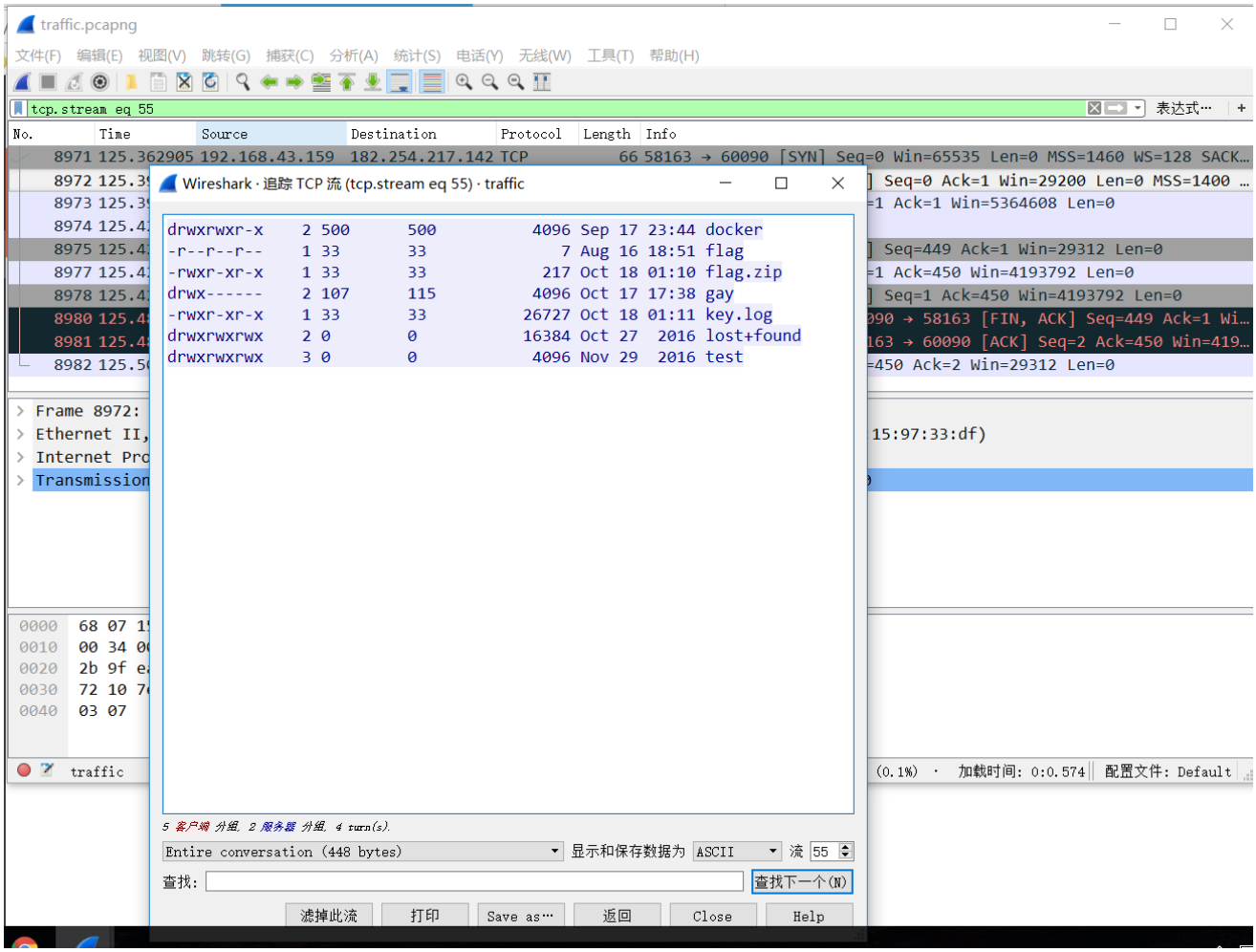
流量分析

一：下载下来题目发现是一个wirehark流量包。想都没想，扔到wireshark查看。作为一个萌新第一想法是不是在某个协议的数据流中，于是找，找啊。没有wtf。看了下题目300分呢，怎么可能呢么简单。于是继续分析。

二：分析ftp数据流。于是在 tcp.streameq 55 追踪TCP流中发现 了一些有用的东西



shujiu：

我们发现传输了名为key.log和zip文件

三：我们继续分析，在tcp流中找到一下内容

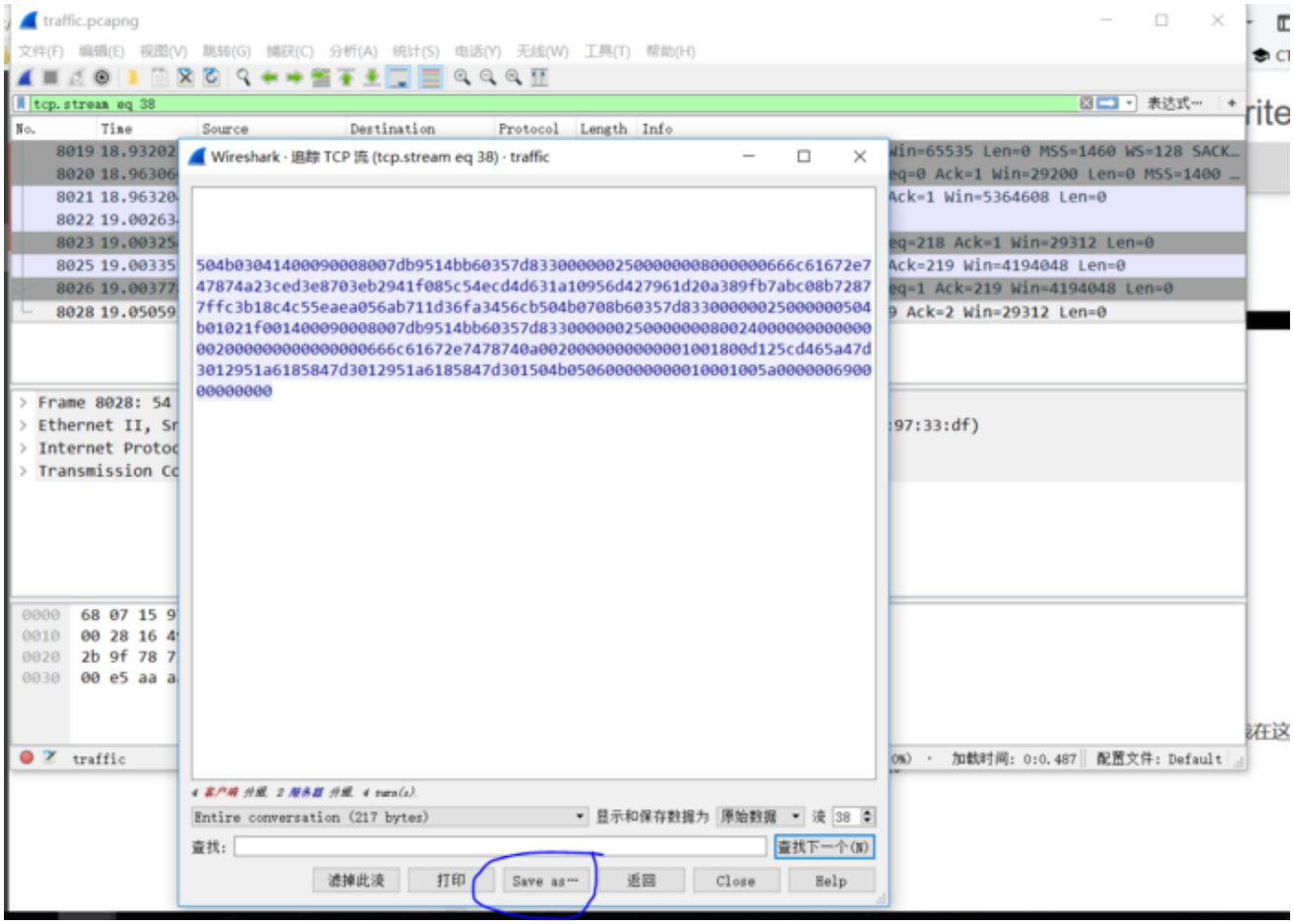
其中flag.zip两个

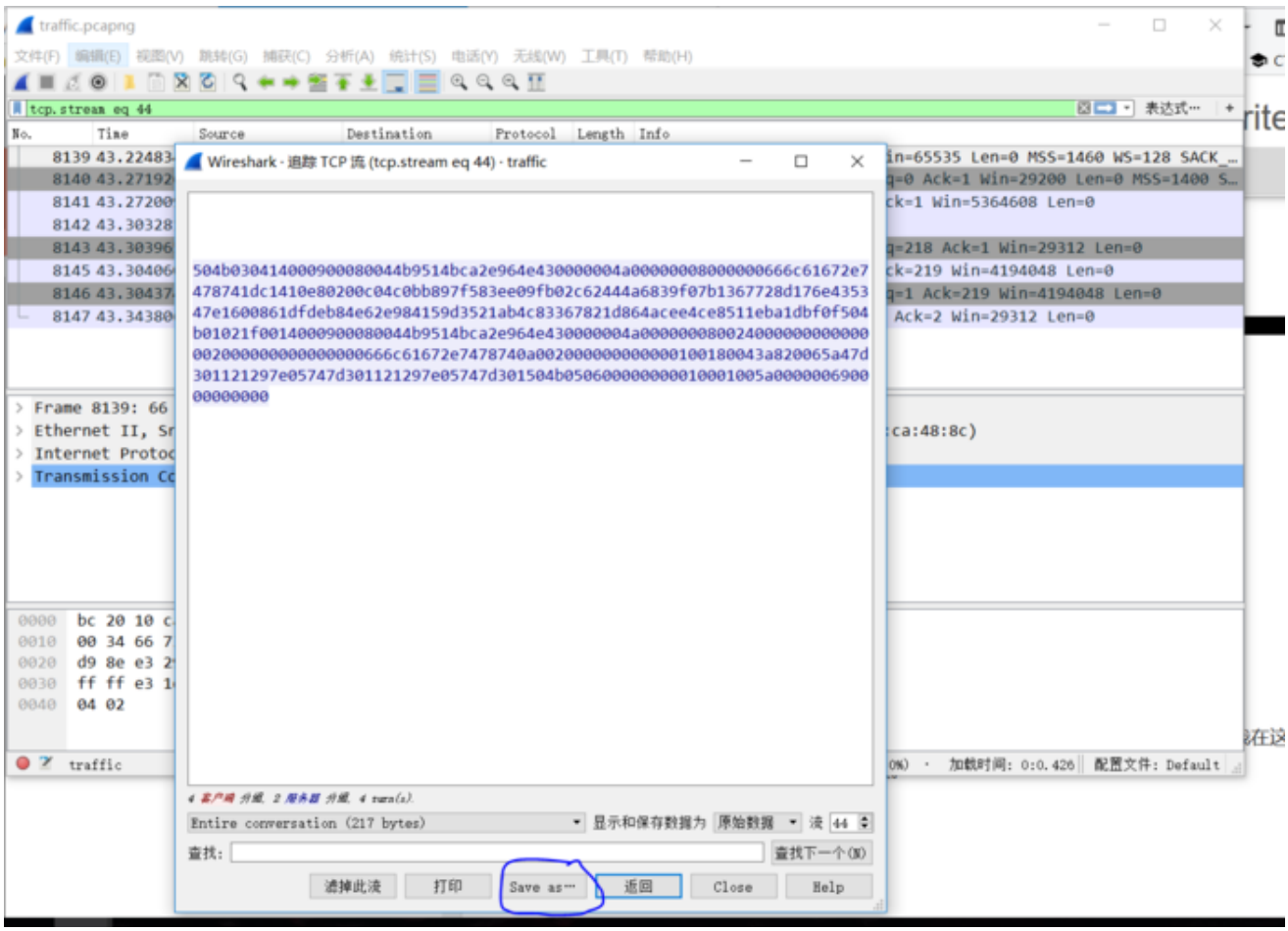
tcp.streameq38

tcp.streameq44

我们现在就可以猜出flag就在zip文件中，可是我们怎么把压缩文件拿出来啊。。萌新的我在这个地方卡了老久，百度，谷歌问大佬。终于得到导出文件的方法发。

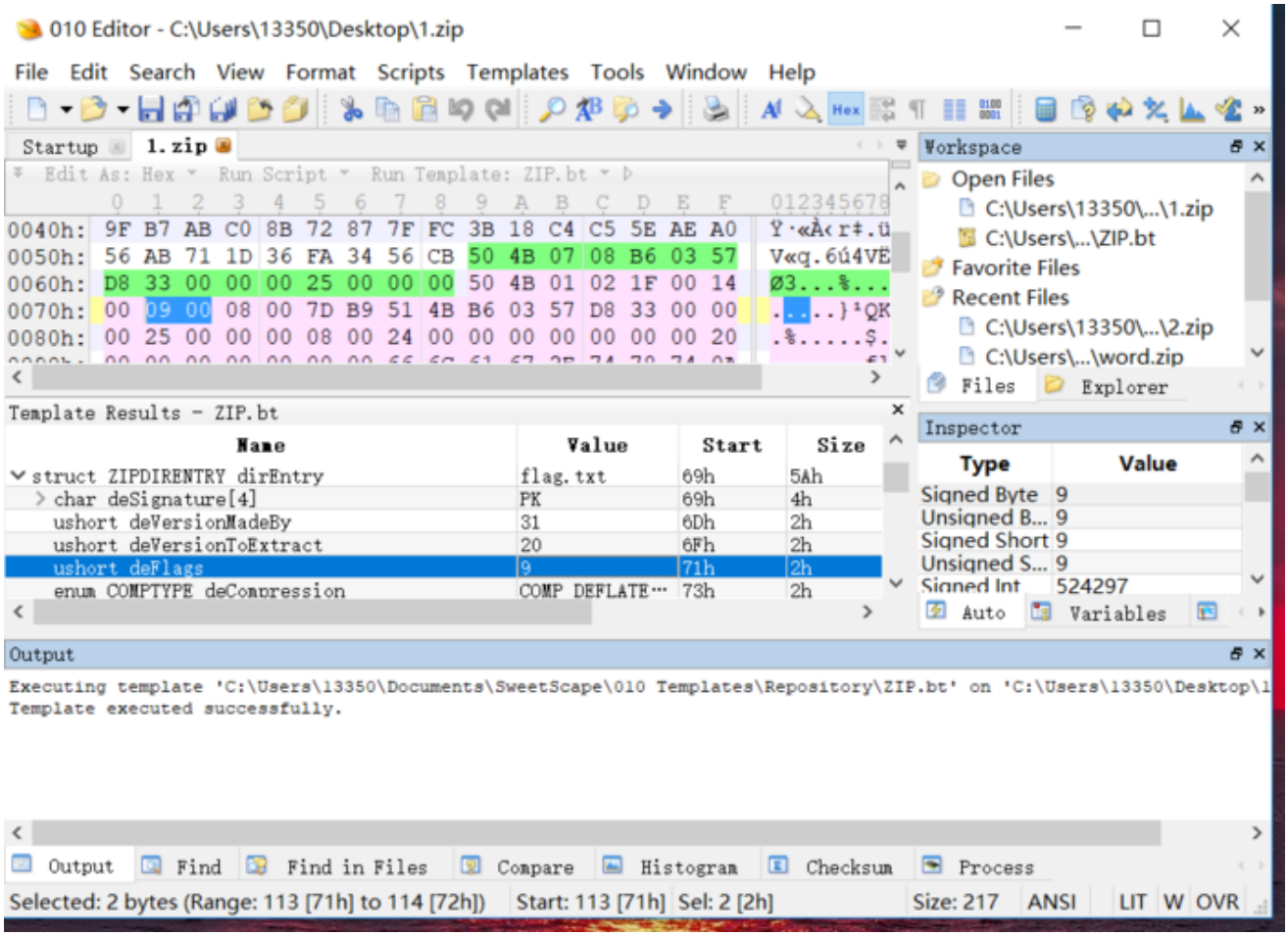
及 右键 -> 追踪流 -> TCP 流 -> Save as （注意显示和保存数据那一栏更改为原始数据） 安先后顺序分别另存为1.zip与2.zip


traffic.pcapng
tcp.stream eq 38
Wireshark · 追踪 TCP 流 (tcp.stream eq 38) · traffic
504b0304140009000800 7db9514bb60357d8330000002500000008000000666c61672e7
47874a23ced3e8703eb2941f085c54ecd4d631a10956d427961d20a389fb7abc08b7287
7ffc3b18c4c55eaea056ab711d36fa3456cb504b0708b60357d8330000002500000050 4
b01021f00140009000800 7db9514bb60357d8330000002500000008002400000000000
0020000000000000000666c61672e7478740a00200000000000001001800d125cd465a47d
3012951a6185847d3012951a6185847d301504b0506000000000100010005a0000006900
00000000
Entire conversation (217 bytes)
显示和保存数据为 原始数据
流 38
查找:
查找下一个(N)
滤掉此流
打印
Save as…
返回
Close
Help

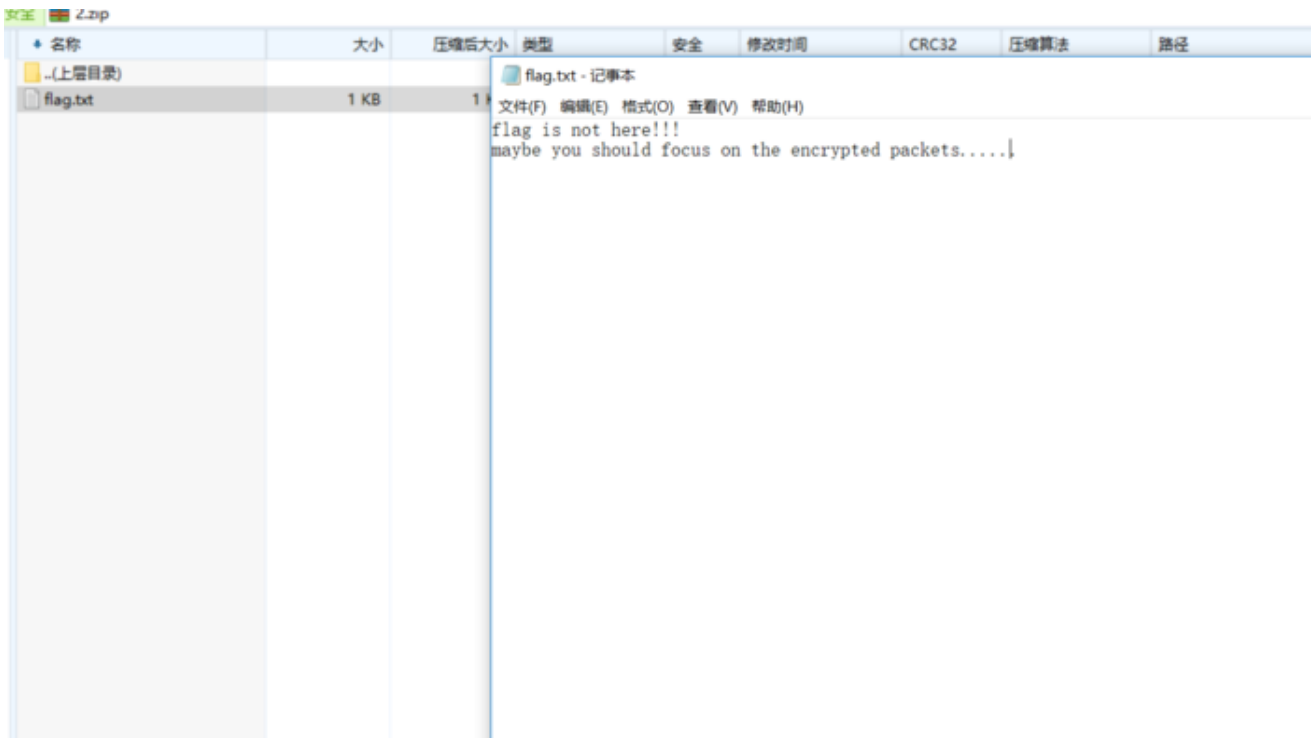

traffic.pcapng
tcp.stream eq 44
Wireshark · 追踪 TCP 流 (tcp.stream eq 44) · traffic
504b0304140009000800 44b9514bca2e964e430000004a0000000800000666c61672e7
478741dc1410e80200c04c0bb897f583ee09fb02c62444a6839f07b1367728d176e4353
47e1600861dfdeb84e62e984159d3521ab4c83367821d864acee4ce8511eba1dbf0f504
b01021f00140009000800 44b9514bca2e964e430000004a0000000800240000000000
0020000000000000000666c61672e7478740a00200000000000010018004 3a820065a47d
301121297e0747d301121297e0747d301504b0506000000000100010005a0000006900
00000000
Entire conversation (217 bytes)
显示和保存数据为 原始数据
流 44
查找:
查找下一个(N)
滤掉此流
打印
Save as…
返回
Close
Help

四：接下来我们查看两个数据包的内容 发现都是名为flag.txt的加密文件。

第一想法暴力破解拉出暴力破解工具Ziperello破解，结果貌似破解不开。经过刷题经验和大佬的口口相传，难道是所谓的伪加密于是我打算拉出工具尝试一番 010 Editor 导入压缩文件



既然是伪加密我们就将ushortdeflags的Value项的9改为0，经测试发现有一个压缩包使用的伪加密打开txt文档来看看



mmp 假的flag欲哭无泪，没办法继续做白，但另一个压缩包该镇么解密呢，

五：对了刚才我们分析到tcp流中还有一个key.log什么东西于是百度谷歌

百度答：能发现这是一份NSS Key Log Format的文件，而这个文件是能解密出 Wireshark 里面的 https 流量的。

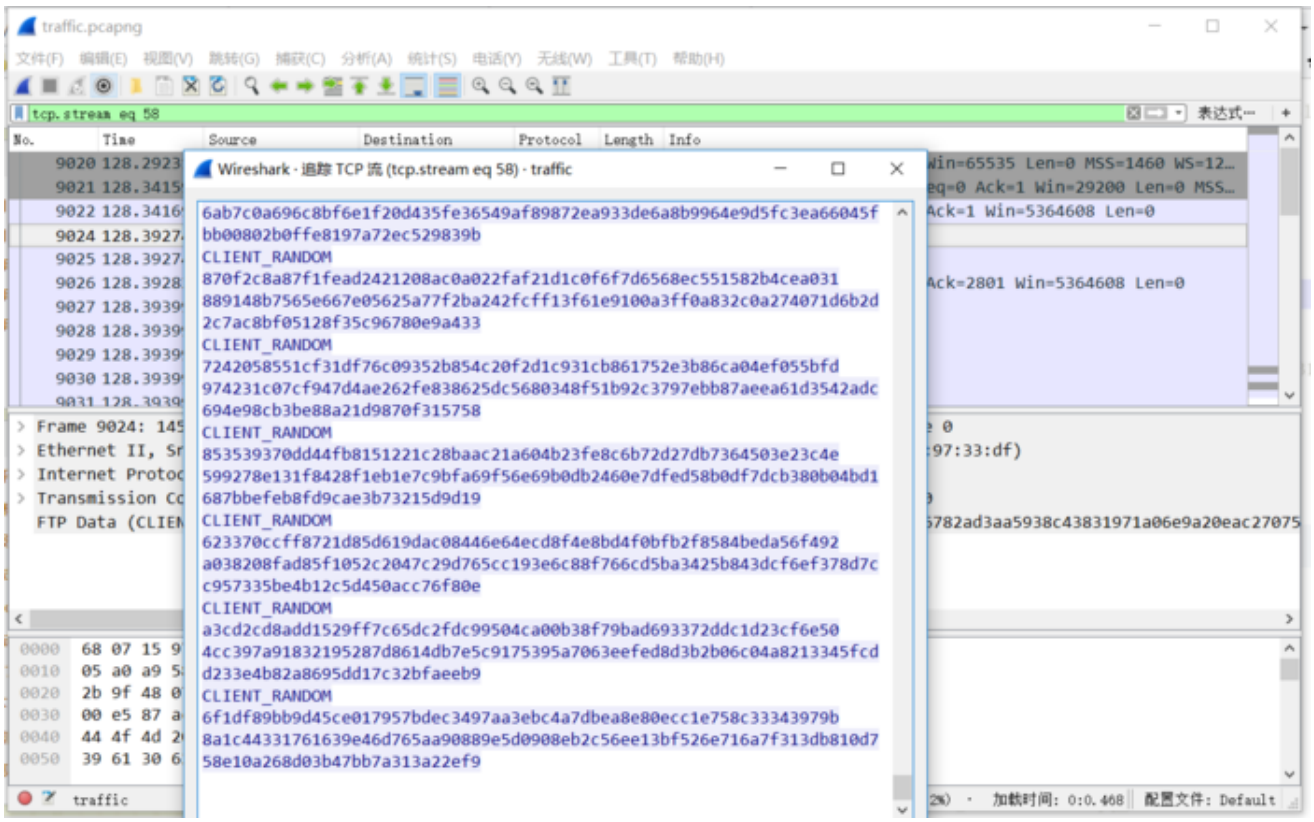
Firefox、Chrome可以通过设置SSLKEYLOGFILE环境变量导出所有的会话密钥，估计是为了方便调试。Wireshark可以通过这种格式的密钥来解密。

资料参考：

NSS Key Log Format - Mozilla | MDN

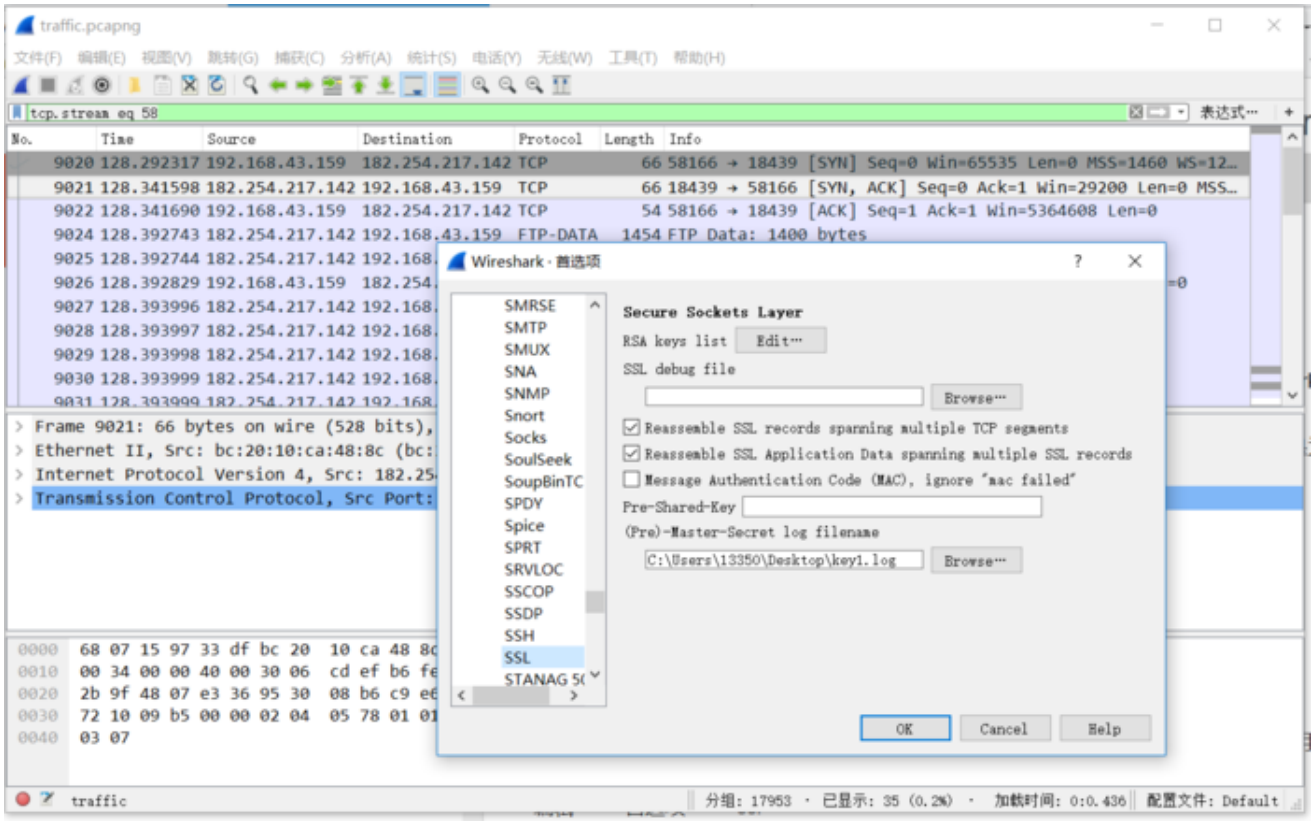
27.3 如何用Wireshark解密HTTPS报文

我们把 tcp.stream eq 58 右键 -> 追踪流 -> TCP 流 -> Save as 导出为key.log

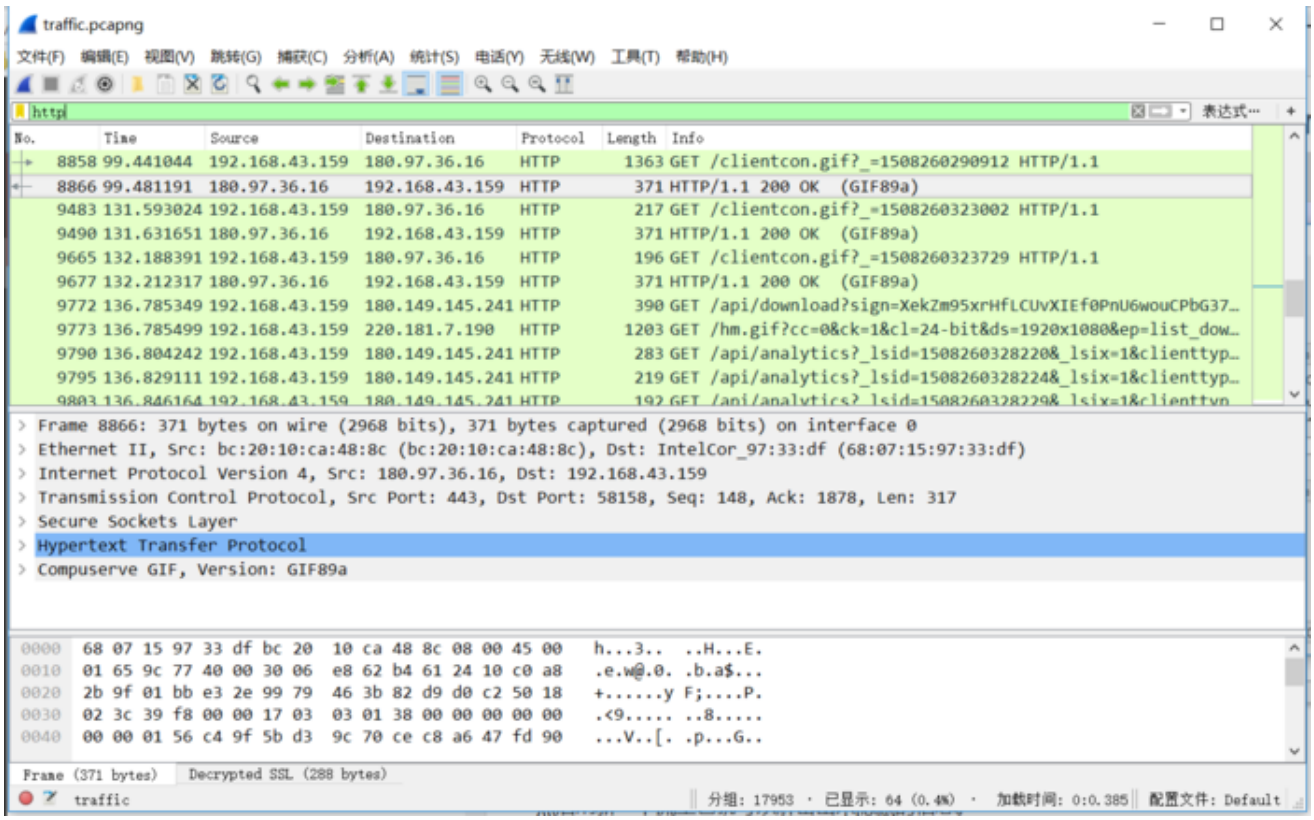


再导入密钥

编辑——>首选项——>ssl

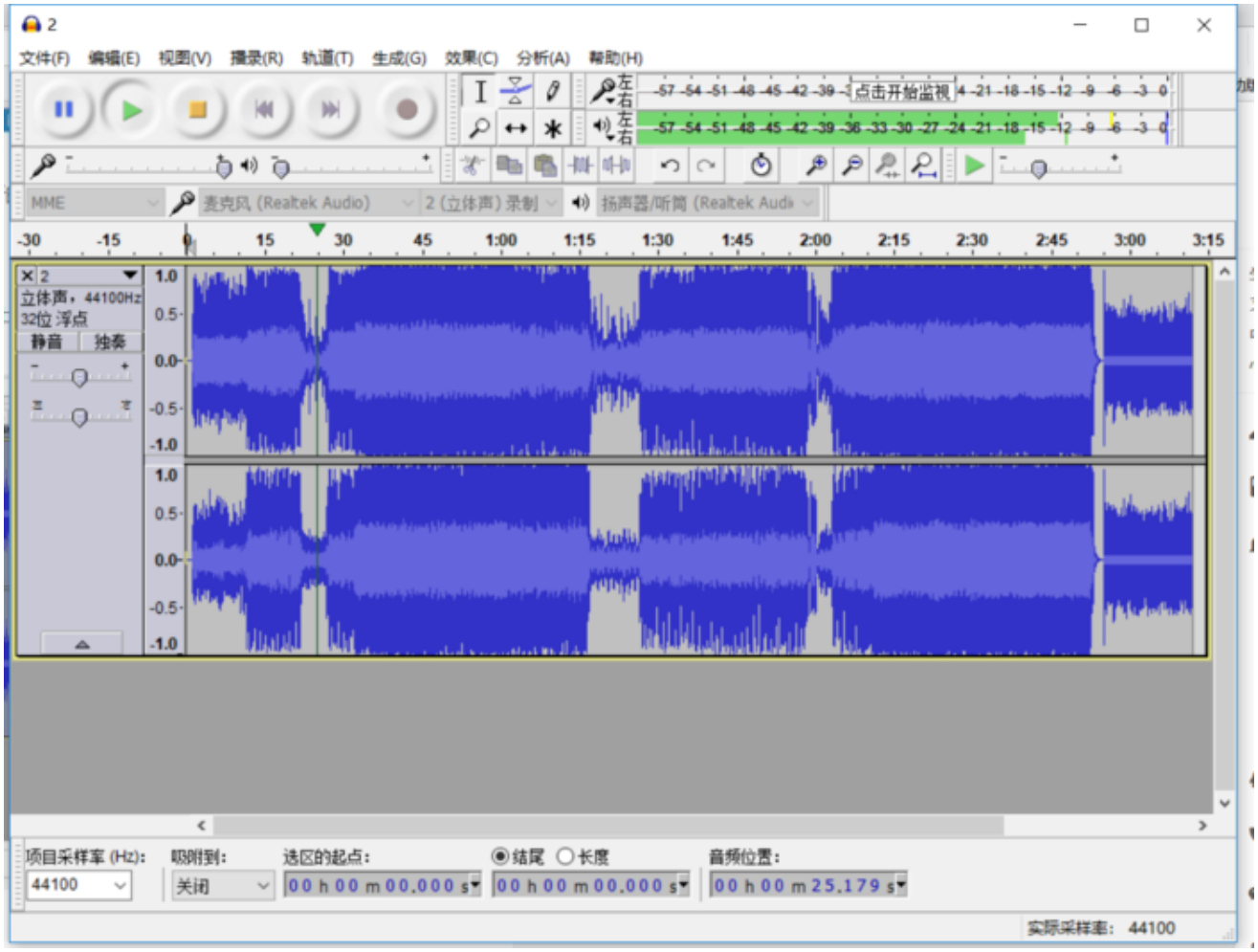


然后刷新一下流量包就可以解密出来隐藏的信息。

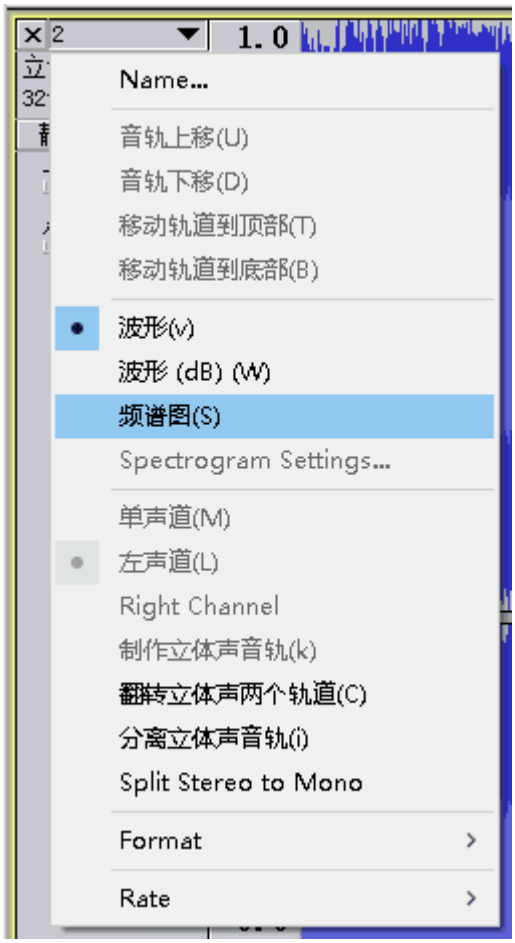


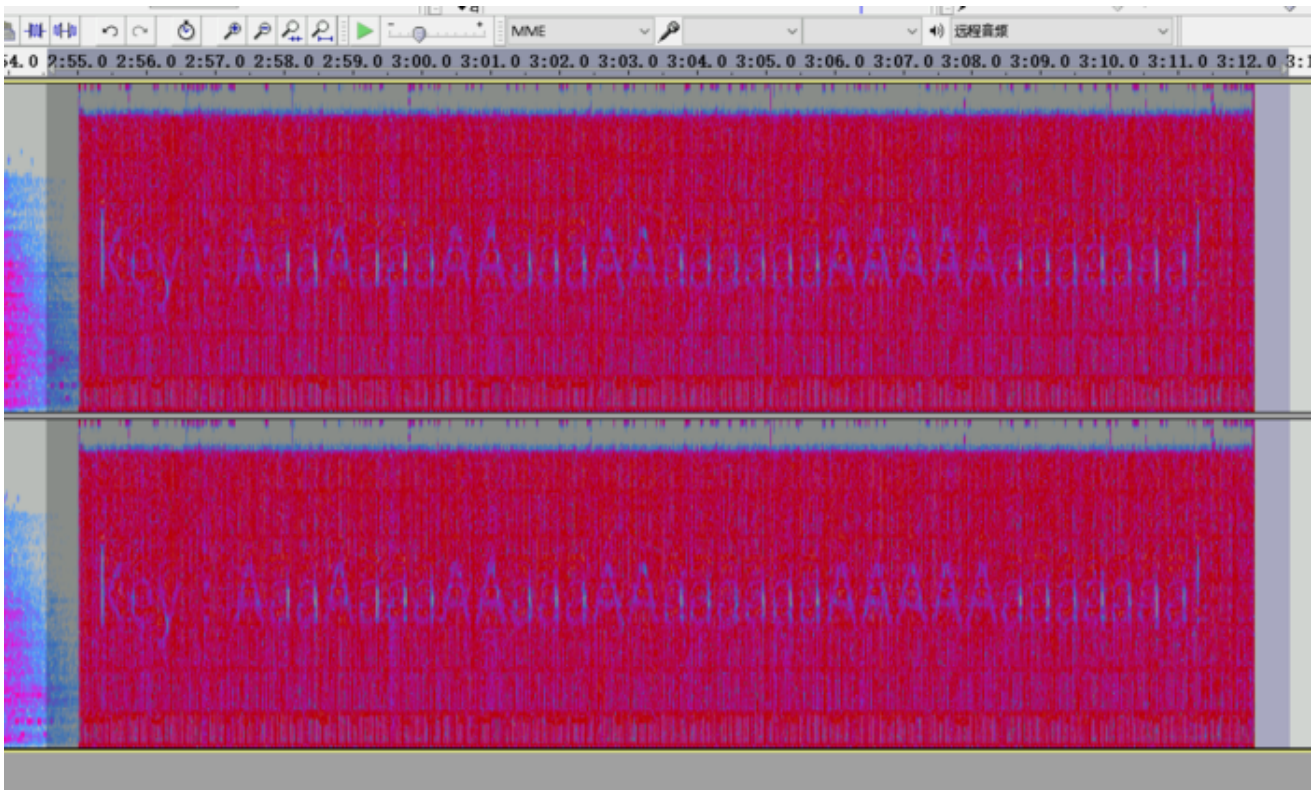
瞄了一眼大概就是上了百度的网盘下载了个东西，直接通过导出对象 -> HTTP进行导出

解压之后发现是一首歌，用Audacity打开挺好听的，听到了最后剩下了莫名其妙的杂音，数据应该就在这个部分了。


2
文件(F) 编辑(E) 视图(V) 播录(R) 轨道(T) 生成(G) 效果(C) 分析(A) 帮助(H)
点击开始监视
MME
麦克风 (Realtek Audio)
2 (立体声) 录制
扬声器/听筒 (Realtek Audi
立体声，44100Hz
32位 浮点
静音
独奏
项目采样率 (Hz):
44100
吸附到:
关闭
选区的起点:
00 h 00 m 00.000 s
结尾
长度
00 h 00 m 00.000 s
音频位置:
00 h 00 m 25.179 s
实际采样率: 44100

切换到频谱图分析





隐约能看到写着

Key：AaaAaaaAAaaaAAaaaaaaAAAAAaaaaaaa!

key及上述中没有运用伪加密的压缩文件密码解压文件打开txt及为flag

名称	大小	压缩后大小	类型	安全	修改时间	CRC32	压缩算法
..(上层目录)							
2.mp3	7.33 MB	6.53 MB	爱奇艺多媒体文件	安全	2017-10-17 22:07:...	59C49A8C	Deflate