厦門大學

信息学院软件工程系

《计算机网络》实验报告

题　　目 实验四　观察 TCP 报文段并侦听分析 FTP 协议

班　　级 软件工程 2018 级 B 班

姓　　名 彭书浩

学　　号 24320182203251

实验时间 2020 年 3 月 25 日

2020 年 3 月 29 日

1 实验目的

用 Wireshark 侦听并观察 TCP 数据段。观察其建立和撤除连接的过程，观察段 ID、 窗口机制和拥塞控制机制等。将该过程截图在报告中。

用 Wireshark 侦听并观察 FTP 数据，分析其用户名密码所在报文的上下文特征， 再总结出提取用户名密码的有效方法。基于 WinPCAP 工具包制作程序，实现监听网 络上的 FTP 数据流，解析协议内容，并作记录与统计。对用户登录行为进行记录。

最终在文件上输出形如下列 CSV 格式的日志：

时间、源 MAC、源 IP、目标 MAC、目标 IP、登录名、口令、成功与否

2015-03-14 13:05:16,60-36-DD-7D-D5-21,192.168.33.1,60-36-DD-7DD5-72,192.168.33.2,student,software,SUCCEED

2015-03-14 13:05:16,60-36-DD-7D-D5-21,192.168.33.1,60-36-DD-7DD5-72,192.168.33.2,student,software1,FAILED

2 实验环境

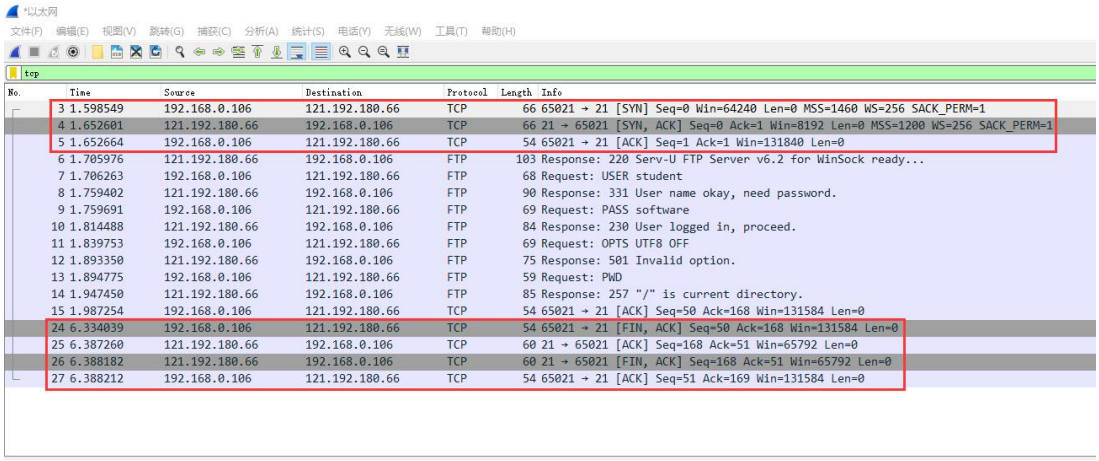
Windows 10 操作系统

-WinPCAP；WireShark；科来数据包播放器

3 实验结果

1、用 Wireshark 侦听并观察 TCP/FTP 数据段：

收到开头 3 次握手数据包、结尾 4 次握手数据包：



开始 3 次标志位结果：

```
[Next sequence number: 0    (relative sequence number)]
Acknowledgment number: 0
1000 .... = Header Length: 32 bytes (8)
Flags: 0x002 (SYN)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): Not set
   .... .0.. .... = ECN-Echo: Not set
   .... ..0. .... = Urgent: Not set
   .... ...0 .... = Acknowledgment: Not set
   .... .... 0... = Push: Not set
   .... .... .0.. = Reset: Not set
   .... .... ..1. = Syn: Set
   .... .... ...0 = Fin: Not set
```

```
Flags: 0x012 (SYN, ACK)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): Not set
   .... .0.. .... = ECN-Echo: Not set
   .... ..0. .... = Urgent: Not set
   .... ...1 .... = Acknowledgment: Set
   .... .... 0... = Push: Not set
   .... .... .0.. = Reset: Not set
   .... .... ..1. = Syn: Set
   .... .... ...0 = Fin: Not set
```

```
Flags: 0x010 (ACK)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): Not set
   .... .0.. .... = ECN-Echo: Not set
   .... ..0. .... = Urgent: Not set
   .... ...1 .... = Acknowledgment: Set
   .... .... 0... = Push: Not set
   .... .... .0.. = Reset: Not set
   .... .... ..0. = Syn: Not set
   .... .... ...0 = Fin: Not set
```

结束两对标志位结果：

```
Flags: 0x011 (FIN, ACK)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): Not set
   .... .0.. .... = ECN-Echo: Not set
   .... ..0. .... = Urgent: Not set
   .... ...1 .... = Acknowledgment: Set
   .... .... 0... = Push: Not set
   .... .... .0.. = Reset: Not set
   .... .... ..0. = Syn: Not set
>  .... .... ...1 = Fin: Set
```

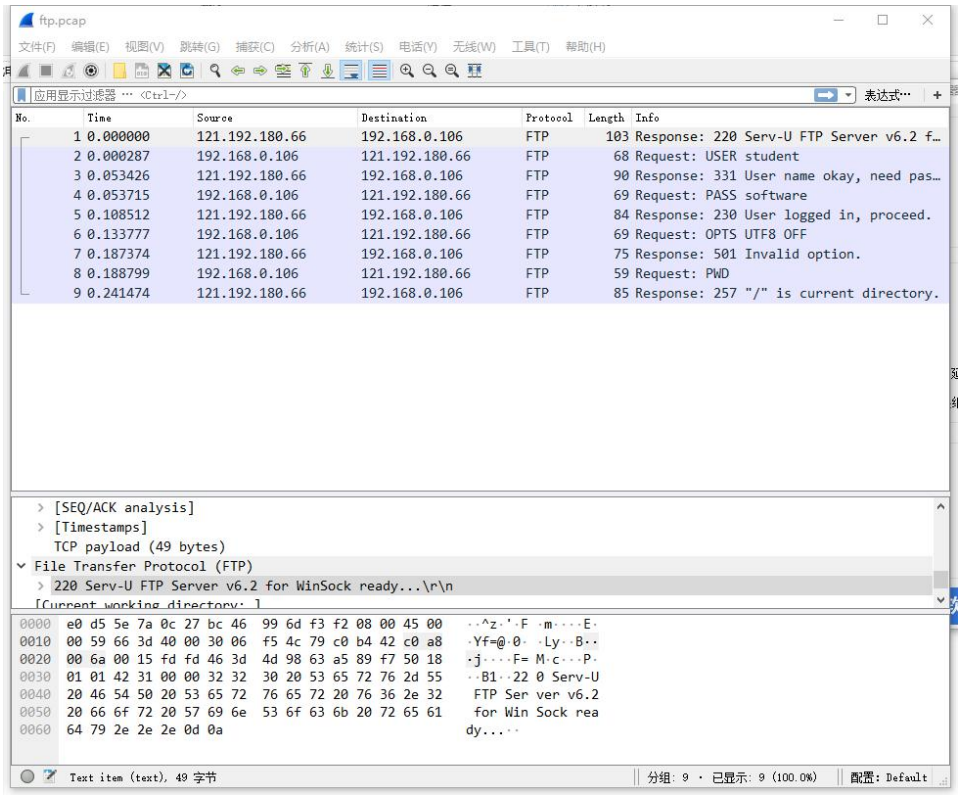
```
Flags: 0x010 (ACK)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): Not set
   .... .0.. .... = ECN-Echo: Not set
   .... ..0. .... = Urgent: Not set
   .... ...1 .... = Acknowledgment: Set
   .... .... 0... = Push: Not set
   .... .... .0.. = Reset: Not set
   .... .... ..0. = Syn: Not set
   .... .... ...0 = Fin: Set
```

2、用 Wireshark 侦听并观察 FTP 数据进行分析

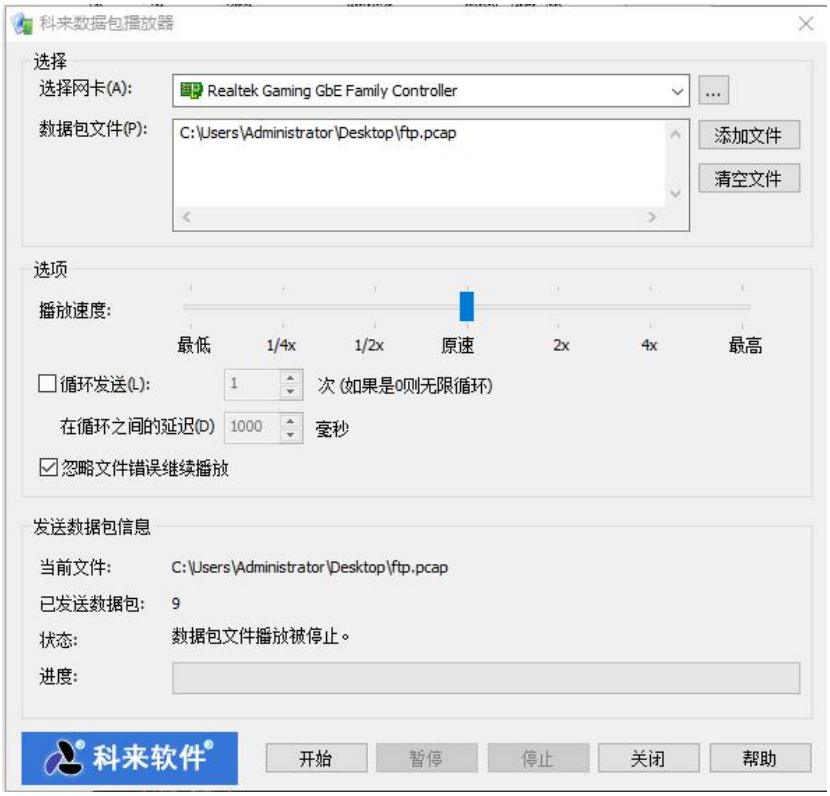
前 54 字的内容是信息头，后面内容为数据段：

```
0000  e0 d5 5e 7a 0c 27 bc 46  99 6d f3 f2 08 00 45 00   ··^z·'·F ·m····E·
0010  00 59 66 3d 40 00 30 06  f5 4c 79 c0 b4 42 c0 a8   ·Yf=@·0· ·Ly··B··
0020  00 6a 00 15 fd fd 46 3d  4d 98 63 a5 89 f7 50 18   ·j····F= M·c···P·
0030  01 01 42 31 00 00 32 32  30 20 53 65 72 76 2d 55   ··B1··22 0 Serv-U
0040  20 46 54 50 20 53 65 72  76 65 72 20 76 36 2e 32    FTP Ser ver v6.2
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 0d 0a                               dy...··
```

3、WireShark 将连接 ftp 产生的 ftp 数据包生成 pcap 文件由程序读取。



4、用科来播放器重发 pcap 文件，由程序网卡接收。



5、用 wincap 的过滤器来过滤数据包，只接收 ftp 数据包。

```
char packet_filter[] = "tcp port ftp";  //只捕获ftp数据包
```

```
//compile the filter
if (pcap_compile(adhandle, &fcode, packet_filter, 1, netmask) <0 )
{
    fprintf(stderr,"\nUnable to compile the packet filter. Check the syntax.\n");
    /* Free the device list */
    pcap_freealldevs(alldevs);
    return -1;
}
```

6、根据 ftp 数据分析结果，改写程序。

```
/*FTP header*/
typedef struct ftp_header {
    u_short src_port;
    u_short des_port;
    u_short seq_num[2];
    u_short ack_num[2];
    u_char bit_len;          // Total bit length
    u_char flags;            //标志位
    u_short win_size;        //window size value
    u_short checksum;        //Checksum
    u_short urg_pointer;     //urgent pointer
} ftp_header;
```

7、程序运行，通过科来重新发送 ftp 数据包，输出结果。

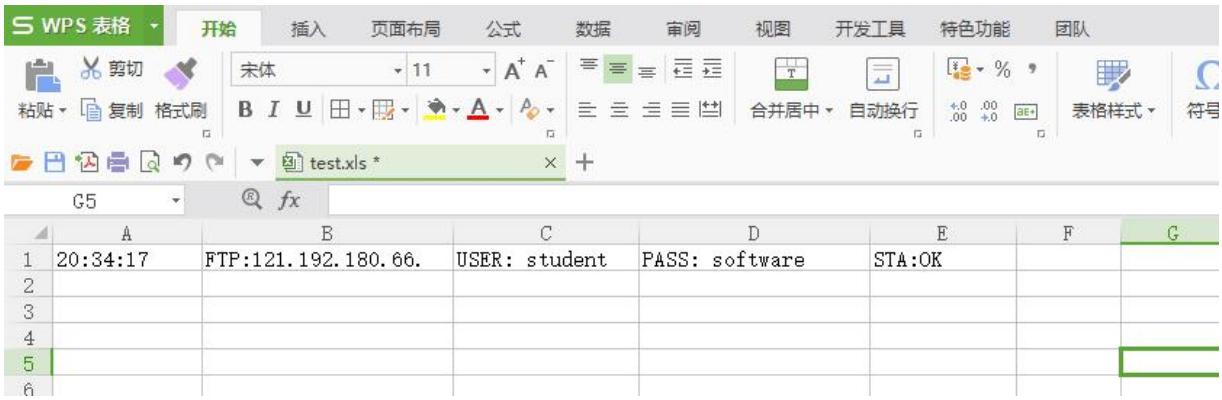
以发送用户名的 ftp 数据包为例：

```
0000  bc 46 99 6d f3 f2 e0 d5  5e 7a 0c 27 08 00 45 00   ·F·m···· ^z·'··E·
0010  00 36 ee 8f 40 00 80 06  00 00 c0 a8 00 6a 79 c0   ·6··@··· ·····jy·
0020  b4 42 fd fd 00 15 63 a5  89 f7 46 3d 4d c9 50 18   ·B····c· ··F=M·P·
0030  02 03 ef 3d 00 00 55 53  45 52 20 73 74 75 64 65   ···=··US ER stude
0040  6e 74 0d 0a                                       nt··
```

```
00 36 EE 8F 40 00 80 06 00 00 C0 A8 00 6A 79 C0
B4 42
mac_header:
        dest_addr: BC 46 99 6D F3 F2
        src_addr: E0 D5 5E 7A 0C 27
        type: 0800
ip_header
        ver_ihl    : 45
        tos        : 00
        tlen       : 0036
        identification: EE8F
        flags_fo   : 4000
        ttl        : 80
        proto      : 06
        crc        : 0000
        saddr:     : C0 A8 00 6A  192.168.0.106.
        daddr      : 79 C0 B4 42  121.192.180.66.
ftp_header
        src_port   : FDFD
        des_port   : 0015
        seq_num    : 63A589F7
        ack_num    : 463D4DC9
        bit_len    : 50
        flags      : 18
        win_size   : 0203
        checksum   : EF3D
        urg_pointer: 0000

data is:USER student
```

最后输出到表格中的数据：



4 实验总结

1、代码中采用了一个 ntohs（）函数，用于 16 进制数据的输出。由于数据的储存方式是字节大端序位小端序，故用此函数以原格式输出数据。

2、 FTP 协议是 TCP/IP 协议的一部分，严格意义上来说是应用层协议，它定义了本地登录户机与远程服务器之间的交互过程。采用过滤器过滤时有其特有的格式。

3、ftp 数据段的结尾是以/r/n 结尾的，输出数据时应把它们排除。

4、拔除网线后，使用科来数据播放器重新发送数据包时，网卡不会发送与接收任何数据。