

网络安全实验报告

题 目 捕包软件的使用与实现

专 业 信息安全

学 号 1190202401

学 生 陈豪

指 导 教 师 王彦

**一、实验目的**

理解捕包程序捕包过程，可以自己编程捕包并从数据包中解析出需要的信息。

# 二、实验内容

1. 熟练使用 sniffer 或 wireshark 软件，对协议进行还原（能够找访问网页的四元组）；只需要写报告，不需要在实验课检查。
2. 利用 libpcap 或 winpcap 进行编程，能够对本机的数据包进行捕获分析（比如将本机所有数据包的四元组写到指定文件），按照自己的设想撰写需求分析和详细设计。（实验课检查程序）

# 三、实验过程

**（一）使用 wireshark 软件对协议进行还原**

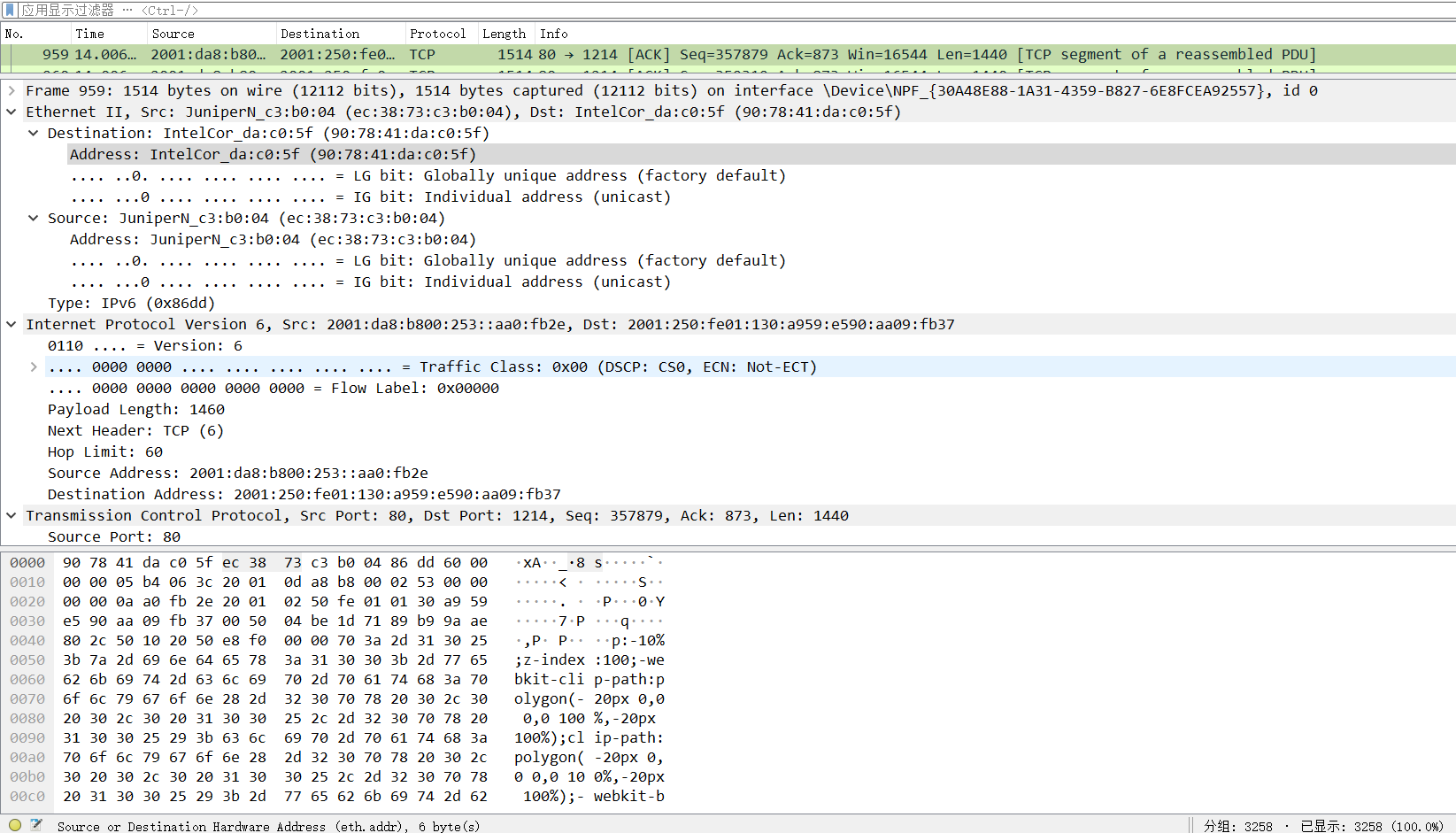
## 实验基本信息：

实验环境：Windows10 x64

WireShark2.6.4

## 1. 捕包并分析四元组

(1) TCP 分析



截图中的数据包为TCP数据包，并且使用IPV6协议作为网络层协议，源IP为2001:da8:b800:253::aa0:fb2e，目的IP为2001:250:fe01:130:a959:e590:aa09:fb37，源端口号是80，目的端口是1214.

分析：

1. 以太网帧

前六个字节90 78 41 da c0 5f是目的Mac地址

接下来六个字节ec 38 73 c3 b0 04是源MAC地址

然后两个字节 86 dd指明使用的是IPV6协议

以太网帧结束

1. IP协议头

前四个字节是 60 00 00 00，第一个字节的前四位是6说明使用的是ipv6协议，接下来是优先级和流标签，它们的值都是0。

接下来两个字节是 05 b4表示载荷大小是1460个字节。

接下来一个字节是06表示下一个首部的协议是TCP协议。

下一个字节3c是跳步限制，相当于IPV4的TTL。

然后是16个字节的源地址IP：

20 01 0d a8 b8 00 02 53 00 00 00 00 0a a0 fb 2e代表的IPV6地址值是2001:da8:b800:253::aa0:fb2e

接着是16个字节的目的地址IP：

20 01 02 50 fe 01 01 30 a9 59 e5 90 aa 09 fb 37代表的IPV6地址是2001:250:fe01:130:a959:e590:aa09:fb37

IPV6分析结束，下面是TCP协议

1. TCP协议头

首先两个字节00 50代表的是源端口的值是80

接着两个字节04 be说明目的端口的值是1214

然后四个字节是1d 71 89 b9说明seq的值是493980089

然后四个字节是9a ae 80 2c说明ack的值是2595127340

往后一个字节的前四位是为头部长度0x5 说明头部长度是20个字节

然后后六位0x00是保留字节

然后后六位是是标志位值是0x10表示ACK

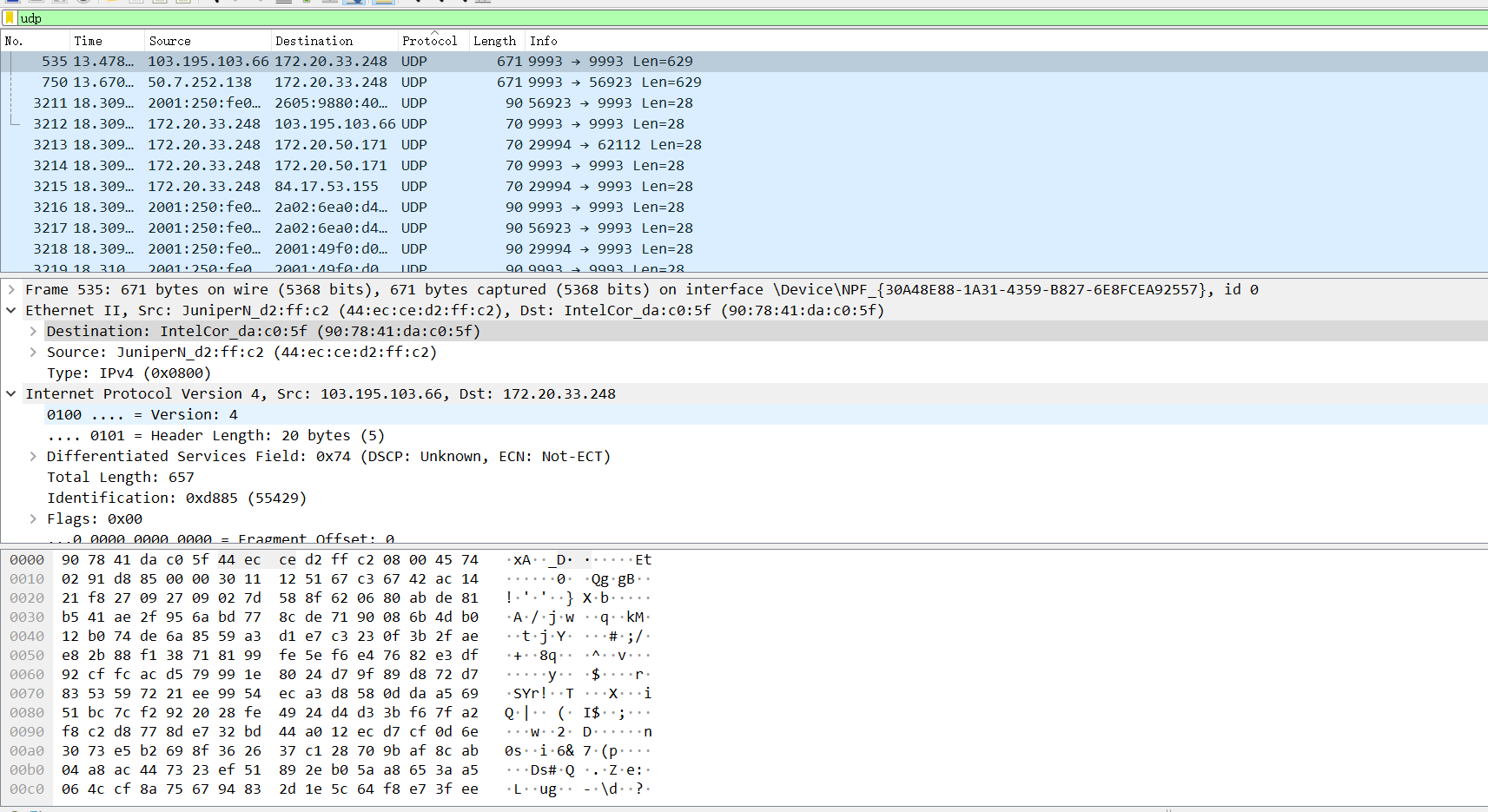
往后两个字节为窗口大小20 50表示窗口大小是8272字节

往后两个字节为校验和 e8 f0 表示校验和是0x8f0

最后两个字节是紧急指针0x00 00

TCP头部结束，接下来是数据

(2) UDP 分析



截图为UDP协议的数据包，使用IPV4作为网络层协议，源IPV4地址是103.195.103.66，目的IP地址是172.20.33.248.源端口号是9993，目的端口号也是9993。

分析：

1. 以太网帧分析

前六个字节90 78 41 da c0 5f是目的Mac地址

接下来六个字节44 ec ce d2 ff c2是源MAC地址

然后两个字节 08 00指明网络层使用的是IPV4协议

以太网帧结束

1. IPV4分析

第一个字节的前四位是0x4，说明使用的是IPV4协议

第一个字节的后四位是0x5，说明头部长度是20个字节

下一个字节是服务类型值0x74

下两个字节 02 97是总长度是657字节

然后是两字节 d8 85的标识

下两个个字节的前四位是标志位是0x0

后12位是片偏移0x000

下一个字节0x30是TTL，值是48

接着一个字节0x11指明使用UDP协议

接着两个字节0x1251是首部校验和

最后的8个字节前四个字节 67 c3 67 42是源IP地址

后四个字节ac 14 21 f8是目的IP地址

IPV4分析结束

1. UDP分析

前两个字节0x27 09是源端口，值是9993

后两个字节0x27 09是目的端口，值是9993

然后两个字节0x02 7d是长度，值是637个字节

最后是两个字节0x588f的校验和。

**（二）利用 libpcap 编写捕包软件实验基本信息：**

实验环境：Ubuntu16.04 x64

编程语言：C 语言

## 1. 需求分析

本程序需要运用 libpcap 来捕获本机数据包，并获取数据包中的四元组，将其展示给

用户。

程序功能：

(1)捕获本机数据包（可以自定义过滤条件）；

(2)逐层解析数据包，获得 IPv4 数据包的源 ip、目的 ip、源端口、目的端口；

(3)将上述四元组写入文件（每次运行程序都新生成一个文件）。

## 2. 环境配置

(1)从官网下载安装包：m4-1.4.18.tar.gz

bison-3.3.tar.gz flex-2.6.4.tar.gz

libpcap-1.9.0.tar.gz

(2)对以上安装包逐个使用以下命令来解压安装包：

tar -zxvf xxx.tar.gz

(3)进入解压后的文件夹后执行以下命令来安装库文件：

./configure

make

make install

(4)配置好环境后，可正常编译，但运行时提示

error while loading shared libraries: libpcap.so.1: cannot open shared object

file: No such file or directory

百度发现是因为没有自动生成链接文件，于是在/usr/lib/文件夹下输入命令：

ln -s libpcap.so.1.9.0 libpcap.so.1

以上命令可生成 libpcap.so.1 文件，但此时仍不能正常运行，还需要将这个文件的地址加入配置文件中，具体做法为执行命令：

locate libpcap.so.1

发现 libpcap.so.1 的路径为/usr/lib/x86\_64-linux-gnu，执行命令： sudo vi /etc/ld.so.conf

打开配置文件，在末尾添加/usr/lib/x86\_64-linux-gnu，保存后退出，再执行命令： sudo ldconfig

这样就编译器就可以正常链接到库了，关于 libpcap 的环境配置也就完成了。

## 3. 数据结构设计

由于是逐层解析以太网数据帧，所以需要准备至少三种数据结构：以太网数据帧头、

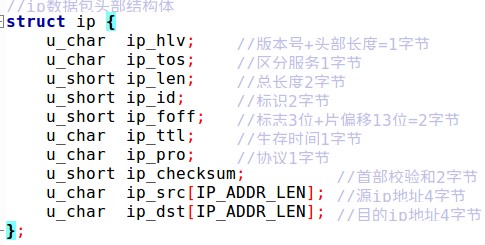
IPv4 数据报头、传输层报文头。具体如下：

数据结构的定义原则：1 字节数据定义为 u\_char，2 字节数据定义为 u\_short，其他 2 的倍数字节的数据（MAC 地址和 IP 地址）定义为 u\_char 数组（TCP 的序列号和 ack 定义为 u\_int，因为它们的表现形式就是一个数字，但地址我们通常是一个一个字节分开解析的）。

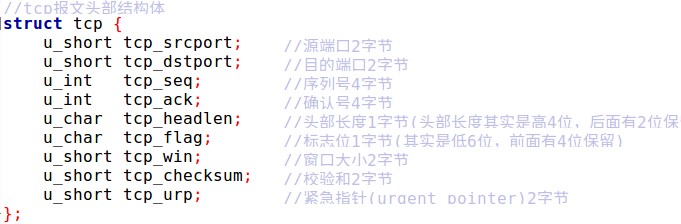
1. 以太网数据帧头



1. IPv4 数据报头



1. TCP 报文头



1. UDP 报文头



## 4. 回调函数设计

## 5. 主函数设计

第一步：使用 pcap\_findalldevs()函数来获取网络设备。此步不用 pcap\_lookupdev() 是因为官方并不推荐使用这个函数，有时第一个位置的网卡是一个虚拟网卡，用它进行下面的步骤会出现错误。

第二步：使用 pcap\_open\_live()函数来获得捕包描述字，由于只需要捕获本机数据包所以设置为非混杂模式。

第三步：如果有过滤条件的话（作为程序运行参数读入），设置过滤条件。

第四步：生成本次捕包的 txt 文件（格式如“capture\_yyyy\_mm\_dd\_hh\_mm\_ss.txt”， capture 后是生成文件的时间），并写入过滤条件和标题栏。

第五步：使用 pcap\_loop()函数和回调函数 ethernet\_callback()来循环捕包。

## 6. 编译运行

使用下列命令编译：

gcc pcap.c -Wall -lpcap -o pcap

其中-Wall 参数指打印所有警告信息，-lpcap 用来链接 pcap 库。

使用下列命令运行：

**sudo** ./pcap

捕包程序在运行时需要 root 权限，否则无法正常打开。

# 四、实验结果

1. **使用 wireshark 软件对协议进行还原**
2. **利用 libpcap 编写捕包软件**

以下为捕包时文件中部分输出：

# 五、心得体会