



警示

- 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	数据科学与计算机学院	班 级	软件工程教二班	组长	郑卓民
学号	18342138	18342077			
学生	郑卓民	南樟			

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	FTP 客户端的 mac 地址是 00:14:2a:20:12:96
截图	<pre>> Frame 4: 103 bytes on wire (824 bits), 103 bytes captured (824 bits) ✓ Ethernet II, Src: DigitalC_02:b7:57 (00:03:0f:02:b7:57), Dst: Elitegro_20:12:96 (00:14:2a:20:12:96) ▾ Destination: Elitegro_20:12:96 (00:14:2a:20:12:96) Address: Elitegro_20:12:96 (00:14:2a:20:12:96) 0. = LG bit: Globally unique address (factory default) 0. = IG bit: Individual address (unicast) > Source: DigitalC_02:b7:57 (00:03:0f:02:b7:57) Type: IPv4 (0x0800) > Internet Protocol Version 4, Src: 172.16.28.58, Dst: 172.16.39.73 > Transmission Control Protocol, Src Port: 21, Dst Port: 1372, Seq: 1, Ack: 1, Len: 49 0000 00 14 2a 20 12 96 00 03 0f 02 b7 57 08 00 45 00 ..*..W..E.. 0010 00 59 3b 8b 40 00 7d 06 26 70 ac 10 1c 3a ac 10 ..Y;@.}.&p.... 0020 27 49 00 15 05 5c 7a 78 43 ac 65 ea 9b 57 50 18 'I...zx C-e..WP.. 0030 ff ff 51 4f 00 00 32 32 30 20 53 65 72 76 2d 55 ..QO...22 0 Serv-U 0040 20 46 54 50 20 53 65 72 76 65 72 20 76 36 2e 34 FTP Ser ver v6.4 0050 20 66 6f 72 20 57 69 6e 53 6f 63 6b 20 72 65 61 for Win Sock rea 0060 64 79 2e 2e 2e 00 0a dy....</pre>
分析	通过抓包我们可以得到其 mac 地址
2	第 1、2、3 号报文的作用是什么？
答案	建立 TCP 连接
截图	<pre>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 SAC... 3 0.000385 172.16.39.73 172.16.28.58 TCP 54 1372 → 21 [ACK] Seq=1 Ack=1 Win=65535 Len=0</pre>
分析	三次挥手
3	该数据包中共有多少个 TCP 流？
答案	5 个



截图

No.	Time	Source	Destination	Protocol	Length	Info
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
132	149.974406	172.16.28.58	172.16.39.73	TCP	60	20 → 1384 [ACK] Seq=1 Ack=1 Win=65535 Len=0
134	149.976040	172.16.28.58	172.16.39.73	FTP-D.	1514	FTP Data: 1460 bytes (PORT) (RETR 888.xls)
135	149.976156	172.16.28.58	172.16.39.73	FTP-D.	1514	FTP Data: 1460 bytes (PORT) (RETR 888.xls)
136	149.976191	172.16.39.73	172.16.28.58	TCP	54	1384 → 20 [ACK] Seq=1 Ack=2921 Win=65535 Len=0
137	149.977267	172.16.28.58	172.16.39.73	FTP-D.	1514	FTP Data: 1460 bytes (PORT) (RETR 888.xls)
138	149.977317	172.16.39.73	172.16.28.58	TCP	54	1384 → 20 [ACK] Seq=1 Ack=4381 Win=65535 Len=0
139	149.977515	172.16.39.73	172.16.28.58	FTP-D.	1514	FTP Data: 1460 bytes (PORT) (RETR 888.xls)

Destination: 172.16.39.73

Transmission Control Protocol, Src Port: 20, Dst Port: 1384, Seq: 0, Len: 0

Source Port: 20
Destination Port: 1384
[Stream index: 4]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
Sequence number (raw): 653758928
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 0

分析

TCP 流的数量是根据有多少次客户端与服务器的连接而判断的,可以根据过滤器 tcp.stream 然后用 eq 来判断最后一次是多少,最后确定有几次。然后根据过滤器的结果,我们得知 4 是最后一次连接,而 tcp 流的序号是从 0 开始的,所以总共有 5 次。

4

用什么用户和密码登录成功?

答案

wlx2008 wlx2008

截图

No.	Time	Source	Destination	Protocol	Length	Info
6	17.542571	172.16.39.73	172.16.28.58	FTP	68	Request: USER wlx2008
9	21.617636	172.16.39.73	172.16.28.58	FTP	68	Request: PASS wlx2008

分析

通过对于 ftp 命令的抓取,可以发现其用户名和密码均为 wlx2008

5

该 FTP 的命令连接和数据连接分别是什么样的连接?

答案

第一次为命令连接,后四次为数据连接

截图

命令连接:

No.	Time	Source	Destination	Protocol	Length	Info
41	104.701805	172.16.28.58	172.16.39.73	FTP	112	Response: 150 Opening ASCII mode data connection for xs2009-9...
104	104.814541	172.16.39.73	172.16.28.58	TCP	54	1372 → 21 [ACK] Seq=136 Ack=534 Win=65002 Len=0
105	104.814922	172.16.28.58	172.16.39.73	FTP	183	Response: 226-Maximum disk quota limited to 307200 kBytes
106	105.017679	172.16.39.73	172.16.28.58	TCP	54	1372 → 21 [ACK] Seq=136 Ack=663 Win=64873 Len=0
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/l...
119	111.822609	172.16.39.73	172.16.28.58	TCP	54	1372 → 21 [ACK] Seq=170 Ack=746 Win=64790 Len=0
120	111.822991	172.16.28.58	172.16.39.73	FTP	183	Response: 226-Maximum disk quota limited to 307200 kBytes
121	112.025742	172.16.39.73	172.16.28.58	TCP	54	1372 → 21 [ACK] Seq=170 Ack=875 Win=64661 Len=0
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...
124	131.654130	172.16.39.73	172.16.28.58	FTP	68	Request: RNT0 888.xls
125	131.657140	172.16.28.58	172.16.39.73	FTP	84	Response: 250 RNT0 command successful.
126	131.831171	172.16.39.73	172.16.28.58	TCP	54	1372 → 21 [ACK] Seq=203 Ack=963 Win=64573 Len=0
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 ...
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



数据连接:

No.	Time	Source	Destination	Protocol	Length	Info
110	111.708415	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	111.708455	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
112	111.708976	172.16.28.58	172.16.39.73	TCP	60	20 → 1381 [ACK] Seq=1 Ack=1 Win=65535 Len=0
114	111.709494	172.16.28.58	172.16.39.73	FTP-DL	367	FTP Data: 313 bytes (PORT) (NLST -1)
115	111.709519	172.16.28.58	172.16.39.73	TCP	60	20 → 1381 [FIN, ACK] Seq=314 Ack=1 Win=65535 Len=0
116	111.709548	172.16.39.73	172.16.28.58	TCP	54	1381 → 20 [ACK] Seq=1 Ack=315 Win=65222 Len=0
117	111.717526	172.16.39.73	172.16.28.58	TCP	54	1381 → 20 [FIN, ACK] Seq=1 Ack=315 Win=65222 Len=0
118	111.717839	172.16.28.58	172.16.39.73	TCP	60	20 → 1381 [ACK] Seq=315 Ack=2 Win=65535 Len=0

分析 可以通过是由客户端还是服务器发起的不同连接来判断是数据连接还是控制连接

6 该 FTP 的连接模式是那种？为什么？

答案 主动模式

截图 14 31.308878 172.16.39.73 172.16.28.58 FTP 63 Request: NLST -1

分析 由于传回为-1，所以 ftp 的连接模式为主动模式

7 最后四个报文的作用是什么？

答案 断开连接，与一开始的三次挥手相对应

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

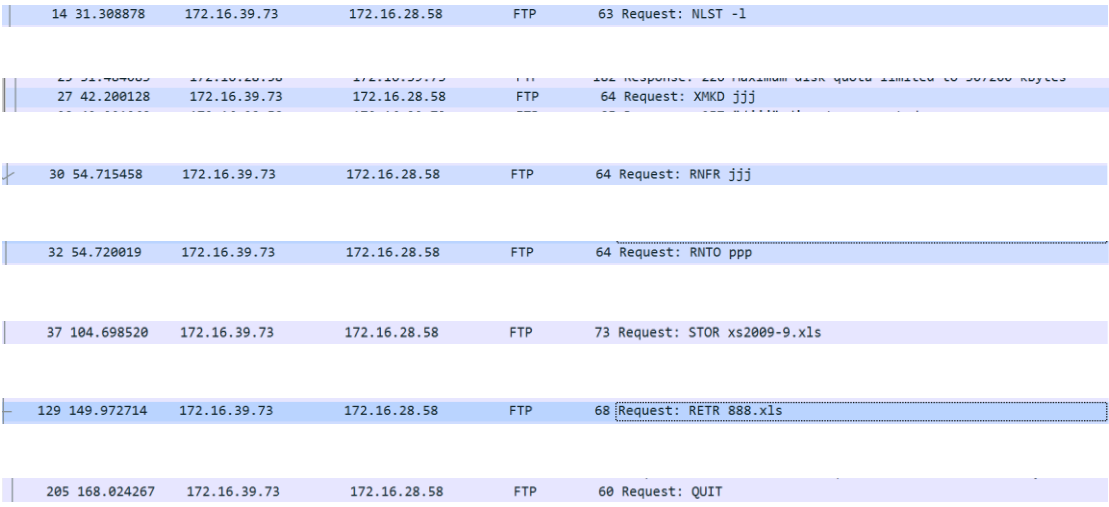
分析 断开连接的四次挥手

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
USER 用户名
PASS 密码
NLST 被动模式
RNT0 重命名后
XMKD 新建目录
SOTR 上传存储
PORT 主动模式
RNFR 重命名前
RETR 下载接收
QUIT 退出

6	17.542571	172.16.39.73	172.16.28.58	FTP	68	Request: USER wlx2008
9	21.617636	172.16.39.73	172.16.28.58	FTP	68	Request: PASS wlx2008
12	31.305692	172.16.39.73	172.16.28.58	FTP	78	Request: PORT 172,16,39,73,5,97



	
分析	通过对于全部 ftp 命令的收集，可以得到其有多少种命令

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	FTP 服务器的 ip 是：172.16.3.240 FTP 客户端的 mac 地址是：00:14:2a:20:12:96
截图	
分析	此数据信息来自客户端向服务器发送的第一个 TCP 请求连接包，故 source 代表客户端， destination 代表服务器，对应的 ip 地址和 mac 地址即所求。
2	该数据包中共有多少个 TCP 流？
答案	该数据包中共有 9 个 TCP 流。
截图	

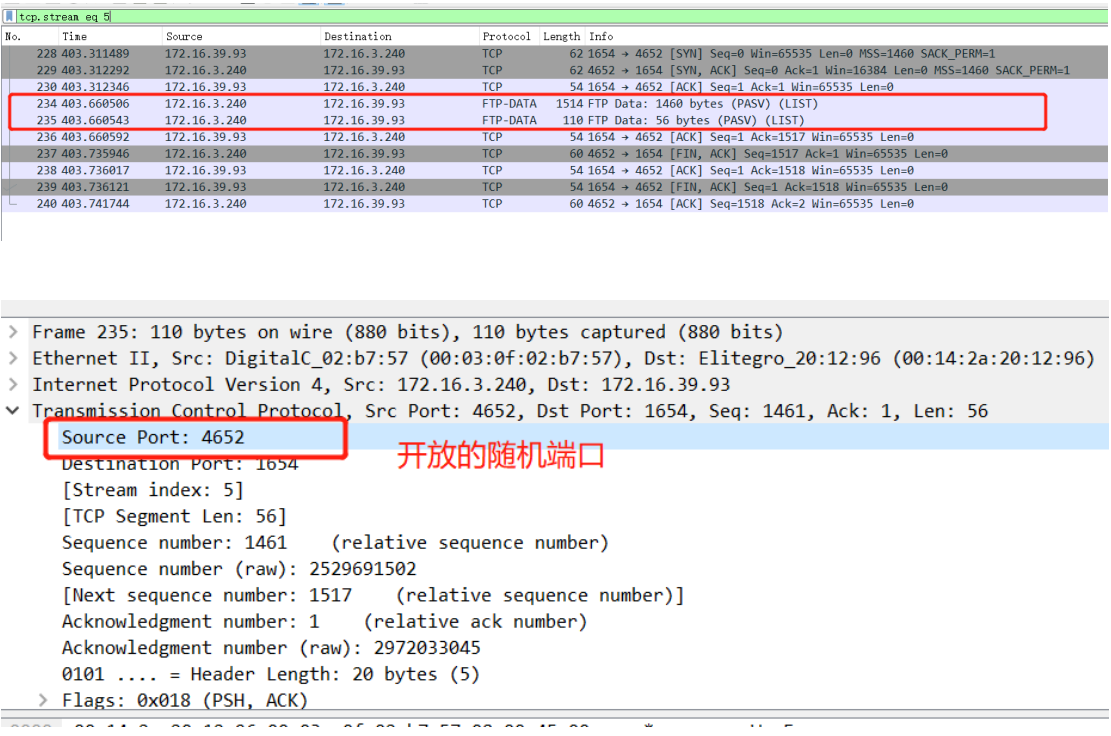


	<p>Destination: 172.16.3.240</p> <p>Transmission Control Protocol, Src Port: 2097, Dst Port: 2118, Seq: 0, Len: 0</p> <p>Source Port: 2097</p> <p>Destination Port: 2118</p> <p>[Stream index: 8] TCP流序号</p> <p>[TCP Segment Len: 0]</p> <p>Sequence number: 0 (relative sequence number)</p> <p>Sequence number (raw): 3913367742</p> <p>[Next sequence number: 1 (relative sequence number)]</p> <p>Acknowledgment number: 0</p> <p>Acknowledgment number (raw): 0</p> <p>0111 = Header Length: 28 bytes (7)</p> <p>Flags: 0x002 (SYN)</p> <p>Window size value: 65535</p> <p>[Calculated window size: 65535]</p>																																																																																																																																																																															
分析	<p>TCP 流的数量为客户端和服务端之间建立了多少次 TCP 连接，一次 TCP 连接包括从握手到分手的过程。因此，计算该数据包中有多少个 TCP 流，可计算该包中 TCP 流对应序号的个数。此处使用过滤器和命令 tcp.stream eq 加一个数字（序号）查看该 TCP 流是否存在，根据截图可见，最高可到达序号 8，由于序号从 0 开始计算，故该数据包中共有 9 个 TCP 流。</p>																																																																																																																																																																															
3	最后用什么用户和密码登录成功？																																																																																																																																																																															
答案	<p>用户：kjdown</p> <p>密码：kjdown</p>																																																																																																																																																																															
截图	<pre> USER kjdown 331 User name okay, need password. PASS kjdown 230 User logged in, proceed. </pre>																																																																																																																																																																															
分析	<p>我们追踪 TCP 流信息，最后成功的那个控制连接为 tcp stream 4，可以看到最后是使用用户名 kjdown 和 kjdown 作为密码。</p>																																																																																																																																																																															
4	该 FTP 的命令连接和数据连接分别是什么？																																																																																																																																																																															
答案	<p>命令（控制）连接：TCP 流 0、1、2、3、4 （数字代表对应 TCP 流序号）。</p> <p>数据连接：TCP 流 5、6、7、8。</p>																																																																																																																																																																															
截图	<p>命令（控制）连接：只选取 TCP 流 0 截图，其他类似：</p> <table border="1"> <thead> <tr> <th>No.</th> <th>Time</th> <th>Source</th> <th>Destination</th> <th>Protocol</th> <th>Length</th> <th>Info</th> </tr> </thead> <tbody> <tr> <td>30</td> <td>0.006731</td> <td>172.16.39.93</td> <td>172.16.3.240</td> <td>TCP</td> <td>62</td> <td>3995 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1</td> </tr> <tr> <td>40</td> <td>0.009137</td> <td>172.16.3.240</td> <td>172.16.39.93</td> <td>TCP</td> <td>62</td> <td>21 → 3995 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1</td> </tr> <tr> <td>50</td> <td>0.009192</td> <td>172.16.39.93</td> <td>172.16.3.240</td> <td>TCP</td> <td>54</td> <td>3995 → 21 [ACK] Seq=1 Ack=1 Win=65535 Len=0</td> </tr> <tr> <td>63</td> <td>6.601481</td> <td>172.16.3.240</td> <td>172.16.39.93</td> <td>FTP</td> <td>79</td> <td>Response: 220-FTP Server ready...</td> </tr> <tr> <td>73</td> <td>8.514666</td> <td>172.16.39.93</td> <td>172.16.3.240</td> <td>TCP</td> <td>54</td> <td>3995 → 21 [ACK] Seq=1 Ack=26 Win=65510 Len=0</td> </tr> <tr> <td>86</td> <td>6.634317</td> <td>172.16.3.240</td> <td>172.16.39.93</td> <td>FTP</td> <td>105</td> <td>Response: 220-USER kjdown</td> </tr> <tr> <td>96</td> <td>7.968083</td> <td>172.16.39.93</td> <td>172.16.3.240</td> <td>TCP</td> <td>54</td> <td>3995 → 21 [ACK] Seq=1 Ack=77 Win=65459 Len=0</td> </tr> <tr> <td>106</td> <td>16.921936</td> <td>172.16.3.240</td> <td>172.16.39.93</td> <td>FTP</td> <td>96</td> <td>Response: 220-PASS kjdown</td> </tr> <tr> <td>117</td> <td>0.959883</td> <td>172.16.39.93</td> <td>172.16.3.240</td> <td>TCP</td> <td>54</td> <td>3995 → 21 [ACK] Seq=1 Ack=119 Win=65417 Len=0</td> </tr> <tr> <td>127</td> <td>0.081263</td> <td>172.16.3.240</td> <td>172.16.39.93</td> <td>FTP</td> <td>92</td> <td>Response: 220-USER kjdown</td> </tr> <tr> <td>137</td> <td>0.258144</td> <td>172.16.39.93</td> <td>172.16.3.240</td> <td>TCP</td> <td>54</td> <td>3995 → 21 [ACK] Seq=1 Ack=157 Win=65379 Len=0</td> </tr> <tr> <td>149</td> <td>0.71181</td> <td>172.16.3.240</td> <td>172.16.39.93</td> <td>FTP</td> <td>106</td> <td>Response: 220-PASS kjdown</td> </tr> <tr> <td>159</td> <td>0.797372</td> <td>172.16.39.93</td> <td>172.16.3.240</td> <td>TCP</td> <td>54</td> <td>3995 → 21 [ACK] Seq=1 Ack=209 Win=65327 Len=0</td> </tr> <tr> <td>166</td> <td>22.761989</td> <td>172.16.3.240</td> <td>172.16.39.93</td> <td>FTP</td> <td>88</td> <td>Response: 220-USER kjdown</td> </tr> <tr> <td>172</td> <td>0.945884</td> <td>172.16.39.93</td> <td>172.16.3.240</td> <td>TCP</td> <td>54</td> <td>3995 → 21 [ACK] Seq=1 Ack=235 Win=65301 Len=0</td> </tr> <tr> <td>183</td> <td>23.13921</td> <td>172.16.3.240</td> <td>172.16.39.93</td> <td>FTP</td> <td>87</td> <td>Response: 220-PASS kjdown</td> </tr> <tr> <td>193</td> <td>23.250596</td> <td>172.16.39.93</td> <td>172.16.3.240</td> <td>TCP</td> <td>54</td> <td>3995 → 21 [ACK] Seq=1 Ack=268 Win=65268 Len=0</td> </tr> <tr> <td>206</td> <td>26.164432</td> <td>172.16.3.240</td> <td>172.16.39.93</td> <td>FTP</td> <td>104</td> <td>Response: 220-USER kjdown</td> </tr> <tr> <td>216</td> <td>26.297562</td> <td>172.16.39.93</td> <td>172.16.3.240</td> <td>TCP</td> <td>54</td> <td>3995 → 21 [ACK] Seq=1 Ack=318 Win=65218 Len=0</td> </tr> <tr> <td>226</td> <td>0.904043</td> <td>172.16.3.240</td> <td>172.16.39.93</td> <td>FTP</td> <td>104</td> <td>Response: 220-PASS kjdown</td> </tr> <tr> <td>237</td> <td>0.008573</td> <td>172.16.39.93</td> <td>172.16.3.240</td> <td>TCP</td> <td>54</td> <td>3995 → 21 [ACK] Seq=1 Ack=368 Win=65168 Len=0</td> </tr> <tr> <td>249</td> <td>22.9977</td> <td>172.16.3.240</td> <td>172.16.39.93</td> <td>FTP</td> <td>104</td> <td>Response: 220-USER kjdown</td> </tr> <tr> <td>259</td> <td>24.454508</td> <td>172.16.39.93</td> <td>172.16.3.240</td> <td>TCP</td> <td>54</td> <td>3995 → 21 [ACK] Seq=1 Ack=418 Win=65118 Len=0</td> </tr> <tr> <td>262</td> <td>23.053320</td> <td>172.16.3.240</td> <td>172.16.39.93</td> <td>FTP</td> <td>106</td> <td>Response: 220-PASS kjdown</td> </tr> </tbody> </table> <p>数据连接：只选取 TCP 流 5 截图，其他类似：</p>	No.	Time	Source	Destination	Protocol	Length	Info	30	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	40	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	50	0.009192	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=1 Win=65535 Len=0	63	6.601481	172.16.3.240	172.16.39.93	FTP	79	Response: 220-FTP Server ready...	73	8.514666	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=26 Win=65510 Len=0	86	6.634317	172.16.3.240	172.16.39.93	FTP	105	Response: 220-USER kjdown	96	7.968083	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=77 Win=65459 Len=0	106	16.921936	172.16.3.240	172.16.39.93	FTP	96	Response: 220-PASS kjdown	117	0.959883	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=119 Win=65417 Len=0	127	0.081263	172.16.3.240	172.16.39.93	FTP	92	Response: 220-USER kjdown	137	0.258144	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=157 Win=65379 Len=0	149	0.71181	172.16.3.240	172.16.39.93	FTP	106	Response: 220-PASS kjdown	159	0.797372	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=209 Win=65327 Len=0	166	22.761989	172.16.3.240	172.16.39.93	FTP	88	Response: 220-USER kjdown	172	0.945884	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=235 Win=65301 Len=0	183	23.13921	172.16.3.240	172.16.39.93	FTP	87	Response: 220-PASS kjdown	193	23.250596	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=268 Win=65268 Len=0	206	26.164432	172.16.3.240	172.16.39.93	FTP	104	Response: 220-USER kjdown	216	26.297562	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=318 Win=65218 Len=0	226	0.904043	172.16.3.240	172.16.39.93	FTP	104	Response: 220-PASS kjdown	237	0.008573	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=368 Win=65168 Len=0	249	22.9977	172.16.3.240	172.16.39.93	FTP	104	Response: 220-USER kjdown	259	24.454508	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=418 Win=65118 Len=0	262	23.053320	172.16.3.240	172.16.39.93	FTP	106	Response: 220-PASS kjdown
No.	Time	Source	Destination	Protocol	Length	Info																																																																																																																																																																										
30	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																																																																																																																																																																										
40	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																																																																																																																																																																										
50	0.009192	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=1 Win=65535 Len=0																																																																																																																																																																										
63	6.601481	172.16.3.240	172.16.39.93	FTP	79	Response: 220-FTP Server ready...																																																																																																																																																																										
73	8.514666	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=26 Win=65510 Len=0																																																																																																																																																																										
86	6.634317	172.16.3.240	172.16.39.93	FTP	105	Response: 220-USER kjdown																																																																																																																																																																										
96	7.968083	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=77 Win=65459 Len=0																																																																																																																																																																										
106	16.921936	172.16.3.240	172.16.39.93	FTP	96	Response: 220-PASS kjdown																																																																																																																																																																										
117	0.959883	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=119 Win=65417 Len=0																																																																																																																																																																										
127	0.081263	172.16.3.240	172.16.39.93	FTP	92	Response: 220-USER kjdown																																																																																																																																																																										
137	0.258144	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=157 Win=65379 Len=0																																																																																																																																																																										
149	0.71181	172.16.3.240	172.16.39.93	FTP	106	Response: 220-PASS kjdown																																																																																																																																																																										
159	0.797372	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=209 Win=65327 Len=0																																																																																																																																																																										
166	22.761989	172.16.3.240	172.16.39.93	FTP	88	Response: 220-USER kjdown																																																																																																																																																																										
172	0.945884	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=235 Win=65301 Len=0																																																																																																																																																																										
183	23.13921	172.16.3.240	172.16.39.93	FTP	87	Response: 220-PASS kjdown																																																																																																																																																																										
193	23.250596	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=268 Win=65268 Len=0																																																																																																																																																																										
206	26.164432	172.16.3.240	172.16.39.93	FTP	104	Response: 220-USER kjdown																																																																																																																																																																										
216	26.297562	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=318 Win=65218 Len=0																																																																																																																																																																										
226	0.904043	172.16.3.240	172.16.39.93	FTP	104	Response: 220-PASS kjdown																																																																																																																																																																										
237	0.008573	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=368 Win=65168 Len=0																																																																																																																																																																										
249	22.9977	172.16.3.240	172.16.39.93	FTP	104	Response: 220-USER kjdown																																																																																																																																																																										
259	24.454508	172.16.39.93	172.16.3.240	TCP	54	3995 → 21 [ACK] Seq=1 Ack=418 Win=65118 Len=0																																																																																																																																																																										
262	23.053320	172.16.3.240	172.16.39.93	FTP	106	Response: 220-PASS kjdown																																																																																																																																																																										



	
分析	FTP 在客户和服务端之间要建立 双重 TCP 连接。一条由客户端发起的“控制连接”，用来传输 FTP 命令，在整个会话期间一直保持打开。一条是 FTP 服务器端发起的“数据连接”，用来传输 FTP 数据。
5	哪几个报文是 FTP 数据连接的三次握手报文？
答案	No: 228-230、256-258、286-288、324-326
截图	
分析	使用 tcp.stream eq 5、tcp.stream eq 6、tcp.stream eq 7、tcp.stream eq 8 过滤器 命令来筛选出数据连接，然后找到三次握手报文
6	哪几个报文是 FTP 数据连接的挥手报文（结束报文）？
答案	No: 237-240、270-273、293-297、620-623
截图	
分析	使用 tcp.stream eq 5、tcp.stream eq 6、tcp.stream eq 7、tcp.stream eq 8 过滤器 命令来筛选出数据连接，然后找到四次挥手报文
7	该 FTP 的连接模式是那种？为什么？



答案	<p>PASV（被动）方式。被动模式主要是服务端打开某个随机端口，被动等待服务端来连接。由截图可见，三次握手后，服务器发送报文告诉客户端开放了 XXXX 随机端口，然后客户端向服务器的 XXXX 端口发出请求。</p>
截图	 <p>Source Port: 4652 Destination Port: 1654 [Stream index: 5] [TCP Segment Len: 56] Sequence number: 1461 (relative sequence number) Sequence number (raw): 2529691502 [Next sequence number: 1517 (relative sequence number)] Acknowledgment number: 1 (relative ack number) Acknowledgment number (raw): 2972033045 0101 = Header Length: 20 bytes (5) Flags: 0x018 (PSH, ACK)</p>
分析	<p>客户端向服务器的 FTP 端口（默认是 21）发送连接请求，服务器接受连接，建立一条命令链路。当需要传送数据时，服务器在命令链路上用 PASV 命令告诉客户端：我打开了 XXXX 端口，你过来连接我。于是客户端向服务器的 XXXX 端口发送连接请求，建立一条数据链路来传送数据。</p>

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？
捕获到了 2784 个数据包。

2778	22.474437	192.168.5.5	61.151.180.239	OICQ	97 OICQ Protocol
2779	22.734956	61.151.180.239	192.168.5.5	OICQ	129 OICQ Protocol
2780	22.760068	192.168.5.5	216.58.200.234	TCP	66 52905 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_...
2781	22.772120	216.58.200.234	192.168.5.5	TCP	66 443 → 52905 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1412 W...
2782	22.772228	192.168.5.5	216.58.200.234	TCP	54 52905 → 443 [ACK] Seq=1 Ack=1 Win=66304 Len=0
2783	22.772556	192.168.5.5	216.58.200.234	TLSv1	571 Client Hello
2784	22.785648	216.58.200.234	192.168.5.5	TCP	54 443 → 52905 [RST] Seq=1 Win=0 Len=0

（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。

通过命令行输入 ipconfig/all 查看本机 ip 地址，为 192.168.5.5。得到了本机 ip 地址就可以知道哪些数据包是自己发



出的，哪些是别人发过来的。

```
连接特定的 DNS 后缀 . . . . . :  
本地链接 IPv6 地址. . . . . : fe80::c84c:4482:52fe:abed%9  
IPv4 地址 . . . . . : 192.168.5.5  
子网掩码 . . . . . : 255.255.255.0  
默认网关. . . . . : 192.168.5.1
```

发出的数据包：Source 为本机 ip 地址即 192.168.5.5 的数据包都是自己发出的。

3	0.164370	61.151.180.239	192.168.5.5	OICQ	129 OICQ Protocol
4	0.652327	61.151.180.239	192.168.5.5	OICQ	129 OICQ Protocol
5	1.113057	192.168.5.5	192.168.5.1	DNS	73 Standard query 0xf718 A www.baidu.com
6	1.116010	192.168.5.1	192.168.5.5	DNS	105 Standard query response 0xf718 A www.baidu.com A 180.101.49.1...
7	1.120643	192.168.5.5	180.101.49.12	TCP	66 52808 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_...
8	1.121008	192.168.5.5	180.101.49.12	TCP	66 52809 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_...

发过来的数据包：Destination 为本机 ip 地址即 192.168.5.5 的数据包都是发过来的。

18	1.168236	180.101.49.12	192.168.5.5	TLSv1...	122 Server Hello
19	1.168237	180.101.49.12	192.168.5.5	TCP	1466 443 → 52808 [ACK] Seq=69 Ack=518 Win=30208 Len=1412 [TCP segm...
20	1.168237	180.101.49.12	192.168.5.5	TCP	1466 443 → 52808 [ACK] Seq=1481 Ack=518 Win=30208 Len=1412 [TCP se...
21	1.168238	180.101.49.12	192.168.5.5	TLSv1...	864 Certificate
22	1.168240	180.101.49.12	192.168.5.5	TCP	54 443 → 52809 [ACK] Seq=1 Ack=518 Win=30208 Len=0
23	1.168240	180.101.49.12	192.168.5.5	TLSv1...	122 Server Hello
24	1.168240	180.101.49.12	192.168.5.5	TCP	1466 443 → 52809 [ACK] Seq=69 Ack=518 Win=30208 Len=1412 [TCP segm...

通过 www.ip138.com 网站，查到下列 ip 地址的地理位置：

180.101.49.12	查询	61.151.180.239	查询
180.101.49.12		61.151.180.239	
rDNS: 即将展示		rDNS: 即将展示	
转换IPv6地址	iP反查网站	转换IPv6地址	iP反查网站
旁站查询		旁站查询	
ASN归属地	江苏省南京市 电信	ASN归属地	上海市上海市 电信
参考数据1	江苏南京 电信	参考数据1	上海上海 电信
参考数据2	江苏省南京市 电信	参考数据2	上海市 电信

180.101.49.13: 中国 江苏省 南京市

61.151.180.239: 中国 上海市

(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 200 a.b.c.d 和 ping -s 4 -l 200 a.b.c.d 命令并捕获数据包。

使用 ipconfig /all 命令，查到所在网络的网关 ip 地址为：192.168.5.1

```
连接特定的 DNS 后缀 . . . . . :  
本地链接 IPv6 地址. . . . . : fe80::c84c:4482:52fe:abed%9  
IPv4 地址 . . . . . : 192.168.5.5  
子网掩码 . . . . . : 255.255.255.0  
默认网关. . . . . : 192.168.5.1
```




运行 ping -r 6 -l 200 192.168.5.1:

```
C:\Users\67068>ping -r 6 -l 200 192.168.5.1

正在 Ping 192.168.5.1 具有 200 字节的数据:
来自 192.168.5.1 的回复: 字节=200 时间=4ms TTL=64
    路由: 192.168.5.1 ->
            192.168.5.1
来自 192.168.5.1 的回复: 字节=200 时间=4ms TTL=64
    路由: 192.168.5.1 ->
            192.168.5.1
来自 192.168.5.1 的回复: 字节=200 时间=2ms TTL=64
    路由: 192.168.5.1 ->
            192.168.5.1
来自 192.168.5.1 的回复: 字节=200 时间=4ms TTL=64
    路由: 192.168.5.1 ->
            192.168.5.1

192.168.5.1 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 = 2ms, 最长 = 4ms, 平均 = 3ms
```

93	1.826036	192.168.5.5	192.168.5.1	ICMP	270 Echo (ping) request	id=0x0001, seq=7189/5404, ttl=64 (reply ...
94	1.827418	192.168.5.1	192.168.5.5	ICMP	270 Echo (ping) reply	id=0x0001, seq=7189/5404, ttl=64 (reques...
118	2.842884	192.168.5.5	192.168.5.1	ICMP	270 Echo (ping) request	id=0x0001, seq=7190/5660, ttl=64 (reply ...
119	2.856394	192.168.5.1	192.168.5.5	ICMP	270 Echo (ping) reply	id=0x0001, seq=7190/5660, ttl=64 (reques...
134	3.852200	192.168.5.5	192.168.5.1	ICMP	270 Echo (ping) request	id=0x0001, seq=7191/5916, ttl=64 (reply ...
135	3.854969	192.168.5.1	192.168.5.5	ICMP	270 Echo (ping) reply	id=0x0001, seq=7191/5916, ttl=64 (reques...
149	4.868584	192.168.5.5	192.168.5.1	ICMP	270 Echo (ping) request	id=0x0001, seq=7192/6172, ttl=64 (reply ...

运行 ping -s 4 -l 200 192.168.5.1

```
C:\Users\67068>ping -s 4 -l 200 192.168.5.1

正在 Ping 192.168.5.1 具有 200 字节的数据:
来自 192.168.5.1 的回复: 字节=200 时间=3ms TTL=64
    时间戳: 192.168.5.1 : 2936634 ->
            192.168.5.1 : 2936634 ->
            192.168.5.5 : 21623124
来自 192.168.5.1 的回复: 字节=200 时间=2ms TTL=64
    时间戳: 192.168.5.1 : 2937660 ->
            192.168.5.1 : 2937660 ->
            192.168.5.5 : 21624149
来自 192.168.5.1 的回复: 字节=200 时间=2ms TTL=64
    时间戳: 192.168.5.1 : 2938683 ->
            192.168.5.1 : 2938684 ->
            192.168.5.5 : 21625173
来自 192.168.5.1 的回复: 字节=200 时间=4ms TTL=64
    时间戳: 192.168.5.1 : 2939704 ->
            192.168.5.1 : 2939704 ->
            192.168.5.5 : 21626196

192.168.5.1 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 = 2ms, 最长 = 4ms, 平均 = 2ms
```

288	13.381101	192.168.5.1	192.168.5.5	ICMP	278 Echo (ping) reply	id=0x0001, seq=7193/6428, ttl=64 (reques...
300	14.404749	192.168.5.5	192.168.5.1	ICMP	282 Echo (ping) request	id=0x0001, seq=7194/6684, ttl=64 (reply ...
301	14.407850	192.168.5.1	192.168.5.5	ICMP	278 Echo (ping) reply	id=0x0001, seq=7194/6684, ttl=64 (reques...
314	15.425915	192.168.5.5	192.168.5.1	ICMP	282 Echo (ping) request	id=0x0001, seq=7195/6940, ttl=64 (reply ...
315	15.427676	192.168.5.1	192.168.5.5	ICMP	278 Echo (ping) reply	id=0x0001, seq=7195/6940, ttl=64 (reques...
326	16.443593	192.168.5.5	192.168.5.1	ICMP	282 Echo (ping) request	id=0x0001, seq=7196/7196, ttl=64 (reply ...
327	16.445204	192.168.5.1	192.168.5.5	ICMP	278 Echo (ping) reply	id=0x0001, seq=7196/7196, ttl=64 (reques...

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



1	0.000000	61.151.180.239	192.168.5.5	OICQ	129 OICQ Protocol
2	0.093382	192.168.5.5	61.151.180.239	OICQ	89 OICQ Protocol
3	0.111878	192.168.5.5	61.151.180.239	OICQ	89 OICQ Protocol
4	0.113465	61.151.180.239	192.168.5.5	OICQ	689 OICQ Protocol
5	0.114650	192.168.5.5	61.151.180.239	UDP	89 4028 → 8000 Len=47
6	0.138278	61.151.180.239	192.168.5.5	OICQ	689 OICQ Protocol
7	0.159963	61.151.180.239	192.168.5.5	UDP	97 8000 → 4028 Len=55

(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。

捕获的数据中协议有：ARP、DNS、HTTP、SSDP、TCP、UDP、TLSv1.2、ICMP。

Echo 的请求和响应分组：

314	15.425915	192.168.5.5	192.168.5.1	ICMP	282 Echo (ping) request id=0x0001, seq=7195/6940, ttl=64 (reply ...)
315	15.427676	192.168.5.1	192.168.5.5	ICMP	278 Echo (ping) reply id=0x0001, seq=7195/6940, ttl=64 (request ...)

Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0

Checksum: 0xa1a1 [correct]

[Checksum Status: Good]

Identifier (BE): 1 (0x0001)

Identifier (LE): 256 (0x0100)

Sequence number (BE): 7195 (0x1c1b)

Sequence number (LE): 6940 (0x1b1c)

[\[Response frame: 315\]](#)

Type 为 8，Code 为 0，意思是 Echo request——回显请求（Ping 请求）

Internet Control Message Protocol

Type: 0 (Echo (ping) reply)

Code: 0

Checksum: 0xa9a1 [correct]

[Checksum Status: Good]

Identifier (BE): 1 (0x0001)

Identifier (LE): 256 (0x0100)

Sequence number (BE): 7195 (0x1c1b)

Sequence number (LE): 6940 (0x1b1c)

[\[Request frame: 314\]](#)

[Response time: 1.761 ms]

Type 为 0，Code 为 0，意思是 Echo Reply——回显应答（Ping 应答）



Stamp 的请求和响应分组:

```
Options: (40 bytes), Time Stamp
  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
  IP Option - End of Options List (EOL)
    Type: 0
      0... .... = Copy on fragmentation: No
      .00. .... = Class: Control (0)
      ...0 0000 = Number: End of Option List (EOL) (0)
```

Time stamp 的请求: 以发送方的主机发送请求的时间为基准, 所以请求的 time stamp 为 0.

Options: (36 bytes), Time Stamp

```
IP Option - Time Stamp (36 bytes)
  Type: 68
    Length: 36
    Pointer: 21
    0000 .... = Overflow: 0
    .... 0001 = Flag: Time stamp and address (0x1)
    Address: 192.168.5.1
    Time stamp: 2983431
    Address: 192.168.5.1
    Time stamp: 2983431
    Address: -
    Time stamp: 0
    Address: -
    Time stamp: 0
```

Time stamp 的应答: 接收方的主机接收到请求后, 计算时间戳并填充接收时间戳并返还给发送方的主机。



实验思考：

(1) 捕获网络上的数据可谓轻而易举，网络嗅探可以说无处不在，如何发现网络中的嗅探行为？

答：

假如网络通讯丢包率非常高或者网络带宽出现反常，那么极有可能你被网络嗅探了。

(2) 如何防范被嗅探？

答：

1. 做好对于私人信息的保护，将数据进行隐藏，如使用安全的拓扑结构。
2. 观察自己的电子产品的信号，如果丢包率突然上升，可能是被攻击。
3. 时常使用检测嗅探器的软件进行检测。

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）

学号	学生	自评分
18342138	郑卓民	100
18342077	南樟	100

【交实验报告】

上传实验报告： aceralon@qq.com

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！