



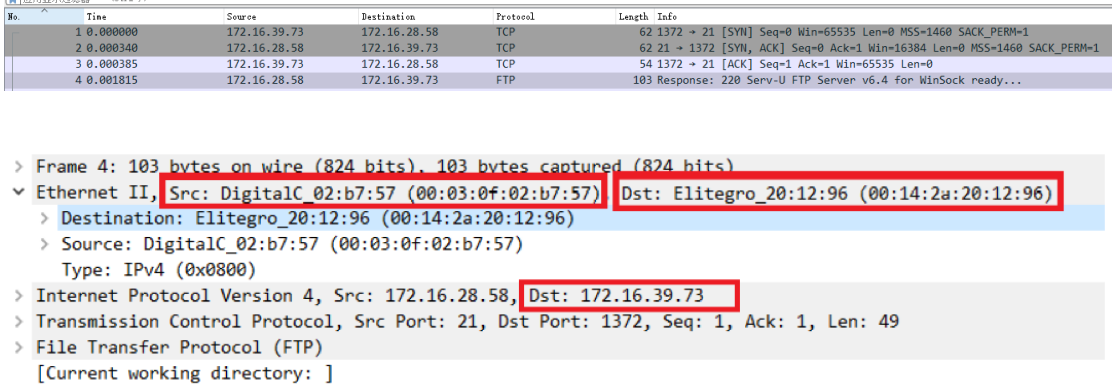
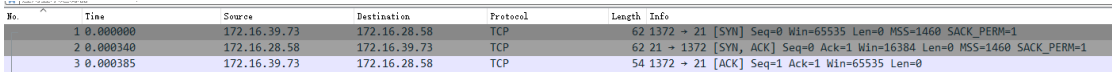
警

- 1.实验报告如有雷同，雷同各方当次实验成绩均以 0 分计。
- 2.当次小组成员成绩只计学号、姓名登录在下表中的。
- 3.在规定时间内未上交实验报告的，不得以其他方式补交，当次成绩按 0 分计。
- 4.实验报告文件以 PDF 格式提交。

院系	计算机学院	班 级	19 级软件工程	组长	冼子婷
学号	18338072	18346019	18322043		
学生	冼子婷	胡文浩	廖雨轩		

Ftp 协议分析实验

一、打开“FTP 数据包”的“ftp 例 1.cap”文件，进行观察分析，回答以下问题(见附件)

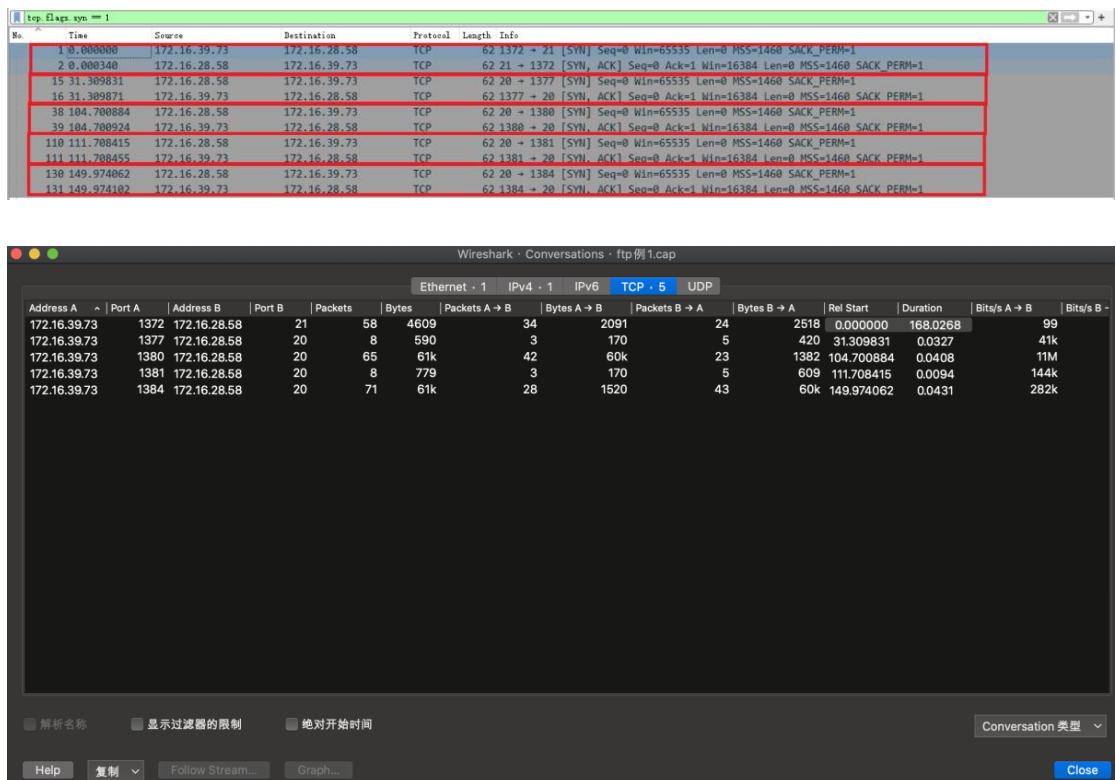
题号	
1	FTP 客户端的 mac 地址是多少？
答案	00:14:2a:20:12:96
截图	 <p>Wireshark packet capture showing Ethernet II frame details. The source MAC address is 00:03:0f:02:b7:57 and the destination MAC address is 00:14:2a:20:12:96.</p>
分析	TCP 的连接和建立都是采用客户/服务器的方式。主动发起建立连接的应用进程称为客户，被动等待连接建立的应用进程称为服务器。发送建立 TCP 连接请求的是 172.16.39.73 地址的机器，因此该机器是客户端，其 mac 地址为 00:14:2a:20:12:96
2	第 1、2、3 号报文的作用是什么？
答案	客户端 172.16.39.73 和服务端 172.16.28.58 用三次握手建立 TCP 连接。具体来看：1 号报文是客户端首先在服务器 21 号端口与服务器端发起一个用于控制的 TCP 连接，并且把自己的 MSS (Maximum Segment Size) 告诉对方。2 号报文是服务端收到连接请求报文端后，同意并发回确认。3 号报文是客户端收到服务端的确认后，建立 TCP 连接。
截图	 <p>Wireshark packet capture showing the first three packets of the TCP connection. Packet 1 is a SYN request from 172.16.39.73 to 172.16.28.58. Packet 2 is a SYN/ACK response from 172.16.28.58 to 172.16.39.73. Packet 3 is an ACK response from 172.16.39.73 to 172.16.28.58.</p>
分析	<p>TCP 连接建立的过程：</p> <p>客户端的 TCP 向服务器发送连接请求报文段，一个建立连接的同步 (SYN) 请求。</p> <p>服务器的 TCP 收到连接请求报文段后，如同意则发回确认 (SYN/ACK) 应答。</p> <p>客户端收到此报文段后，向服务器给出确认 (ACK)。</p> <p>也就是俗称的 TCP 连接“三次握手”过程。</p>



3 该数据包中共有多少个 TCP 流？

答案 5 个

截图



分析

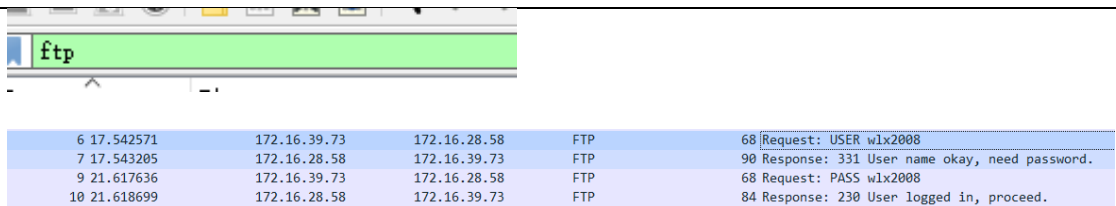
第一种方法：SYN 置为 1 时表示一个 TCP 连接请求，通过筛选 SYN 为 1 的字段，可以筛选出客户端向服务器发出连接请求报文段和服务器的 TCP 收到连接请求报文段后同意的报文段。可以看出有 5 次 TCP 连接请求和连接同意，即共有 5 个 TCP 流。

第二种方法，也可以通过 Wireshark 通过两端的 IP 加 port 过滤出一个 TCP/UDP 流，即通过 Wireshark 的 Statistics->Conversations，点击 TCP 标签即可看到所有的 TCP 流。

4 用什么用户和密码登录成功？

答案 USER: wlx2008
PASS: wlx2008

截图



分析

由于 ftp 协议是以明文方式发送用户名和口令，只需要筛选出 ftp 的数据包即可捕获到用户名和用户密码

5 该 FTP 的命令连接和数据连接分别是什么样的连接？

答案 第一次连接属于命令连接，后四次连接都是数据连接



截图

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.39.73	172.16.28.58	TCP	62	1372 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
2	0.000340	172.16.28.58	172.16.39.73	TCP	62	21 → 1372 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
15	31.309831	172.16.28.58	172.16.39.73	TCP	62	20 → 1377 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
16	31.309871	172.16.39.73	172.16.28.58	TCP	62	1377 → 20 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
38	104.780884	172.16.28.58	172.16.39.73	TCP	62	20 → 1380 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
39	104.780924	172.16.39.73	172.16.28.58	TCP	62	1380 → 20 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
110	111.780415	172.16.28.58	172.16.39.73	TCP	62	20 → 1381 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
111	111.780455	172.16.39.73	172.16.28.58	TCP	62	1381 → 20 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
130	149.974062	172.16.28.58	172.16.39.73	TCP	62	20 → 1384 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
131	149.974102	172.16.39.73	172.16.28.58	TCP	62	1384 → 20 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1

分析

FTP 是一个客户/服务端系统，客户端和服务端通过两个连接进行通信，其一是控制连接，客户端发出 FTP 命令，服务器给出应答。在命令连接中，FTP 服务器使用的端口号是 21。其二是数据连接，真正的文件传输是在这个连接上进行的。服务器端的数据连接端口号是 20。

6

该 FTP 的连接模式是那种？为什么？

答案

主动模式，因为数据连接是由服务器端主动发起的，客户端利用控制连接将客户端端口号通告给服务器，客户端发送 PORT 命令。

截图

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.39.73	172.16.28.58	TCP	62	1372 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
2	0.000340	172.16.28.58	172.16.39.73	TCP	62	21 → 1372 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
3	0.000385	172.16.39.73	172.16.28.58	TCP	54	1372 → 21 [ACK] Seq=1 Ack=1 Win=65535 Len=0
4	0.001815	172.16.28.58	172.16.39.73	FTP	103	Response: 220 Serv-U FTP Server v6.4 for WinSock ready...
5	0.201287	172.16.39.73	172.16.28.58	TCP	54	1372 → 21 [ACK] Seq=1 Ack=50 Win=65486 Len=0
6	17.542571	172.16.39.73	172.16.28.58	FTP	68	Request: USER wlx2008
7	17.543205	172.16.28.58	172.16.39.73	FTP	90	Response: 331 User name okay, need password.
8	17.670784	172.16.39.73	172.16.28.58	TCP	54	1372 → 21 [ACK] Seq=15 Ack=86 Win=65450 Len=0
9	21.617636	172.16.39.73	172.16.28.58	FTP	68	Request: PASS wlx2008
10	21.618699	172.16.28.58	172.16.39.73	FTP	84	Response: 230 User logged in, proceed.
11	21.733350	172.16.39.73	172.16.28.58	TCP	54	1372 → 21 [ACK] Seq=29 Ack=116 Win=65420 Len=0
12	31.305692	172.16.39.73	172.16.28.58	FTP	78	Request: PORT 172,16,39,73,5,97
13	31.306179	172.16.28.58	172.16.39.73	FTP	84	Response: 200 PORT Command successful.
14	31.306798	172.16.39.73	172.16.28.58	FTP	62	Request: QUIT
15	31.306838	172.16.28.58	172.16.39.73	FTP	62	Response: 221 Goodbye!

分析

FTP 的数据连接支持两种模式：主动模式和被动模式，区别在于数据连接是由谁发起的。

主动模式即 Port 方式，收到数据传送请求后，服务器主动与客户端建立连接。服务器必须获得客户端的端口号，在此模式下，客户端利用控制连接，将客户端号通告给服务器。客户端发送的命令是 PORT n1,n2,n3,n4,n5,n6，其中前四位表示客户端的 IP 地址，后两位确定端口号为 $n5*256+n6$ 。

被动模式是发送 Pasv 命令到 FTP 服务器，服务端随机打开一个高端端口，并通知客户端在该端口上传数据的请求。

7

最后四个报文的作用是什么？

答案

最后四个包是四次挥手过程，表示数据传输结束，TCP 连接释放。

截图

No.	Time	Source	Destination	Protocol	Length	Info
198	149.983318	172.16.28.58	172.16.39.73	FTP-DATA	970	FTP Data: 916 bytes (PORT) (RETR 088.xls)
199	149.983346	172.16.39.73	172.16.28.58	TCP	54	1384 → 20 [ACK] Seq=1 Ack=57858 Win=64619 Len=0
200	150.016332	172.16.39.73	172.16.28.58	TCP	54	1384 → 20 [FIN, ACK] Seq=1 Ack=57858 Win=64619 Len=0
201	150.017147	172.16.28.58	172.16.39.73	TCP	60	20 → 1384 [ACK] Seq=57858 Ack=2 Win=65535 Len=0
202	150.113091	172.16.39.73	172.16.28.58	TCP	54	1372 → 21 [ACK] Seq=242 Ack=1060 Win=64476 Len=0
203	150.113474	172.16.28.58	172.16.39.73	FTP	183	Response: 226-Maximum disk quota limited to 307200 kbytes
204	150.316222	172.16.39.73	172.16.28.58	TCP	54	1372 → 21 [ACK] Seq=242 Ack=1189 Win=64347 Len=0
205	168.024267	172.16.39.73	172.16.28.58	FTP	60	Request: QUIT
206	168.024673	172.16.28.58	172.16.39.73	FTP	68	Response: 221 Goodbye!
207	168.026381	172.16.39.73	172.16.28.58	TCP	54	1372 → 21 [FIN, ACK] Seq=248 Ack=1203 Win=64333 Len=0
208	168.026708	172.16.28.58	172.16.39.73	TCP	60	21 → 1372 [ACK] Seq=1203 Ack=249 Win=65288 Len=0
209	168.026762	172.16.28.58	172.16.39.73	TCP	60	21 → 1372 [FIN, ACK] Seq=1203 Ack=249 Win=65288 Len=0
210	168.026800	172.16.39.73	172.16.28.58	TCP	54	1372 → 21 [ACK] Seq=249 Ack=1204 Win=64333 Len=0

分析

最后四个数据包前两个数据包中可以看出，客户端向服务端发送 quit 请求。

TCP 连接的释放需要双方都发送释放连接的报文，等待对方确认。

即客户端释放连接，置 FIN 位，服务端发送 ACK 确认。

服务器释放连接置 FIN 位，客户端发送 ACK 确认。

8

该数据包中有多少个 ftp 的命令及应答，其含义分别是什么？



答案

共十次命令和响应，其中分别为

命令（Request）：

USER：指定登陆的用户名，以便服务器进行身份验证

PASS：指定用户口令，该命令必须跟在登陆用户命令之后

PORT：该命令告诉 FTP 服务器，客户端监听的端口号是 address，让 FTP 服务器采用主动模式连接客户端

XMKD：新建目录

RNFR：重新命名文件，该命令的下一条命令应该用 RNTD 指定新的文件名

RNTD：该命令和 RNFR 命令共同完成对文件的重命名，紧跟在 RNFR 命令后

STOR：上传一个指定的文件，并将其存储在指定的位置

NLST：返回指定路径下的目录列表，省略<路径>时，返回当前目录

RETR：请求服务器将指定路径内的文件复制到客户端，也即下载指定的文件

QUIT：关闭与服务器的连接

应答（Response）：

220：客户端与服务端完成连接的建立

331：用户名正确，需要用户密码

230：用户登入成功

200：命令执行成功

150：文件状态正常，开始数据连接

226：结束数据连接，数据传输完成

257：路径创建成功

350：请求文件成功，需要下一步的操作命令

250：请求文件操作完成

221：服务端断开控制连接

截图

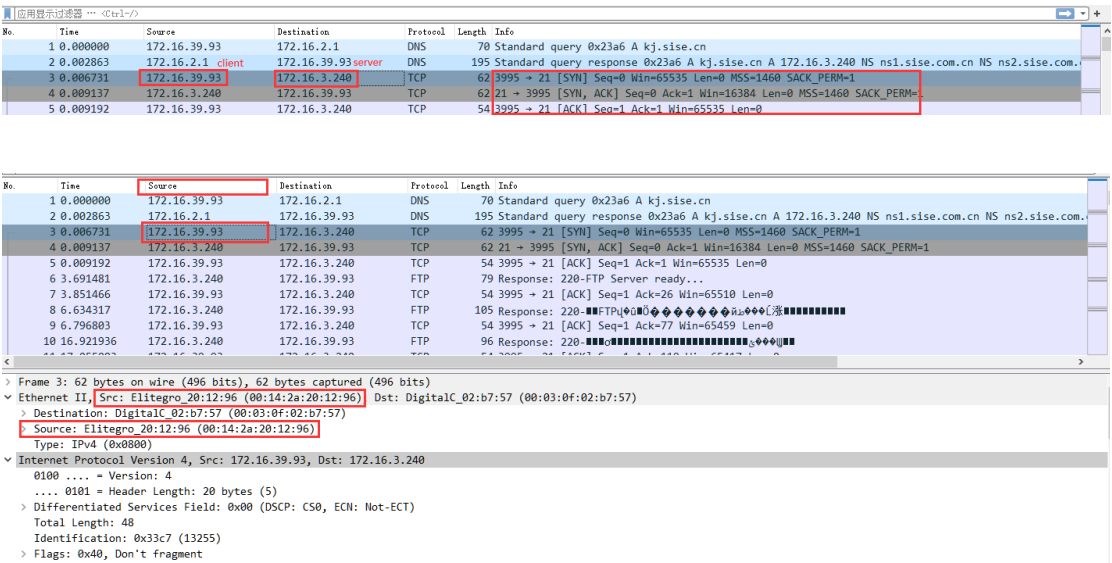
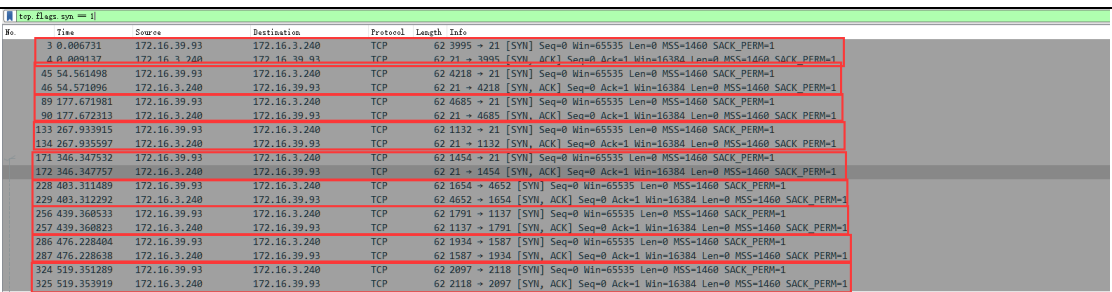
No.	Time	Source	Destination	Protocol	Length	Info
4	0.001815	172.16.28.58	172.16.39.73	FTP	103	Response: 220 Serv-U FTP Server v6.4 for Winsock ready...
6	17.542571	172.16.39.73	172.16.28.58	FTP	68	Request: USER wlx2008
7	17.543205	172.16.28.58	172.16.39.73	FTP	90	Response: 331 User name okay, need password.
9	21.617636	172.16.39.73	172.16.28.58	FTP	68	Request: PASS wlx2008
10	21.618699	172.16.28.58	172.16.39.73	FTP	84	Response: 230 User logged in, proceed.
12	31.305692	172.16.39.73	172.16.28.58	FTP	78	Request: PORT 172,16,39,73,5,97
13	31.306179	172.16.28.58	172.16.39.73	FTP	84	Response: 200 PORT Command successful.
14	31.308878	172.16.39.73	172.16.28.58	FTP	63	Request: NLST -l
18	31.310880	172.16.28.58	172.16.39.73	FTP	107	Response: 150 Opening ASCII mode data connection for /bin/ls.
25	31.484083	172.16.28.58	172.16.39.73	FTP	182	Response: 226-Maximum disk quota limited to 307200 kBytes
27	42.200128	172.16.39.73	172.16.28.58	FTP	64	Request: XMKD jjj
28	42.201268	172.16.28.58	172.16.39.73	FTP	85	Response: 257 "/jjj" directory created.
30	54.715458	172.16.39.73	172.16.28.58	FTP	64	Request: RNFR jjj
31	54.716541	172.16.28.58	172.16.39.73	FTP	112	Response: 350 File or directory exists, ready for destination name
32	54.720019	172.16.39.73	172.16.28.58	FTP	64	Request: RNTD ppp
33	54.723253	172.16.28.58	172.16.39.73	FTP	84	Response: 250 RNTD command successful.
35	104.695575	172.16.39.73	172.16.28.58	FTP	79	Request: PORT 172,16,39,73,5,100
36	104.696037	172.16.28.58	172.16.39.73	FTP	84	Response: 200 PORT Command successful.
37	104.698520	172.16.39.73	172.16.28.58	FTP	73	Request: STOR xs2009-9.xls
41	104.701805	172.16.28.58	172.16.39.73	FTP	112	Response: 150 Opening ASCII mode data connection for xs2009-9.xls.
105	104.814922	172.16.28.58	172.16.39.73	FTP	183	Response: 226-Maximum disk quota limited to 307200 kBytes
107	111.703852	172.16.39.73	172.16.28.58	FTP	79	Request: PORT 172,16,39,73,5,101
108	111.704411	172.16.28.58	172.16.39.73	FTP	84	Response: 200 PORT Command successful.
109	111.707423	172.16.39.73	172.16.28.58	FTP	63	Request: NLST -l
113	111.709282	172.16.28.58	172.16.39.73	FTP	107	Response: 150 Opening ASCII mode data connection for /bin/ls.
120	111.822991	172.16.39.73	172.16.28.58	FTP	183	Response: 226-Maximum disk quota limited to 307200 kBytes
122	131.649709	172.16.39.73	172.16.28.58	FTP	73	Request: RNFR xs2009-9.xls
123	131.650613	172.16.28.58	172.16.39.73	FTP	112	Response: 350 File or directory exists, ready for destination name
124	131.654130	172.16.39.73	172.16.28.58	FTP	68	Request: RNTD 888.xls
125	131.657140	172.16.28.58	172.16.39.73	FTP	84	Response: 250 RNTD command successful.
127	149.968452	172.16.39.73	172.16.28.58	FTP	79	Request: PORT 172,16,39,73,5,104
128	149.968908	172.16.28.58	172.16.39.73	FTP	84	Response: 200 PORT Command successful.
129	149.972714	172.16.39.73	172.16.28.58	FTP	68	Request: RETR 888.xls
133	149.975126	172.16.28.58	172.16.39.73	FTP	121	Response: 150 Opening ASCII mode data connection for 888.xls (57856 Bytes).
203	150.113474	172.16.28.58	172.16.39.73	FTP	183	Response: 226-Maximum disk quota limited to 307200 kBytes
205	168.024267	172.16.39.73	172.16.28.58	FTP	60	Request: QUIT
206	168.024673	172.16.28.58	172.16.39.73	FTP	68	Response: 221 Goodbye!

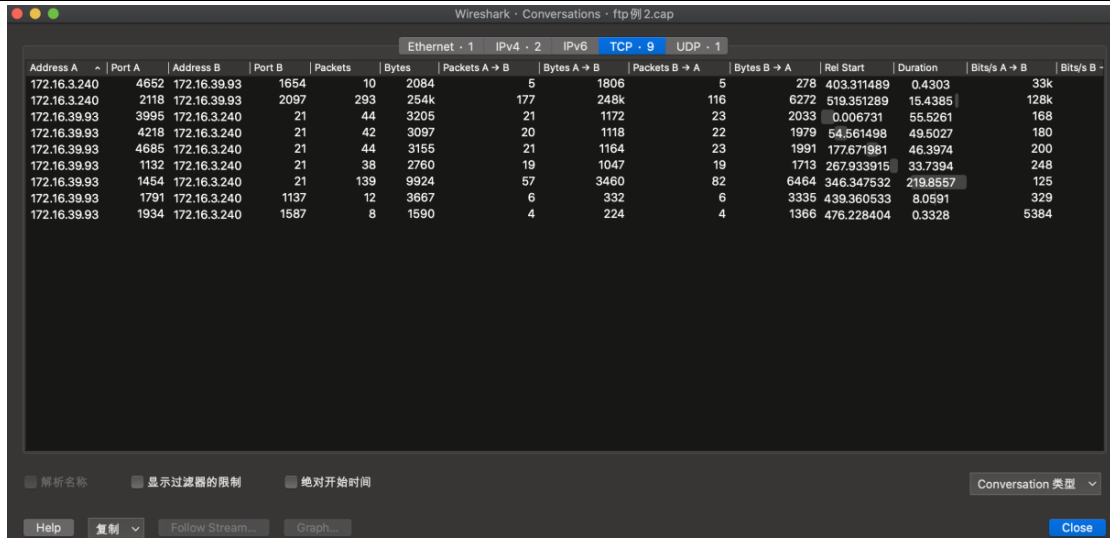
分析

筛选出通过 ftp 协议传送的数据包，查看通过 ftp 发起的命令请求和对应的响应



二、打开“FTP 数据包”的“ftp 例 2.cap”文件，进行观察分析，回答以下问题

题号	
1	FTP 服务器的 ip 是多少？FTP 客户端的 mac 地址是多少？
答案	服务器 IP 地址：172.16.3.240 客户端 mac 地址：00:14:2a:20:12:96
截图	 <p>The screenshot shows a Wireshark capture of an FTP session. The packet list pane shows several packets. Packet 3 is a SYN packet from 172.16.3.93 to 172.16.3.240. Packet 4 is an ACK packet from 172.16.3.240 to 172.16.3.93. Packet 5 is a SYN packet from 172.16.3.93 to 172.16.3.240. The packet details pane for packet 3 shows the Ethernet II and Internet Protocol Version 4 fields. The Ethernet II field shows the source MAC address as 00:14:2a:20:12:96 and the destination MAC address as 02:00:00:00:00:00. The Internet Protocol Version 4 field shows the source IP as 172.16.3.93 and the destination IP as 172.16.3.240.</p>
分析	通过 SYN 置位的标志，找到建立 TCP 连接的三次握手过程，其中 source 是发起连接的客户端，destination 是服务器。
2	该数据包中共有多少个 TCP 流？
答案	9 个
截图	 <p>The screenshot shows a Wireshark capture of an FTP session. The packet list pane shows 32 packets. All packets are TCP flows between 172.16.3.93 and 172.16.3.240. The packet details pane for packet 3 shows the Ethernet II and Internet Protocol Version 4 fields. The Ethernet II field shows the source MAC address as 00:14:2a:20:12:96 and the destination MAC address as 02:00:00:00:00:00. The Internet Protocol Version 4 field shows the source IP as 172.16.3.93 and the destination IP as 172.16.3.240.</p>



Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
172.16.3.240	4652	172.16.3.240	1654	10	2084	5	1806	5	278	403.311489	0.4303	33k	
172.16.3.240	2118	172.16.3.240	2097	293	254k	177	248k	116	6272	519.351289	15.4385	128k	
172.16.3.240	3995	172.16.3.240	21	44	3205	21	1172	23	2033	0.006731	55.5261	168	
172.16.3.240	4218	172.16.3.240	21	42	3097	20	1118	22	1979	54.561498	49.5027	180	
172.16.3.240	4685	172.16.3.240	21	44	3155	21	1164	23	1991	177.671981	46.3974	200	
172.16.3.240	1132	172.16.3.240	21	38	2760	19	1047	19	1713	267.933915	33.7394	248	
172.16.3.240	1454	172.16.3.240	21	139	9924	57	3460	82	6464	346.347532	219.8557	125	
172.16.3.240	1791	172.16.3.240	1137	12	3667	6	332	6	3335	439.360533	8.0591	329	
172.16.3.240	1934	172.16.3.240	1587	8	1590	4	224	4	1366	476.228404	0.3328	5384	

分析 SYN 置为 1 时表示一个 TCP 连接请求，通过筛选 SYN 为 1 的字段，可以筛选出客户端向服务器发出连接请求报文段和服务器的 TCP 收到连接请求报文段后同意的报文段。

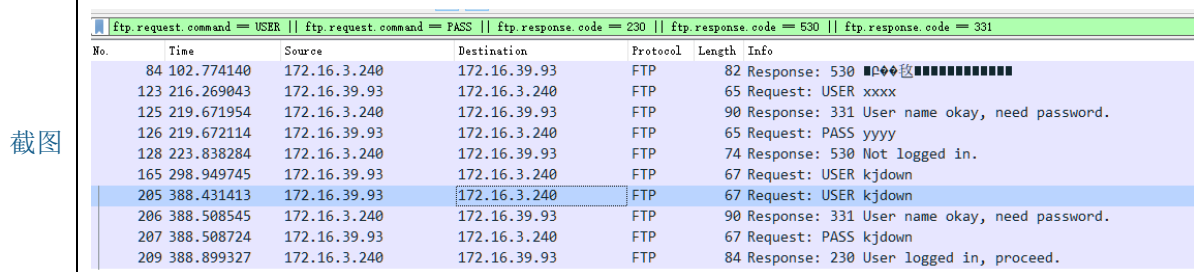
可以看出有 9 次 TCP 连接请求和连接同意，即共有 9 个 TCP 流。

或者在 Wireshark 的 Statistics->Conversations，点击 TCP 标签即可看到所有的 TCP 流

3 最后用什么用户和密码登录成功？

答案 USER: kjdown

PASS: kjdown



No.	Time	Source	Destination	Protocol	Length	Info
84	102.774140	172.16.3.240	172.16.3.240	FTP	82	Response: 530 Not logged in.
123	216.269043	172.16.3.240	172.16.3.240	FTP	65	Request: USER xxxx
125	219.671954	172.16.3.240	172.16.3.240	FTP	90	Response: 331 User name okay, need password.
126	219.672114	172.16.3.240	172.16.3.240	FTP	65	Request: PASS yyyy
128	223.838284	172.16.3.240	172.16.3.240	FTP	74	Response: 530 Not logged in.
165	298.949745	172.16.3.240	172.16.3.240	FTP	67	Request: USER kjdown
205	388.431413	172.16.3.240	172.16.3.240	FTP	67	Request: USER kjdown
206	388.508545	172.16.3.240	172.16.3.240	FTP	90	Response: 331 User name okay, need password.
207	388.508724	172.16.3.240	172.16.3.240	FTP	67	Request: PASS kjdown
209	388.899327	172.16.3.240	172.16.3.240	FTP	84	Response: 230 User logged in, proceed.

分析 通过过滤出 ftp 客户端的请求（USER 和 PASS）和服务器的响应码（230 登陆成功，331 用户名正确需要密码，530 登陆失败），可以过滤出登录的操作，最终使用用户名 kjdown 和密码 kjdown 登录成功。

4 该 FTP 的命令连接和数据连接分别是什么？

答案 命令连接：图中数据包序号小于等于 172 的包

数据连接：图中数据包序号大于等于 228 的包



截图

No.	Time	Source	Destination	Protocol	Length	Info
3	0.006731	172.16.39.93	172.16.3.240	TCP	62	3995 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
4	0.009137	172.16.3.240	172.16.39.93	TCP	62	21 → 3995 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
45	54.561498	172.16.39.93	172.16.3.240	TCP	62	4218 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
46	54.571096	172.16.3.240	172.16.39.93	TCP	62	21 → 4218 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
89	177.671981	172.16.39.93	172.16.3.240	TCP	62	4685 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
90	177.672313	172.16.3.240	172.16.39.93	TCP	62	21 → 4685 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
133	267.933915	172.16.39.93	172.16.3.240	TCP	62	1132 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
134	267.935597	172.16.3.240	172.16.39.93	TCP	62	21 → 1132 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
171	346.347532	172.16.39.93	172.16.3.240	TCP	62	1454 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
172	346.347757	172.16.3.240	172.16.39.93	TCP	62	21 → 1454 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
228	403.311489	172.16.39.93	172.16.3.240	TCP	62	1654 → 4652 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
229	403.312292	172.16.3.240	172.16.39.93	TCP	62	4652 → 1654 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
256	439.360533	172.16.39.93	172.16.3.240	TCP	62	1791 → 1137 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
257	439.360823	172.16.3.240	172.16.39.93	TCP	62	1137 → 1791 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
286	476.228404	172.16.39.93	172.16.3.240	TCP	62	1934 → 1587 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
287	476.228638	172.16.3.240	172.16.39.93	TCP	62	1587 → 1934 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
324	519.351289	172.16.39.93	172.16.3.240	TCP	62	2097 → 2118 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
325	519.353919	172.16.3.240	172.16.39.93	TCP	62	2118 → 2097 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1

205	388.431413	172.16.3.240	172.16.39.93	FTP	67	Request: USER kjdown
206	388.508545	172.16.3.240	172.16.39.93	FTP	90	Response: 331 User name okay, need password.
207	388.508724	172.16.3.240	172.16.39.93	FTP	67	Request: PASS kjdown
209	388.899327	172.16.3.240	172.16.39.93	FTP	84	Response: 230 User logged in, proceed.

分析

通过筛选同步位 SYN 置为 1 的数据包，可以得到连接请求或连接接受的报文。由于控制连接中 FTP 服务器使用的端口号是 21，连接由客户端发起。图中序号小于等于 172 号的数据包都访问到了服务器的 21 端口，即命令连接。

由于本次 FTP 连接模式是被动模式，客户端的数据连接端口是随机的，FTP 服务器收到 Pasv 命令后随机打开一个高端端口并且通知客户端在该端口上传送数据的请求。并且在 205-209 数据包之后，用户登陆成功后才能进行数据连接，所以图中序列号大于等于 228 的数据包进行的 4 次连接是数据连接。

5 哪几个报文是 FTP 数据连接的三次握手报文？

答案

- ① 228 229 230
- ② 256 257 258
- ③ 286 287 288
- ④ 324 325 326

截图

228	403.311489	172.16.39.93	172.16.3.240	TCP	62	1654 → 4652 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
229	403.312292	172.16.3.240	172.16.39.93	TCP	62	4652 → 1654 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
230	403.312346	172.16.39.93	172.16.3.240	TCP	54	1654 → 4652 [ACK] Seq=1 Ack=1 Win=65535 Len=0

256	439.360533	172.16.39.93	172.16.3.240	TCP	62	1791 → 1137 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
257	439.360823	172.16.3.240	172.16.39.93	TCP	62	1137 → 1791 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
258	439.360876	172.16.39.93	172.16.3.240	TCP	54	1791 → 1137 [ACK] Seq=1 Ack=1 Win=65535 Len=0

286	476.228404	172.16.39.93	172.16.3.240	TCP	62	1934 → 1587 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
287	476.228638	172.16.3.240	172.16.39.93	TCP	62	1587 → 1934 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
288	476.228669	172.16.39.93	172.16.3.240	TCP	54	1934 → 1587 [ACK] Seq=1 Ack=1 Win=65535 Len=0

324	519.351289	172.16.39.93	172.16.3.240	TCP	62	2097 → 2118 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
325	519.353919	172.16.3.240	172.16.39.93	TCP	62	2118 → 2097 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
326	519.353959	172.16.39.93	172.16.3.240	TCP	54	2097 → 2118 [ACK] Seq=1 Ack=1 Win=65535 Len=0

分析

FTP 的控制连接不论是主动模式还是被动模式都会涉及到服务端的 21 端口。因此找出同步位 SYN 置位且不涉及 21 端口的数据包，并且由于本次 FTP 连接模式是被动模式，客户端的数据连接端口是随机的，按照三次握手的规则即可找出。

6 哪几个报文是 FTP 数据连接的挥手报文（结束报文）？

答案

- ① 237 238 239 240
- ② 270 271 272 273
- ③ 293 294 296 297

④ 620 621 622 623

截图

No.	Time	Source	Destination	Protocol	Length	Info
43	54.427655	172.16.3.93	172.16.3.240	TCP	54	3995 → 21 [FIN, ACK] Seq=31 Ack=760 Win=64776 Len=0
48	55.532799	172.16.3.240	172.16.3.93	TCP	60	21 → 3995 [FIN, ACK] Seq=760 Ack=32 Win=65505 Len=0
85	102.808153	172.16.3.93	172.16.3.240	TCP	54	4218 → 21 [FIN, ACK] Seq=31 Ack=760 Win=64776 Len=0
87	104.064150	172.16.3.240	172.16.3.93	TCP	60	21 → 4218 [FIN, ACK] Seq=760 Ack=32 Win=65505 Len=0
129	223.838438	172.16.3.93	172.16.3.240	TCP	54	4685 → 21 [FIN, ACK] Seq=23 Ack=718 Win=64818 Len=0
131	224.069301	172.16.3.240	172.16.3.93	TCP	60	21 → 4685 [FIN, ACK] Seq=718 Ack=24 Win=65513 Len=0
167	301.669758	172.16.3.240	172.16.3.93	TCP	60	21 → 1132 [FIN, ACK] Seq=662 Ack=14 Win=65522 Len=0
169	301.670062	172.16.3.93	172.16.3.240	TCP	54	1132 → 21 [FIN, ACK] Seq=14 Ack=663 Win=64874 Len=0
237	403.735946	172.16.3.240	172.16.3.93	TCP	60	4652 → 1654 [FIN, ACK] Seq=1517 Ack=1 Win=65535 Len=0
239	403.736121	172.16.3.93	172.16.3.240	TCP	54	1654 → 4652 [FIN, ACK] Seq=1 Ack=1518 Win=65535 Len=0
270	447.419304	172.16.3.240	172.16.3.93	TCP	60	1137 → 1791 [FIN, ACK] Seq=2992 Ack=1 Win=65535 Len=0
272	447.419475	172.16.3.93	172.16.3.240	TCP	54	1791 → 1137 [FIN, ACK] Seq=1 Ack=2993 Win=65464 Len=0
293	476.501474	172.16.3.240	172.16.3.93	TCP	60	1587 → 1934 [FIN, ACK] Seq=1131 Ack=1 Win=65535 Len=0
296	476.561030	172.16.3.93	172.16.3.240	TCP	54	1934 → 1587 [FIN, ACK] Seq=1 Ack=1132 Win=64405 Len=0
620	534.787848	172.16.3.240	172.16.3.93	TCP	60	2118 → 2097 [FIN, ACK] Seq=239105 Ack=1 Win=65535 Len=0
622	534.788371	172.16.3.93	172.16.3.240	TCP	54	2097 → 2118 [FIN, ACK] Seq=1 Ack=239106 Win=65535 Len=0
629	565.983884	172.16.3.93	172.16.3.240	TCP	54	1454 → 21 [FIN, ACK] Seq=375 Ack=1843 Win=65161 Len=0
631	566.203149	172.16.3.240	172.16.3.93	TCP	60	21 → 1454 [FIN, ACK] Seq=1843 Ack=376 Win=65161 Len=0
237	403.735946	172.16.3.240	172.16.3.93	TCP	60	4652 → 1654 [FIN, ACK] Seq=1517 Ack=1 Win=65535 Len=0
238	403.736017	172.16.3.93	172.16.3.240	TCP	54	1654 → 4652 [ACK] Seq=1 Ack=1518 Win=65535 Len=0
239	403.736121	172.16.3.93	172.16.3.240	TCP	54	1654 → 4652 [FIN, ACK] Seq=1 Ack=1518 Win=65535 Len=0
240	403.741744	172.16.3.240	172.16.3.93	TCP	60	4652 → 1654 [ACK] Seq=1518 Ack=2 Win=65535 Len=0
270	447.419304	172.16.3.240	172.16.3.93	TCP	60	1137 → 1791 [FIN, ACK] Seq=2992 Ack=1 Win=65535 Len=0
271	447.419373	172.16.3.93	172.16.3.240	TCP	54	1791 → 1137 [ACK] Seq=1 Ack=2993 Win=65464 Len=0
272	447.419475	172.16.3.93	172.16.3.240	TCP	54	1791 → 1137 [FIN, ACK] Seq=1 Ack=2993 Win=65464 Len=0
273	447.419643	172.16.3.240	172.16.3.93	TCP	60	1137 → 1791 [ACK] Seq=2993 Ack=2 Win=65535 Len=0
293	476.501474	172.16.3.240	172.16.3.93	TCP	60	1587 → 1934 [FIN, ACK] Seq=1131 Ack=1 Win=65535 Len=0
294	476.501536	172.16.3.93	172.16.3.240	TCP	54	1934 → 1587 [ACK] Seq=1 Ack=1132 Win=64405 Len=0
295	476.541711	172.16.3.93	172.16.3.240	TCP	54	1454 → 21 [ACK] Seq=178 Ack=1362 Win=64174 Len=0
296	476.561030	172.16.3.93	172.16.3.240	TCP	54	1934 → 1587 [FIN, ACK] Seq=1 Ack=1132 Win=64405 Len=0
297	476.561201	172.16.3.240	172.16.3.93	TCP	60	1587 → 1934 [ACK] Seq=1132 Ack=1 Win=65535 Len=0
620	534.787848	172.16.3.240	172.16.3.93	TCP	60	2118 → 2097 [FIN, ACK] Seq=239105 Ack=1 Win=65535 Len=0
621	534.787917	172.16.3.93	172.16.3.240	TCP	54	2097 → 2118 [ACK] Seq=1 Ack=239106 Win=65535 Len=0
622	534.788371	172.16.3.93	172.16.3.240	TCP	54	2097 → 2118 [FIN, ACK] Seq=1 Ack=239106 Win=65535 Len=0
623	534.789817	172.16.3.240	172.16.3.93	TCP	60	2118 → 2097 [ACK] Seq=239106 Ack=2 Win=65535 Len=0

分析

控制连接不论是主动模式还是被动模式都会涉及到服务端的 21 端口。因此找出终止位 FIN 置位且不涉及 21 端口的数据包，并且由于本次 FTP 连接模式是被动模式，客户端的数据连接端口是随机的，按照四次挥手的规则即可找出。

7

该 FTP 的连接模式是那种？为什么？

答案

被动模式

截图

No.	Time	Source	Destination	Protocol	Length	Info
305	506.223658	172.16.3.240	172.16.39.93	FTP	99	Response: 250 Directory changed to /■■■■■■■■■■/
306	506.281431	172.16.39.93	172.16.3.240	FTP	60	Request: noop
308	508.532958	172.16.3.240	172.16.39.93	FTP	73	Response: 200 Command okay.
309	508.533194	172.16.39.93	172.16.3.240	FTP	79	Request: CWD /■■■■■■■■■■/
311	510.594758	172.16.3.240	172.16.39.93	FTP	99	Response: 250 Directory changed to /■■■■■■■■■■/
312	510.595099	172.16.39.93	172.16.3.240	FTP	59	Request: PWD
314	512.965124	172.16.3.240	172.16.39.93	FTP	102	Response: 257 "/■■■■■■■■■■/" is current directory.
315	512.965326	172.16.39.93	172.16.3.240	FTP	79	Request: CWD /■■■■■■■■■■/
317	515.608359	172.16.3.240	172.16.39.93	FTP	99	Response: 250 Directory changed to /■■■■■■■■■■/
318	515.616639	172.16.39.93	172.16.3.240	FTP	62	Request: TYPE I
320	517.493653	172.16.3.240	172.16.39.93	FTP	74	Response: 200 Type set to I.
321	517.494019	172.16.39.93	172.16.3.240	FTP	66	Request: PASV
323	519.286491	172.16.3.240	172.16.39.93	FTP	101	Response: 227 Entering Passive Mode (172,16,3,240,8,70)
327	519.354845	172.16.39.93	172.16.3.240	FTP	87	Request: SIZE ■■■■■■■■■■.doc
329	521.263283	172.16.3.240	172.16.39.93	FTP	66	Response: 213 239104
330	521.361808	172.16.39.93	172.16.3.240	FTP	87	Request: RETR ■■■■■■■■■■.doc
335	523.337233	172.16.3.240	172.16.39.93	FTP	142	Response: 150 Opening BINARY mode data connection for ■■■■■■■■■■.doc (239104 bytes)
624	534.794583	172.16.3.240	172.16.39.93	FTP	78	Response: 226 Transfer complete.
625	534.832176	172.16.39.93	172.16.3.240	FTP	79	Request: CWD /■■■■■■■■■■/
627	535.117956	172.16.3.240	172.16.39.93	FTP	99	Response: 250 Directory changed to /■■■■■■■■■■/

分析

被动模式即 **Pasv** 方式，FTP 的客户端发送 **Pasv** 命令到 FTP 服务器，FTP 服务器收到 **Pasv** 命令后，随机打开一个高端端口，并且通知客户端在该端口上传送数据的请求。

三、在线捕获数据包实验

1. 阅读教材 P64-69 内容, 熟悉 FTP 协议。



2. 完成 P51 的实例 2-1。

【实验内容】

(1) 单击 Wireshark 工具栏左起第一个图标,在接口上开始侦听,片刻后停止侦听。这时捕获的数据量有多少?

(2) 观察捕获数据的源 IP 地址和目的 IP 地址,这些数据是发出的还是发过来的? 选择几个 IP 地址,通过网站 www.ip138.com 查询这些 IP 地址的地理位置。

(3) 查看所在网络的网关 IP 地址,假设查到的 IP 地址是 a. b. c. d,在命令窗口运行 ping -r 6 -l a. b. c. d 和 ping -s 4 -l a. b. c. d 命令并捕获数据包。

(4) 执行 filter: ip. addr==a. b. c. d 命令查看,截屏运行结果。

(5) 捕获的数据中都有哪些协议? 分别找出 Echo 和 Stamp 的请求和响应分组,分析这些数据主要字段的含义。

【解答】

(1)

592 6.363357

在监听过程中使用搜索引擎搜索, 6.36 秒捕获了 592 个数据包

(2)

556 4.788251	202.89.233.101	172.18.198.216
<div><div><input type="text" value="202.89.233.101"/><input type="button" value="X"/><input type="button" value="查询"/></div><div><input type="text" value="172.18.198.216"/><input type="button" value="X"/><input type="button" value="查询"/></div></div> <div><div>202.89.233.101</div><div>172.18.198.216</div></div> <div><div><input type="button" value="转换IPv6地址"/><input type="button" value="IP反查网站"/><input type="button" value="旁站查询"/></div><div><input type="button" value="转换IPv6地址"/><input type="button" value="IP反查网站"/><input type="button" value="旁站查询"/></div></div> <div><div>ASN归属地 北京市海淀区 微软 (中国) 有限公司 微软云</div><div>ASN归属地 本地局域网</div></div>		
540 4.737935	222.200.254.2	172.18.198.216
<div><div><input type="text" value="222.200.254.217"/><input type="button" value="X"/><input type="button" value="查询"/></div><div><input type="text" value="172.18.198.216"/><input type="button" value="X"/><input type="button" value="查询"/></div></div> <div><div>222.200.254.217</div><div>172.18.198.216</div></div> <div><div><input type="button" value="转换IPv6地址"/><input type="button" value="IP反查网站"/><input type="button" value="旁站查询"/></div><div><input type="button" value="转换IPv6地址"/><input type="button" value="IP反查网站"/><input type="button" value="旁站查询"/></div></div> <div><div>ASN归属地 广东省广州市 大学城网络互联汇接中心 教育网</div><div>ASN归属地 本地局域网</div></div>		



491 4.315721

120.241.179.34

172.18.198.216

120.241.179.34

查询

172.18.198.216

查询

120.241.179.34

172.18.198.216

转换IPv6地址

IP反查网站

旁站查询

ASN归属地

广东省深圳市
移动

转换IPv6地址

IP反查网站

旁站查询

ASN归属地

本地局域网

375 2.993409

59.82.33.213

172.18.198.216

59.82.33.213

查询

172.18.198.216

查询

59.82.33.213

172.18.198.216

转换IPv6地址

IP反查网站

旁站查询

ASN归属地

北京市北京市
阿里云 数据中心

转换IPv6地址

IP反查网站

旁站查询

ASN归属地

本地局域网

这些数据都是发过来的

(3)

默认网关为：172.18.198.254

```
PS C:\Users\Adbea> ipconfig
```

```
Windows IP 配置
```

```
以太网适配器 cfw-tap:
```

```
媒体状态 . . . . . : 媒体已断开连接
连接特定的 DNS 后缀 . . . . . :
```

```
以太网适配器 以太网:
```

```
连接特定的 DNS 后缀 . . . . . :
IPv6 地址 . . . . . : 2001:250:3002:4b9d:a9:5c96:ed0d:2957
临时 IPv6 地址 . . . . . : 2001:250:3002:4b9d:80e3:b7ed:4c85:399c
临时 IPv6 地址 . . . . . : 2001:250:3002:4b9d:ec8c:3dcc:cf9b:36b4
本地连接 IPv6 地址 . . . . . : fe80::a9:5c96:ed0d:2957%5
IPv4 地址 . . . . . : 172.18.198.229
子网掩码 . . . . . : 255.255.255.192
默认网关 . . . . . : fe80::5ee8:83ff:fec4:ece9%5
                      172.18.198.254
```



Ping 的命令出现错误-l 需要后面添加一个字节大小的参数, 因为-l 默认为 32 字节大小, 此处添加了 32 作为参数。

```
PS C:\Users\MXDAM> ping -r 6 -l 32 172.18.198.254
正在 Ping 172.18.198.254 具有 32 字节的数据:
路由: 172.18.198.254
来自 172.18.198.254 的回复: 字节=32 时间=1ms TTL=255
路由: 172.18.198.254
来自 172.18.198.254 的回复: 字节=32 时间=2ms TTL=255
路由: 172.18.198.254
来自 172.18.198.254 的回复: 字节=32 时间=4ms TTL=255
路由: 172.18.198.254

172.18.198.254 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 1ms, 最长 = 4ms, 平均 = 2ms

PS C:\Users\MXDAM> ping -s 4 -l 32 172.18.198.254
正在 Ping 172.18.198.254 具有 32 字节的数据:
来自 172.18.198.254 的回复: 字节=32 时间=4ms TTL=255
    时间戳: 172.18.198.254 : 48173788 ->
        172.18.198.216 : 48174239
来自 172.18.198.254 的回复: 字节=32 时间=1ms TTL=255
    时间戳: 172.18.198.254 : 48174788 ->
        172.18.198.216 : 48175242
来自 172.18.198.254 的回复: 字节=32 时间<1ms TTL=255
    时间戳: 172.18.198.254 : 48175798 ->
        172.18.198.216 : 48176247
来自 172.18.198.254 的回复: 字节=32 时间=1ms TTL=255
    时间戳: 172.18.198.254 : 48176798 ->
        172.18.198.216 : 48177256

172.18.198.254 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 4ms, 平均 = 1ms
```

(4)

ping -r 6 -l 32 172.18.198.254

No.	Time	Source	Destination	Protocol	Length	Info
10	1.034136	172.18.198.216	172.18.198.254	ICMP	102	Echo (ping) request id=0x0001, seq=84/21504, ttl=128 (reply in 11)
11	1.034992	172.18.198.254	172.18.198.216	ICMP	102	Echo (ping) reply id=0x0001, seq=84/21504, ttl=255 (request in 10)
28	2.037680	172.18.198.216	172.18.198.254	ICMP	102	Echo (ping) request id=0x0001, seq=85/21760, ttl=128 (reply in 29)
29	2.038734	172.18.198.254	172.18.198.216	ICMP	102	Echo (ping) reply id=0x0001, seq=85/21760, ttl=255 (request in 28)
42	3.042220	172.18.198.216	172.18.198.254	ICMP	102	Echo (ping) request id=0x0001, seq=86/22016, ttl=128 (reply in 43)
43	3.043402	172.18.198.254	172.18.198.216	ICMP	102	Echo (ping) reply id=0x0001, seq=86/22016, ttl=255 (request in 42)
52	4.047027	172.18.198.216	172.18.198.254	ICMP	102	Echo (ping) request id=0x0001, seq=87/22272, ttl=128 (reply in 53)
53	4.048088	172.18.198.254	172.18.198.216	ICMP	102	Echo (ping) reply id=0x0001, seq=87/22272, ttl=255 (request in 52)
139	9.267266	172.18.198.254	224.0.0.5	OSPF	78	Hello Packet
774	20.159603	172.18.198.254	224.0.0.5	OSPF	78	Hello Packet
1014	31.043471	172.18.198.254	224.0.0.5	OSPF	78	Hello Packet
1536	41.933974	172.18.198.254	224.0.0.5	OSPF	78	Hello Packet
2019	52.831647	172.18.198.254	224.0.0.5	OSPF	78	Hello Packet
2377	63.713848	172.18.198.254	224.0.0.5	OSPF	78	Hello Packet

ping -s 4 -l 32 172.18.198.254

No.	Time	Source	Destination	Protocol	Length	Info
9	1.083666	172.18.198.216	172.18.198.254	ICMP	114	Echo (ping) request id=0x0001, seq=88/22528, ttl=128 (reply in 10)
10	1.112211	172.18.198.254	172.18.198.216	ICMP	110	Echo (ping) reply id=0x0001, seq=88/22528, ttl=255 (request in 9)
32	2.114402	172.18.198.216	172.18.198.254	ICMP	114	Echo (ping) request id=0x0001, seq=89/22784, ttl=128 (reply in 33)
33	2.115495	172.18.198.254	172.18.198.216	ICMP	110	Echo (ping) reply id=0x0001, seq=89/22784, ttl=255 (request in 32)
39	3.120713	172.18.198.216	172.18.198.254	ICMP	114	Echo (ping) request id=0x0001, seq=90/23040, ttl=128 (reply in 40)
40	3.121570	172.18.198.254	172.18.198.216	ICMP	110	Echo (ping) reply id=0x0001, seq=90/23040, ttl=255 (request in 39)
49	4.128171	172.18.198.216	172.18.198.254	ICMP	114	Echo (ping) request id=0x0001, seq=91/23296, ttl=128 (reply in 50)
50	4.129178	172.18.198.254	172.18.198.216	ICMP	110	Echo (ping) reply id=0x0001, seq=91/23296, ttl=255 (request in 49)
88	7.079707	172.18.198.254	224.0.0.5	OSPF	78	Hello Packet
177	17.962920	172.18.198.254	224.0.0.5	OSPF	78	Hello Packet
269	28.854889	172.18.198.254	224.0.0.5	OSPF	78	Hello Packet
376	39.743683	172.18.198.254	224.0.0.5	OSPF	78	Hello Packet
487	50.638643	172.18.198.254	224.0.0.5	OSPF	78	Hello Packet

(5)

协议类型: ICMP 与 OSPF

ICMP: 在 IP 通信中, 经常有数据包到达不了对方的情况。原因是, 在通信途中的某处的一个路由器由于不能处理所有的数据包, 就将数据包一个一个**丢弃**了。或者, 虽然到达了对方, 但是由于搞错了端口号, 服务器软件可能**不能接受**它。这时, 可以使用 **ICMP 报文**来进行故障定位, 将故障信息传递给源端。

OSPF: 开放式最短路径优先 (Open Shortest Path First, OSPF) 是广泛使用的一种动态路由协议, 它属于链路状态路由协议, 具有路由变化收敛速度快、无路由环路、支持变长子网掩码 (VLSM) 和汇总、层次区域划分等优点。

Echo 字段: 回送选项, 协商终端是否将接收的内容返回给发送者

Stamp 字段: 并没有找到, 但是有一个类似的字段, 只有 ICMP 协议才有。



```

√ IP Option - Time Stamp (36 bytes)
  √ Type: 68
    0... .... = Copy on fragmentation: No
    .10. .... = Class: Debugging and measurement (2)
    ...0 0100 = Number: Time stamp (4)

  Length: 36
  Pointer: 5
  0000 .... = Overflow: 0
  .... 0001 = Flag: Time stamp and address (0x1)
  Address: -
  Time stamp: 0
  Address: -
  Time stamp: 0
  Address: -
  Time stamp: 0
  Address: -
  Time stamp: 0

```

【实验思考】

(2) 如何防范被嗅探?

1. 检查网络接口卡是否为混杂模式 (PROMISC)。

3. 发送一个带有网络中不存在的 MAC 地址的广播包到网络中的所有主机。正常情况下, 网络中的主机接口卡在收到带有不存在的 MAC 地址的数据包时, 会将它丢弃, 而当某台主机中的网络接口卡处于混杂模式时, 它就会回应一个带有 RST 标志的包。这样, 就可以认为网络中已经有嗅探器在运行。

5. 使用 Honeypot（蜜罐）技术来设计一个陷阱，以此来诱骗嗅探者对它进行嗅探，并通过它来找到嗅探的源头。

7. 在 Linux 发行版本中运行 ARPWatch 来监控网络中是否有新的 MAC 地址加入。

一、在以太网中防御网络嗅探的方法

1. 尽量在网络中使用交换机和路由器。

3. 对于 E-mail 的内容进行加密后再传输。应用于 E-Mail 加密的方法主要有数字认证与数字签名。

5. 在网络中布置入侵检测系统 (IDS) 或入侵防御系统 (IPS), 以网络防火墙等安全设备。它们对于许多针对交换机和路由器的攻击方法, 很容易就识别出来。

6. 强化安全策略，加强安全培训和管理工作的。



7. 在内部关键位置布置防火墙和 IDS，防止来自内部的嗅探。
8. 如果要在的网络中布置网络分析器，应当保证网络分析器本身的安全，最好事先制定一个网络分析策略来规范使用。

二、在无线局域网中防御无线网络嗅探的方法

1. 禁止 SSID 广播；
2. 对数据进行加密。你可以在无线访问点（AP）后再连接一个×××网关，通过×××强大的数据加密功能来保护无线数据传输；
3. 使用 MAC 地址过滤，强制访问控制；
4. 使用定向天线；
5. 采取屏蔽无线信号方法，将超出使用范围的无线信号屏蔽得；
6. 使用无线嗅探软件实时监控无线局域网中无线访问点（AP）和无线客户连入情况。

本次实验完成后，请根据组员在实验中的贡献，请实事求是，自评在实验中应得的分数。（按百分制）

【交实验报告】

上传实验报告：<ftp://172.18.178.1/> 用户名/口令：netjob/d502 截止日期（不迟于）：1 周之内
上传包括两个文件：

（1）小组实验报告。上传文件名格式：小组号_Ftp 协议分析实验.pdf （由组长负责上传）

例如：文件名“10_Ftp 协议分析实验.pdf”表示第 10 组的 Ftp 协议分析实验报告

（2）小组成员实验体会。每个同学单独交一份只填写了实验体会的实验报告。只需填写自己的学号和姓名。

文件名格式：小组号_学号_姓名_Ftp 协议分析实验.pdf （由组员自行上传）

例如：文件名“10_05373092_张三_Ftp 协议分析实验.pdf”表示第 10 组的 Ftp 协议分析实验报告。

注意：不要打包上传！

学号	学生	自评分
<u>18338072</u>	冼子婷	<u>98</u>
<u>18322043</u>	廖雨轩	<u>98</u>
<u>18346019</u>	胡文浩	<u>98</u>