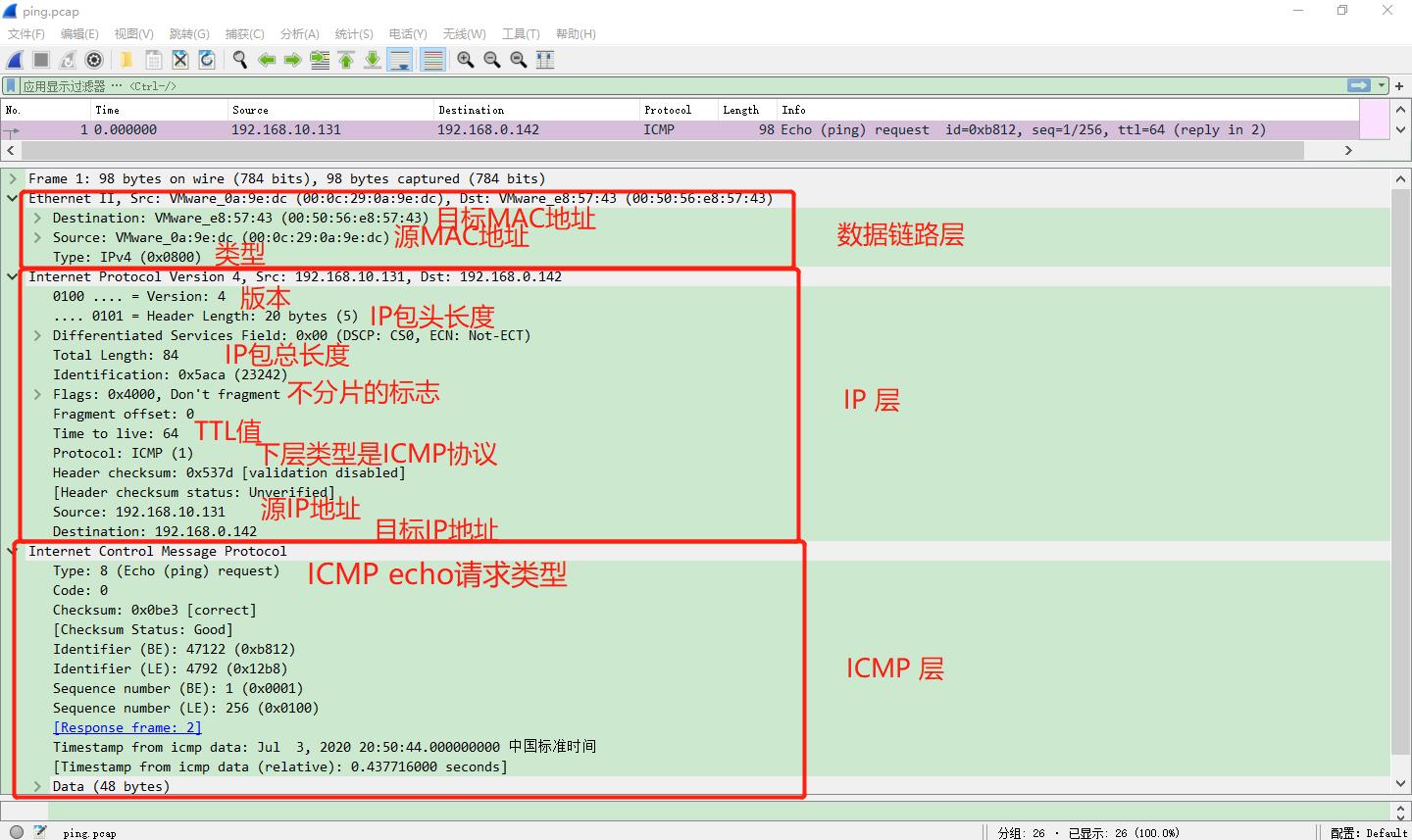
信息安全课程实验报告

# 1. 对ICMP协议的请求和应答包实施抓包



计算Header checksum：

(1)将校验和字段置为0：

将53 7d改成00 00，如下：

45,00,00,54,5a,ca,40,00,40,01,00,00,c0,a8,0a,83,c0,a8,00,8e

(2)求和

4500+0054+5aca+4000+4001+0000+c0a8+0a83+c0a8+008e

将将进位加到低16位上

(3)取反码

取反得:checksum=537d

代码如下：

#include<stdio.h>

int main()

{

int buff[10]={0x4500,0x0054,0x5aca,0x4000,0x4001,0x0000,0xc0a8,0x0a83,0xc0a8,0x008e};

int checksum=0,i=0;

for(i=0;i<10;i++)

{

checksum+=buff[i];

}

checksum=(checksum>>16)+(checksum & 0xffff);

checksum+=(checksum>>16);

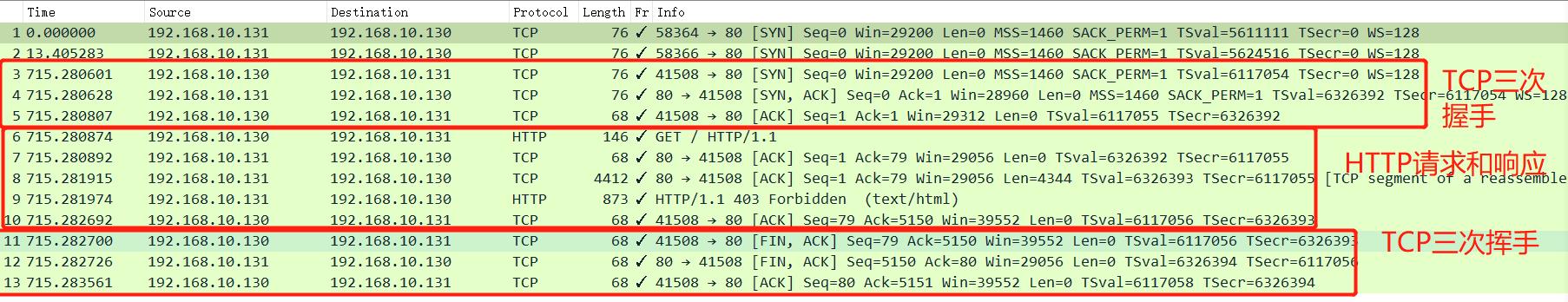
checksum=0xffff-checksum;

printf("checksum=%04x\n",checksum);

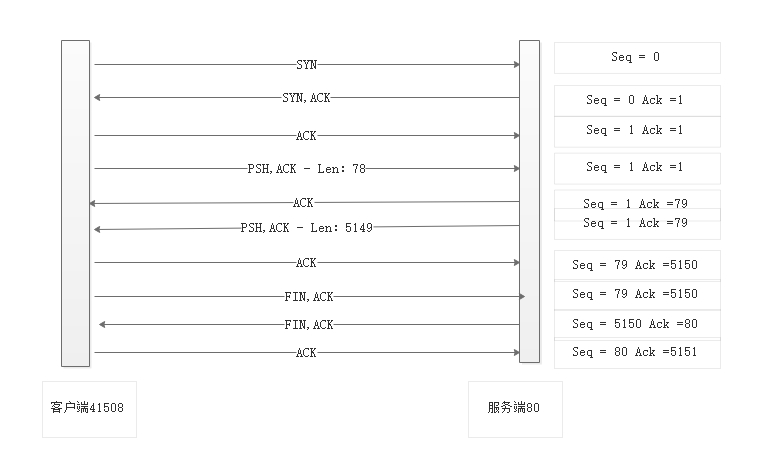
return 0;

}

# 2.一次TCP协议的三次握手和四次挥手抓包实验



分析：

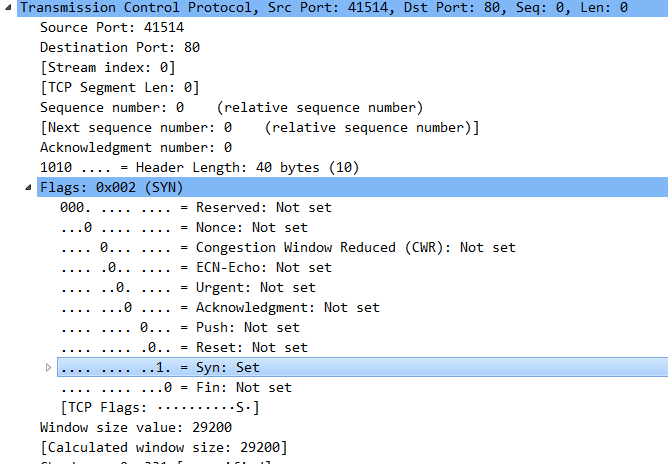
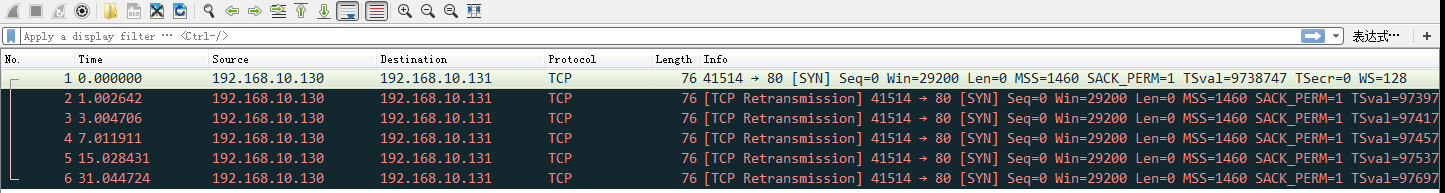


# **3.TCP丢包分析**

**TCP第一次握手的SYN丢包：**

操作：在服务端通过iptables -I INPUT -s 192.168.10.130 -j DROP 命令阻挡客户端的请求

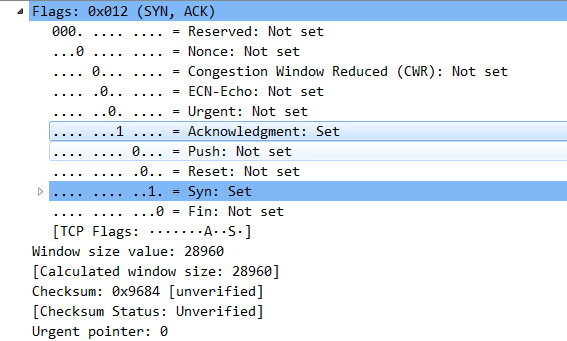
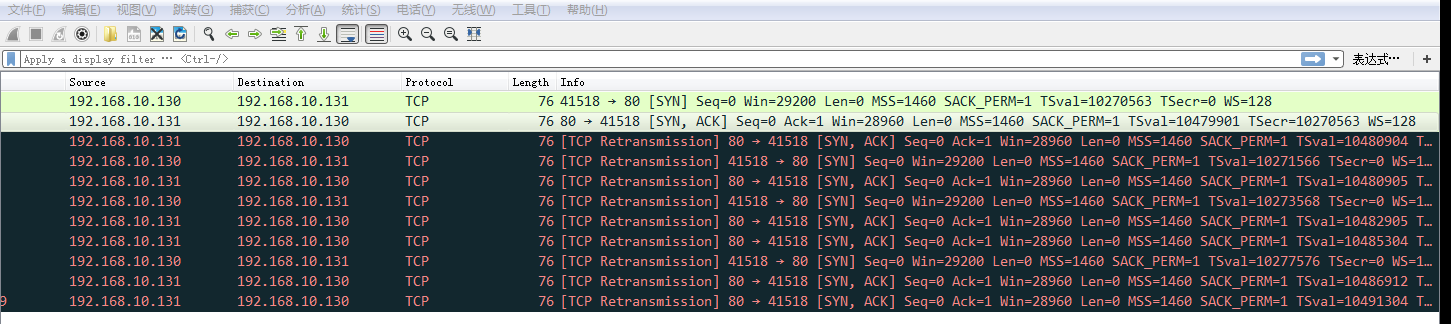
结果：客户端第一次握手请求重传五次，重传时间间隔增大。



**TCP第二次握手的SYN、ACK丢包：**

操作：在客户端通过iptables -I INPUT -s 192.168.10.131 -j DROP 命令阻挡服务端的请求

结果：服务端第二次握手请求重传，客户端继续第一次握手请求，并且客户端请求时间间隔在增大。



**TCP 第三次握手的 ACK 丢包**：

操作：在服务端通过iptables -I INPUT -s 192.168.10.130 -p tcp –tcp-flags ALL SYN,ACK -j REJECT 命令阻挡客户端第三次发来的SYN和ACK

结果：服务器端无法进入ESTABLISH状态，会定时发送第二步SYN+ACK数据包，而客户端暂时认为连接成功，一直维护此连接，但有超时设置，超时后服务端给客户端发RTS报文，并主动关闭连接，客户端也关闭连接。

