

復旦大學

数据通信与计算机网络 Lab2:应用层

王傲

15300240004

数据通信与计算机网络

COMP130017. 01

指导教师：肖晓春

目录:

0. 目录	2
1. HTTP 与 DNS 协议分析	3
1. 1	3
1. 2	7
1. 3	8
1. 4	9
2. FTP 协议分析	11
控制通道	11
2. 1. 1	11
2. 1. 2	12
2. 1. 3	12
2. 1. 4	13
数据通道	14
2. 2. 1	14
2. 2. 2	14
2. 2. 3	14
2. 2. 4	15
2. 2. 5	15
3. SMTP 与 POP3 协议分析	16
3. 1	16
3. 2	19
3. 3	21
3. 4	21
4. 讨论与研究	23
5. 参考资料	26

1. HTTP 与 DNS 协议分析:

实验步骤:

实验平台为 Macbook Pro (Mid 2015)，使用 Wireshark 版本为 2.4.1。在使用 sudo dscacheutil -flushcache 指令清空 DNS 缓存后，在浏览器输入 http://www.baidu.com/，打开 Wireshark 开始监听，并在浏览器输入回车。捕获的分组为 HTTP_DNS.pcapng 和 HTTP_GET.pcapng。

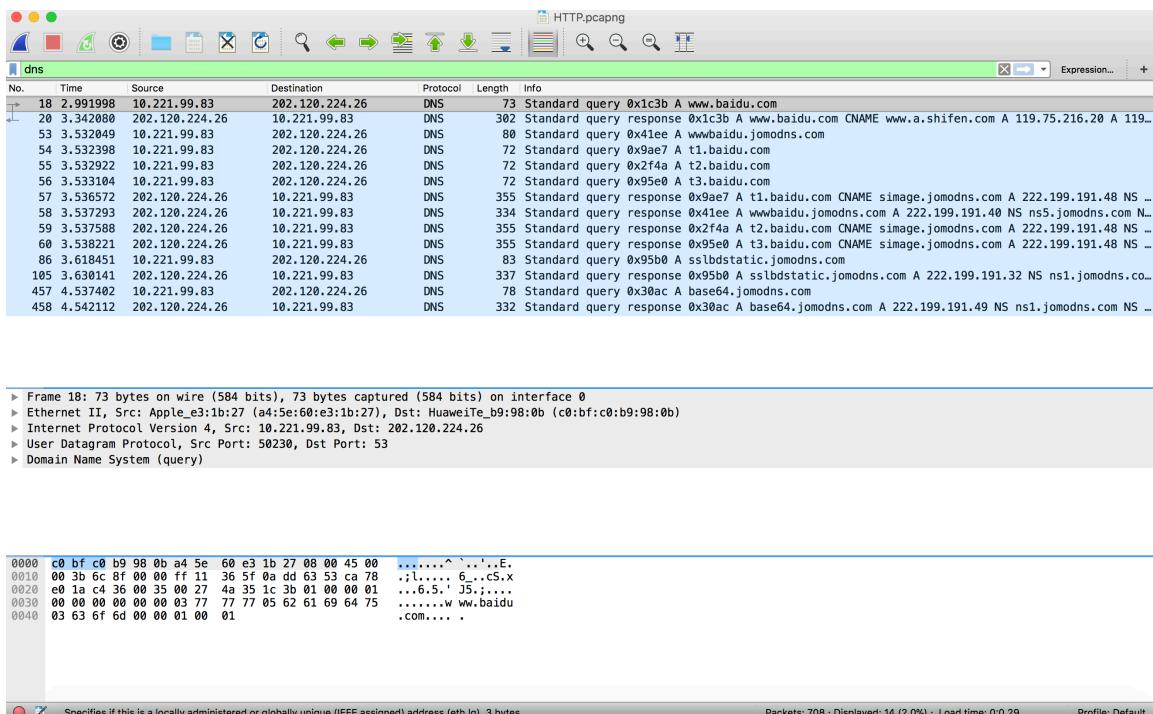
本机 IP 地址为 10.221.99.83。

实验分析:

1.1 分析发送到域名服务器的请求分组及响应分组，将相关信息截图下来，并指出

<http://www.baidu.com> 的 IP 地址

在对捕获的分组加入“dns”的 filter 后，结果如下:



从 Source 和 Destination 信息可以看出，本机的 IP 地址为 10.221.99.83，本地 DNS 服务器的 IP 地址为 202.120.224.26。

在终端查询本机的 IP 地址，如下，可见 IP 地址是正确的。

```
[~] ~ ifconfig | grep "inet" | grep -v 127.0.0.1
      inet 10.221.99.83 netmask 0xfffffc000 broadcast 10.221.127.255
```

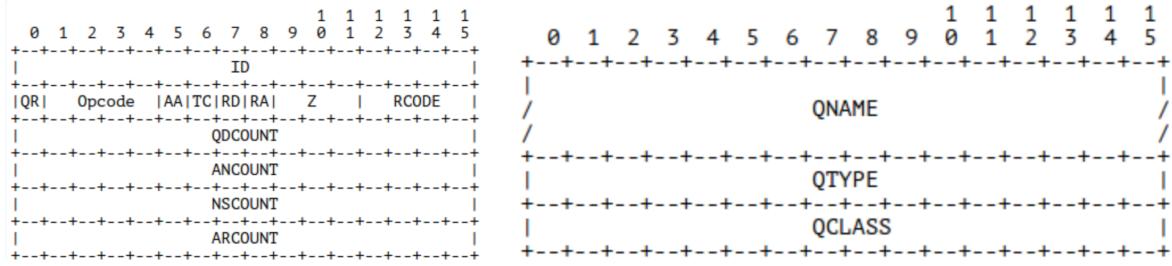
第一个分组是发送给域名服务器请求 www.baidu.com 的 IP 地址的。分组内信息如下:

```

> Frame 18: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
> Ethernet II, Src: Apple_e3:1b:27 (a4:5e:60:e3:1b:27), Dst: HuaweiTe_b9:98:0b (c0:bf:c0:b9:98:0b)
> Internet Protocol Version 4, Src: 10.221.99.83, Dst: 202.120.224.26
> User Datagram Protocol, Src Port: 50230, Dst Port: 53
▼ Domain Name System (query)
  [Response In: 20]
  Transaction ID: 0x1c3b
  ▼ Flags: 0x0100 Standard query
    0... .... .... = Response: Message is a query
    .000 0... .... = Opcode: Standard query (0)
    .... .0. .... = Truncated: Message is not truncated
    .... ..1 .... = Recursion desired: Do query recursively
    .... ..0. .... = Z: reserved (0)
    .... .... ..0 .... = Non-authenticated data: Unacceptable
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  ▼ Queries
    ▼ www.baidu.com: type A, class IN
      Name: www.baidu.com
      [Name Length: 13]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)

```

DNS 报文的 Header 和 Question 结构如下：



逐个分析分组内的信息：首先，Transaction ID（16位，由请求程序生成）标明了这次事务的编号（可以与 response 信息进行对应，用于区分 DNS 请求）。在 Header 栏中，一位的 QR code 为 0，表示这是一次请求；Opcode，占四位，指示请求的类型，这里是 0000，表示是标准查询；一位的 Truncated，这里为 0，表明分组没有由于过长而被截断（DNS 协议在运输层使用 UDP 协议，UDP 的包超过 512 位将被截断）；由于不是服务器发出的响应消息，Opcode 和 TC 之间的 AA 位没有被设置；一位的 Recursion desired 设置为 1，表明希望查询时为 DNS 的递归查询（尽管服务器不一定支持）；同样的，由于不是响应消息，RD 之后的 RA 位未被设置。之后，预留了空的三位为 Z，后一位为 0，表明非认证数据不被接受。最后四位 RCODE 在查询分组中不被设置。后面 Questions 为 1，其他为 0，表明 Questions 中有一个实体（RR）。

在 Queries 中的要查询的一个实体是 www.baidu.com，type 为 A，class 为 IN。具体信息在 Questions 部分：Name 为 www.baidu.com，长度为 13 个字符，被“.”划分为三个部分。type 为 A，表明要查询的是主机地址，class 为 IN(0x0001)，表明类别为 Internet。

从 ID 信息可知，这个 DNS 请求分组的 ID 为 0x1c3b。针对这个请求信息的 DNS 响应分组的内容如下：

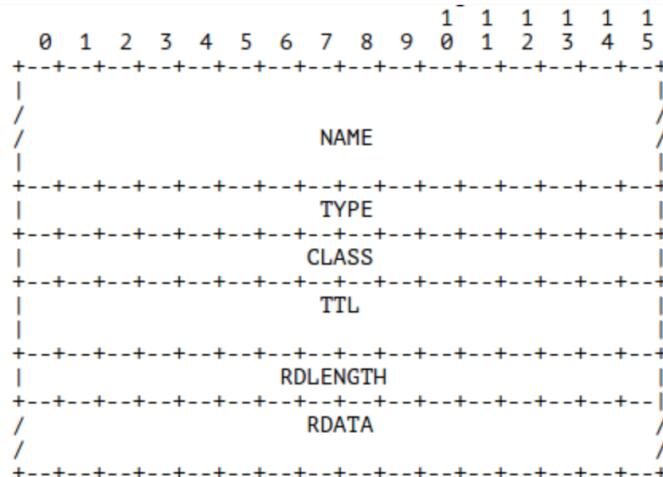
```

▼ Domain Name System (response)
[Request In: 18]
[Time: 0.350082000 seconds]
Transaction ID: 0x1c3b
▼ Flags: 0x8180 Standard query response, No error
  1... .... .... .... = Response: Message is a response
  .000 0... .... .... = Opcode: Standard query (0)
  .... 0.. .... .... = Authoritative: Server is not an authority for domain
  .... ..0. .... .... = Truncated: Message is not truncated
  .... ...1 .... .... = Recursion desired: Do query recursively
  .... .... 1... .... = Recursion available: Server can do recursive queries
  .... .... .0. .... = Z: reserved (0)
  .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
  .... .... ...0. .... = Non-authenticated data: Unacceptable
  .... .... .... 0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 3
Authority RRs: 5
Additional RRs: 5
▼ Queries
  ▶ www.baidu.com: type A, class IN
    Name: www.baidu.com
    [Name Length: 13]
    [Label Count: 3]
    Type: A (Host Address) (1)
    Class: IN (0x0001)
▼ Answers
  ▶ www.baidu.com: type CNAME, class IN, cname www.a.shifen.com
  ▶ www.a.shifen.com: type A, class IN, addr 119.75.216.20
  ▶ www.a.shifen.com: type A, class IN, addr 119.75.213.61
▼ Authoritative nameservers
  ▶ a.shifen.com: type NS, class IN, ns ns1.a.shifen.com
  ▶ a.shifen.com: type NS, class IN, ns ns2.a.shifen.com
  ▶ a.shifen.com: type NS, class IN, ns ns4.a.shifen.com
  ▶ a.shifen.com: type NS, class IN, ns ns5.a.shifen.com
  ▶ a.shifen.com: type NS, class IN, ns ns3.a.shifen.com
▼ Additional records
  ▶ ns1.a.shifen.com: type A, class IN, addr 61.135.165.224
  ▶ ns2.a.shifen.com: type A, class IN, addr 180.149.133.241
  ▶ ns3.a.shifen.com: type A, class IN, addr 61.135.162.215
  ▶ ns4.a.shifen.com: type A, class IN, addr 115.239.210.176
  ▶ ns5.a.shifen.com: type A, class IN, addr 119.75.222.17

```

DNS 的请求报文和响应报文的格式是一致的。这里，Header 的信息与请求报文类似，不同的地方有：QR 位为 1，表明这是一次响应；AA 位为 0，表明不是权威 DNS 服务器；RA 位为 1，表明服务器支持 DNS 的递归查询；最后四位 Reply code 为 0000，表明没有出错。之后可以看见，Questions 有 1 个资源记录（RR），Answer 有 3 个，Authority 有 5 个，Additional 有 5 个。Questions 的 RR 与请求报文的相同，Answer、Authority、Additional 各举一例：

Answer、Authority、Additional 的格式相同：



Answer:

```
▼ www.baidu.com: type CNAME, class IN, cname www.a.shifen.com
  Name: www.baidu.com
  Type: CNAME (Canonical NAME for an alias) (5)
  Class: IN (0x0001)
  Time to live: 1190
  Data length: 15
  CNAME: www.a.shifen.com

▼ www.a.shifen.com: type A, class IN, addr 119.75.216.20
  Name: www.a.shifen.com
  Type: A (Host Address) (1)
  Class: IN (0x0001)
  Time to live: 300
  Data length: 4
  Address: 119.75.216.20
```

这条 RR 表明了查询的结果: name 为 www.baidu.com, type 为 CNAME 表明 name 中的信息是一个别名 (alias), 规范名 (canonical name) 在 RDATA 中 (这里为 CNAME), 表明 www.baidu.com 的规范名为 www.a.shifen.com。TTL 表明了这条记录在 DNS 缓存中存活的时间; 数据长度为 15。Answer 的后两条 RR 的 type 为 A, 表明了主机的 IP 地址, 说明 www.a.shifen.com 的 IP 地址为 **119.75.216.20** 和 **119.75.213.61**。

使用 nslookup 和 ping 指令, 证明了这两个 IP 地址的准确性:

```
~ nslookup www.baidu.com
Server: 202.120.224.26
Address: 202.120.224.26#53

Non-authoritative answer:
www.baidu.com canonical name = www.a.shifen.com.
Name: www.a.shifen.com
Address: 119.75.213.61
Name: www.a.shifen.com
Address: 119.75.216.20

~ ping www.baidu.com
PING www.a.shifen.com (119.75.216.20): 56 data bytes
64 bytes from 119.75.216.20: icmp_seq=0 ttl=46 time=33.028 ms
64 bytes from 119.75.216.20: icmp_seq=1 ttl=46 time=30.498 ms
```

Authority:

```
▼ Authoritative nameservers
  ▼ a.shifen.com: type NS, class IN, ns ns1.a.shifen.com
    Name: a.shifen.com
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 208
    Data length: 6
    Name Server: ns1.a.shifen.com
```

Authority 有 5 个 RR, type 为 NS, 表明了这个域名的权威服务器的地址, 5 个 RR 表明了权威服务器有 5 个, 分别是 ns1.a.shifen.com 至 ns5.a.shifen.com。

Addition:

```
▼ Additional records
  ▼ ns1.a.shifen.com: type A, class IN, addr 61.135.165.224
    Name: ns1.a.shifen.com
    Type: A (Host Address) (1)
    Class: IN (0x0001)
    Time to live: 600
    Data length: 4
    Address: 61.135.165.224
```

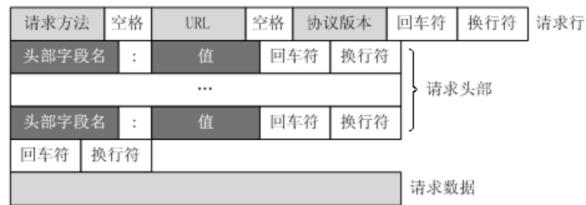
Addition 信息中包含了五台权威服务器的 IP 地址。

最终得到 www.baidu.com 的 IP 地址为 119.75.216.20 和 119.75.213.61。

1.2 将“HTTP GET”请求的分组的截图，并分析协议层框中 HTTP 层字段的含义

由于在 HTTP_DNS.pcapng 中只捕获到了一个包含 GET 的包，所以重新对 HTTP 协议的包进行捕获，结果存储在 HTTP_GET.pcapng 中。

HTTP 的请求消息的结构如下：



以捕捉到的一个请求方式为 GET 的包为例：

```

▼ Hypertext Transfer Protocol
  ▼ GET / HTTP/1.1\r\n
    ▼ [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      [GET / HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
      Host: www.bilibili.com\r\n
      Connection: keep-alive\r\n
      Upgrade-Insecure-Requests: 1\r\n
      Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
      User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_1) AppleWebKit/604.3.5 (KHTML, like Gecko) Version/11.0.1 Safari/604.3.5\r\n
      Accept-Language: en-us\r\n
      DNT: 1\r\n
      Accept-Encoding: gzip, deflate\r\n
      \r\n
      [Full request URI: http://www.bilibili.com/]
      [HTTP request 1/1]
      [Response in frame: 673]
  
```

请求行表明，请求方法为 GET；请求 URL 为 /，与 HOST 组合后完整的 URL 为

http://www.bilibili.com/；协议版本为 1.1。Expert Info 中表明验证等级为 chat。

之后，在请求头部中：1. Host 表明了服务器的域名（也可以是 IP 地址），为 www.bilibili.com；2. Connection 为 keep-alive，说明浏览器希望与服务器的通信连接会被持续保存；3. Upgrade-Insecure-Requests:1 表明浏览器理解并能支持升级到 https 协议；4. Accept 说明了浏览器可接受的文件类型，这里包括 html、xhtml+xml 和 xml 文件；q=0.9, /*, q=0.8 表明了对相应文件类型的喜好程度；5. User-Agent 表明了用户的浏览器的信息，这里可以看出计算机类型为 Mac，客户端的操作系统为 Mac OS X 10.13.1，浏览器为 604.3.5 版本的 Safari；6. Accept-Language 表明了浏览器接受的语言，这里为英语；7. DNT 是 Do Not Track，为 1 表示开启，不允许服务器跟踪记录用户信息；8. Accept-Encoding 表示浏览器接受的编码方式，这里为 gzip 和 deflate。这个 GET 请求不包含请求数据，为空。

对应这个 GET 请求的 HTTP 的应答分组内容如下：

```

▼ Hypertext Transfer Protocol
▶ HTTP/1.1 302 Found\r\n
Server: Tengine\r\n
Date: Tue, 14 Nov 2017 13:01:13 GMT\r\n
Content-Type: text/html\r\n
Content-Length: 258\r\n
Connection: keep-alive\r\n
Location: https://www.bilibili.com/\r\n
Expires: Tue, 14 Nov 2017 13:01:43 GMT\r\n
Cache-Control: max-age=30\r\n
X-Cache: from cn-fjxm-dx-w-02.hdslb.com\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.136318000 seconds]
[Request in frame: 669]
File Data: 258 bytes
▼ Line-based text data: text/html
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">\r\n
<html>\r\n
<head><title>302 Found</title></head>\r\n
<body bgcolor="white">\r\n
<h1>302 Found</h1>\r\n
<p>The requested resource resides temporarily under a different URI.</p>\r\n
<hr/>Powered by Tengine</body>\r\n
</html>\r\n

```

与 DNS 协议不同，HTTP request 和 response 的格式不同。在这里，response 的状态行表明 HTTP 协议版本为 1.1，状态码为 302，状态消息为 Found。消息报头中，Server 表明服务器名称为 Tengine；Date 记录了发送消息的时间；Expires 表明了这个响应的过期时间，一般被客户端缓存使用；之后是对浏览器相应缓存的设置。

这里，响应正文共有 258 个字节，是一篇不大的 html 文件，文件名为 text.html。

1.3 通过浏览分组来阅读HTML页面源文件：找一个有“明文数据”的分组进行截图并分析，在协议层框中选择HTTP层的Data部分，观察原始框中的信息并截图

在 HTTP_GET.pcapng 中有这样一个 response 分组，其携带的 data 信息如下：

```

▼ Line-based text data: text/html
<!DOCTYPE html>\n<html lang="en">\n<head>\n<meta charset="UTF-8">\n<title></title>\n<style>\n*{\n    margin:0;\n    padding:0;\n}\n</style>\n</head>\n<body>\n<\n<script type="text/javascript">\n    var ad_winWidth = document.documentElement.clientWidth || document.body.clientWidth;\n    \n    if(ad_winWidth>859){\n        \tigAdWidth = 860;\n        \tigAdHeight = 125;\t\n    }else{\n        \tigAdWidth = 630;\n        \tigAdHeight = 91;\n    }\n    \n    google_ad_width = igAdWidth;\n    google_ad_height = igAdHeight;\n</script>\n<script type="text/javascript" src="//js.idgdmg.com.cn/s/a002ab0020171013.js"></script>\n<\n</body>\n</html>

```

这里的数据是明文的，是一个名为 text.html 的 html 文件，表明了相应的网页布局。其对应的 Wireshark 的原始框的信息如下：

```

0000 3c 21 44 4f 43 54 59 50 45 20 68 74 6d 6c 3e 0a <!DOCTYPE html>.
0010 3c 68 74 6d 6c 20 6c 61 6e 67 3d 22 65 6e 22 3e <html lang="en">
0020 0a 3c 68 65 61 64 3e 0a 20 20 20 3c 6d 65 74 <head>.
0030 61 20 63 68 61 72 73 65 74 3d 22 55 54 46 2d 38 <meta charset="UTF-8">
0040 22 3e 0a 20 20 20 20 3c 74 69 74 6c 65 3e 3c 2f <title>.
0050 74 69 74 6c 65 3e 0a 20 20 20 3c 73 74 79 6c title>.
0060 65 3e 0a 20 20 20 20 2a 7b 0a 20 20 20 20 20 20 2e>.
0070 20 20 6d 61 72 67 69 6e 3a 30 3b 0a 20 20 20 20 margin:0;.
0080 20 20 20 70 61 64 64 69 66 67 3a 3b 0a 20 20 20 padding:0;.
0090 20 20 20 7d 0a 20 20 20 20 3c 2f 73 34 79 6c 65 3e 0a 2f 62 6f 64 79 3e </style>
00a0 3e 0a 2f 68 65 61 64 >.</head>.<body>
00b0 0a 0e 3c 73 63 72 69 70 74 20 74 79 70 65 3d 22 .<script type="text/javascript">
00c0 74 65 78 74 2f 6d 61 76 61 73 63 72 69 70 74 22 <!-->.
00d0 3e 0e 20 20 20 20 7e 61 72 20 61 64 5f 77 69 6e 57 69 6e <!-->.
00e0 57 69 64 74 68 20 3d 20 64 6f 63 75 6d 65 6e 74 45 6c 65 6d 65 6e 74 width = document.documentElement.clientWidth || clientWidth || 1000;
00f0 2e 64 6f 63 75 6d 65 6e 74 57 69 64 74 68 20 7c 72 20 64 6f 63 75 6d 65 6e 74 <!-->.
0100 64 6f 63 69 65 66 74 57 69 64 74 68 20 7c 72 20 64 6f 63 75 6d 65 6e 74 <!-->.
0110 64 6f 63 75 6d 65 6e 74 6e 62 6f 64 79 2e 63 6c document.body.clientWidth || clientWidth || 1000;
0120 69 65 6e 74 57 69 64 74 68 3e 0a 20 20 0a 20 20 64 6f 63 75 6d 65 6e 74 <!-->.
0130 20 20 69 66 28 61 64 5f 77 69 6e 57 69 64 74 68 if(adWidth <= 850){ adWidth = 800; }.
0140 3e 38 35 39 29 7b 0a 20 20 20 09 69 67 41 64 6f 57 69 64 74 68 <!-->.
0150 57 69 64 74 68 20 3d 20 38 36 30 3b 0a 20 20 20 64 6f 63 75 6d 65 6e 74 <!-->.
0160 20 09 69 67 41 64 48 65 69 67 68 74 20 3d 20 31 64 57 69 64 74 68 20 3d 20 25;.. <!-->.
0170 32 35 3b 09 0a 20 20 20 20 20 7d 65 6c 73 65 67 7b 0a 64 57 69 64 74 68 20 3d 20 25;.. <!-->.
0180 20 20 20 20 09 69 67 41 64 57 69 64 74 68 20 3d 20 20 09 69 67 41 64 48 630;.. <!-->.
0190 20 36 33 30 3b 0a 20 20 39 31 3b 0a 20 20 20 64 6f 63 75 6d 65 5f 61 64 <!-->.
01a0 65 69 67 68 74 20 3d 20 7d 0a 0a 20 20 20 67 6f 6f 67 6c 65 5f 61 64 <!-->.
01b0 7d 0a 0a 20 20 20 20 67 6f 6f 67 6c 65 5f 61 64 <!-->.
01c0 5f 77 69 64 74 68 20 3d 20 69 67 41 64 57 69 64 <!-->.
01d0 74 68 3b 0a 20 20 20 20 67 6f 6f 67 6c 65 5f 61 <!-->.
01e0 64 5f 68 65 69 67 68 74 20 3d 20 69 67 41 64 48 <!-->.
01f0 65 69 67 68 74 3b 0a 3c 2f 73 63 72 69 70 74 3e 64 5f 68 65 69 67 68 74 <!-->.
0200 0a 3c 73 63 72 69 70 74 20 74 79 70 65 3d 22 74 <!-->.
0210 65 78 74 2f 6a 61 76 61 73 63 72 69 70 74 22 20 <!-->.
0220 73 72 63 3d 22 2f 2f 6a 73 2e 69 64 67 64 6d 67 ext/java script" src="http://www.163.com/s/a002ab
0230 2e 63 6f 6d 62 63 6e 2f 73 2f 61 30 30 32 61 62 <!-->.
0240 30 30 32 30 31 37 31 30 31 33 2e 6a 73 22 3e 3c 00201710 13.js"><!-->.
0250 2f 73 63 72 69 70 74 3e 0a 20 20 20 0a 0a 3c /script>.
0260 2f 62 6f 64 79 3e 0a 3c 2f 68 74 6d 6c 3e </body>.</html>

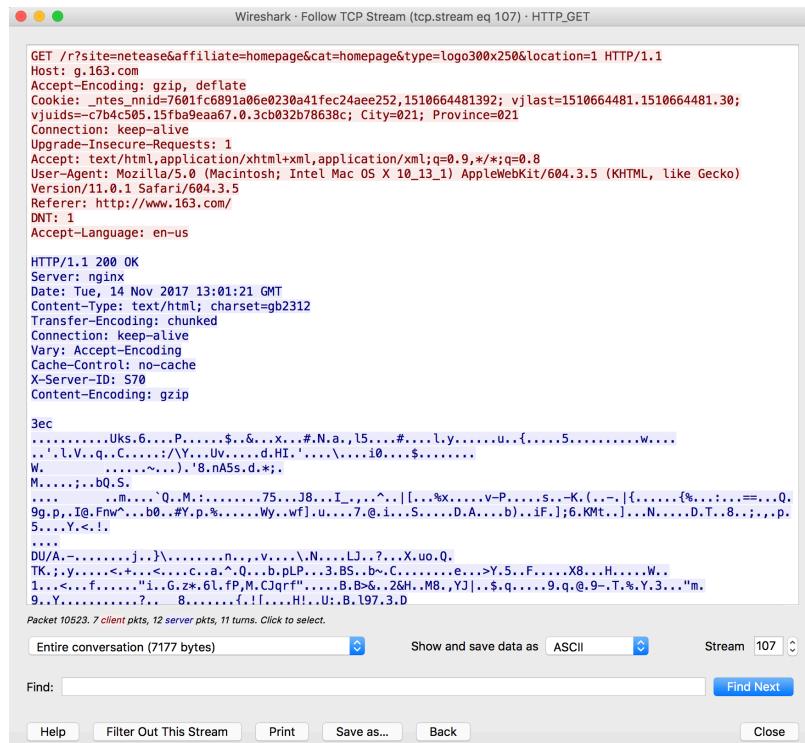
```

可以看出，经过 ASCII 编码得到的信息就是 data 中的信息。

1.4 观察 Web 浏览器和 Web 服务器整个会话过程

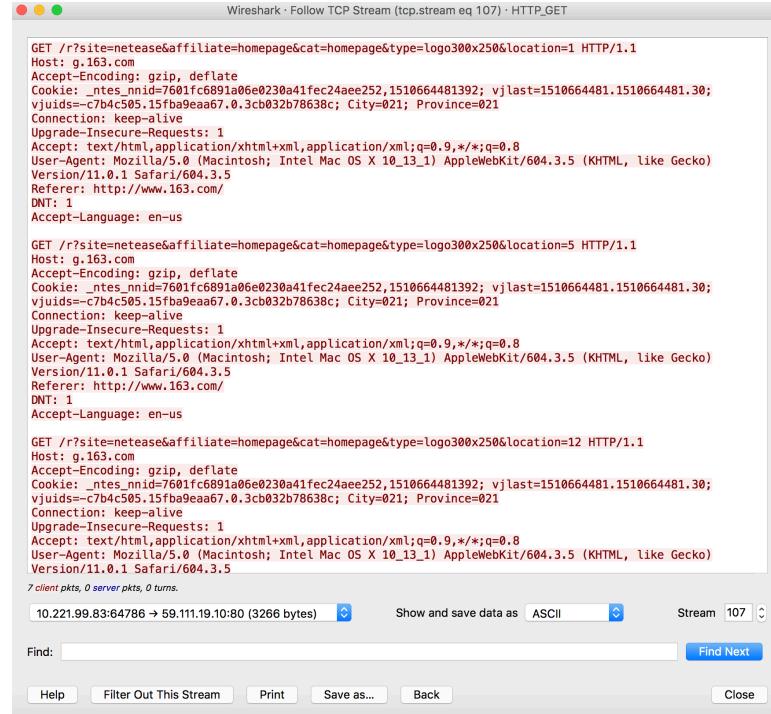
对于 1.3 中的分组，通过 Analyze/Follow/TCP Stream 可以得到客户端和服务器之间完整的数据流。

这里截图获得其一部分：

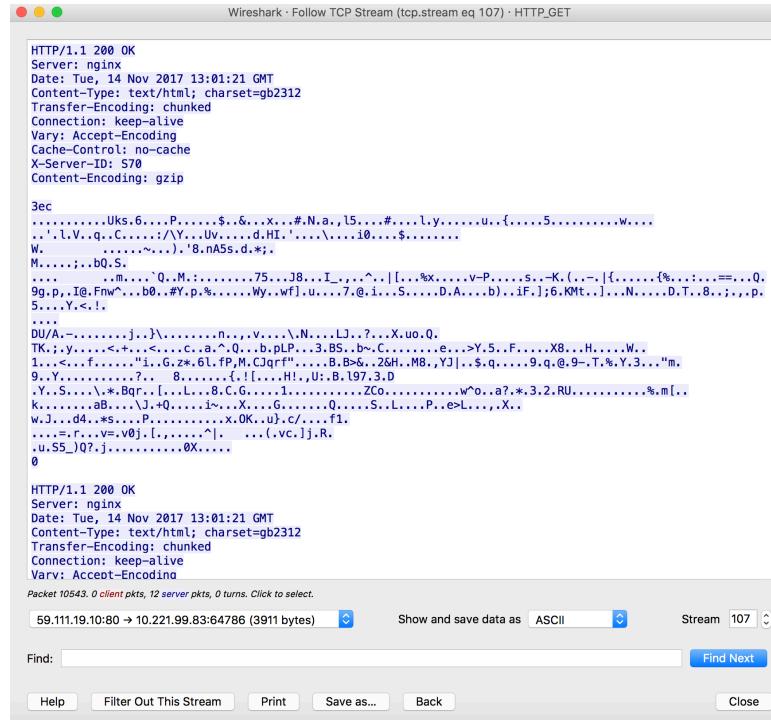


这里，Web 浏览器发送的信息为红色，Web 服务器发送的数据为紫色。

在左下方菜单选择 浏览器 -> 服务器，窗口显示如下：



选择 服务器 -> 浏览器，窗口显示如下：



这样，便可以详细的看出浏览器和服务器之间详细的数据流动。

此时，filter 的过滤条件为：tcp.stream eq 107，表示追踪编号为 107 的 TCP 流。

Bonus: TCP 的三次握手的包:

23 5.374200	10.221.99.83	119.75.213.61	TCP	78 64688 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=1198566760 TSecr=0 SACK_
24 5.422493	119.75.213.61	10.221.99.83	TCP	78 80 → 64688 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1
25 5.422605	10.221.99.83	119.75.213.61	TCP	54 64688 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0

2. FTP 协议分析:

实验步骤:

实验平台与之前相同。使用 FTP 协议连接到 IP 地址为 10.131.243.51 的服务器，下载内容为“computer networks-ftp protocol”的名为 readme.txt 的文本，分析 FTP 协议的工作情况。

实验分析:

控制通道:

2.1.1 在分组中，本地客户端为控制通道初始化了一个到 ftp 服务器上的 FTP 端口的 TCP 连接。观察发送用户名和密码的请求分组和响应分组

FTP 客户端与服务器建立连接时，客户端连接到了服务器的 21 号端口，这是控制连接。

发送用户名的请求分组内容如下:

```
► Internet Protocol Version 4, Src: 10.221.99.83, Dst: 10.131.243.51
► Transmission Control Protocol, Src Port: 59263, Dst Port: 21, Seq: 1, Ack: 1, Len: 16
▼ File Transfer Protocol (FTP)
  ▼ USER anonymous\r\n
    Request command: USER
    Request arg: anonymous
```

可以看见，客户端的 IP 地址为 10.221.99.83，FTP 服务器的 IP 地址为 10.131.243.51；客户端的端口为 59263，服务器的端口为 21。FTP 服务器的控制链接端口均为 21。FTP 命令为 USER，参数为 anonymous，即匿名登录。

对用户名的响应分组内容如下:

```
► Internet Protocol Version 4, Src: 10.131.243.51, Dst: 10.221.99.83
► Transmission Control Protocol, Src Port: 21, Dst Port: 59263, Seq: 1, Ack: 1, Len: 27
▼ File Transfer Protocol (FTP)
  ▼ 220 Microsoft FTP Service\r\n
    Response code: Service ready for new user (220)
    Response arg: Microsoft FTP Service

► Internet Protocol Version 4, Src: 10.131.243.51, Dst: 10.221.99.83
► Transmission Control Protocol, Src Port: 21, Dst Port: 59263, Seq: 28, Ack: 17, Len: 72
▼ File Transfer Protocol (FTP)
  ▼ 331 Anonymous access allowed, send identity (e-mail name) as password.\r\n
    Response code: User name okay, need password (331)
    Response arg: Anonymous access allowed, send identity (e-mail name) as password.
```

服务器的相应信息有两条：220 表示服务就绪，331 表示用户名正确，需要密码。这里第二个分组表明允许用户匿名登录，需要身份（邮箱地址）作为密码输入。

发送密码的请求分组内容如下:

```
► Internet Protocol Version 4, Src: 10.221.99.83, Dst: 10.131.243.51
► Transmission Control Protocol, Src Port: 59263, Dst Port: 21, Seq: 17, Ack: 100, Len: 26
▼ File Transfer Protocol (FTP)
  ▼ PASS cfnetwork@apple.com\r\n
    Request command: PASS
    Request arg: cfnetwork@apple.com
```

这里说明输入的指令为 PASS，即传递密码；传递的密码为一邮箱地址。

对密码的响应分组内容如下:

```

▶ Internet Protocol Version 4, Src: 10.131.243.51, Dst: 10.221.99.83
▶ Transmission Control Protocol, Src Port: 21, Dst Port: 59263, Seq: 100, Ack: 43, Len: 21
▼ File Transfer Protocol (FTP)
  ▼ 230 User logged in.\r\n
    Response code: User logged in, proceed (230)
    Response arg: User logged in.

```

230 表示用户注册完毕，即已经登录。

2.1.2 一旦用户名和密码交换完成后，FTP客户端就会发送命令“PWD”来指定当前工作的目录。

PWD请求和响应分组如下：

```

▶ Internet Protocol Version 4, Src: 10.221.99.83, Dst: 10.131.243.51
▶ Transmission Control Protocol, Src Port: 59263, Dst Port: 21, Seq: 64, Ack: 178, Len: 5
▼ File Transfer Protocol (FTP)
  ▼ PWD\r\n
    Request command: PWD

▶ Internet Protocol Version 4, Src: 10.131.243.51, Dst: 10.221.99.83
▶ Transmission Control Protocol, Src Port: 21, Dst Port: 59263, Seq: 178, Ack: 69, Len: 31
▼ File Transfer Protocol (FTP)
  ▼ 257 "/" is current directory.\r\n
    Response code: PATHNAME created (257)
    Response arg: "/" is current directory.

```

PWD指令要求显示当前目录，这里当前目录是“/”，即根目录。

CWD请求和响应分组如下：

```

▶ Internet Protocol Version 4, Src: 10.221.99.83, Dst: 10.131.243.51
▶ Transmission Control Protocol, Src Port: 59263, Dst Port: 21, Seq: 77, Ack: 229, Len: 7
▼ File Transfer Protocol (FTP)
  ▼ CWD /\r\n
    Request command: CWD
    Request arg: /

▶ Internet Protocol Version 4, Src: 10.131.243.51, Dst: 10.221.99.83
▶ Transmission Control Protocol, Src Port: 21, Dst Port: 59263, Seq: 229, Ack: 84, Len: 29
▼ File Transfer Protocol (FTP)
  ▼ 250 CWD command successful.\r\n
    Response code: Requested file action okay, completed (250)
    Response arg: CWD command successful.

```

CWD指令为 Change Working Directory，即更改工作目录，这里根据分组信息，将工作目录更改到“/”，即根目录。响应分组表示更改成功。

2.1.3 下一条命令“PASV”通知服务器准备第一个即将到来的数据连接，观察该请求的响应分组“227

Entering Passive Mode”。分析服务器所发送6个数字的含义，指出端口号（用10进制表示）

PASV指令的请求分组和响应分组内容如下：

```

▶ Internet Protocol Version 4, Src: 10.221.99.83, Dst: 10.131.243.51
▶ Transmission Control Protocol, Src Port: 59263, Dst Port: 21, Seq: 84, Ack: 258, Len: 6
▼ File Transfer Protocol (FTP)
  ▼ PASV\r\n
    Request command: PASV

▶ Internet Protocol Version 4, Src: 10.131.243.51, Dst: 10.221.99.83
▶ Transmission Control Protocol, Src Port: 21, Dst Port: 59263, Seq: 258, Ack: 90, Len: 49
▼ File Transfer Protocol (FTP)
  ▼ 227 Entering Passive Mode (10,131,243,51,57,6).\r\n
    Response code: Entering Passive Mode (227)
    Response arg: Entering Passive Mode (10,131,243,51,57,6).
    Passive IP address: 10.131.243.51
    Passive port: 14598

```

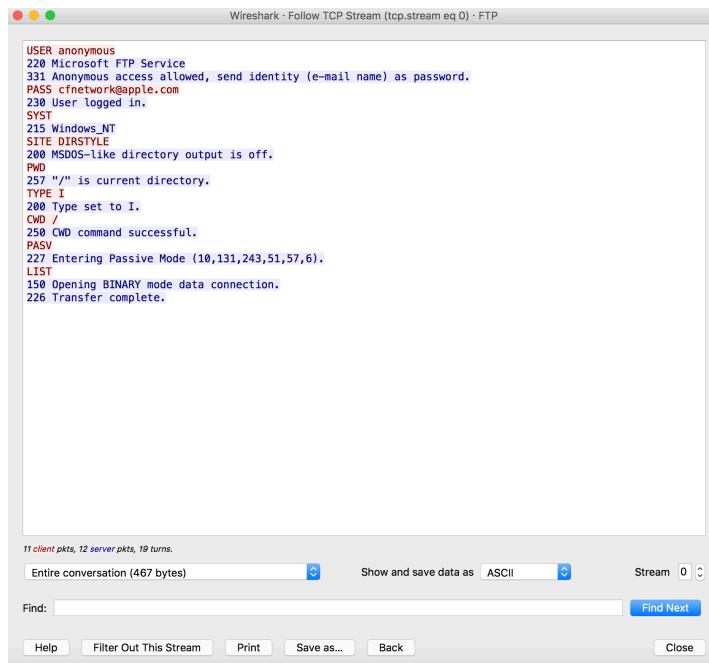
PASV指令通知服务器进入被动模式，准备数据连接。

这里响应分组的返回值为227，表示进入被动模式，并且返回一个六个数字的组。这里前四个数字是服务器的IP地址，后两个数字为端口，十进制表示为 $a * 256 + b$ 。这里服务器的IP地址为10.131.243.51，端口号为 $57 * 256 + 6 = 14598$ 。分组中的Passive port也是14598，证明了这一点。

由于进入了被动模式（被动是指服务器被动等待客户端建立数据连接），服务器通知客户端用这个端口号与自己建立数据连接，连接建立好后便可以开始传输（如果在PORT模式下，则是服务器主动用20号端口去连接客户端。PASV的目的是防止有些防火墙将来自服务器的数据连接请求拒绝掉）。

2.1.4 使用 Follow TCP Stream 检查控制通道，服务器发送多少数据到客户端？客户端发送多少数据到服务器？

TCP Stream表示的控制通道结果如下：



根据统计，服务器在控制通道发送了11个包，客户端在控制通道发送了9个包（TCP Stream显示服务器发送了12个包，客户端发送了11个包，但这是包括数据通道的结果，这里没有显示）。

控制通道截图：

9	13.282319	10.221.99.83	10.131.243.51	FTP	82 Request: USER anonymous
12	13.283020	10.221.99.83	10.131.243.51	FTP	82 Request: USER anonymous
13	13.284599	10.131.243.51	10.221.99.83	FTP	93 Response: 220 Microsoft FTP Service
15	13.285432	10.131.243.51	10.221.99.83	FTP	138 Response: 331 Anonymous access allowed, send identity (e-mail name) as password.
17	13.285543	10.221.99.83	10.131.243.51	FTP	92 Request: PASS cfnetwork@apple.com
18	13.286292	10.131.243.51	10.221.99.83	FTP	93 Response: 220 Microsoft FTP Service
20	13.286604	10.131.243.51	10.221.99.83	FTP	138 Response: 331 Anonymous access allowed, send identity (e-mail name) as password.
22	13.286745	10.221.99.83	10.131.243.51	FTP	92 Request: PASS cfnetwork@apple.com
23	13.288736	10.131.243.51	10.221.99.83	FTP	87 Response: 230 User logged in.
25	13.288827	10.221.99.83	10.131.243.51	FTP	72 Request: SYST
26	13.290321	10.131.243.51	10.221.99.83	FTP	87 Response: 230 User logged in.
28	13.290408	10.221.99.83	10.131.243.51	FTP	72 Request: SYST
168	19.076811	10.221.99.83	10.131.243.51	FTP	72 Request: PASV
169	19.080183	10.131.243.51	10.221.99.83	FTP	115 Response: 227 Entering Passive Mode (10,131,243,51,57,8).

数据通道:

2.2.1 数据通道用于列出当前工作目录的内容（开始是根目录）。客户端通过控制通道发送LIST命令，请求列出当前工作目录的内容

截图如下：

```

▶ Internet Protocol Version 4, Src: 10.221.99.83, Dst: 10.131.243.51
▶ Transmission Control Protocol, Src Port: 59263, Dst Port: 21, Seq: 90, Ack: 307, Len: 6
▼ File Transfer Protocol (FTP)
  ▼ LIST\r\n
    Request command: LIST
  
```

客户端通过控制通道发送LIST命令，服务器通过数据通道返回结果（显示在ftp-data中）。

响应报文如下：

```

▶ Internet Protocol Version 4, Src: 10.131.243.51, Dst: 10.221.99.83
▶ Transmission Control Protocol, Src Port: 14598, Dst Port: 59266, Seq: 1, Ack: 1, Len: 71
  FTP Data (-rwxrwxrwx 1 owner group 30 Oct 22 21:56 readme.txt\r\n)
  
```

2.2.2 在文件传送前，有一个命令“TYPE I”，表示传送的数据表示格式，找到该分组并分析

该请求分组和响应分组内容如下：

```

▶ Internet Protocol Version 4, Src: 10.221.99.83, Dst: 10.131.243.51
▶ Transmission Control Protocol, Src Port: 59263, Dst Port: 21, Seq: 69, Ack: 209, Len: 8
▼ File Transfer Protocol (FTP)
  ▼ TYPE I\r\n
    Request command: TYPE
    Request arg: I

  ▶ Internet Protocol Version 4, Src: 10.131.243.51, Dst: 10.221.99.83
  ▶ Transmission Control Protocol, Src Port: 21, Dst Port: 59263, Seq: 209, Ack: 77, Len: 20
  ▼ File Transfer Protocol (FTP)
    ▼ 200 Type set to I.\r\n
      Response code: Command okay (200)
      Response arg: Type set to I.
  
```

TYPE指令有A和I两种：A表示ASCII码，I表示Binary（二进制）。这里设置为按照二进制传输。返回值为200，表示设置成功。

2.2.3 分析文件传输的分组

将filter设置为ftp-data，找到四个分组：

ftp-data						
No.	Time	Source	Destination	Protocol	Length	Info
77	13.318849	10.131.243.51	10.221.99.83	FTP-D...	137	FTP Data: 71 bytes
79	13.318856	10.131.243.51	10.221.99.83	FTP-D...	125	FTP Data: 71 bytes
124	13.369690	10.131.243.51	10.221.99.83	FTP-D...	137	FTP Data: 71 bytes
177	19.086411	10.131.243.51	10.221.99.83	FTP-D...	96	FTP Data: 30 bytes

刚开始比较奇怪为什么会有四个分组，后来经过分析，发现情况如下：

62	13.307921	10.221.99.83	10.131.243.51	FTP	72	Request: LIST
68	13.310759	10.221.99.83	10.131.243.51	FTP	72	Request: LIST
118	13.362908	10.221.99.83	10.131.243.51	FTP	72	Request: LIST
17	13.285543	10.221.99.83	10.131.243.51	FTP	92	Request: PASS cfnetwork@apple.com
22	13.286745	10.221.99.83	10.131.243.51	FTP	92	Request: PASS cfnetwork@apple.com
97	13.334914	10.221.99.83	10.131.243.51	FTP	92	Request: PASS cfnetwork@apple.com
150	19.048912	10.221.99.83	10.131.243.51	FTP	92	Request: PASS cfnetwork@apple.com
55	13.304330	10.221.99.83	10.131.243.51	FTP	72	Request: PASV
115	13.357054	10.221.99.83	10.131.243.51	FTP	72	Request: PASV
168	19.076811	10.221.99.83	10.131.243.51	FTP	72	Request: PASV
59	13.306772	10.221.99.83	10.131.243.51	FTP	93	Request: PORT 10,221,99,83,231,129

FTP共连接了四次，其中三次为PASV，一次为PORT；三次PASV中，每次都发送了LIST指令，在数据通道中均接收到了相应的包，其中服务器的端口分别为14598，14599，14600。最后一个包为传送的文件。

含有传送文件的包内容如下：

```
▶ Internet Protocol Version 4, Src: 10.131.243.51, Dst: 10.221.99.83
▶ Transmission Control Protocol, Src Port: 14600, Dst Port: 59270, Seq: 1, Ack: 1, Len: 30
FTP Data (computer networks-ftp protocol)
```

FTP每传送一个文件就要新建一个数据连接。前面的连接用于传递LIST的返回结果，这里传输文件的连接IP地址不变，服务器端口号为14600，客户端端口号为59270。传送的数据为“computer networks-ftp protocol”，这正是readme.txt文件的内容。

2.2.4 使用Follow TCP Stream检查每一个数据通道，在每个数据通道中，服务器发送多少数据到客户端？客户端发送多少数据到服务器？哪个请求的数据最多，传送文本文件还是目录列表信息？

(内容较多，截图不一放上来)

PORT模式下服务器20端口到客户端59265端口：服务器发送了2个包，客户端发送了3个包。

PASV模式下服务器14598端口到客户端59266端口：服务器发送了2个包，客户端发送了5个包。

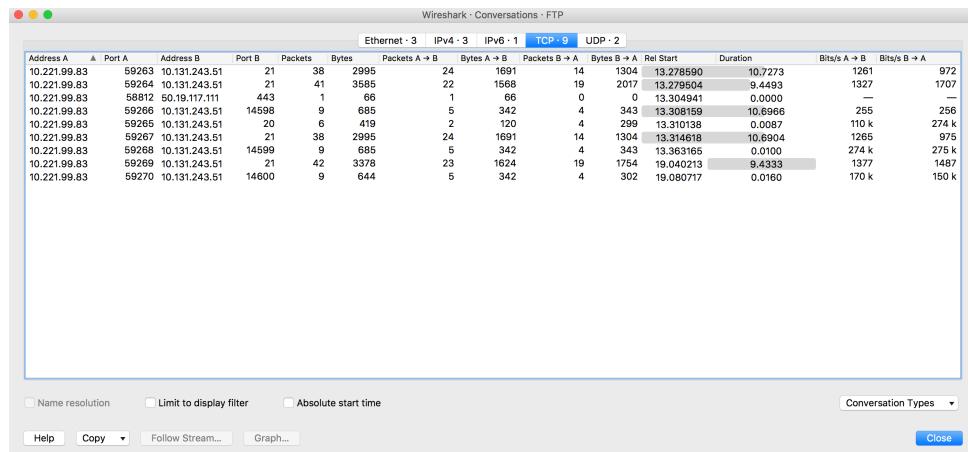
PASV模式下服务器14599端口到客户端59268端口：服务器发送了2个包，客户端发送了5个包。

PASV模式下服务器14600端口到客户端59270端口：服务器发送了2个包，客户端发送了5个包。

这里只有一个分组是文本文件，其余全是目录列表信息，所以请求的目录列表信息多。

2.2.5 在菜单“Statistics/Conversation”中选则“TCP” Conversation，分析本机与ftp服务器之间的跟踪记录，截图，分析其中的几个TCP连接，写出各位什么类型的通道（其中是数据通道的要指明是列举目录的数据通道还是文件传输的数据通道）

结果如下：



主机IP地址为10.221.99.83和10.131.243.51的为FTP通信。其中，FTP服务器端口号为21的是控制通道，为20的是数据通道中的PORT模式，为14598、14599、14600的是数据通道中的PASV模式。其中，服务器端口号为20、14598、14599的是列举目录的数据，14600是文件传输的数据。

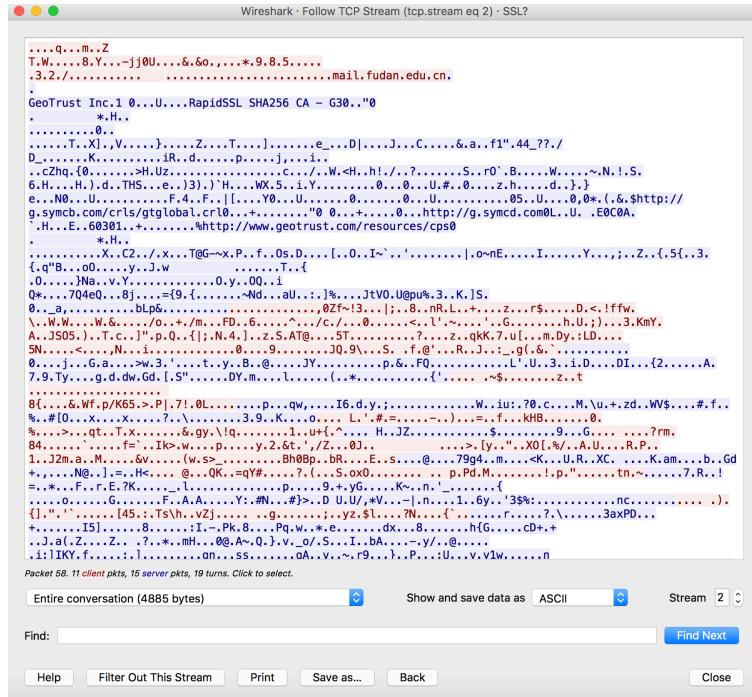
3. SMTP 与 POP3 协议分析:

实验步骤:

实验平台与之前相同。考虑到使用 SMTP 协议，使用了几个邮件客户端，抓取相关包并进行了比较。测试使用了 FoxMail、网易邮件大师和 Thunderbird，发现 FoxMail 和网易邮件大师均对邮件正文进行了加密且无法手动设置不加密，而 Thunderbird 可以手动选择不加密，因此使用 Thunderbird 作为实验用的客户端。手动设置使用 SMTP 和 POP3 协议，邮件服务器均使用 mail.fudan.edu.cn，使用自己的学邮（15300240004@fudan.edu.cn）向自己的学邮发送测试信息。

本机 IP 地址为 10.221.99.83，邮件服务器 IP 地址为 202.120.224.10。

加密后的邮件(无法用于实验):



实验分析:

3.1 根据捕获的分组，了解发送邮件和接收邮件的过程。

SMTP主要用于发送邮件，是一个“推”的过程：

建立TCP连接后，客户端发送EHLO标记发件人的身份，并要求进行安全认证：

```

▶ Internet Protocol Version 4, Src: 10.221.99.83, Dst: 202.120.224.10
▶ Transmission Control Protocol, Src Port: 54418, Dst Port: 25, Seq: 1, Ack: 70, Len: 21
▼ Simple Mail Transfer Protocol
  ▼ Command Line: EHLO Wong-Mac.local\r\n
    Command: EHLO
    Request parameter: Wong-Mac.local
  
```

可以看见，客户端IP地址为10.221.99.83，端口号为54418；邮件服务器IP地址为202.120.224.10，端口号为25，这也是SMTP协议使用的端口号。

客户端发送EHLO指令到服务器后，服务器返回一个列表：

```

▶ Internet Protocol Version 4, Src: 202.120.224.10, Dst: 10.221.99.83
▶ Transmission Control Protocol, Src Port: 25, Dst Port: 54418, Seq: 70, Ack: 22, Len: 199
▼ Simple Mail Transfer Protocol
  ▶ Response: 250-mail\r\n
  ▶ Response: 250-PIPELINING\r\n
  ▶ Response: 250-AUTH LOGIN\r\n
  ▶ Response: 250-AUTH=LOGIN PLAIN\r\n
  ▶ Response: 250-coremail 1uXr2xkj7kG0xkI17xGrUDI0s8FY2U3Uj8Cz28x1UUUUU7Ic2I0Y2UFhFVclUCa0xDrUUUUj\r\n
  ▶ Response: 250-STARTTLS\r\n
  ▶ Response: 250-SMTPUTF8\r\n
  ▶ Response: 250-8BITMIME\r\n

```

这里服务器返回值为250，表示请求的操作成功；列表中的AUTH一值表示了身份验证的方式，这里PLAIN指的是明文验证。

之后，客户端发送BASE64编码的验证信息，服务器检验后发送成功信息：

```

▶ Internet Protocol Version 4, Src: 10.221.99.83, Dst: 202.120.224.10
▶ Transmission Control Protocol, Src Port: 54418, Dst Port: 25, Seq: 22, Ack: 269, Len: 49
▼ Simple Mail Transfer Protocol
  ▶ Command Line: AUTH PLAIN ADE1MzAwMjQwMDA0Ahdbmdhbzk3MDIyMA==\r\n
    Command: AUTH
    Request parameter: PLAIN ADE1MzAwMjQwMDA0Ahdbmdhbzk3MDIyMA==
  ▶ Username/Password: ADE1MzAwMjQwMDA0Ahdbmdhbzk3MDIyMA==

▶ Internet Protocol Version 4, Src: 202.120.224.10, Dst: 10.221.99.83
▶ Transmission Control Protocol, Src Port: 25, Dst Port: 54418, Seq: 269, Ack: 71, Len: 31
▼ Simple Mail Transfer Protocol
  ▶ Response: 235 Authentication successful\r\n
    Response code: Authentication successful (235)
    Response parameter: Authentication successful

```

之后，使用SMTP发送邮件的三个主要命令：MAIL FROM, RCPT TO, DATA发送邮件正文，服务器正常响应：

```

MAIL FROM:<15300240004@fudan.edu.cn>
250 Mail OK
RCPT TO:<15300240004@fudan.edu.cn>
250 Mail OK
DATA
354 End data with <CR><LF>.<CR><LF>

```

邮件正文（以“.”结束）：

```

Message-ID: <5A0D76D0.70603@fudan.edu.cn>
Date: Thu, 16 Nov 2017 19:30:24 +0800
From: apple <15300240004@fudan.edu.cn>
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:24.0) Gecko/20100101 Thunderbird/24.3.0
MIME-Version: 1.0
To: 15300240004@fudan.edu.cn
Subject: Test Mail
Content-Type: text/plain; charset=GB2312
Content-Transfer-Encoding: 7bit

This is the test mail for protocol SMTP & POP3.
.

```

最后，客户端使用QUIT指令退出，服务器响应：

```

▶ Internet Protocol Version 4, Src: 10.221.99.83, Dst: 202.120.224.10
▶ Transmission Control Protocol, Src Port: 54418, Dst Port: 25, Seq: 573, Ack: 419, Len: 6
▼ Simple Mail Transfer Protocol
  ▶ Command Line: QUIT\r\n
    Command: QUIT

▶ Internet Protocol Version 4, Src: 202.120.224.10, Dst: 10.221.99.83
▶ Transmission Control Protocol, Src Port: 25, Dst Port: 54418, Seq: 419, Ack: 579, Len: 9
▼ Simple Mail Transfer Protocol
  ▶ Response: 221 Bye\r\n
    Response code: <domain> Service closing transmission channel (221)
    Response parameter: Bye

```

POP3协议主要用于接收邮件，是一个“收”的过程：

类似的，客户端使用AUTH登陆，邮件服务器响应：

```

▶ Internet Protocol Version 4, Src: 10.221.99.83, Dst: 202.120.224.10
▶ Transmission Control Protocol, Src Port: 54429, Dst Port: 110, Seq: 1, Ack: 87, Len: 6
▼ Post Office Protocol
  ▼ AUTH\r\n
    Request command: AUTH

▶ Internet Protocol Version 4, Src: 202.120.224.10, Dst: 10.221.99.83
▶ Transmission Control Protocol, Src Port: 110, Dst Port: 54429, Seq: 87, Ack: 7, Len: 25
▼ Post Office Protocol
  ▼ +OK core mail\r\n
    Response indicator: +OK
    Response description: core mail
    PLAIN\r\n
    .\r\n
  .

```

随后，客户端使用CAPA指令，服务器返回所支持的拓展服务列表：

```

CAPA
+OK Capability list follows
TOP
USER
PIPELINING
UIDL
LANG
UTF8
SASL PLAIN
STLS
.

```

客户端进行登陆：

```

▶ Internet Protocol Version 4, Src: 10.221.99.83, Dst: 202.120.224.10
▶ Transmission Control Protocol, Src Port: 54429, Dst Port: 110, Seq: 25, Ack: 207, Len: 38
▼ Post Office Protocol
  ▼ ADE1MzAwMjQwMDA0Ahdbmdhbzk3MDIyMA==\r\n
    Request command: ADE1MzAwMjQwMDA0Ahdbmdhbzk3MDIyMA==

▶ Internet Protocol Version 4, Src: 202.120.224.10, Dst: 10.221.99.83
▶ Transmission Control Protocol, Src Port: 110, Dst Port: 54429, Seq: 207, Ack: 63, Len: 38
▼ Post Office Protocol
  ▼ +OK 13 message(s) [23275981 byte(s)]\r\n
    Response indicator: +OK
    Response description: 13 message(s) [23275981 byte(s)]

```

客户端使用LIST指令列出所有邮件的个数、序号和大小，服务器响应：

```

LIST
+OK 13 23275981
1 1681
2 4598
3 5363
4 6112
5 11833
6 78200
7 103479
8 462640
9 513204
10 914216
11 1946008
12 4494151
13 14734496
.

```

客户端使用UIDL指令列出所有邮件，服务器响应：

```

UIDL
+OK 13 23275981
1 1tbIA0cGE1kp0cOyXQAAAsS
2 1tbIAgUPEVKpW0ZeaAmJ
3 1tbIAQEPEVKp0REfYgAAmw
4 1tbIAQkHBVKp0UN5qAAmR
5 1tbIAQkAE1Kp0VHZqAAasQ
6 1tbIAQkEA1Kp0VUC8AAmD
7 1tbIAQMGD1kp8UfggAAm3
8 1tbIAgkBE1kpw5f0xAAsG
9 1tbIAQcKB1kp0bYEJgAAm1
10 1tbIAgkJBVKp03940gAAmq
11 1tbIAgkRBVKp04VgnwAAm1
12 1tbIAgkLA1Kp05BxnwAAmE
13 1tbIAgkKA1Kp05A21wAAmW
.

```

客户端要求取出第一封邮件，服务器回应并给出数据：

```

RETR 1
+OK 1681 octets
Received: from Wong-Mac.local (unknown [10.221.99.83])
by app1 (Coremail) with SMTP id AQUFCkAJr0LzdglabBvvzAg--.29910S2;
Thu, 16 Nov 2017 19:31:05 +0800 (CST)
Message-ID: <5A0D76D0.70603@fudan.edu.cn>
Date: Thu, 16 Nov 2017 19:30:24 +0800
From: apple <15300240004@fudan.edu.cn>
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:24.0) Gecko/20100101 Thunderbird/24.3.0
MIME-Version: 1.0
To: 15300240004@fudan.edu.cn
Subject: Test Mail
Content-Type: text/plain; charset=GB2312
Content-Transfer-Encoding: 7bit
X-CM-TRANSID:AQUFCkAJr0LzdglabBvvzAg--.29910S2
X-Coremail-Antispam: 1UD129KbjDUh29KB7ZKAUJUUUUU529EdanIXcx71UUUUU7v73
VFW2AGmfu7bjvjm3AaLaJ3UjIYCTnIWjp_UUUYg7CY07T20VC2zvCF04k26cxkx2IYs7xG
6rwj6s0DM28LY4IEwIIxxk0rvA271Y1VAk24vEj48ve4K1BwA24x0Y4e21x8c181cV
AfW10_tr0E3s1184ACjcx6xLIjxv20xEc7cjxVAFwI0_GcCE3s1l84ACjcx618E87Iv
67AKxW0oVcQ3wA2z0xY4vEx4A2jz1Ec7cjxVAFwI0_GcCE3s1lnxKEFVAIw20f6cxK64
vIFxWle2I262IYc4CY6c81j28ICvAay2xGBwq4xG64yxF2Iw4CE518CrC2j2WLYx0E
74AG7CV6cx26ry5x10Jr10x856xCAFVCj44Y6t1j6r4UM4x0Y481cvAKT48MAX6x7
Aq67IIx4CEVc8vx2IErcIfxwCYj10s3jkx162A11cE67vIY487Mxk1ecxBwCm-wCF04k2
0xyY0x0EwlxGrwCF04k20xv0xI1j40Ec7cjxwF04k20xvE74AG7Vc6c26ry5x10Jr
1l41B1310E41kC6x0y7zJr0_Gr1lx21qxVAqx4xG67AKxWUJWUGwC20s026x8gjcxK
67AKxWUGWNUWwC2zVAF1VAY17Ce14v26r1j6r15MIYxrkI7VAKI48JMIIf0xvE21x0cI
8ICvAFwI0_Jr0_JF41IxA1cVCOI7IYx21Y6xkF7I0E14v26r1j6r4UMIIf0xvE42xk8Vav
wI81Ck0rWrZr1j6s0DMIf0xvEx4A2jz1E14v26r1j6r4UMIIf0xvEx4A2jsIEc7CjxV
AFwI0_Jr0_GrUvc5s0vfc2KfnxnUU143zEx7VU0Ap5UUUUUJ=
X-CM-SenderInfo: arvtiasuqim6i3vldqovvfx0/

```

This is the test mail for protocol SMTP & POP3.

```

► Internet Protocol Version 4, Src: 202.120.224.10, Dst: 10.221.99.83
► Transmission Control Protocol, Src Port: 110, Dst Port: 54429, Seq: 2249, Ack: 89, Len: 224
► Post Office Protocol
▼ Internet Message Format
  Received: from Wong-Mac.local (unknown [10.221.99.83]) \r\n\tby app1 (Coremail) with SMTP id AQUFCkAJr0LzdglabBvvzAg--.29910S2;\r\n\tThu, 16 Nov 2017 19
  Message-ID: <5A0D76D0.70603@fudan.edu.cn>
  Date: Thu, 16 Nov 2017 19:30:24 +0800
  From: apple <15300240004@fudan.edu.cn>, 1 item
  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:24.0) Gecko/20100101 Thunderbird/24.3.0
  MIME-Version: 1.0
  To: 15300240004@fudan.edu.cn, 1 item
  Subject: Test Mail
  Content-Type: text/plain; charset=GB2312
  Content-Transfer-Encoding: 7bit
  Unknown-Extension: X-CM-TRANSID:AQUFCkAJr0LzdglabBvvzAg--.29910S2 (Contact Wireshark developers if you want this supported.)
  Unknown-Extension [truncated]: X-Coremail-Antispam: 1UD129KbjDUh29KB7ZKAUJUUUUU529EdanIXcx71UUUUU7v73\r\n\tVFW2AGmfu7bjvjm3AaLaJ3UjIYCTnIWjp_UUUYg7CY0
  Unknown-Extension: X-CM-SenderInfo: arvtiasuqim6i3vldqovvfx0/ (Contact Wireshark developers if you want this supported.)
  ▼ Line-based text data: text/plain
    This is the test mail for protocol SMTP & POP3.\r\n
    .\r\n

```

可以看见，邮件正文为：“This is the test mail for protocol SMTP & POP3.”

最后，客户端使用QUIT指令退出，服务器响应：

```

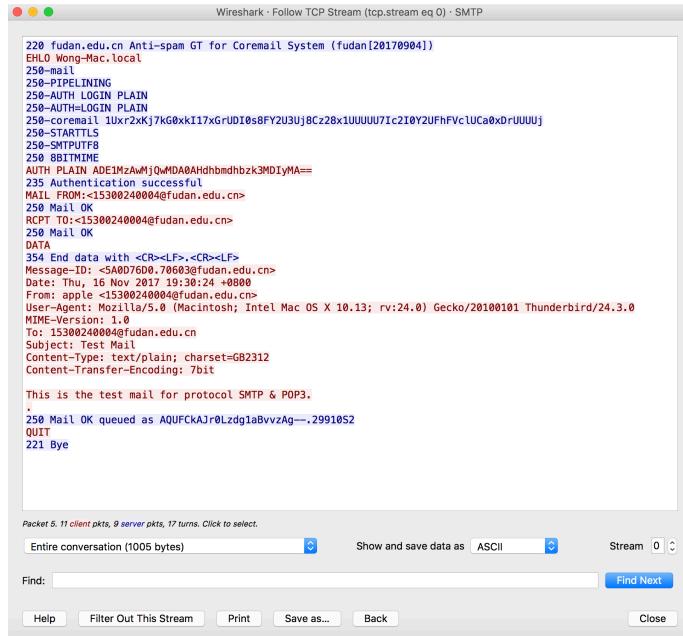
► Internet Protocol Version 4, Src: 10.221.99.83, Dst: 202.120.224.10
► Transmission Control Protocol, Src Port: 54429, Dst Port: 110, Seq: 89, Ack: 2473, Len: 6
► Post Office Protocol
  ▼ QUIT\r\n
    Request command: QUIT

► Internet Protocol Version 4, Src: 202.120.224.10, Dst: 10.221.99.83
► Transmission Control Protocol, Src Port: 110, Dst Port: 54429, Seq: 2473, Ack: 95, Len: 15
► Post Office Protocol
  ▼ +OK core mail\r\n
    Response indicator: +OK
    Response description: core mail

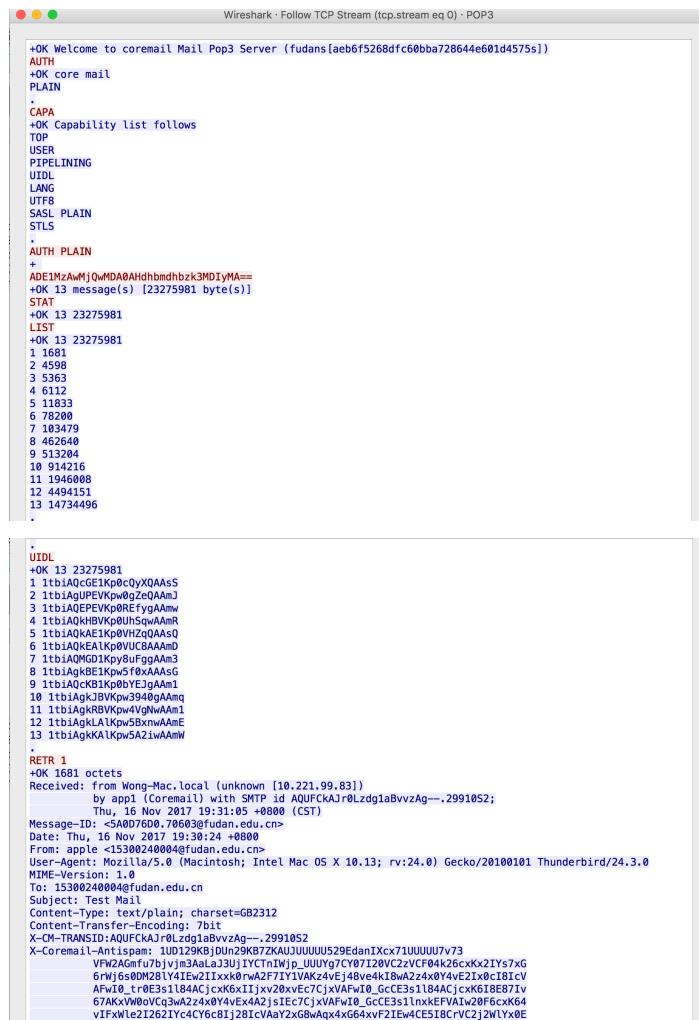
```

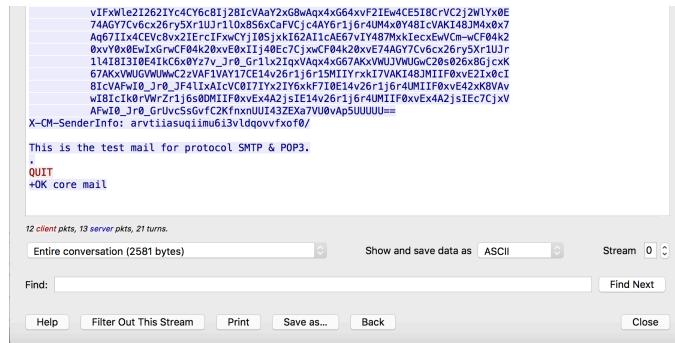
3.2 在发送的分组中，选择一个，使用Follow TCP Stream来观察控制通道的全部内容

SMTP的通道如图：



POP3协议如图：





3.3 基于SMTP的电子邮件报文传输包括三个主要命令：MAIL FROM, RCPT TO 及 DATA，将 TCP Stream 截图

```

MAIL FROM:<15300240004@fudan.edu.cn>
250 Mail OK
RCPT TO:<15300240004@fudan.edu.cn>
250 Mail OK
DATA
354 End data with <CR><LF>,<CR><LF>
Message-ID: <5A0D76D0.70603@fudan.edu.cn>
Date: Thu, 16 Nov 2017 19:30:24 +0800
From: apple <15300240004@fudan.edu.cn>
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:24.0) Gecko/20100101 Thunderbird/24.3.0
MIME-Version: 1.0
To: 15300240004@fudan.edu.cn
Subject: Test Mail
Content-Type: text/plain; charset=GB2312
Content-Transfer-Encoding: 7bit

This is the test mail for protocol SMTP & POP3.
.
```

MAIL FROM 指令指出了发件人的地址，这里为15300240004@fudan.edu.cn；RCPT TO 指出了收件人的地址，这里也为15300240004@fudan.edu.cn(所以SMTP和POP3的邮件服务器是一台)；DATA向服务器请求发送数据，服务器返回值为354，表示开始邮件正文输入，以“.”结束。

3.4 分析POP连接的分组，使用Follow TCP Stream来观察全部的数据交换

POP3协议的分析见3.1。全部的数据交换见3.2。

这里报文由于比较长，被截成了许多块，每块含正文的一部分：

```

RETR 1
+OK 1681 octets
Received: from Wong-Mac.local (unknown [10.221.99.83])
    by app1 (Coremail) with SMTP id AQUFCkAJr0LzdglabvvzAg--.29910S2;
    Thu, 16 Nov 2017 19:31:05 +0800 (CST)
Message-ID: <5A0D76D0.70603@fudan.edu.cn>
Date: Thu, 16 Nov 2017 19:30:24 +0800
From: apple <15300240004@fudan.edu.cn>
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:24.0) Gecko/20100101 Thunderbird/24.3.0
MIME-Version: 1.0
To: 15300240004@fudan.edu.cn
Subject: Test Mail
Content-Type: text/plain; charset=GB2312
Content-Transfer-Encoding: 7bit
X-COREMAIL-TRANSACTIONID:AQUFCkAJr0LzdglabvvzAg--.29910S2
X-COREMAIL-ANTISPAM: 1UD129KBJdUJn29KB7ZKAUJUUUUU529EdanIxcx71UUUUU7v73
VFW2AGmfu7bjvjm3aLaJ3UjIYCTnIwjp_UUUYg7CY07I20Vc2zVCF04k26cxKx2IYs7xG
6rWj6s0DM28lY4IEw2IIxx0rwA2F7IY1VKz4Ej48ve4k18wA2z4x0v4vE2Ix0cI8IcV
AFw10_tr0Essl84ACjcxK6x1Ijxv20xvEc7jxVAFw10_GCcE3s1l84ACjcxK618E871v
67AKxVW0oVCq3wA2z4x0v4vEx4A2jzIc7CjxVAFw10_GCcE3s1lnxKEFVaiW20f6cxK64
vIFxWle212621Yc4CY6c8Ij28ICVaA2y2x8wAqx4xG64xvF2Iw4CE51B7vC2j2Wlyx0E
74AGY7Cv6cx26ry5vX1u1r1l0x856xCaFVcjc4Y6r1j64M4x0v48IcVAKI48JM4x0x7
Aq67IIx4CEVc8v2IErcIxFwCYI05jkxI62A1IcAE67vIY487MxkIecxEwvCm-wcf04k2
0xv0x0xEwIxGrwCF04k20xvEx0Ij40Ec7jxwCF04k20xvE74AGY7Cv6cx26ry5r1Uj
1141813I0E4IkC6x0y7z7v_Jr0_Grlv2IxqVAq4xG67AKxWUJWvUGwC20s026x8GjcxK
67AKxVWUGvUWwC2zAF1VAY17CE14v26l1j6r15MIYrrxk7VAKI48JM4x0vE2Ix0cI
81cVAFw10_Jr0_JF41IxAcVc07IYx2IY6kkF7I0E14v26l1j6r4UMIIF0xvEx4A2jsIEc7CjxV
wI81cIk0rVmrZr1j6s0DMIIFF0xvEx4A2jsIE14v26r1j6r4UMIIF0xvEx4A2jsIEc7CjxV
AFw10_Jr0_GrUvcSsGvfc2KfnxnUI43ZE7VU0vAp5UUUUU==

X-CM-UserInfo: arvtiasuqiimu6i3vldqovvxf0/0

This is the test mail for protocol SMTP & POP3.
.
QUIT
+OK core mail

```

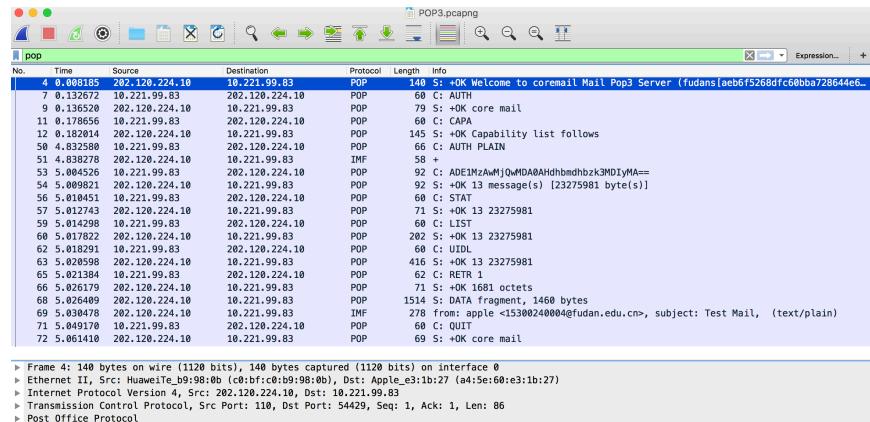
ASCII信息如下：

```

16 d0 2e ee 00 00 52 65 63 65 69 76 65 64 3a 20 .....Received:
66 72 6f 6d 20 57 6f 6e 67 2d 4d 61 63 2e 6c 6f from Wong-Mac.lo
63 61 6c 20 28 75 6e 6b 6e 6f 77 6e 20 5b 31 30 cal (unk nown [10
2e 32 32 31 2e 39 39 2e 38 33 5d 29 0d 0a 09 62 .221.99. 83])...b
79 20 61 70 70 31 20 28 43 6f 72 65 6d 61 69 6c y app1 ( Coremail
29 20 77 69 74 58 20 53 4d 54 50 20 69 64 20 41 ) with S MTP id A
51 55 46 43 6b 41 4a 72 30 4c 7a 64 67 31 61 42 QUFCKAJr 0Lzdq1aB
76 76 7a 41 67 2d 2d 2e 32 39 39 31 30 53 32 3b vVzAg--. 2991052;
0d 0a 09 54 68 75 2c 20 31 36 20 4e 6f 76 20 32 ...Thu, 16 Nov 2
30 31 37 20 31 39 3a 33 31 3a 30 35 20 2b 30 38 017 19:3 1:05 +08
30 30 20 28 43 53 54 29 0d 0a 4d 65 73 73 61 67 00 (CST) ..Message
65 2d 49 44 3a 20 3c 35 41 30 44 37 36 44 30 2e e-ID: <5 A007600.
37 30 36 30 33 40 66 75 64 61 6e 2e 65 64 75 2e 70603fu dan.edu.
63 6e 3e 0d 0a 44 61 74 65 3a 20 54 68 75 2c 20 cn..Dat e: Thu,
31 36 20 4e 6f 76 20 32 30 31 37 20 31 39 3a 33 16 Nov 2 017 19:3
30 3a 32 34 20 2b 30 38 30 30 0d 0a 46 72 6f 6d 0:24 +08 0..From
3a 20 61 70 70 6c 65 20 3c 31 35 33 30 30 32 34 : apple <1530024
30 30 30 34 40 66 75 64 61 6e 65 64 75 26 63 0004@fud an.edu.c
6e 3e 0d 0a 55 73 65 72 2d 41 67 65 64 74 3a 20 n>..User-Agent:
4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 4d 61 63 Mozilla/ 5.0 (Mac
69 6e 74 6f 73 68 3b 20 49 6e 74 65 6c 20 4d 61 intos; Intel Ma
63 20 4f 53 20 58 20 31 30 2e 31 33 3b 20 72 76 c OS X 0.13; rv
3a 32 34 2e 30 29 20 47 65 63 6b 6f 2f 32 30 31 :24.0) G ecko/201
30 30 31 30 31 20 54 68 75 6e 64 65 72 62 69 72 00101 Th underbir
64 2f 32 34 2e 33 2e 30 0d 0a 4d 49 4d 45 2d 56 d/24.3.0 ..MIME-V
65 72 73 69 6f 6e 3a 20 31 2e 30 0d 0a 54 6f 3a ersion: 1.0..To:
20 31 35 33 30 30 32 34 30 30 34 40 66 75 64 1530024 0004@fud
61 6e 2e 65 64 75 2e 63 6e 0d 0a 53 75 62 6a 65 an.edu..n..Subje
63 74 3a 20 54 65 73 74 20 4d 61 69 6c 0d 0a 43 ct: Test Mail..C
6f 6e 74 65 2d 54 79 70 65 3a 20 74 65 78 ontent-Type: tex
74 2f 70 6c 61 69 6e 3b 20 63 68 61 72 73 65 74 t/plain; charset
3d 47 42 32 33 31 32 0d 0a 43 6f 6e 74 65 6e 74 =GB2312. .Content
2d 54 72 61 6e 73 66 65 72 2d 45 6e 63 6f 64 69 -Transfe r-Encoding
6e 67 3a 20 37 62 69 74 0d 0a 58 2d 43 4d 2d 54 ng: 7bit ..X-CM-T
52 41 4e 53 49 44 3a 41 51 55 46 43 6b 41 4a 72 RANSID:A QUFCKAJr
30 4c 7a 64 67 31 61 42 76 76 7a 41 67 2d 2d 2e 0Lzdq1aB vVzAg--.
32 39 39 31 30 53 32 0d 0a 58 2d 43 6f 72 65 6d 2991052. .X-Core
61 69 6c 2d 41 6e 74 69 73 70 61 6d 3a 20 31 55 ail-Anti spam: 1U

```

POP3接收邮件：



4. 讨论与研究

1. 以HTTP为关键字，搜索到了128个文档。最新的一个为《HTTP Immutable Responses》，在2017年9月发布。
2. 这里访问http://www.fudan.edu.cn/foobar，服务器返回404 Not Found：

404 Not Found

nginx

Wireshark抓到的HTTP Response包显示：

```

▶ Transmission Control Protocol, Src Port: 80, Dst Port: 56642, Seq: 256, Ack: 372, Len: 162
▶ [2 Reassembled TCP Segments (417 bytes): #21(255), #23(162)]
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.1 404 Not Found\r\n
    Server: nginx\r\n
    Date: Thu, 16 Nov 2017 13:19:41 GMT\r\n
    Content-Type: text/html\r\n
  ▶ Content-Length: 162\r\n
    X-Cache: MISS from 2ndDomainSrv\r\n
    X-Cache-Lookup: MISS from 2ndDomainSrv:80\r\n
    Via: 1.1 2ndDomainSrv (squid)\r\n
    Connection: keep-alive\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.018394000 seconds]
  [Request in frame: 19]
  File Data: 162 bytes
  ▶ Line-based text data: text/html

```

在RFC 2616中，对返回值404这样定义：

10.4.5 404 Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent. The 410 (Gone) status code SHOULD be used if the server knows, through some internally configurable mechanism, that an old resource is permanently unavailable and has no forwarding address. This status code is commonly used when the server does not wish to reveal exactly why the request has been refused, or when no other response is applicable.

3. (1) HTTP-NG开发组在W3C计划中对下一代HTTP的建议（More-is-Better Axes for Evaluating HTTP）包括更好的网络表现、更低的本地处理消耗、更大的可拓展性、模块化、可进化性、表达能力、安全性、更少的约束、降低对TCP协议的依赖等等。

(2) HTTP/1.1 被应用在越来越广泛的环境中，然而HTTP/1.1 的局限性也逐渐体现出来：结构的复杂性、较差的可拓展性、性能方面的问题以及对TCP/IP协议栈的依赖性等问题。

针对这种情况，HTTP-NG 被提出，可以看做下一代的 HTTP，尽管该版本还没有被投入使用，并且 HTTP-NG 工作组已经解散。HTTP-NG 支持模块化，将协议模块化为三层，而不是将连接管理、报文处理、服务器处理逻辑、协议方法全都混在一起；HTTP-NG 支持分布式对象，HTTP-NG 的很多基本原理和功能目录都是从 CORBA 和 DCOM 这样的结构化、面向对象的分布式对象系统中借鉴来的；HTTP-NG 支持 WebMUX 标准，WebMUX 是一个复杂的高性能报文系统，通过该系统，可以在一个复用的 TCP 连接上并行地传输报文，可以对不同速度产生和消耗的独立报文流进行高效的分组，并将其复用到一条或少数几条 TCP 连接上去；HTTP-NG 支持二进制连接协议，使用二进制连接协议来提高下一代 HTTP 协议支持远程操作的能力。

HTTP-NG 将协议模块化为三层：报文传输层、远程调用层（RPC）和 Web 应用层。第一层为报文传输层，不考虑报文的功能，而注重考虑报文的高效传输；第二层为远程调用层，定义了请求/响应的功能，客户端可以通过这些功能调用对服务器资源的操作；第三层为 Web 应用层，提供了大部分的内容管理逻辑，所有的 HTTP 方法和首部参数都是在这里定义的。

(3) HTTP-NG 并没有被推广或者投入使用，工作组也已经解散，HTTP-NG 的目标并没有达到。

4. 如果FTP客户端尝试使用命令CD进入一个不存在的目录，FTP服务器返回550：

```

▶ Internet Protocol Version 4, Src: 10.131.243.51, Dst: 10.221.99.83
▶ Transmission Control Protocol, Src Port: 21, Dst Port: 58640, Seq: 1, Ack: 14, Len: 49
▼ File Transfer Protocol (FTP)
  ▼ 550 The system cannot find the file specified. \r\n
      Response code: Requested action not taken: File unavailable (550)
      Response arg: The system cannot find the file specified.

```

550表示请求的操作无法执行，文件不可用（例如找不到文件，无访问权限）。

为了防止出现访问不存在或者错误的目录的情况，客户端会利用控制通道发送LIST指令；服务器端从数据通道发送一个包括有效路径和文件规范的列表。

如果FTP客户端尝试使用命令CWD进入一个不存在的目录，FTP服务器返回550：

```

575 2.593347 10.221.99.83      10.131.243.51      FTP      79 Request: CWD /foobar
576 2.596878 10.131.243.51      10.221.99.83      FTP      115 Response: 550 The system cannot find the file specified.

▶ Internet Protocol Version 4, Src: 10.131.243.51, Dst: 10.221.99.83
▶ Transmission Control Protocol, Src Port: 21, Dst Port: 59489, Seq: 121, Ack: 47, Len: 49
▼ File Transfer Protocol (FTP)
  ▼ 550 The system cannot find the file specified. \r\n
      Response code: Requested action not taken: File unavailable (550)
      Response arg: The system cannot find the file specified.

```

RFC959中的定义为：

```

550 Requested action not taken.
File unavailable (e.g., file not found, no access).

```

5. 进行试验尝试：

使用telnet指令连接10.131.243.51的21端口，输入用户名和密码后使用PASV进入被动模式：

```

apple — telnet 10.131.243.51 21 — telnet — telnet 10.131.243.51 21 — 80x24
~ telnet 10.131.243.51 21
Trying 10.131.243.51...
Connected to 10.131.243.51.
Escape character is '^].
220 Microsoft FTP Service
user anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
pass
230 User logged in.
type I
200 Type set to I.
pasv
227 Entering Passive Mode (10,131,243,51,43,54).

```

可以看见，被动模式下服务器开通的端口号为 $43 * 256 + 54 = 11062$ 。另开一个终端窗口，尝试连接这个端口：

```
[*] ~ telnet 10.131.243.51 11062 — telnet — telnet 10.131.243.51 11062 — 80x24
[*] ~ telnet 10.131.243.51 11062
Trying 10.131.243.51...
Connected to 10.131.243.51.
Escape character is '^]'.
```

可以发现，这个端口被成功连接。接下来，用控制通道发送RETR指令取回文件，服务器成功响应，并在数据通道的连接中显示了文件内容：

```
retr readme.txt
125 Data connection already open; Transfer starting.

[*] ~ telnet 10.131.243.51 11062
Trying 10.131.243.51...
Connected to 10.131.243.51.
Escape character is '^]'.
computer networks-ftp protocalConnection closed by foreign host.
```

这说明在相同IP地址下，使用数据连接可以获得文件内容。尝试使用具有不同IP地址的主机进行数据连接：

```
[*] ~ telnet 10.131.243.51 11221
Trying 10.131.243.51...
Connected to 10.131.243.51.
Escape character is '^]'.
Connection closed by foreign host.
```

连接会被服务器自动关闭，这个端口便不可再用。

经过查阅，发现有的FTP服务器在PASV模式中向客户端提供连接端口后，对这个连接的IP地址可能不会进行检查，这样就有可能遭到攻击；攻击者可以猜出相应的端口号（一般是递增的），通过这个端口建立数据连接获得相应的数据。这种攻击称为端口盗用。

FTP协议没有阻止这种情况的发生。

这种攻击是自动化的，同样，处理这种攻击也可以自动化，比如强制检测IP地址、随机选择数据连接端口号等。

FTP协议还有其他漏洞，比如跳转攻击：FTP主动模式中，客户端向服务器发送port命令，指定IP地址和端口号；服务器收到后，会从80端口发起数据连接。PORT命令格式如下：PORT A. B. C. D. E. F。服务器解析后，参数的前四个字段，作为地址A. B. C. D，对应的数据连接的目标端口为E*256+F。FTP的这个特性，被利用的话，可以导致安全问题：假设A. B. C. D是一个邮件服务器，那么如果E, F的值算出来刚好又是SMTP服务的端口的话，此时通过FTP命令下载一个文件，那么FTP服务器就会自动连接A. B. C. D的SMTP端口，然后将文件内容发往这个端口。刚好此时如果文件内容是经过精心构造，满足SMTP协议格式的话，那么就可以用来进行攻击了。

6. 使用Web发送邮件时，浏览器先通过HTTP协议，用80端口与Web服务器交互；Web服务器再通过SMTP协议，用25端口与邮件服务器联系，之后便是正常的SMTP邮件发送；使用Web接收邮件时，浏览器先通过HTTP协议，用80端口与Web服务器交互，Web服务器再通过POP3协议，用110端口与邮件服务器联系并取回邮

件。整个过程与使用邮件客户端十分类似，区别在于用户端不再直接与邮件服务器联系，而是在中间加入了Web服务器作为中转。

尝试使用Wireshark抓包，但是发现邮件内容被SSL或者TSL加密，无法解析。

7. 可以在Data的MAIL FROM中加入伪造的邮件地址，利用MAIL FROM指令输入伪造的地址以进行欺骗。DATE、SMTP服务器域名等信息可以以同样的方式伪造。不过接收方可以看见发送方真实的IP地址。

可以欺骗接收方的垃圾邮件过滤系统、伪造地址进行钓鱼欺诈等。

防护方法有检查邮件来源IP、检查邮件发送的域名、反向DNS查询、登录验证等等。

比较新的反垃圾邮件的技术有邮件电子签名、实时黑名单系统、机器学习方法（如朴素贝叶斯等）等等。这些方法取得了比较好的成果，但是只要垃圾邮件能带来（合法或非法）利润，就会有人尝试新的技术，垃圾邮件很难被终结。

5. 参考资料

1. RFC相关文档
2. CSDN技术博客
3. 《数据通信与计算机网络》课件
4. 计算机网络实验教程 (Computer Networking Internet Protocols in Action) Jeanna Matthews著，李毅超、曹跃、王钰等译，人民邮电出版社