

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 1190201012

学 生 张中界

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**

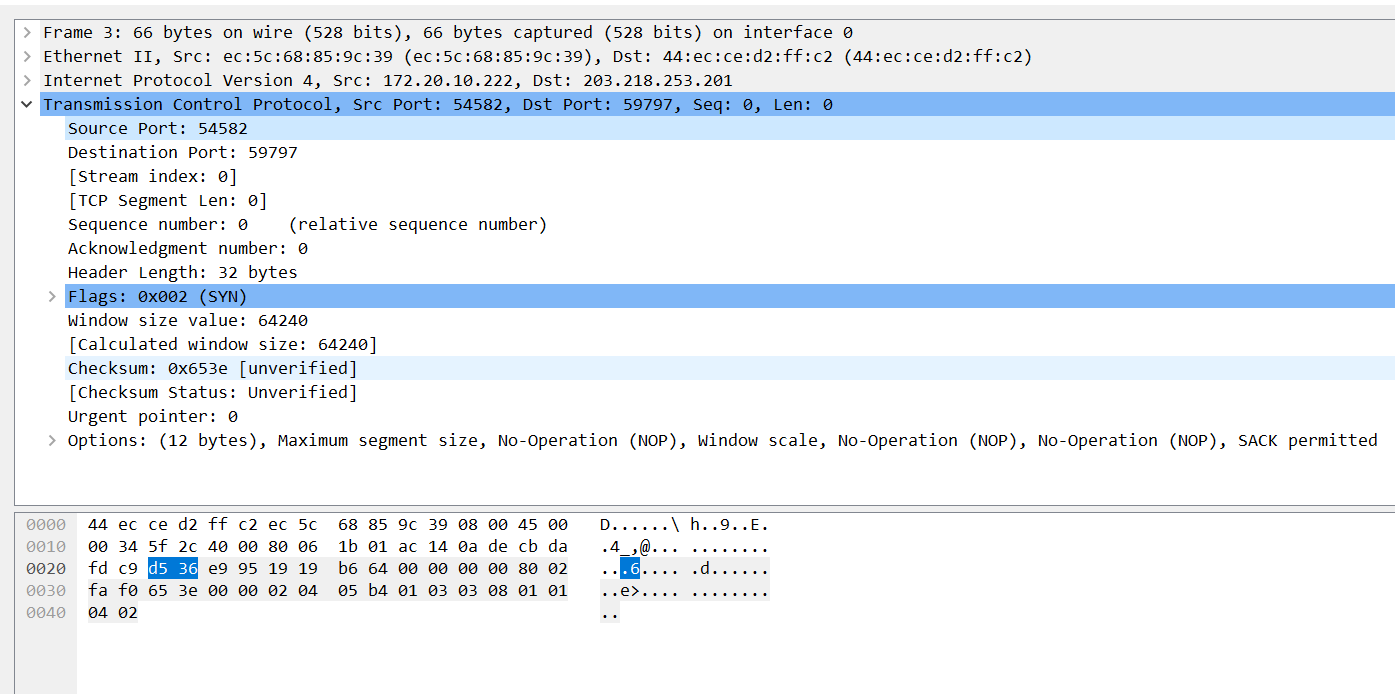
## 实验基本信息：

实验环境：Windows10 x64

WireShark2.6.4

## 1. 捕包并分析四元组

(1) TCP 分析

源 IP 为 172.20.10.222，目的 IP 为 203.218.253.201，源

端口为 54582，目的端口为 59797。

分析：以太网头部：

前 6 个字节 44 ec ce d2 ff c2 为目的主机 MAC，往后 6 个字节 ec 5c 68 85 9c 39 为源主机 MAC，

往后 2 个字节为上层协议，0x0800 表示 IPv4 协议；以太网头部结束，现在是 ip 头部：往后 1 个字节 0x45 表示 IP 版本为 4，头部长度为 5，往后 1 个字节为区分服务，0x00 表示默认，

往后 2 个字节为总长度，0x0034，往后 2 个字节为 id，值为 0x5f2c，

往后 2 个字节为标志位+片偏移，值为 0x4000，

往后 1 个字节为 ttl，值为 0x80 = 128，

往后 1 个字节为上层协议，0x06 表示 TCP 协议，往后 2 个字节为头部校验和，值为 0x1b01，

往后 4 个字节为源 ip 地址,往后4个字节为目的ip地址 ;

ip 头部结束：现在是 TCP 头部：

往后 2 个字节为源端口 0xd536 ，往后 2 个字节为目的端口 0xe995，往后 4 个字节为 seq，往后 4 个字节为 ack，

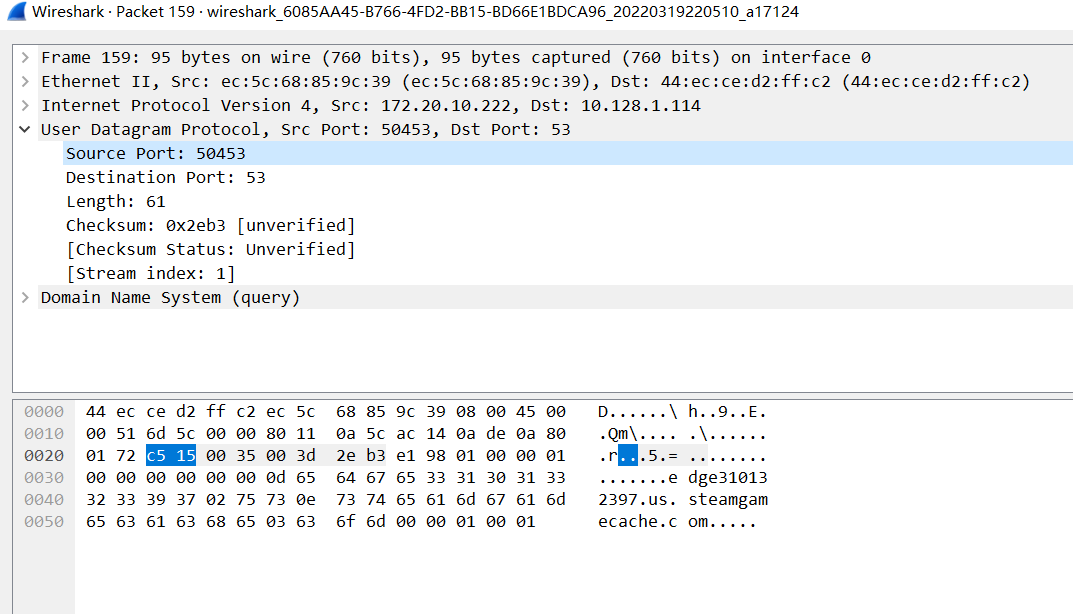
往后 1 个字节为头部长度 0x80，

往后 1 个字节为标志位（其实是 6 位），

往后 2 个字节为窗口大小，往后 2 个字节为校验和，

往后 2 个字节为紧急指针 0x0000； TCP 头部结束，接下来是数据。

(2) UDP 分析



截图中这个 UDP 数据包，源 IP 为 172.20.10.222，目的 IP 为 10.128.1.114，

源端口为 50453，目的端口为 53。

分析：

以太网头部和 ip 头部与 TCP 协议基本相同，直接分析 UDP 头部，从第三行第三个字节开始：

2 个字节源端口 0xc515 ，

2 个字节目的端口 0x0035，

2 个字节总长度 0x003d， 2 个字节校验值：0x2eb3；

后面全都是数据。

**（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：

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

## 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、

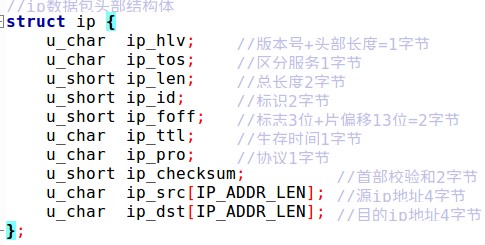
IPv4 数据报头、传输层报文头。具体如下：

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）。

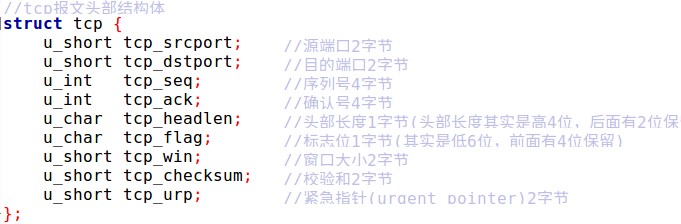
1. 以太网数据帧头



1. IPv4 数据报头



1. TCP 报文头



1. UDP 报文头



## 4. 主函数设计

1. pcap\_findalldevs()获取设备列表，打印所有设备信息，由用户选择要使用的设备；

2. pcap\_open\_live()打开句柄，设置为混杂模式

3. 指定过滤规则，由用户自定义，然后调用pcap\_compile()和pcap\_setfilter()设置过滤器；

4. pcap\_loop()开始捕包 ；

5. 捕包结束后调用pcap\_freealldevs()以及pcap\_close()释放资源。

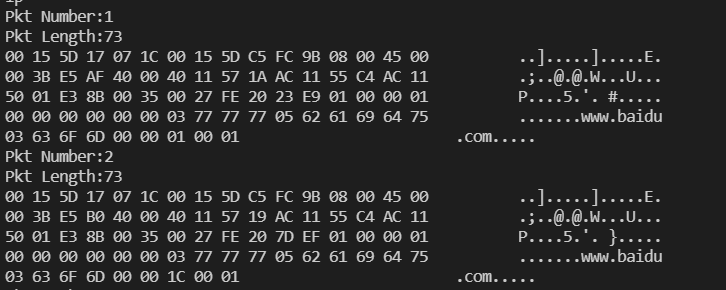
本程序可接收的命令行参数如下：

-a 打印数据包十六进制数据

-o 指定输出文件，输出的内容包括数据包的mac地址、ip地址、协议以及端口

## 5. 回调函数设计

首先读取数据包的长度，若指定了-a参数，则在标准输出打印数据包的十六进制数据和对应的字符串：



然后解析以太网数据帧，提取出协议字段和MAC地址，将MAC地址写到文件中，在IP数据包中解析出IP地址，判断协议字段是tcp协议还是udp协议，然后根据相应的数据结构进一步解析，最后提取出端口号，写入文件。



其中端口号字段要进行字节序的转换，由convert函数实现。

## 6. 编译运行

使用下列命令编译：

gcc sniffer.c -Wall -lpcap -o sniffer

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

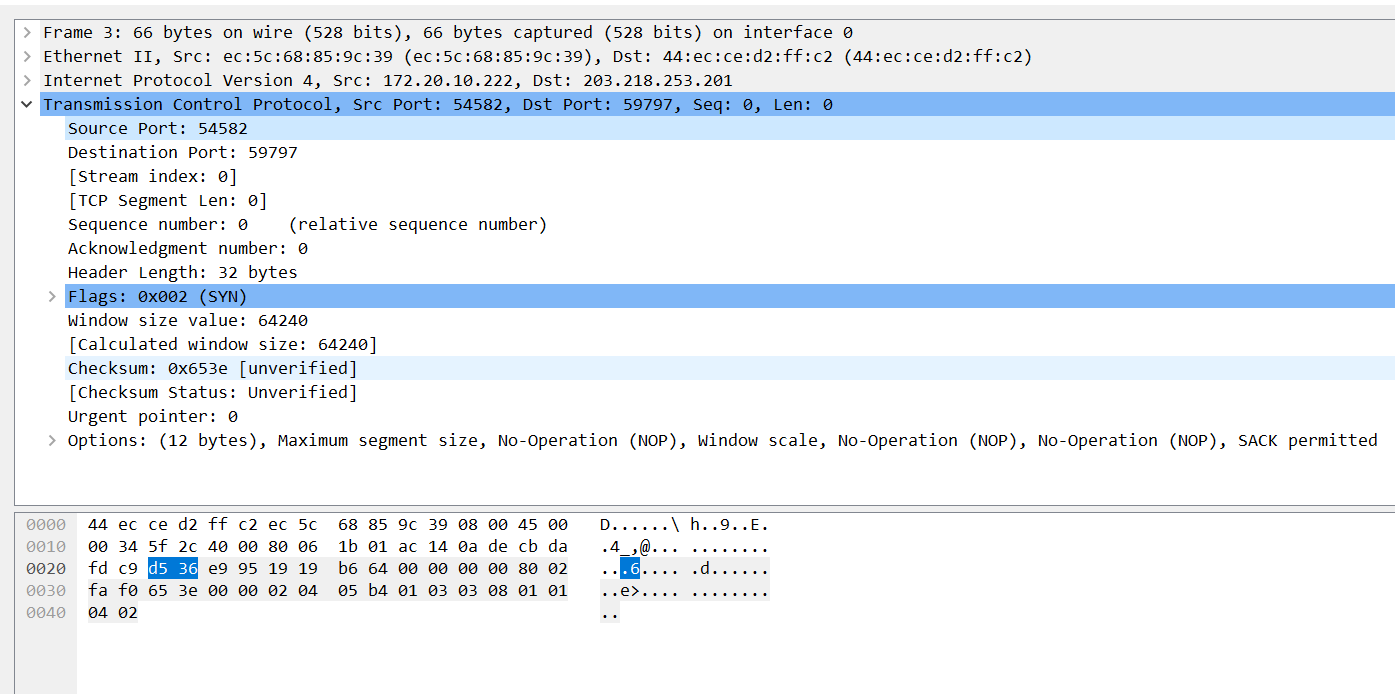
使用下列命令运行：

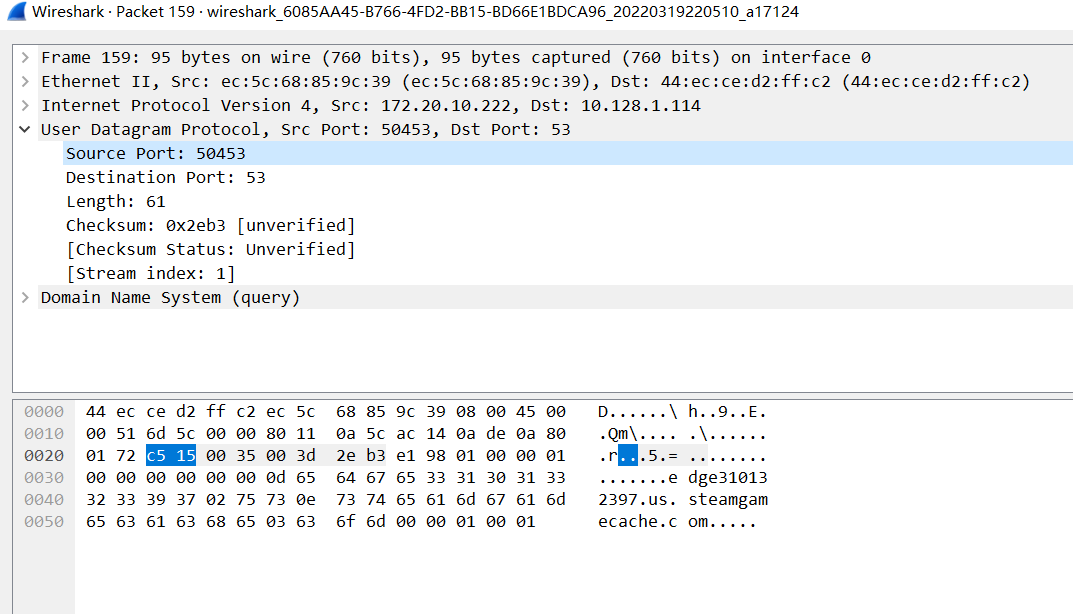
**sudo** ./sniffer

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**



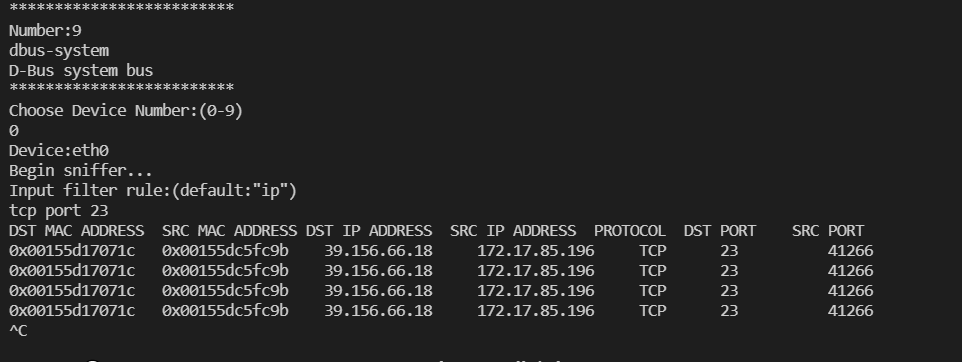


1. **利用 libpcap 编写捕包软件**

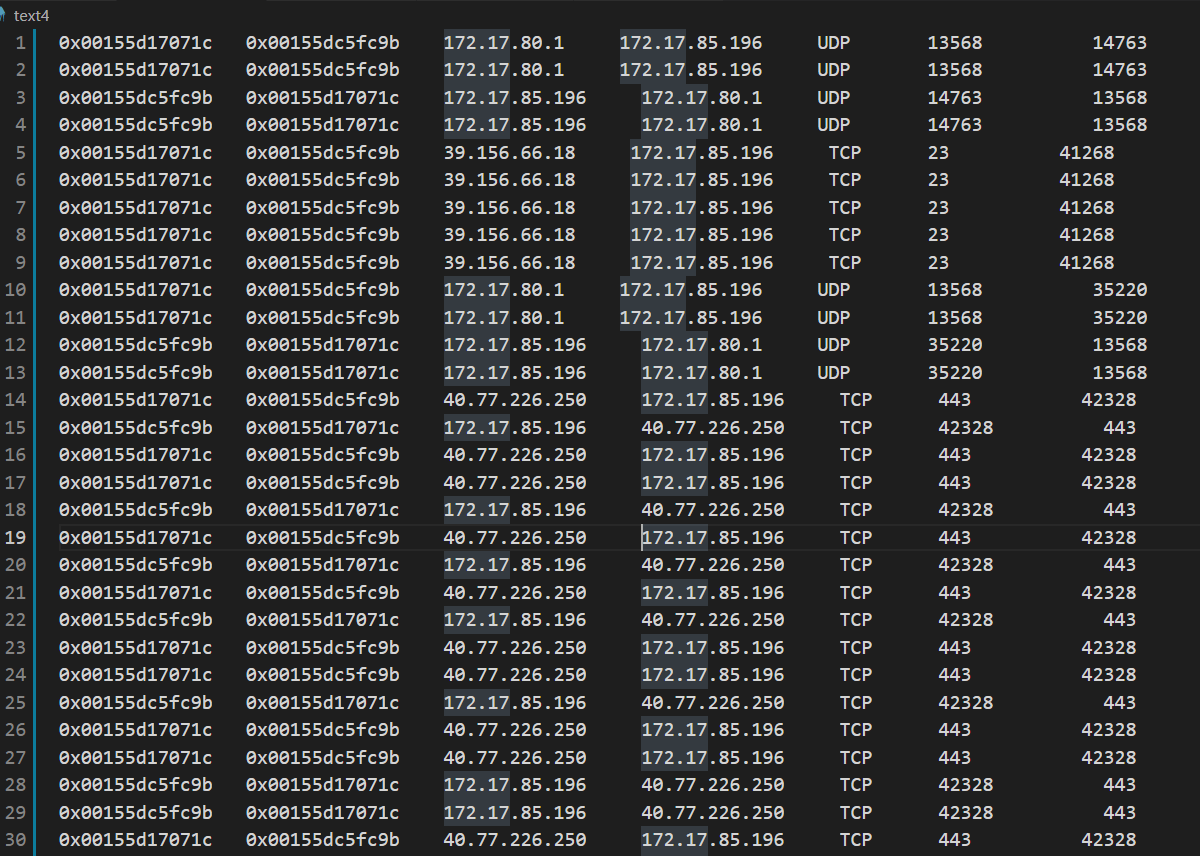
./sniffer -a 打印原始数据包



./sniffer 指定过滤规则：tcp port 23 ，即过滤telnet包



./sniffer -o text4



# 五、心得体会

1. 通过本次实验，我掌握了libpcap开发包的基本使用，对数据包捕获有了更深的了解。