

## 4 用 WinPCAP 监听并解析 FTP 命令

### 4.1 实验目的

通过捕获并解析 FTP 连接的建立与断开过程的数据帧，达到以下目的：

熟悉并学会分析 TCP 连接的建立、断开过程；

熟悉常见的 FTP 命令，熟悉并学会分析 FTP 连接的建立、断开过程；

掌握使用代码实现解析以太网数据帧的技能，提高编码能力。

### 4.2 实验基本要求

基于 WinPCAP 工具包制作程序，捕获与 FTP 连接的建立与断开过程相关的数据帧，并完成分析。

通过监听网络上的 FTP 数据流，解析协议内容，对用户登录行为进行记录。

限制使用 C/C++，拒绝其他语言；允许基于 libpcap 使用 Linux 编程实现。

输出文件类型自选。

IDE 推荐：VS2019。

### 4.3 实验介绍

TCP 连接的建立过程：三次握手；四次挥手；

FTP 连接建立过程：控制连接的建立过程；数据传输连接的建立过程；

### 4.4 事前准备

1. 查阅 TCP 连接建立、断开过程；**4.6 参考的 1**；

2. 查阅 FTP 的通信协议相关的资料，熟悉重点掌握 FTP 登陆环节的通信过程；**4.6 参考的 2, 3, 4**；

3. 了解字符编码，**4.6 参考的 5**；

4. 可选：自行搭建 FTP 服务器，**4.6 参考的 6**。

**安全警告：在测试过程中不要输入敏感的密码，如自己的银行卡密码、QQ 密码等，以防被写入表格。**

### 4.5 实验提交文件

报告和源码一并打包提交，命名格式为：“E4+学号+姓名”。

#### 报告

对于结果图片，可以根据验收项，将对应结果截图。其余要求照旧。

#### 源码

只需提交核心代码部分，使用的库、缓存文件等不用提交。比如你只在 main.cpp 文件里面实现所有功能，那只需提交 main.cpp 文件即可。

### 4.6 参考资料

[1. TCP 连接建立与断开过程](#)

[2. FTP 服务器是什么\(FTP 连接步骤\)](#)

[3. 用命令方式登陆 FTP](#)

[4. FTP 是否可以修改为其它端口](#)

[5. \[字符编码\]彻底理解字符编码](#)

6. 预留，没有东西

[7. 几个字节序转换函数](#)

[8. C 语言中的位段](#)

#### 4.7 验收打分项

实验项	验收依据	分数	备注
-2	以命令行的方式登录 FTP 服务器	10	下载文件“/教学课件/林坤辉/计算机网络与因特网/ftp.txt”
-1	使用 Wireshark 等抓包工具监听网络上的数据流，定位出 TCP 连接建立、断开过程	10	根据要求找出软件抓取到的对应数据帧，如 3 次握手手中的第 2 次握手数据帧；
0	使用 Wireshark 等抓包工具监听网络上的数据流，定位出 FTP 连接建立过程	10	根据要求找出软件抓取到的对应数据帧，如包含登录密码的那一帧；
1	配置好实验环境；设置相应的过滤条件；	10	分析 TCP/FTP 连接建立过程更加方便
2	编写程序，捕获到 TCP 连接建立的数据帧，并能够解析出目的端口、源端口	30	①对遇到的异常情况分析；如为什么会出现“TCP Out-of-Order”；这部分看实际过程中遇到的异常； ②在程序的输出能明确指出握手包的前 2 个 ③在程序输出能明确指出 <b>客户端与 FTP 服务器、FTP 服务器与客户端</b> 断开 TCP 连接而发出的挥手包、
3	编写程序，捕获到 FTP 连接建立的数据帧，并能够解析出对应的 FTP 命令	30	①知道常用 FTP 命令的含义：如 USER，同时打印出对应的用户名；如 PASS，同时打印出对应密码； ②对于数据连接， <b>解析出对应的客户端、FTP 服务器使用的端口号（或者能从你的输出结果中找到判断依据）</b>
4（附加）	将捕获到的数据存储到文件中	0(第 3 次做过) 10（第 3 次没做过）	文件格式自选 数据格式自选

注：-2、0、1 项不给出示例；验收时需要打开 Wireshark 与编写的程序，一同抓包；

-1 项示例

本地主机主动发起连接，本地主机主动断开。

Source	Destination	Protocol	Length	Info
本地主机	ftp服务器	TCP	66	60888 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
ftp服务器	本地主机	TCP	66	21 → 60888 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM=1
本地主机	ftp服务器	TCP	54	60888 → 21 [ACK] Seq=1 Ack=1 Win=132352 Len=0

图 1 Wireshark 抓到的 3 次握手包

本地主机	ftp服务器	TCP	54	60888 → 21 [FIN, ACK] Seq=99 Ack=696 Win=131584 Len=0
ftp服务器	本地主机	TCP	54	21 → 60888 [ACK] Seq=696 Ack=100 Win=66048 Len=0
本地主机	ftp服务器	TCP	54	21 → 60888 [FIN, ACK] Seq=696 Ack=100 Win=66048 Len=0
ftp服务器	本地主机	TCP	54	60888 → 21 [ACK] Seq=100 Ack=697 Win=131584 Len=0

图 2 Wireshark 抓到的 4 次挥手包

2 项示例

1009 30.914043		TCP	66	[TCP Out-Of-Order] 52983 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1010 30.947240		TCP	66	21 → 52983 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM=1
1011 30.947332		TCP	54	52983 → 21 [ACK] Seq=1 Ack=1 Win=132352 Len=0
1012 30.947338		TCP	54	[TCP Dup ACK 1011#1] 52983 → 21 [ACK] Seq=1 Ack=1 Win=132352 Len=0

图 3 TCP 连接建立过程中的异常情况

```
No. 1
Time:20:27:54(883103) Length:66
DESC MAC
SRC MAC
Get an IP packet
DEST IP
SRC IP
->Get a TCP packet
-->sport:61369
-->dport:21
6比特标志位:
URG=0 ACK=0 PSH=0 RST=0 SYN=1 FIN=0
握手包

No. 2
Time:20:27:54(916517) Length:66
DESC MAC
SRC MAC
Get an IP packet
DEST IP
SRC IP
->Get a TCP packet
-->sport:21
-->dport:61369
6比特标志位:
URG=0 ACK=1 PSH=0 RST=0 SYN=1 FIN=0
握手包
```

握手包的前两个

图 4 握手包

```
No. 73
Time:20:34:02(720554) Length:54
DESC MAC
SRC MAC
Get an IP packet
DEST IP 1
SRC IP 192.168.1.1
->Get a TCP packet
-->sport:61606
-->dport:21
6比特标志位:
URG=0 ACK=1 PSH=0 RST=0 SYN=0 FIN=1
挥手包

No. 74
Time:20:34:02(753051) Length:54
DESC MAC
SRC MAC
Get an IP packet
DEST IP
SRC IP
->Get a TCP packet
-->sport:21
-->dport:61606
6比特标志位:
URG=0 ACK=1 PSH=0 RST=0 SYN=0 FIN=0
ftp server-->client
ftp data :CW: |'€?
```

挥手包与应答  
客户端与ftp服务器  
断开TCP连接

图 5 挥手包 1

```

No. 75
Time:20:34:02(754048) Length:54
DESC MAC
SRC MAC
Get an IP packet
DEST IP
SRC IP
->Get a TCP packet
-->sport:21
-->dport:61606
6比特标志位:
URG=0 ACK=1 PSH=0 RST=0 SYN=0 FIN=1
挥手包

No. 76
Time:20:34:02(754165) Length:54
DESC MAC
SRC MAC
Get an IP packet
DEST IP
SRC IP
->Get a TCP packet
-->sport:61606
-->dport:21
6比特标志位:
URG=0 ACK=1 PSH=0 RST=0 SYN=0 FIN=0
client-->ftp server
cmd: yW@
ftp data :yW@

```

挥手包与应答  
服务器与客户端断开TCP连接

图 6 挥手包 2

### 3 项示例

```

No. 5
Time:17:32:42(253469) Length:66
DESC MAC
SRC MAC
Get an IP packet
DEST IP
SRC IP 192.168.1.104
->Get a TCP packet
-->sport:54137
-->dport:21
thl = 0 flag = 0x6
client-->ftp server
cmd: USER
user name:

```

验收时应打印出用户名

图 7 FTP 登录用户名

```

No. 6
Time:20:27:54(986554) Length:90
DESC MA
SRC MAC
Get an IP packet
DEST IP
SRC IP
->Get a TCP packet
-->sport:21
-->dport:61369
6比特标志位:
URG=0 ACK=1 PSH=1 RST=0 SYN=0 FIN=0
ftp server-->client
ftp data :331 User name okay, need password.

```

指出数据传输方向  
数据内容

图 8 FTP 服务器响应

```

No. 7
Time:17:32:42(292454) Length:71
DESC MAC 
SRC MAC 
Get an IP packet
DEST IP 
SRC IP 
->Get a TCP packet
-->sport:54137
-->dport:21
thl = 0  flag = 0x6
client-->ftp server
cmd: PASS
user password: 

```

验收时应打印出密码

图 9 FTP 登录密码

```

No. 8
Time:20:27:55(026936) Length:84
DESC MAC b0:c0:90:3b:97:73
SRC MAC 9c:a6:15:47:22:dc
Get an IP packet
DEST IP 192.168.1.104
SRC IP 121.192.180.66
->Get a TCP packet
-->sport:21
-->dport:61369
6比特标志位:
URG=0  ACK=1  PSH=1  RST=0  SYN=0  FIN=0
ftp server-->client
ftp data :230 User logged in, proceed.

```

登录成功  
指出数据传输方向  
数据内容

图 10 成功登录

```

No. 84
Time:20:56:09(151759) Length:74
DESC MAC 
SRC MAC 
Get an IP packet
DEST IP 
SRC IP 
->Get a TCP packet
-->sport:21
-->dport:62428
6比特标志位:
URG=0  ACK=1  PSH=1  RST=0  SYN=0  FIN=0
ftp server-->client
ftp data :530 Not logged in.

```

图 11 登录失败