厦門大學



信息学院软件工程系

《计算机网络》实验报告

题	目	实验三 基于 PCAP 库侦听并分析网络流量
班	级	<u> </u>
姓	名	廖陈承
学	号	22920192204238
实验时	付间	2021年5月5日

2021 年 5 月 5 日

填写说明

- 1、本文件为 Word 模板文件,建议使用 Microsoft Word 2019 打开, 在可填写的区域中如实填写;
- 2、填表时, 勿破坏排版, 勿修改字体字号, 打印成 PDF 文件提交;
- 3、文件总大小尽量控制在 1MB 以下, 勿超过 5MB;
- 4、应将材料清单上传在代码托管平台上;
- 5、在学期最后一节课前按要求打包发送至 cni21@qq.com。

1 实验目的

通过完成实验,理解数据链路层、网络层、传输层和应用层的基本原理。掌握用 Wireshark 观察网络流量并辅助网络侦听相关的编程;掌握用 Libpcap 或 WinPcap 库侦听并处理以太网帧和 IP 报文的方法;熟悉以太网帧、IP 报文、TCP 段和 FTP 命令的格式概念,掌握 TCP 协议的基本机制;熟悉帧头部或 IP 报文头部各字段的含义。熟悉 TCP 段和 FTP 数据协议的概念,熟悉段头部各字段和 FTP 控制命令的指令和数据的含义。

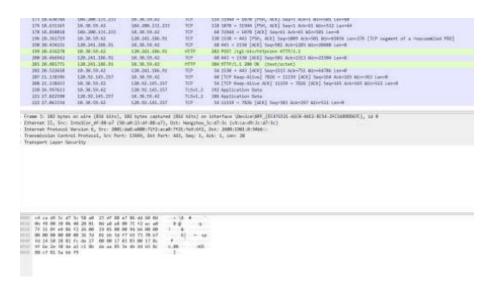
2 实验环境

操作系统: Windows10 编程语言: C/C++ 环境: Visual Studio 2019、Wireshark。

3 实验结果

1、用侦听解析软件观察数据格式

用 Wireshark 或 Omnipeek 等网络侦听软件网络上的数据流,验证理论课 讲 授的网络协议层次嵌套,验证帧格式、IP 报文格式、TCP 段格式和 FTP 协议命 令和响应的格式,验证 MAC 地址、IP 地址、TCP 端口等协议地址格式。



2、用侦听解析软件观察 TCP 机制

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。

TCP 格式:

```
Source Port: 56923
Destination Port: 443
[Stream index: 10]
[TCP Segment Len: 0]
                        (relative sequence number)
Sequence Number: 691
Sequence Number (raw): 738250361
                              (relative sequence number)]
[Next Sequence Number: 691
Acknowledgment Number: 10002
                               (relative ack number)
Acknowledgment number (raw): 3975635134
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
Window: 32747
[Calculated window size: 65494]
[Window size scaling factor: 2]
Checksum: 0xd0df [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[SEQ/ACK analysis]
```

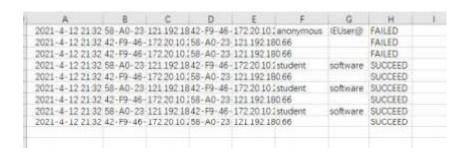
3、用 Libpcap 或 WinPcap 库侦听网络数据

```
/* convert the timestamp to readable format */
local_tv_sec = header->ts.tv_sec;
ltime=localtime(&local_tv_sec);
strftime( timestr, sizeof timestr, "%Y-%m-%d %H:%M:%S", ltime);
mh = (mac header*)pkt data;
if ((file = fopen("Myoutput.csv", "a")) == NULL)
    file = fopen("Myoutput.csv", "w");
if (file == NULL)
    printf("file is null\n");
/* print timestamp and length of the packet */
//printf("%s.%.6d len:%d ", timestr, header->ts.tv_usec, header->len);
fprintf(file, "%s, ", timestr);
for (int i = 0; i < 6; i++)
    fprintf(file, "%02X", mh->dest_addr[i]);
    if (i != 5)
        fprintf(file, "-");
fprintf(file, ", "):
   测试结果导出为 csv:
58-A0-23 8.8.4.4 60-EE-5C 192.168.1.
                                            121
60-EE-5C 192.168.1, 58-A0-23 14.116.137
                                            209
FF-FF-FF-192.168.1. 58-A0-23 192.168.1.
                                            82
58-A0-23 14.116.13760-EE-5C 192.168.1.
                                            409
60-EE-5C 192.168.1. 58-A0-23 14.116.137
                                            97
58-A0-23 14.116.13760-EE-5C 192.168.1.
                                           129
58-A0-23 14.116.13760-EE-5C 192.168.1.
                                            129
58-A0-23 14.116.13760-EE-5C 192.168.1.
                                            129
60-EE-5C 192.168.1. 58-A0-23 14.116.137
                                            209
60-EE-5C 192.168.1. 58-A0-23 58.60.10.4
                                            86
58-A0-23 14.116.13760-EE-5C 192.168.1.
                                            769
58-A0-23 14.116.13760-EE-5C 192.168.1.
                                            129
60-EE-5C 192.168.1, 58-A0-23 14.116.137
                                            97
                                           425
58-A0-23 14.116.13760-EE-5C 192.168.1.
60-EE-5C 192.168.1. 58-A0-23 14.116.137
                                            97
60-EE-5C 192.168.1. 58-A0-23 8.8.8.8
                                            88
58-A0-23 8.8.8.8
                  60-EE-5C 192.168.1.
                                           343
58-A0-23 14.116.13760-EE-5C 192.168.1.
                                            409
60-EE-5C 192.168.1, 58-A0-23 14.116.137
                                            97
                                            89
60-EE-5C 192.168.1. 58-A0-23 14.116.137
58-A0-23 14.116.13760-EE-5C 192.168.1.
                                            873
60-EE-5C 192.168.1. 58-A0-23 14.116.137
                                            89
EO AO 22 14 116 12 60 EE EC 102 160 1
                                            Ω7
```

4、解析侦听到的网络数据

```
if (com == "530 ")
    printf("%s, ", timestr);
    for (int i = 0; i < 6; i++)
        printf("%02X", mh->dest_addr[i]);
        if (i != 5)
             printf("-");
    printf(", "):
    printf("%d. %d. %d. %d, ",
        ih->saddr.bytel,
        ih->saddr. byte2,
        ih->saddr. byte3,
        ih->saddr.byte4);
    for (int i = 0; i < 6; i++)
        printf("%02X", mh->src_addr[i]);
         if (i != 5)
             printf("-");
    printf(", ");
    printf("%d. %d. %d. %d, ",
        ih->daddr. byte1,
        ih->daddr. byte2,
        ih->daddr. byte3,
        ih->daddr.byte4);
    //printf("%d,", header->len);
std::cout << user << "," << pass << ",";
    printf("FAILED\n");
    user.clear();
    pass.clear();
```

导出为 csv:



4 实验代码

本次实验的代码已上传于以下代码仓库: https://github.com/marshcoldboy/Internet-and-Network

5 实验总结

通过本次实验,掌握了 Wireshark 观察网络流量并辅助网络侦听相关的编程。 更加熟悉以太网帧、IP 报文、TCP 段和 FTP 命令的格式概念,掌握 TCP 协议的 基本机制。但实验过程中的困难也很多。