**实验背景：**

TCP/IP协议在设计之初仅考虑了成本和实现功能，并没有过多考虑安全因素。因此TCP/IP协议栈中提供了大量的起关键作用的信息和指令，但是这些信息和指令的执行缺乏认证机制，能够方便地伪造。由于历史原因，这些漏洞一直被保留，最多进行过简单的修补。通过TCP/IP协议栈的安全实验，可以加深TCP/IP协议的具体机制的理解。

**实验原理：**

 ICMP重定向信息是路由器向主机提供实时的路由信息，当一个主机收到ICMP重定向信息时，它就会根据这个信息来更新自己的路由表。由于缺乏必要的合法性检查，如果一个黑客想要被攻击的主机修改它的路由表，黑客就会发送ICMP重定向信息给被攻击的主机，让该主机按照黑客的要求来修改路由表。

在VMware虚拟环境中（Virtual Box不同），.2是充当网关的默认地址（通过route命令可以查看当前的网关和路由信息）；所以攻击者可以冒充.2发出重定向包，通知受害者修改自己的网关为攻击者指定的gw地址；如果伪造的gw是自身，可以实现中间人攻击或者DOS攻击（没有启动IP转发功能）；如果是随意IP（不能到达或不负责转发），则可以导致DOS攻击。

关于VMWare组网的基本原理可以参考：<http://blog.csdn.net/lionhunter2010/archive/2007/12/19/1954276.aspx>

**实验环境：**

启动两台虚拟机，一台作为攻击者，一台作为被攻击者。

**实验目标：**

1. 体验ICMP重定向攻击的效果，加深对ICMP协议功能的理解。
2. 练习使用Raw Socket编程，学习构造包括包头部分的数据包。

**实验步骤：**

1. 使用netwox体会实验效果，使用netwox 86发送ICMP重定向包，

Netwox 86 –gw 192.168.137.220 –src-ip 192.168.137.2

其中-gw指定新的网关地址；--src-ip是当前网关地址；也即攻击者冒充当前网关.2通知受害者修改自己的网关IP。

启动wireshark，观察netwox发出的数据包：

1. 通过wireshark抓包查看所发出的数据包的源IP是.2而不是攻击者真实的IP；
2. 通过抓包，查看攻击数据包的结构

请注意，ICMP重定向报文除了ICMP包中的通用头部之外，还包括原始IP头部信息和数据报文的前8个字节。也即，在构造ICMP重定向包中，除了头部之外，还需要额外的28字节（在IP头部没有可选字段的情况下）

1. 另外，注意观察，netwox发出的ICMP重定向包的目的IP是受害者正通信的IP，也即，netwox先抓到受害者的数据包，根据捕获包的IP地址，再构造攻击包。
2. 在充分了解实验原理的基础上，自己使用raw socket，写出来一个icmp redirect包，达到使受害者不能正常上网的目的。
3. 关于raw socket语法，可以使用man socket；man raw；man packet等查看细节，可以运行参考代码，并结合wireshark抓包查看代码生成的数据包。
4. 原始套接字是一个特殊的套接字类型,但它的创建方式跟TCP/UDP创建方法几乎是一摸一样,例如,通过

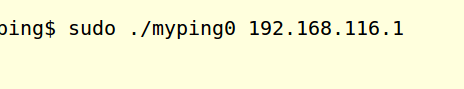
       int sockfd;  
       sockfd = socktet(AF\_INET， SOCK\_RAW， IPPROTO\_ICMP);

就可以创建一个原始套接字。 这种类型的原始套接字可以使用户自行生成IP数据包中的ICMP数据部分。

参考myping0.c进一步理解:

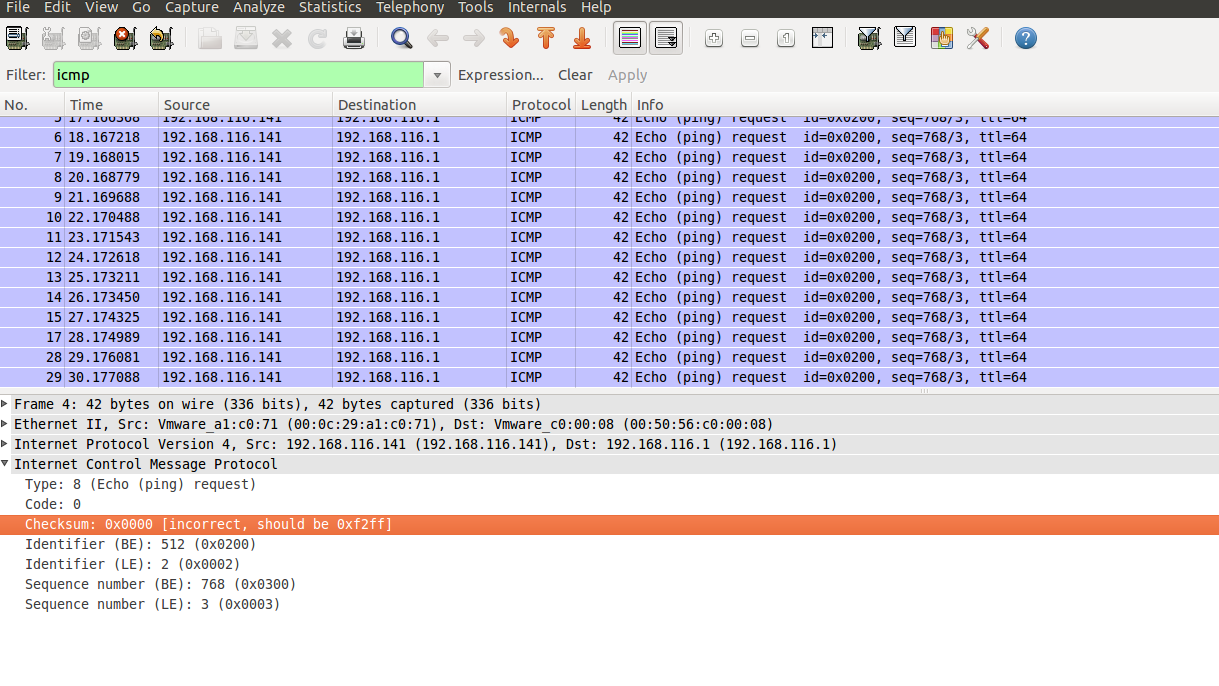
sockfd = socket(AF\_INET,SOCK\_RAW,IPPROTO\_ICMP);

生成一个ICMP原始包，自己仅填充ICMP部分，构造ping包，IP头部部分由系统生成。编译并运行该程序。



（根据自己的Vmware生成的网段修改IP地址）

然后通过Wireshark抓包，可以发现本文件中的代码生成的包已经也被Wireshark识别为ICMP Ping包，但是因为没有进行校验，包不正确，因此不会产生回复。Wireshark抓包时可以重点查看checksum字段，通过wireshark查看可以发现wireshark提示校验和错误。



上一个代码中的socket实际上并不能满足试验需求，因为ICMP重定向包需要冒充真正网关的IP地址（Vmware中虚拟机的默认网关是.2），也即程序中必须要自己设定源IP，所以必须使用另一种类型的socket.

参考myping2.c:

socket(AF\_INET,SOCK\_RAW,IPPROTO\_RAW)，使用IPPROTO\_RAW参数可以填充包括IP头部的内容。

参考myping6.c:

此文件中添加了checksum计算，发送的ping包正确；并且同时生成了一个用于接收icmp数据包的socket；可以打印出icmp数据包。

至此，已经能够成功构造ping包，可以在此基础上进一步生成ICMP重定向包。

1. IP头部和ICMP头部的数据结构可以参考头文件查看详细内容。（可以使用find或whereis等命令查找头文件的位置）。譬如，可以在/usr/include/netinet文件夹下找到ip.h等头文件。
2. 本实验需要先抓包再发包，请使用raw socket实现抓包，获取其中的信息，基于该信息构造ICMP重定向包中的附加原始数据包的内容。请一定注意使用的socket的类型。