

# 20307030135李钧实验2

---

## 一、实验目的

## 二、实验过程

### (一) webmail

抽取http 或https内容信息，提取图片

提取两邮箱之间互发邮件的基本信息

获得附件信息，还原文件

### (二) ftpexample

## 三、遇到的问题与总结

20307130135 李钧

## 一、实验目的

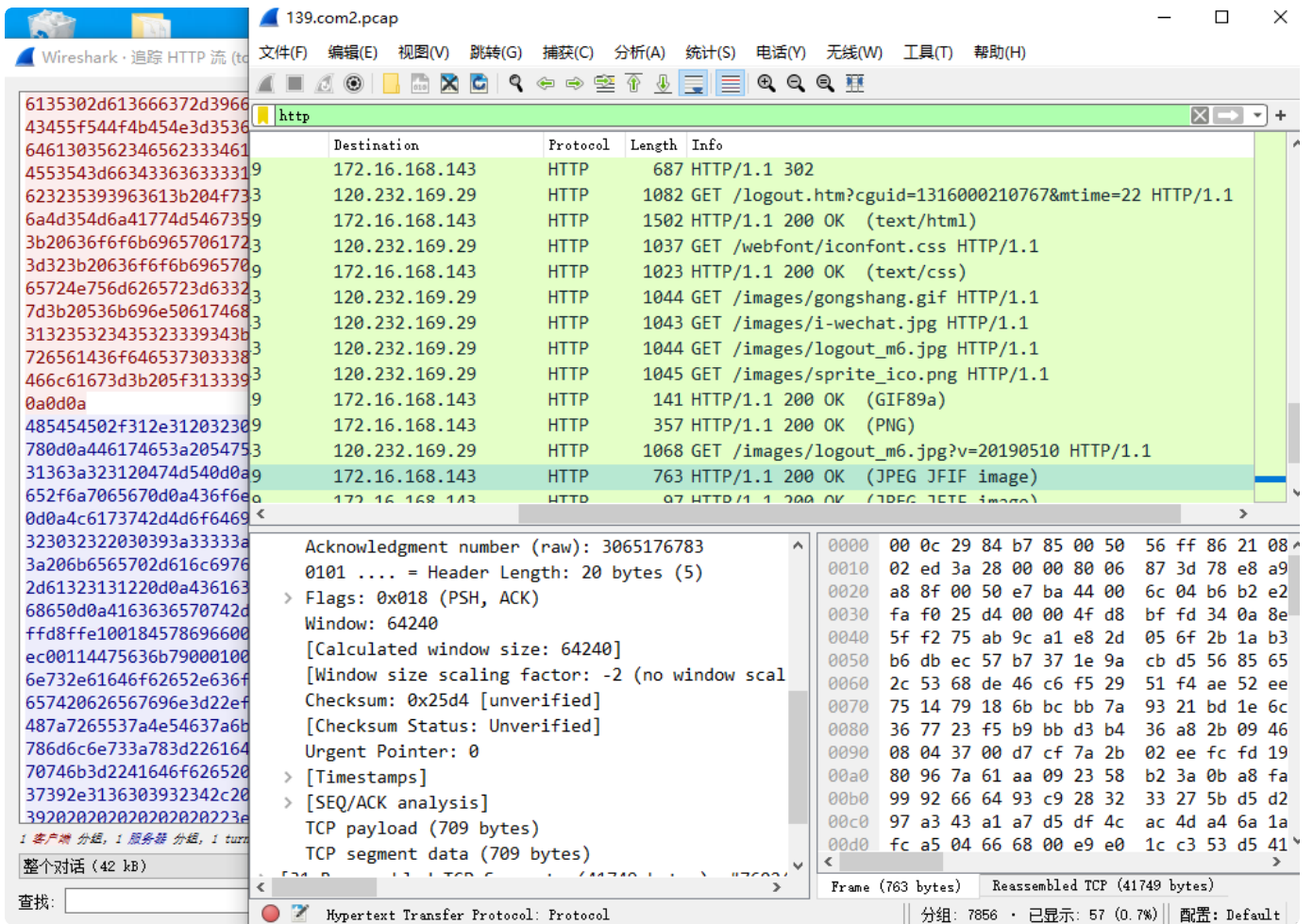
1. 通过嗅探器抽取http或https内容信息（从mail.139.com网站上提取图片）
2. 获取抓包下来两封邮件的基本信息
3. 获取邮件的附件信息，还原附件
4. 还原ftp操作过程记录中的上传文件、下载文件

## 二、实验过程

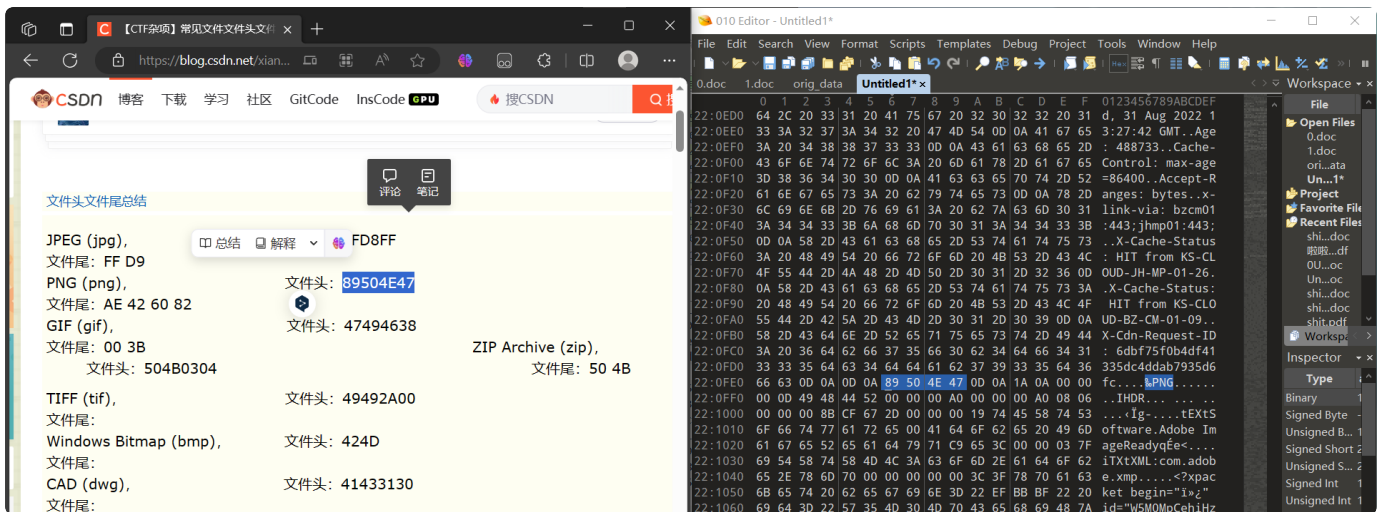
### (一) webmail

抽取http 或https内容信息，提取图片

通过wireshark打开139.com2.pcap文件，在过滤器当中输入http进行过滤，在Info栏中检索和图片有关的信息，如PNG、JPEG，通过右击选择“追踪流”-“HTTP流”-“原始数据”，将追踪到的数据以原数据格式复制到剪切板，转移到010 editor中操作。



在010 editor软件工具栏的“Edit”选项中以十六进制(Hex)方式粘贴上述复制的数据，并在视图(View)中选择“edit as Hex”，我们得到十六进制格式的数据。查阅PNG照片的十六进制文件头，在010 editor中通过“Ctrl + F”定位该文件头的位置，将文件头之前的所有数据删除。除此之外还可以采用相同方法将文件尾之后的数据删除，接着将文件“另存为img1.png”，即导出文件为png格式的图片，双击打开保存的图片文件就是题中要求提取的一张图片。





## 体验139邮箱小程序

除了上述采用010 editor的方法，在wireshark软件中定位到该图片数据包，直接选择“导出分组字节流”，保存文件名后缀".png"也可达到提取图片的效果。

## 提取两邮箱之间互发邮件的基本信息

在继续实验内容之前，需要导入keylog文件以得到解密过的传输数据。如下所示，通过“编辑”-“首选项”-“protocol”-“TLS”将keylog文件导入到wireshark当中。

139.com2.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
1	0.000000	172.16.168.143	222.192.186.36	TCP	55	59229 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=1
2	0.000504	222.192.186.36	172.16.168.143	TCP	60	80 → 59229 [ACK] Seq=1 Ack=2 Win=64240 Len=0
3	4.648445	172.16.168.143	117.18.237.29	HTTP	55	Continuation
4	4.657066	117.18.237.29	172.16.168.143	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
5	5.472882	172.16.168.143	34.107.221.82	HTTP	55	Continuation
6	5.473471	34.107.221.82	172.16.168.143	TCP	60	80 → 59218 [ACK] Seq=1 Ack=2 Win=64240 Len=0
7	5.740108	172.16.168.143	172.16.168.2	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
8	5.759653	172.16.168.2	172.16.168.143	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
9	5.764970	172.16.168.143	182.61.200.7	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
10	5.768926	172.16.168.143	172.16.168.2	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
11	5.774640	172.16.168.2	172.16.168.143	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
12	5.776501	172.16.168.143	172.16.168.2	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
13	5.782870	172.16.168.2	172.16.168.143	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
14	5.796612	182.61.200.7	172.16.168.143	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
15	5.796763	172.16.168.143	182.61.200.7	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
16	5.799580	172.16.168.143	182.61.200.7	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
17	5.804512	182.61.200.7	172.16.168.143	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
18	5.833883	182.61.200.7	172.16.168.143	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
19	5.833883	182.61.200.7	172.16.168.143	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
20	5.833883	182.61.200.7	172.16.168.143	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
21	5.833883	182.61.200.7	172.16.168.143	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
22	5.834100	172.16.168.143	182.61.200.7	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0
23	5.852815	172.16.168.143	182.61.200.7	TCP	60	80 → 59222 [ACK] Seq=1 Ack=2 Win=64240 Len=0

Wireshark - 首选项

Transport Layer Security

RSA keys list Edit...

TLS debug file

Reassemble TLS records spanning multiple TCP segments

Reassemble TLS Application Data spanning multiple TLS records

Message Authentication Code (MAC), ignore "mac failed"

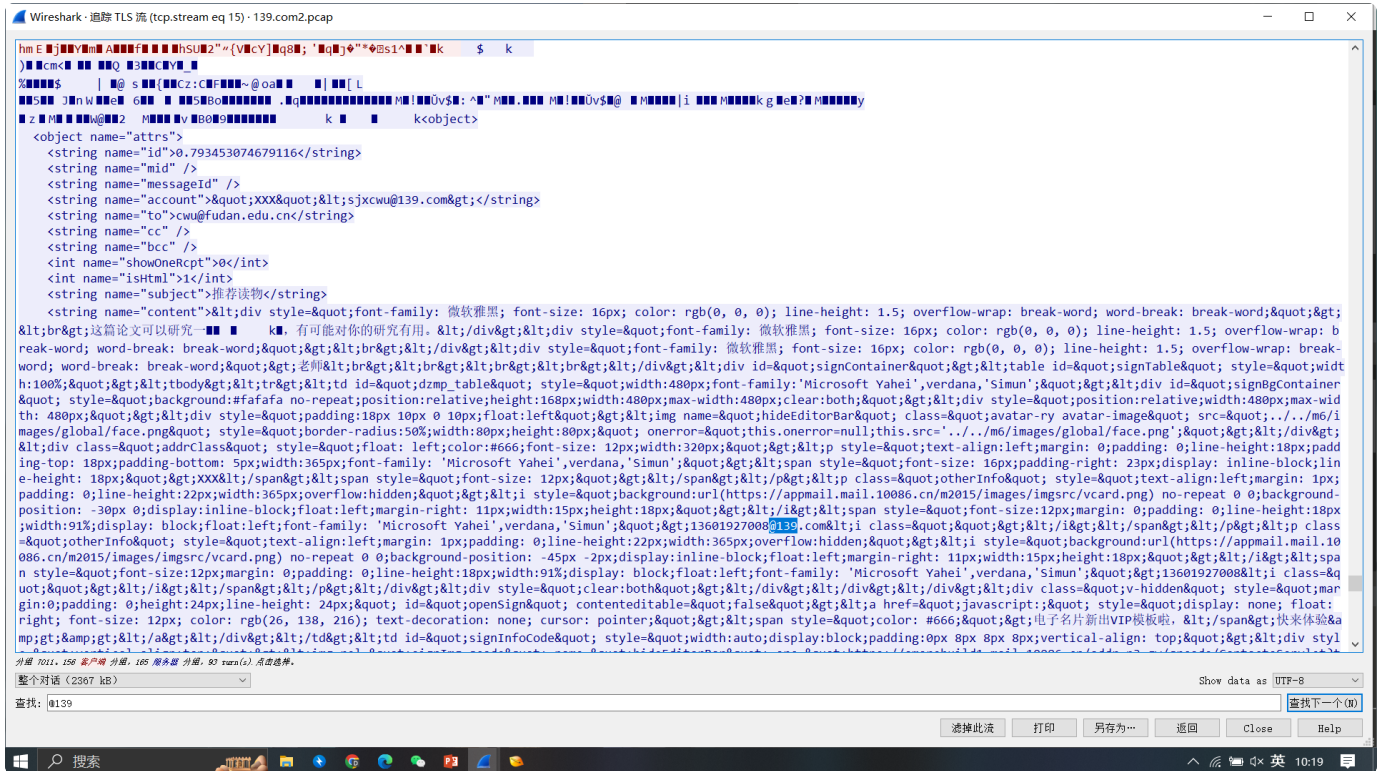
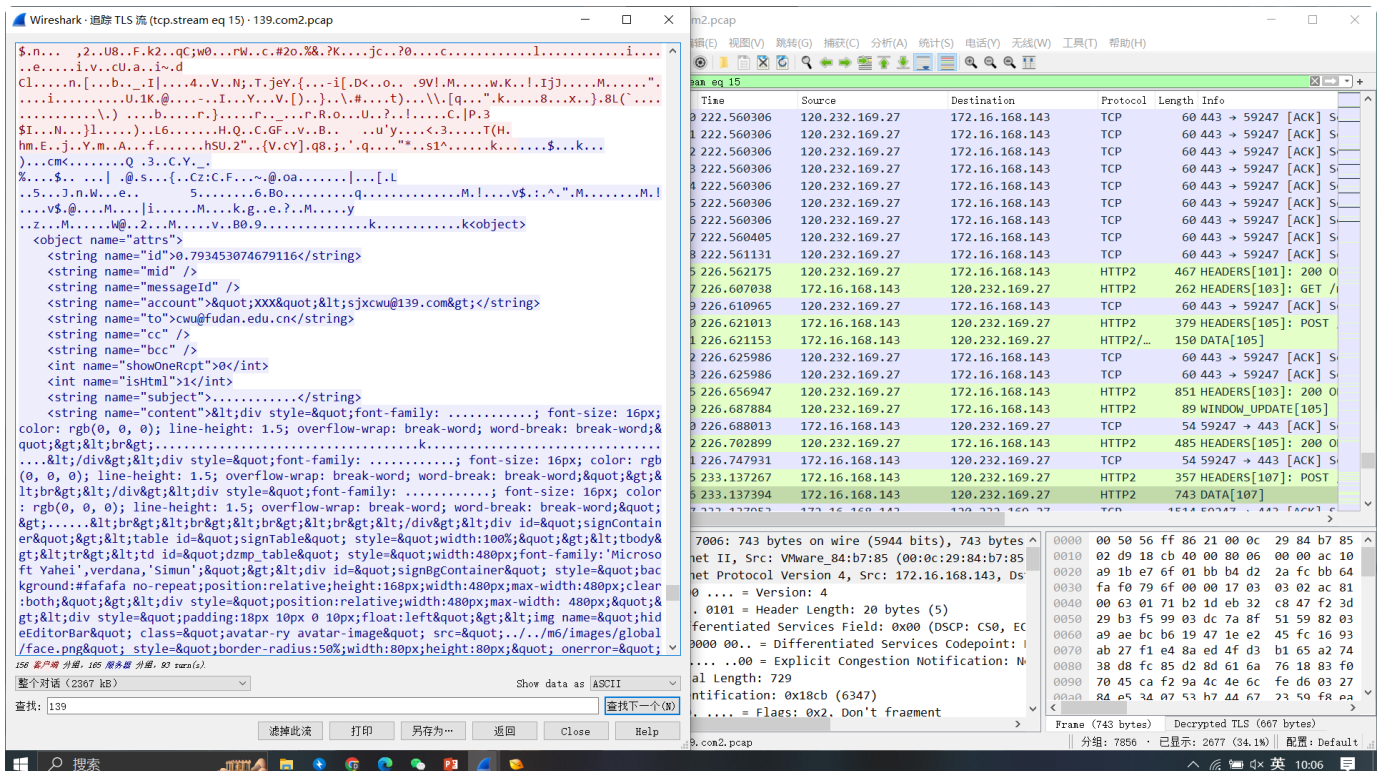
Pre-Shared Key

(Pre)-Master-Secret log filename

j:\webmail\sslkeylog.for139com2.txt

OK Cancel Help

通过过滤器搜索字符串".pdf"并追踪流，在追踪的TLS流中搜索“139”关键词定位到邮件发送方、接收方的位置，由于在流当中数据默认的格式为ASCII码，我们能看到发送方为"sjxcwu@139.com"，接收方为"cwu@fudan.edu.cn"，邮件标题和正文被编码。切换编码格式即可得到邮件标题和正文。

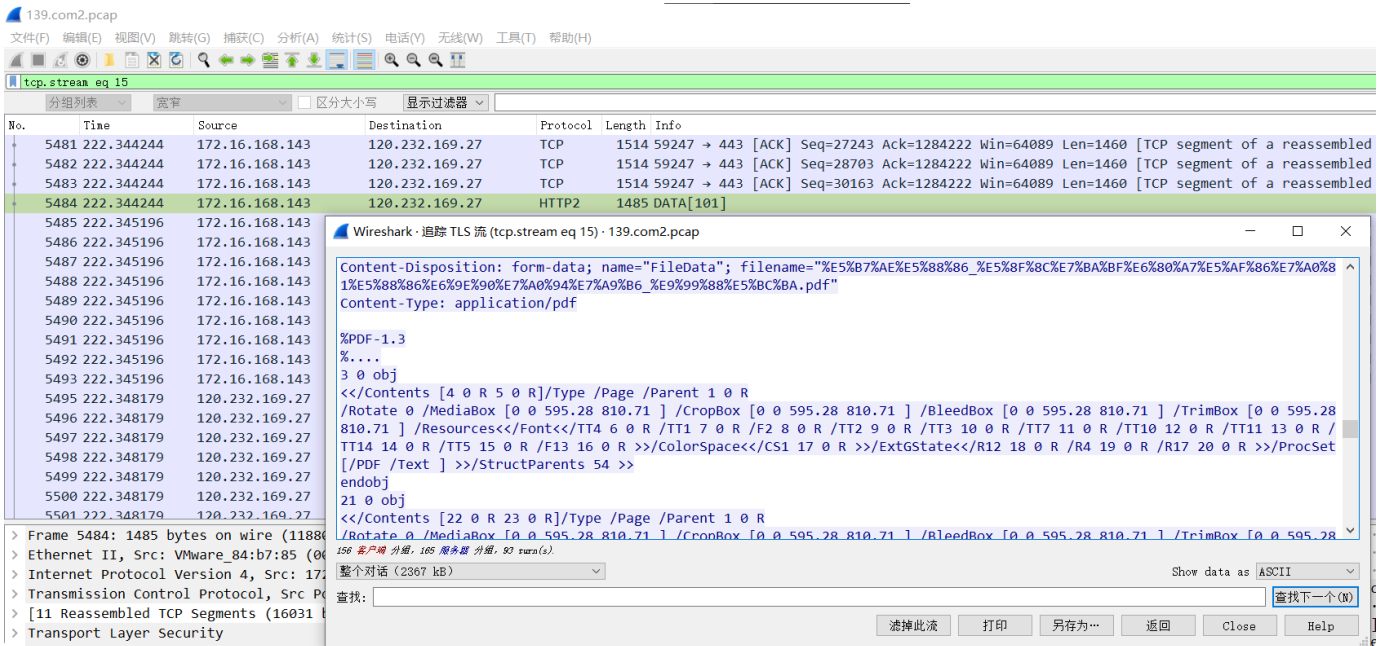


## 获得附件信息，还原文件

在上述操作定位到邮件附件的基础上，点击追踪TLS流中的PDF部分，可以观察到PDF对应的内容在DATA [101] 中，再通过追踪HTTP流、找到最终接收的完整PDF文件DATA数组，导出分组数据流并更改文件后缀为pdf即可得到要求的PDF文件。对于PDF文件名，在附件上方的数据流中可以观察



到“%..%.pdf”样式的字符串，将其复制到网络上解码（或在wireshark数据流追踪中通过更改数据编码类型）得到PDF的文件名。导出doc文件、找出doc文件名的操作过程和PDF类似，不再赘述。



差分\_双线性密码分析研究\_陈强.pdf

字符编码: UTF-8

Url 编码 ↓

Url 解码 ↑

%E5%B7%AE%E5%88%86\_%E5%8F%8C%E7%BA%BF%E6%80%A7%E5%AF%86%E7%A0%81%E5%88%86%E6%9E%90%E7%A0%94%E7%A9%B6\_%E9%99%88%E5%BC%BA.pdf



```

200 Type set to I.
PASV
227 Entering Passive Mode (172,16,168,21,192,27).
SIZE 伪DNA加密算法.pptx
213 3232128
RETR 伪DNA加密算法.pptx
150 Opening data connection for 伪DNA加密算法.pptx.
226 File sent ok

```

```

PASV
227 Entering Passive Mode (172,16,168,21,192,29).
STOR 宋代字验.docx
150 Opening data connection for 宋代字验.docx.
226 File received ok

```

31 客户端 分组, 36 服务器 分组, 62 turn(s).

整个对话 (1582 bytes)

Show data as GBK

由于ftp传输需要控制连接和数据连接的配合才能完成，控制连接中有相关数据连接的端口信息，当客户端与FTP服务器建立控制连接后，客户端会发送PORT命令或者PASV命令来建立数据连接。PORT命令——客户端发送PORT命令指定自己的数据连接端口，指令格式——PORT h1,h2,h3,h4,p1,p2——其中，h1.h2.h3.h4表示客户端的IP地址，而p1\*256+p2则表示客户端的端口号。PASV命令：客户端发送PASV命令请求服务器在一个特定的端口上打开数据连接，并等待服务器的响应。服务器会返回包含服务器的IP地址和端口号的响应消息。

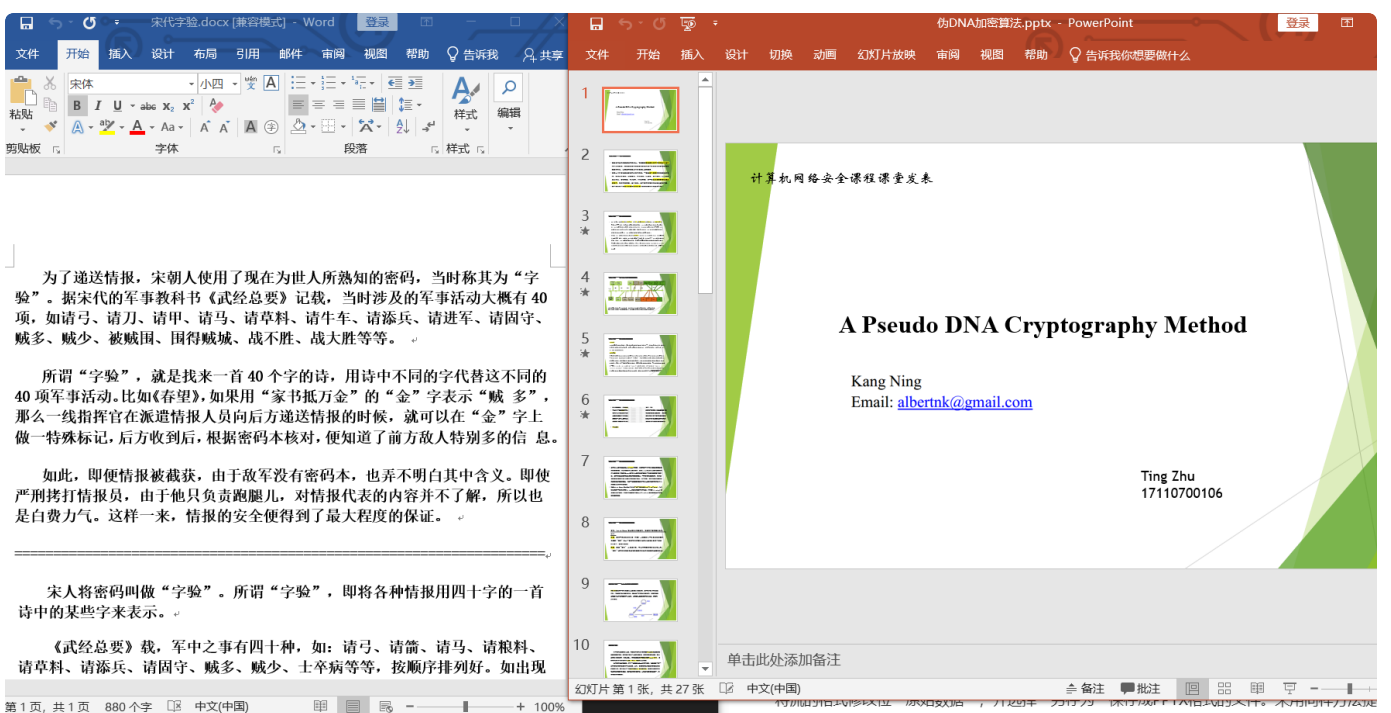
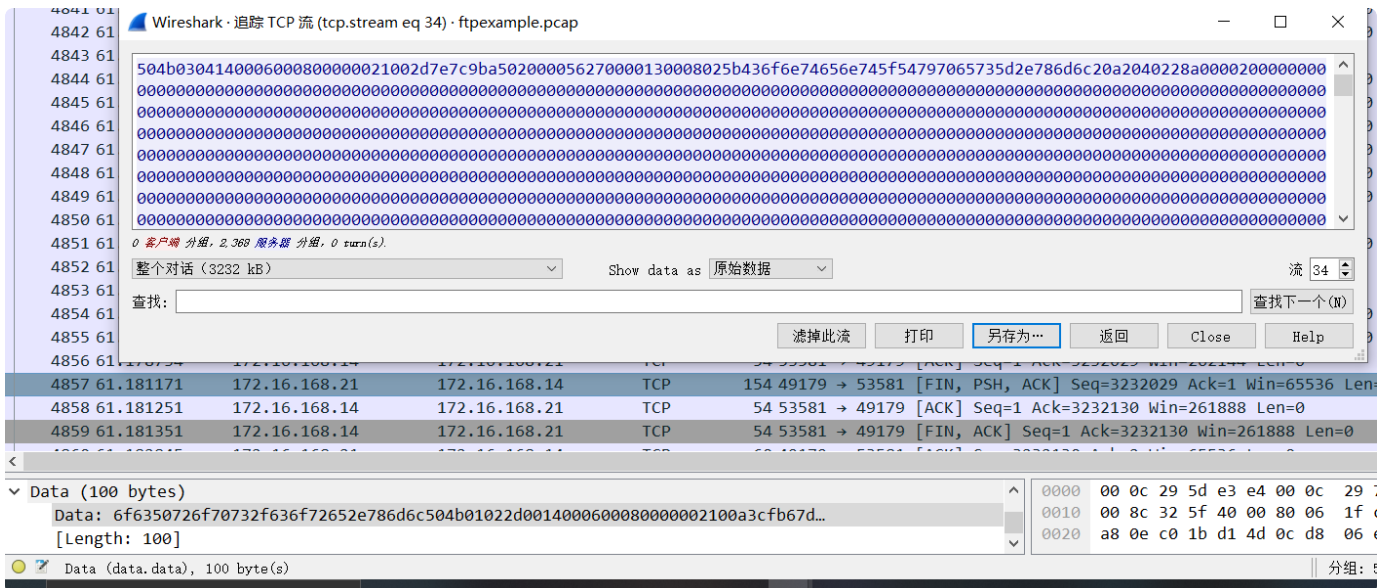
知晓了FTP的相关知识之后便可计算出两个文件的传输端口分别为49179（PPTX）和49181（DOCX），于是可以在 wireshark过滤器中指定TCP传输的端口进行第一遍过滤，接着找到某一条传输数据的信息右击追踪TCP流，找到最下方包含全部数据包的一条记录，右击追踪TCO流，将流的格式修改位“原始数据”，并选择“另存为”保存成PPTX格式的文件。采用同样方法提取出DOCX文件，提取出的结果如下所示。

ftpexample.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)

tcp.port == 49179

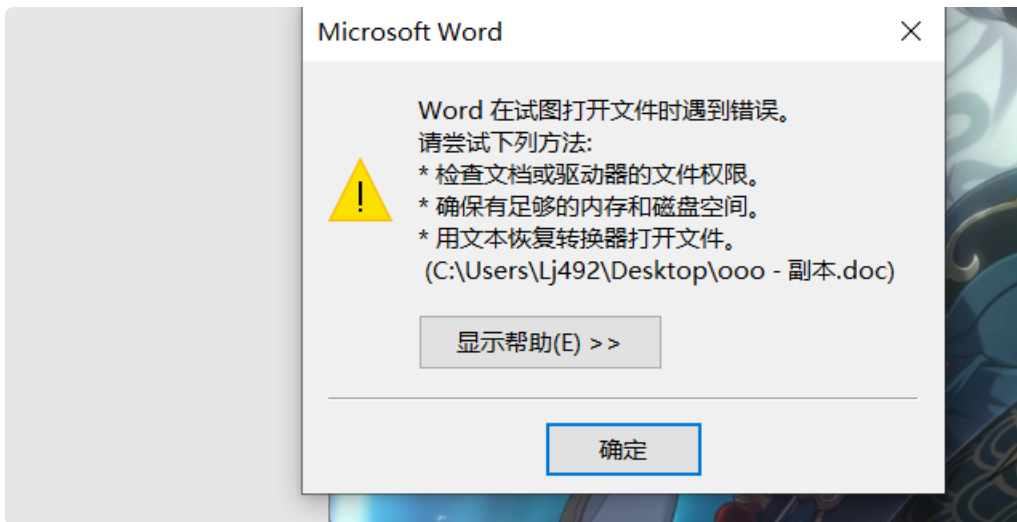
No.	Time	Source	Destination	Protocol	Length	Info
2041	60.664902	172.16.168.14	172.16.168.21	TCP	66	53581 → 49179 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
2042	60.666397	172.16.168.21	172.16.168.14	TCP	66	49179 → 53581 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
2043	60.666667	172.16.168.14	172.16.168.21	TCP	54	53581 → 49179 [ACK] Seq=1 Ack=1 Win=262144 Len=0
2050	60.672120	172.16.168.21	172.16.168.14	TCP	1514	49179 → 53581 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=1460



### 三、遇到的问题与总结

在本次实验的过程中，我在提取附件、提取文件的时候遇到问题——如何将原始数据导出成我们需要的格式文件。在实验开头提取图片时，我采用复制原始数据到010 editor中进行编辑，根据PNG图片文件头、文件尾十六进制的特征截取数据并保存的方法。本想效仿此方法完成PDF文件和DOC文件的提取，但在根据文件头截取的时候发现生成的PDF文件是乱码、生成的DOC文件无法打开，具体报错如下图所示：





为尝试解决该问题，我创建空的doc文件、含有少量内容的doc文件并通过010editor打开查看十六进制数据，并根据文件头、文件尾进行数据的截取，仍得不到正确结果。在回顾该问题时，我认为这是追踪TLS流导致的，PDF和DOC文件的部分关键信息在该流中缺失。