greatest_geek}

下面是杂项的内容：

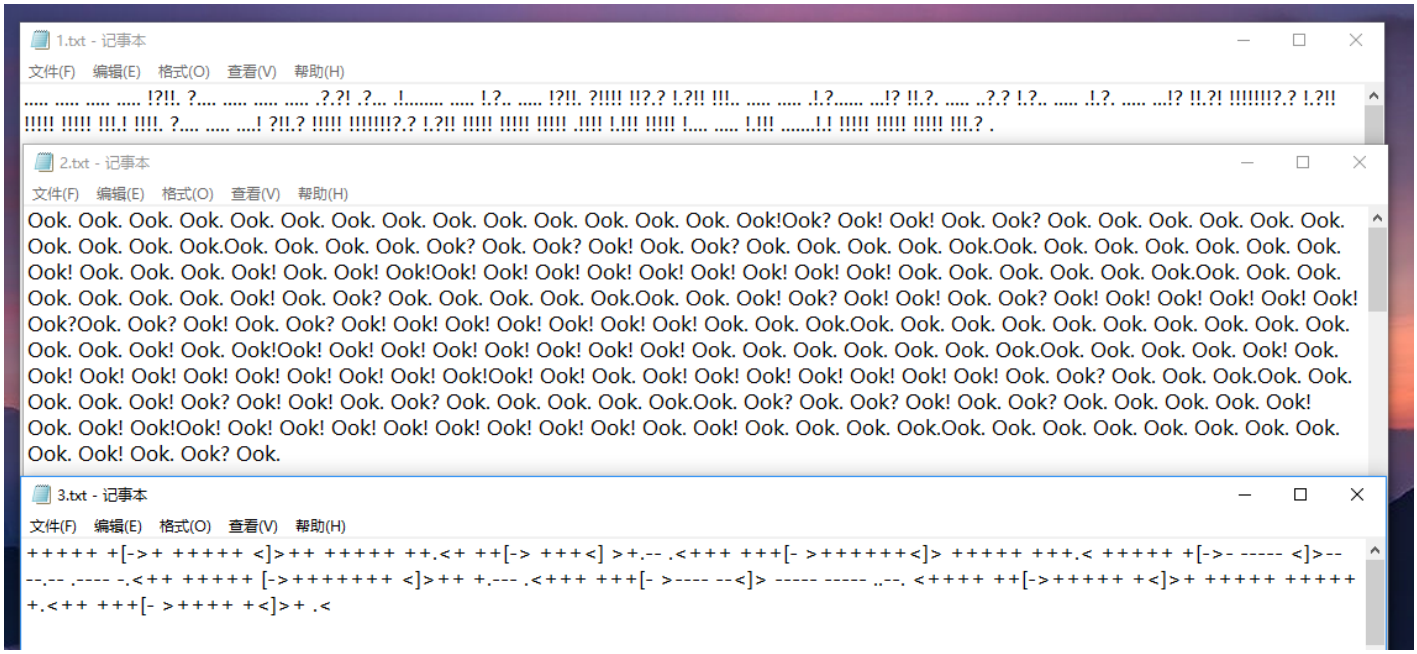
warmup

用stegsolve.jar 打开图片，Analyse→Data Extract，分别选中R、G、B三种最低位，导出bin文件
我举一个例子，直接看图：



就是这样操作的，分别选中R、G、B三种最低位，导出bin文件

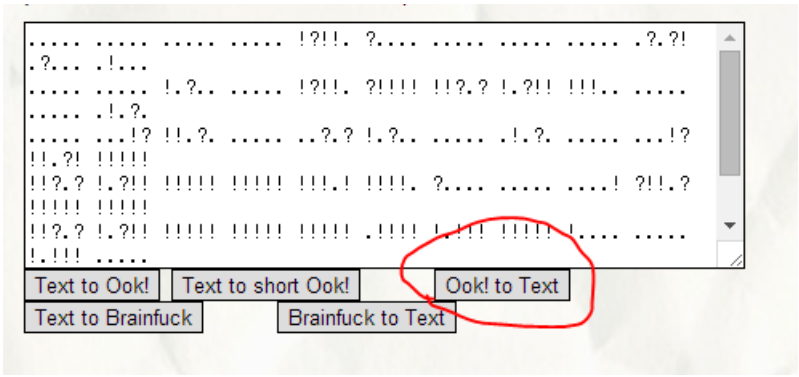
就会得到：

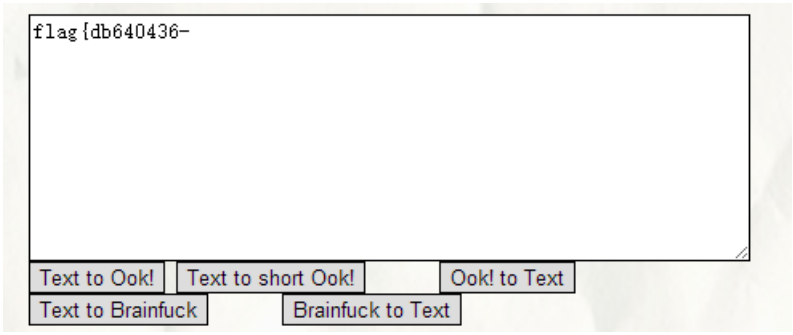


这个是属于十三种最为荒谬的语言（我写的有这个方面的介绍：https://www.cnblogs.com/lxz-1263030049/p/9123775.html

我们可以使用在线工具：https://www.splitbrain.org/services/ook

我也是举个栗子（hhhhh)



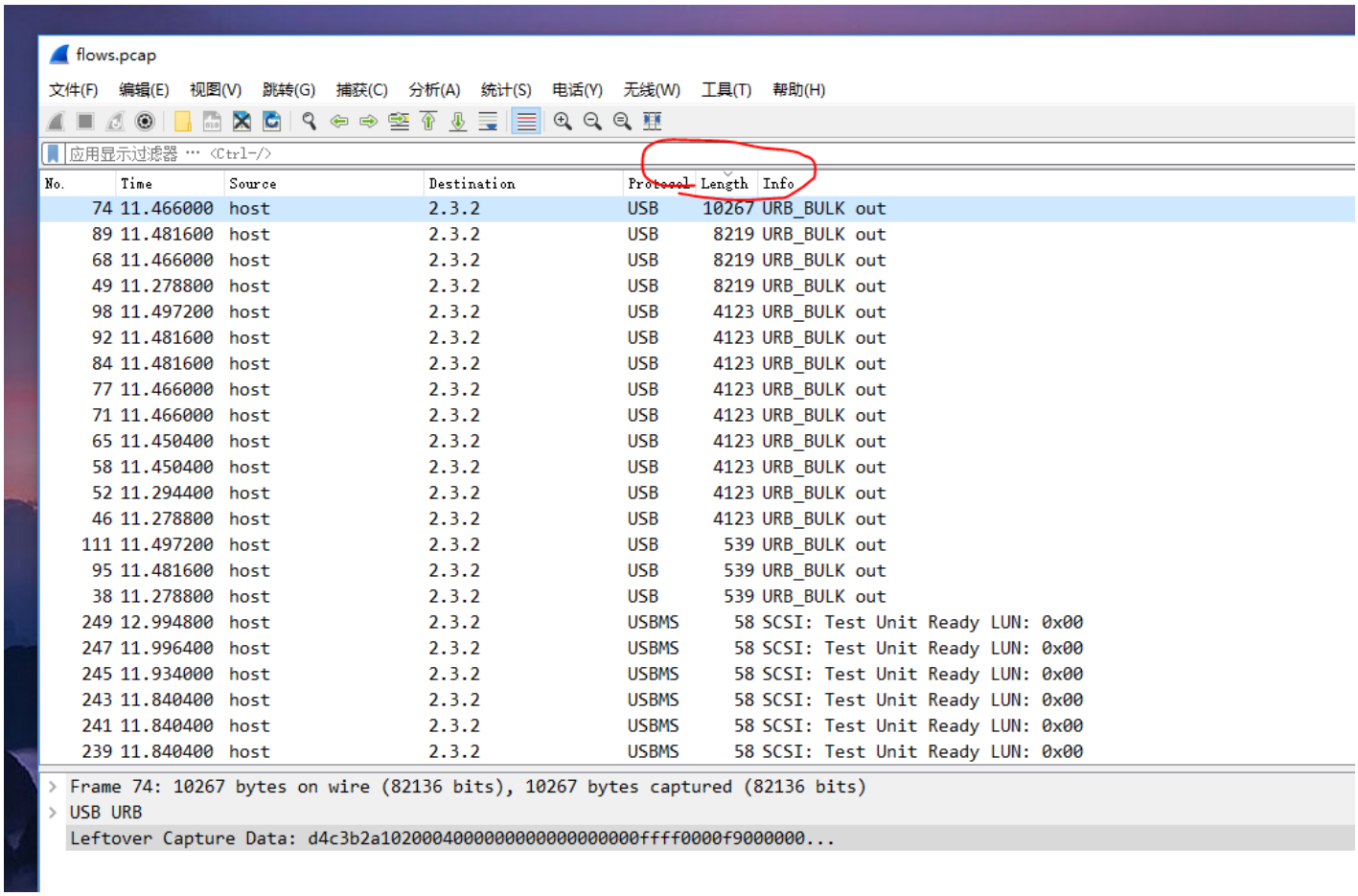


接着把剩下的两个继续解密就会得到组成flag：

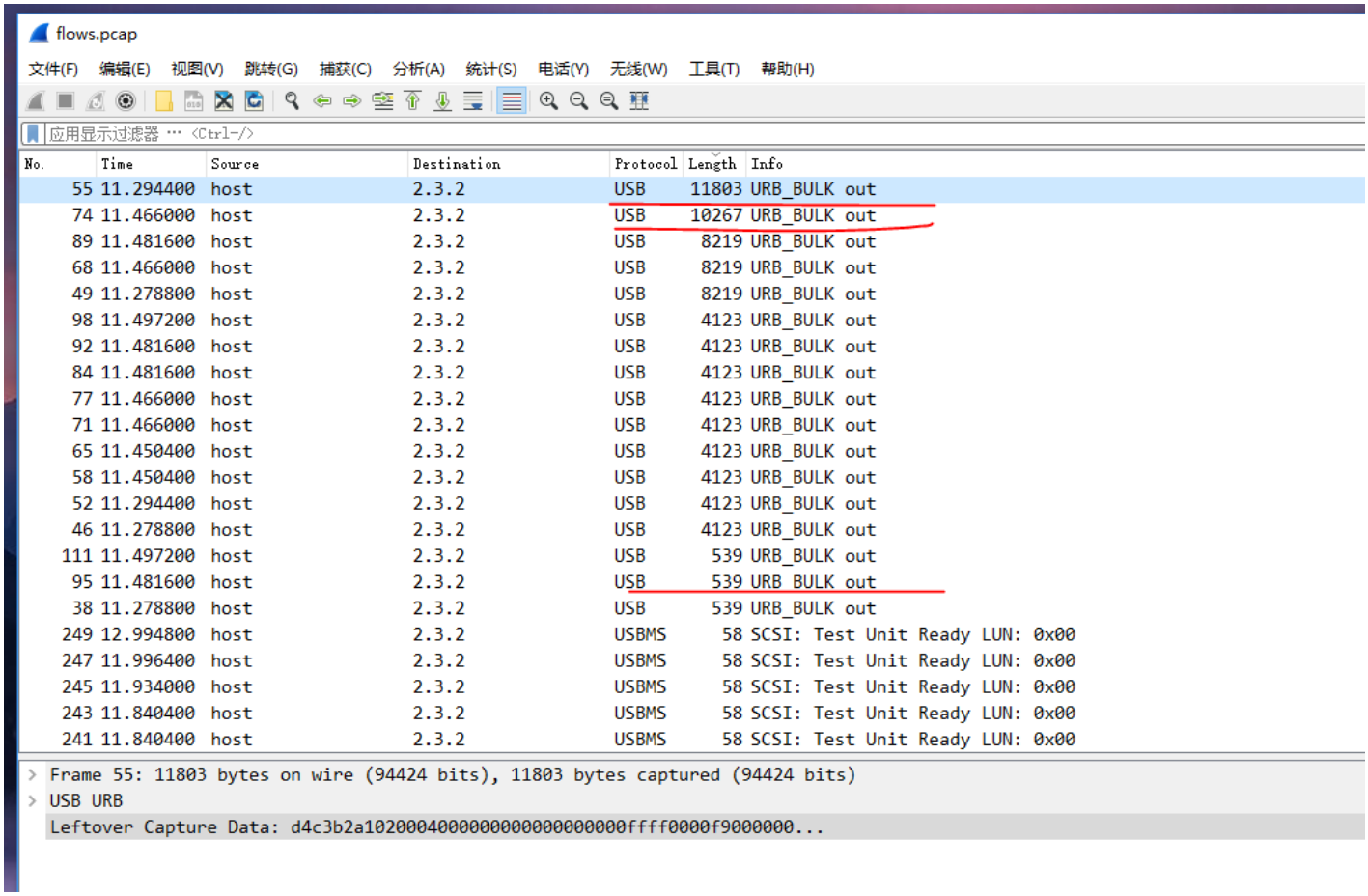
flag{db640436-7839-4050-8339-75a972fc553c}

3：flows

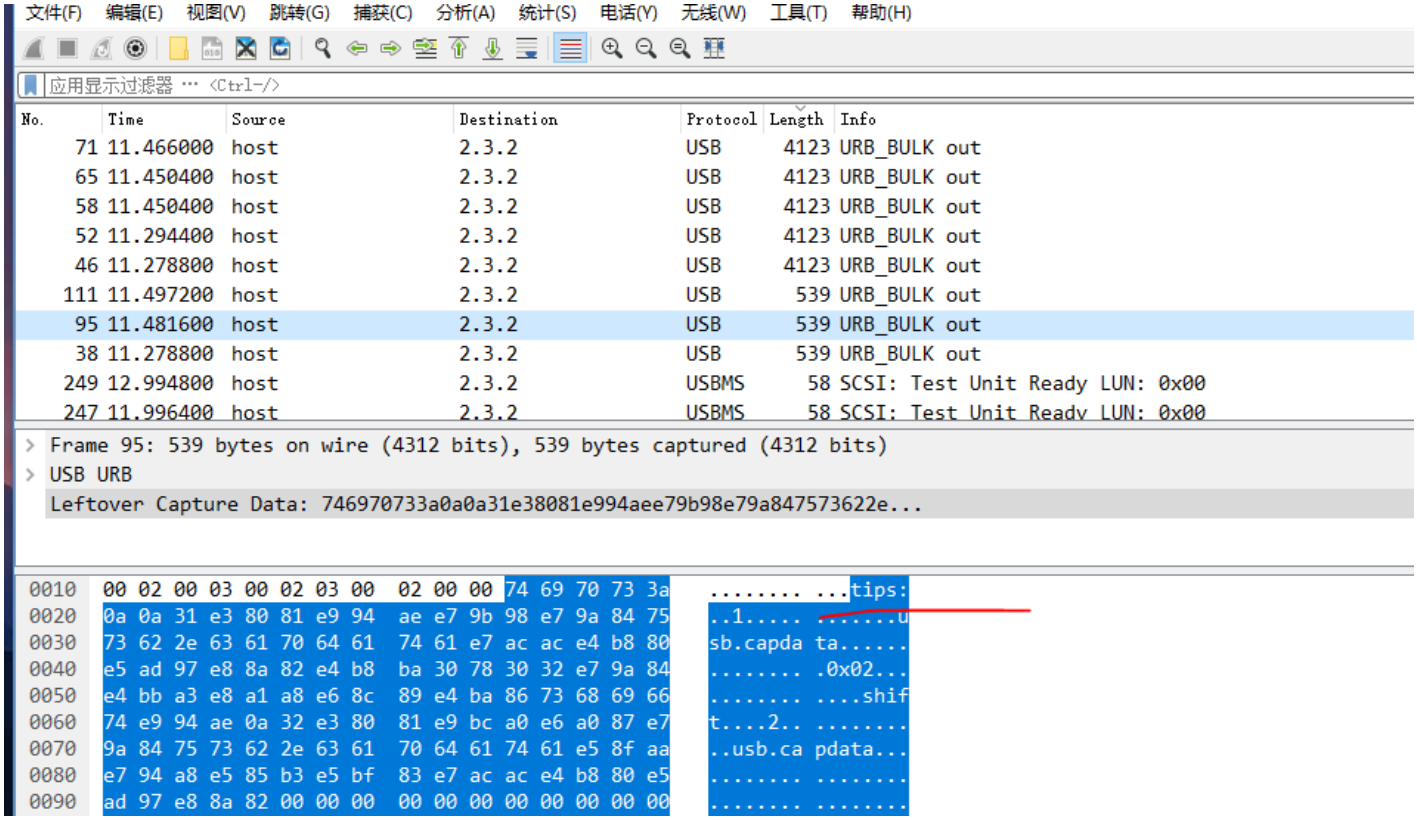
首先：使用Wireshark打开就会得到：（点击Length进行排序）



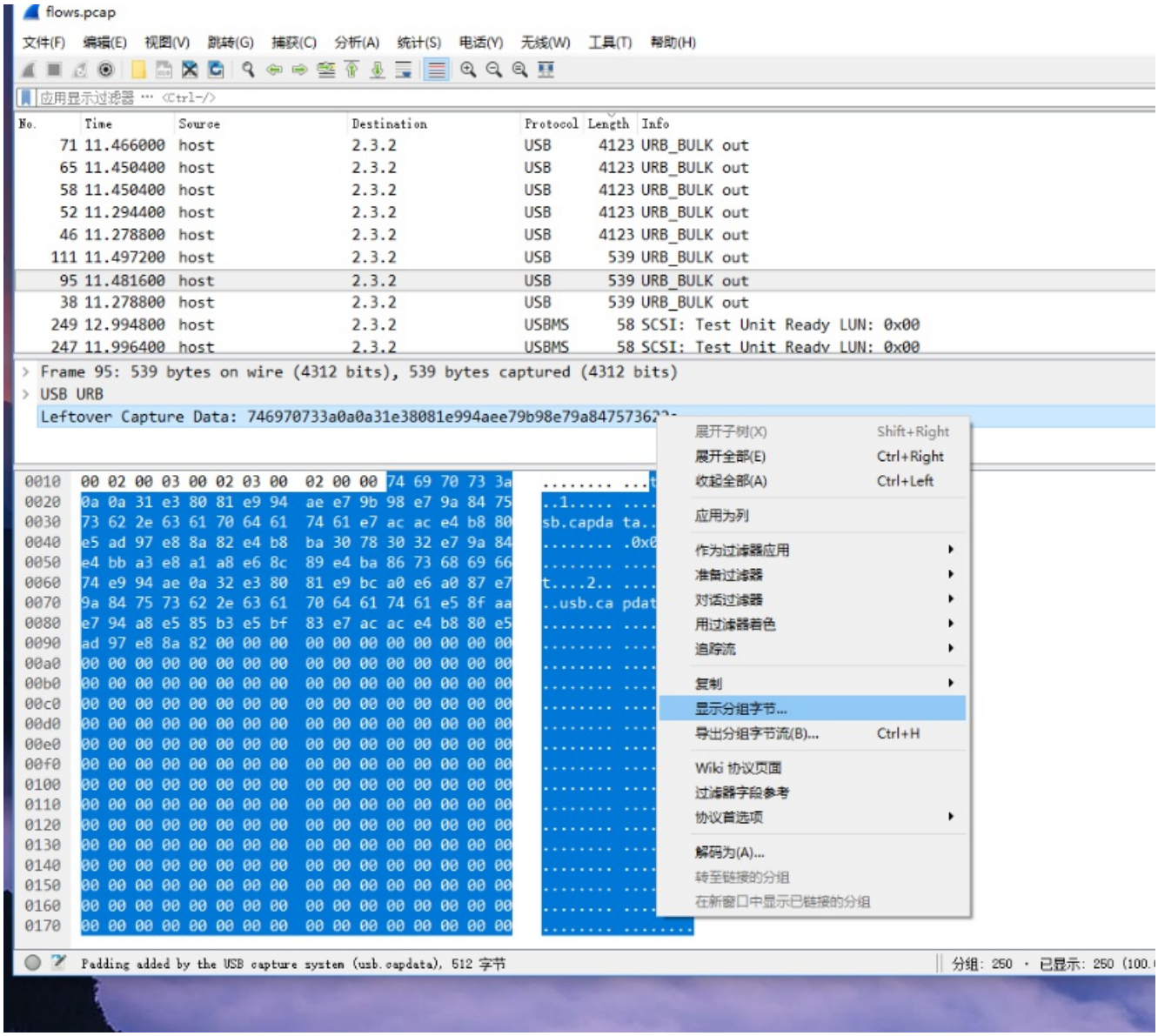
可以看出这是一个抓取usb协议的数据包，看下长度比较大的几个包，可以找到3个文件，这是抓取的将文件复制到u盘上产生的数据包

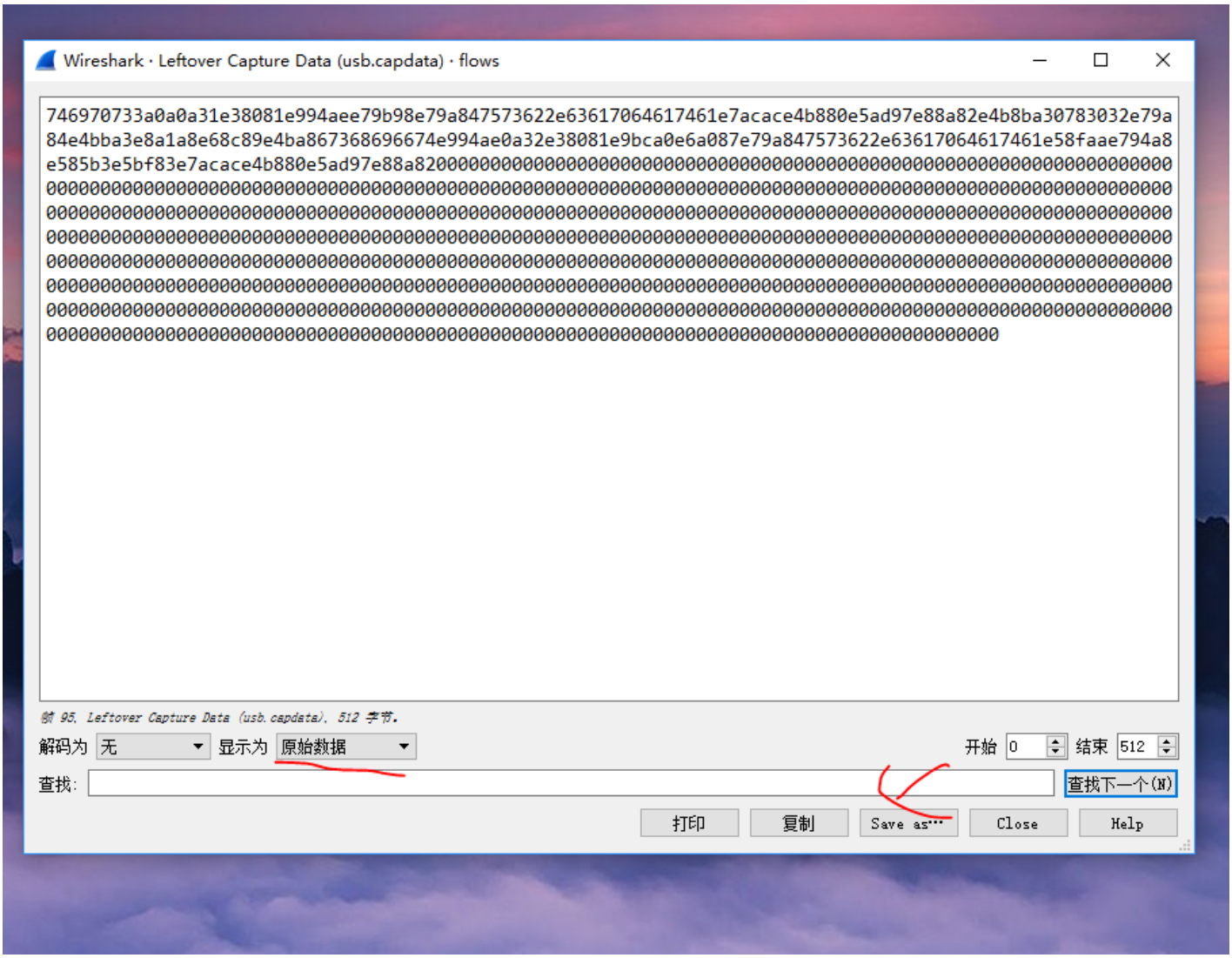


点击下面这一个可以看到：

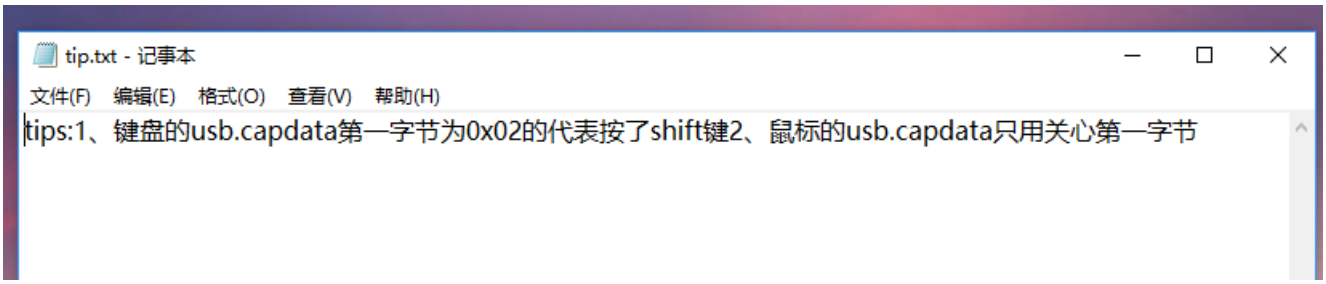


分别为两个pcap文件和一个txt文件，我们可以将这些数据另存下来分析


flows.pcap
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
应用显示过滤器 … <Ctrl-/>
No. Time Source Destination Protocol Length Info
71 11.466000 host 2.3.2 USB 4123 URB_BULK out
65 11.450400 host 2.3.2 USB 4123 URB_BULK out
58 11.450400 host 2.3.2 USB 4123 URB_BULK out
52 11.294400 host 2.3.2 USB 4123 URB_BULK out
46 11.278800 host 2.3.2 USB 4123 URB_BULK out
111 11.497200 host 2.3.2 USB 539 URB_BULK out
95 11.481600 host 2.3.2 USB 539 URB_BULK out
38 11.278800 host 2.3.2 USB 539 URB_BULK out
249 12.994800 host 2.3.2 USBMS 58 SCSI: Test Unit Ready LUN: 0x00
247 11.996400 host 2.3.2 USBMS 58 SCSI: Test Unit Ready LUN: 0x00
Frame 95: 539 bytes on wire (4312 bits), 539 bytes captured (4312 bits)
USB URB
Leftover Capture Data: 746970733a0a0a31e38081e994aee79b98e79a847573622
0010 00 02 00 03 00 02 03 00 02 00 00 74 69 70 73 3a
0020 0a 0a 31 e3 80 81 e9 94 ae e7 9b 98 e7 9a 84 75
0030 73 62 2e 63 61 70 64 61 74 61 e7 ac ac e4 b8 80
0040 e5 ad 97 e8 8a 82 e4 b8 ba 30 78 30 32 e7 9a 84
0050 e4 bb a3 e8 a1 a8 e6 8c 89 e4 ba 86 73 68 69 66
0060 74 e9 94 ae 0a 32 e3 80 81 e9 bc a0 e6 a0 87 e7
0070 9a 84 75 73 62 2e 63 61 70 64 61 74 61 e5 8f aa
0080 e7 94 a8 e5 85 b3 e5 bf 83 e7 ac ac e4 b8 80 e5
0090 ad 97 e8 8a 82 00 00 00 00 00 00 00 00 00 00 00
00a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
展开子树(X) Shift+Right
展开全部(E) Ctrl+Right
收起全部(A) Ctrl+Left
应用为列
作为过滤器应用
准备过滤器
对话过滤器
用过滤器着色
追踪流
复制
显示分组字节...
导出分组字节流(B)... Ctrl+H
Wiki 协议页面
过滤器字段参考
协议首选项
解码为(A)...
转至链接的分组
在新窗口中显示已链接的分组
Padding added by the USB capture system (usb.capdata), 512 字节
分组: 250 · 已显示: 250



保存为txt文件就会得到：

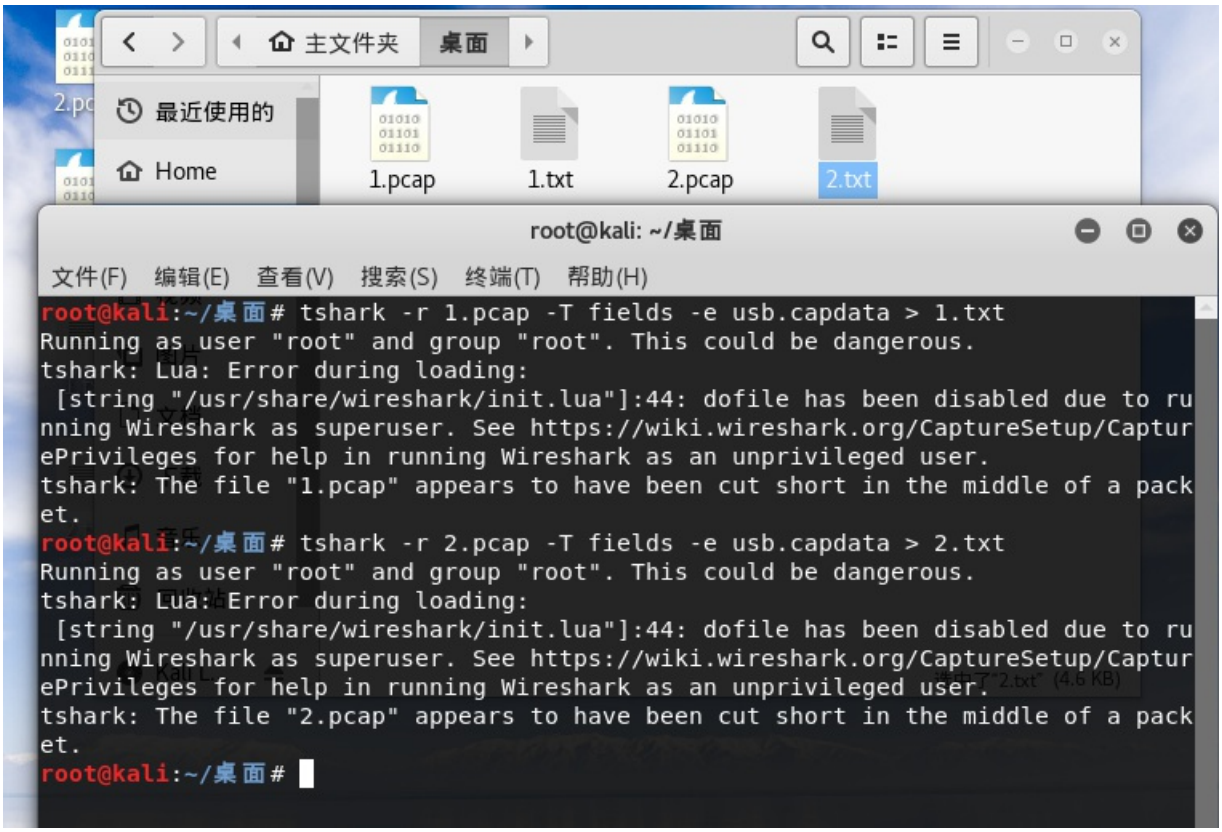


剩下的两个大包就以pcap的格式进行保存，分别为1.pcap ,2.pcap就可以了

打开linux使用命令：

tshark -r 1.pcap -T fields -e usb.capdata > 1.txt

tshark -r 2.pcap -T fields -e usb.capdata > 2.txt



参考：https://www.anquanke.com/post/id/85218

直接脚本将其中一个txt跑出来为：

`flag[u5b-key`Tips里提醒了shirt问题，

则为：`flag{u5b_key`

第二包tshark提取出来都是8字节的数据，代表捕获鼠标的数据。之前提取的txt也有提示，

只用关心第一个字节，有 0x00 0x01 0x02三种，其中当取0x00时，代表没有按键、为0x01时，

代表按左键，为0x02时，代表当前按键为右键。0x00是没意义的，那么剩下鼠标的左键与右键的点击事件，

正好对应0与1，或者1与0，反正就两种情况，写个脚本提取一下，再将01序列转换成字符串就是flag的下半段了

两个合起来就会得到：flag{u5b_keybo4rd_m0use}

我还是太笨了，只能把解题思路写成这样了（文笔不好呀！！）

至于其它的题目，我会把自己会的题目解题思路写出来的

您可以考虑给博主来个小小的打赏以资鼓励，您的肯定将是我最大的动力。