Packet Sniffing and Spoofing Lab Report

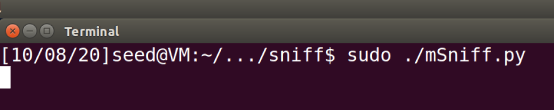
管箫 18307130012 2018级信息安全

**一、Task Set 1: Using tools to Sniff and Spoof Packets**

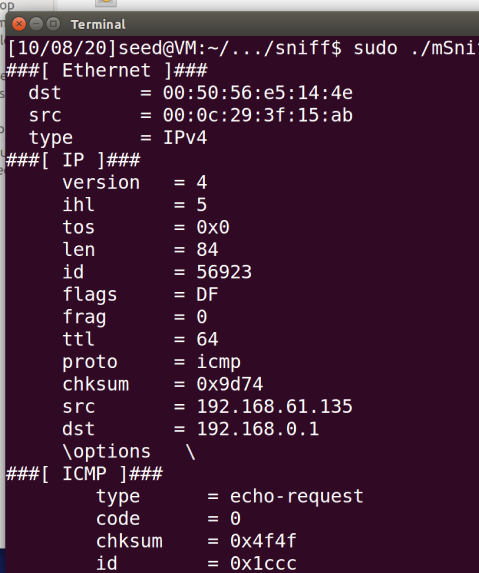
**1、Task1.1：Sniffing Packets**

①1.1A：

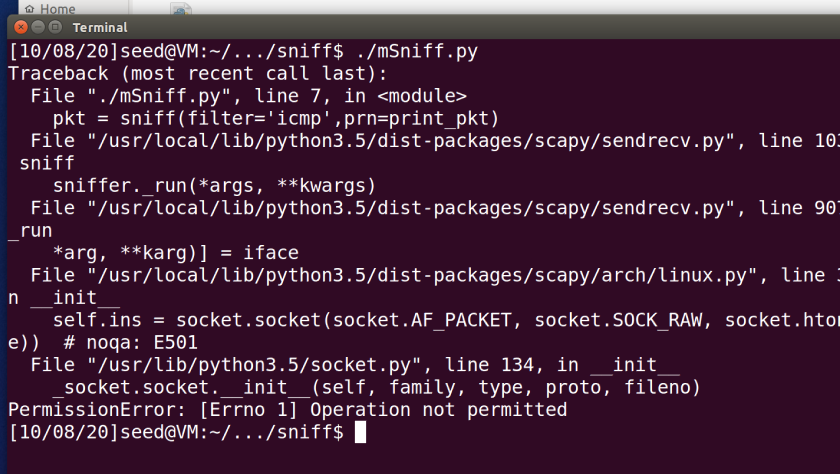
运行抓包程序，可以观察到程序在运行后因为无网络分组交换，进入阻塞状态。



对192.168.0.1执行ping命令，程序捕获icmp回显分组



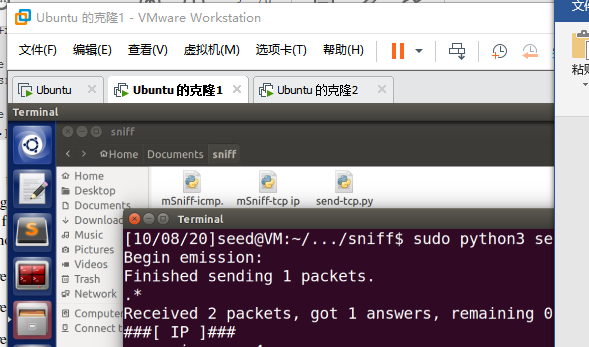
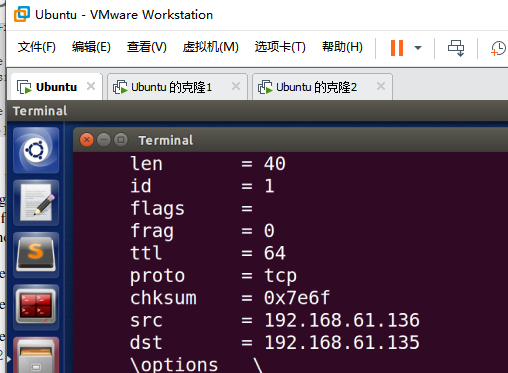
在不使用sudo命令的情况下运行抓包程序，观察到程序无法运行，观察调用栈可以意识到是的函数调用的socket底层指令无法获得系统级权限。



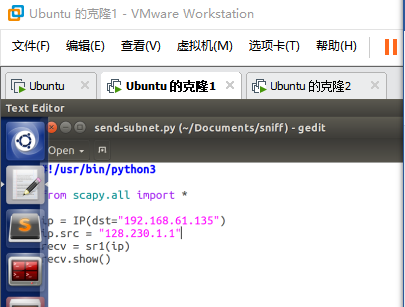
②1.1B：

只捕获icmp包：filter='icmp'，效果图已经在上面给出。

捕获来自特定IP，以及目标端口是23的任何TCP包：filter='ip src 192.168.61.136 and tcp and dst port 23'。

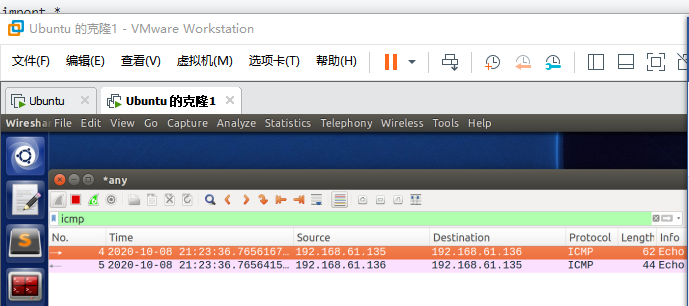
构造发送包：接收效果：

捕获来自或到达特定子网的数据包：filter='net 128.230.0.0/16'

构造发送包：接受效果：

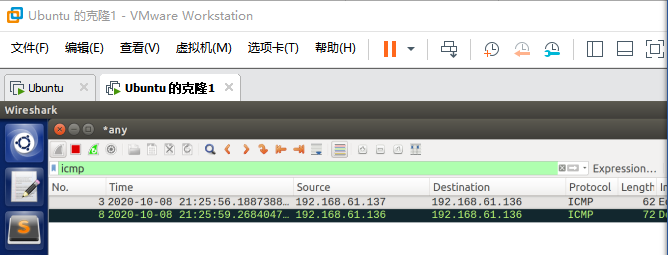
若不伪造源地址，则主机永远无法捕获该子网数据包，因为主机不在该子网内。

**2、Task1.2：Spoofing ICMP Packets**



直接发送icmp请求，使用wireshark捕获到正常的icmp请求和回显包。

此时我们将源地址请求伪造为192.168.61.137，并发送icmp请求



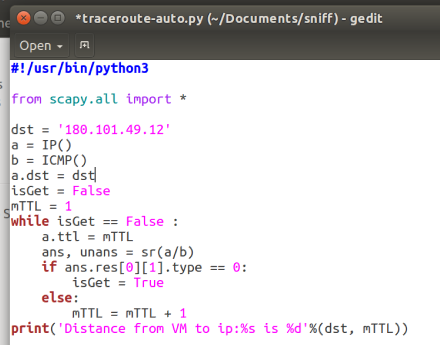
观察到目标主机接收到了包，并试图回复给并不存在的主机192.168.61.137

**3、Task1.3：Traceroute**

利用TTL每经过一个路由器值就会减去1的特性，逐个构造从1开始的TTL的icmp请求包。每个请求包的TTL在网络上减少到0时，就会被丢弃，同时丢弃该包的路由器会回发一个ICMP错误信息。通过在icmp请求包获得目的主机的响应前收到了多少个错误信息，我们可以估算VM到目的主机的路由器跳数。

编写程序，在发送数据包后接收并分析数据包，检查是否由目标主机发送回ICMP的回应，直到获得目的主机响应，并输出结果。

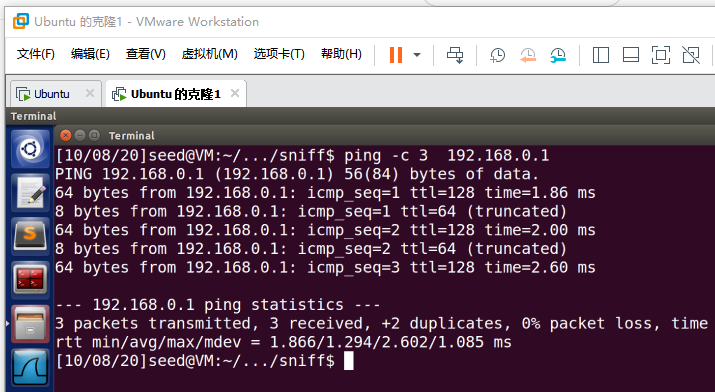
程序理论上可以正常运行，但无法收到TTL大于1的路由器的回应包，怀疑与虚拟机网络设置有关，手动发送icmp遇到同样错误。



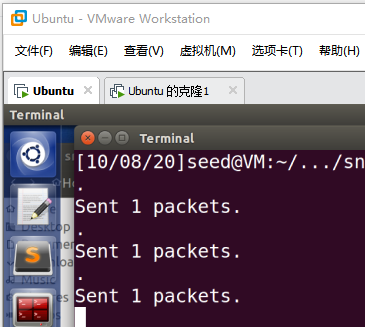
**4、Task1.4：Sniffing and-then Spoofing**

对子网中的所有icmp包进行响应，伪装其存在。

由主机B向192.168.0.1（实际不存在）发送icmp请求：



在主机A上运行欺骗程序：

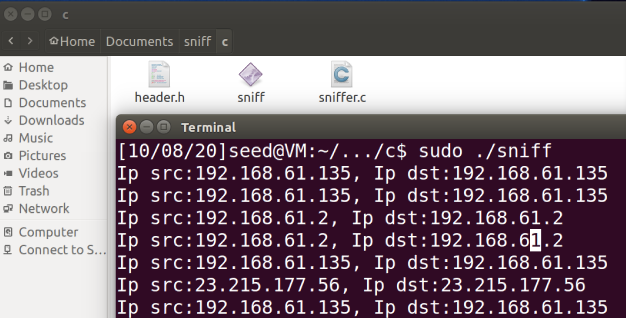


观察到欺骗程序响应了每个icmp请求，表现为每次请求得到欺骗程序响应后都会收到一个实际上的不可达的icmp响应。

**二、Task Set 2: Wirting Program to Sniff and Spoof Packets**

**1、Task2.1：Writing Packet Sniffing Program**

①Task2.1A：编写抓包程序并显示其ip地址



打开火狐，观察到捕获程序正常运行。

问题1：库函数调用顺序及作用

// 打开选定的网卡开始嗅探

handle = pcap\_open\_live("ens33", BUFSIZ, 1, 1000, errbuf);

// 将字符串形式的过滤规则编译成BFP形式，保存在fp中

pcap\_compile(handle, &fp, filter\_exp, 0, net);

// 将filter加入到handle中

pcap\_setfilter(handle, &fp);

// 开始监听，当接收到符合过滤规则的数据包时回调 got\_packet函数

pcap\_loop(handle, -1, got\_packet, NULL);

// 关闭监听，释放网卡

pcap\_close(handle); //Close the handle

问题2：当不使用特权模式时，会发生segmentation fault

可能原因是不使用特权模式时，程序访问由系统控制的内存空间时没有访问权限

问题3：打开关闭混杂模式的区别

通过修改pcap\_open\_live函数的第三个参数，我们可以打开或关闭混杂模式。

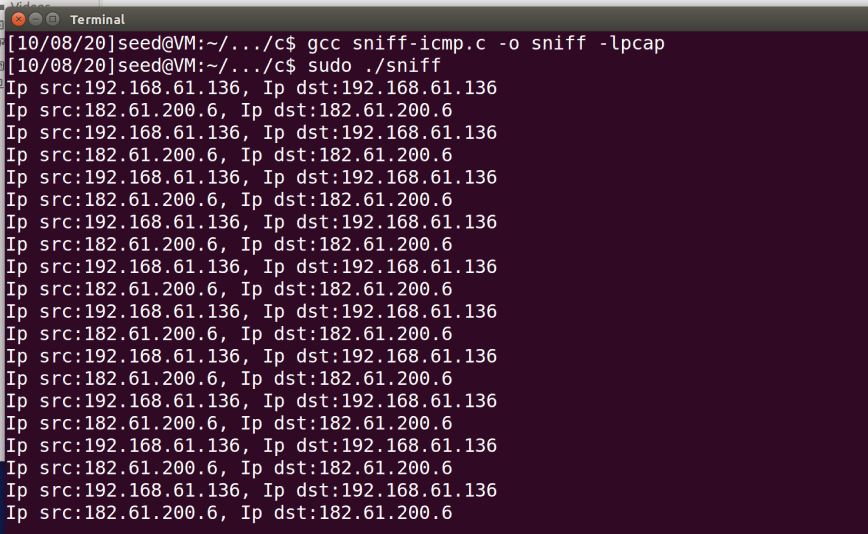
当混杂模式打开时，主机可以监听子网中任一主机的信息，反之只能接收目标为自身的包。

②Task2.1B：编写过滤规则

问题1：捕获特定host之间的ICMP包：

char filter\_exp[] = "icmp && host 192.168.61.136 && host 182.61.200.6";

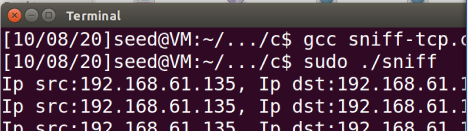
在主机上捕获另一台虚拟机和百度之间的icmp：



问题2：捕获所有端口号在10到100之间的TCP包：

char filter\_exp[] = "tcp && dst portrange 10-100";

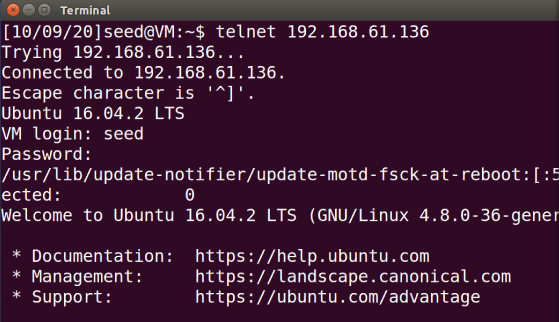
使用火狐访问因特网，发现正常捕获



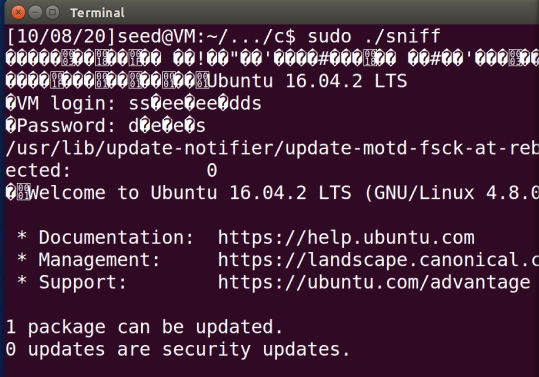
③Task2.1C：嗅探密码

设置过滤规则为tcp port 23

在虚拟机C上远程登录虚拟机B



在虚拟机A上捕获到其发送的内容：

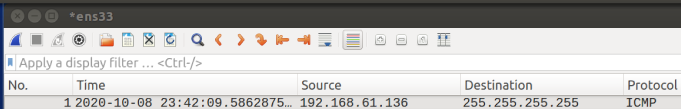


观察到telnet使用明文发送报文内容，很容易泄露账户名密码。

**2、Task2.2：Spoofing**

①Task2.2A：编写一个欺骗程序

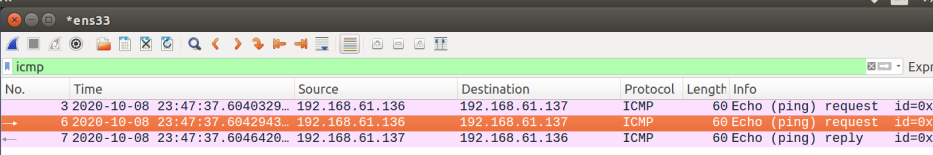
我们使用地址为192.168.61.135的虚拟机，伪造一个由192.168.61.136发送给255.255.255.255的icmp包



观察到在另一台虚拟机上，该包被捕获。

②Task2.2B：伪造icmp请求

我们使用地址为192.168.61.135的虚拟机，伪造一个由192.168.61.136发送给192.168.61.137的icmp包



我们观察到在另一虚拟机上，该包可以被捕获，并且目的主机正常回复了icmp响应给虚构的主机

问题4：是否可以给ip包长度设定任意值？

我们将生成的ip包长度设置为1，但观察到捕获的数据包中，长度仍然为20



似乎并不能进行这样的更改，怀疑socket底层对长度进行了自动校验。

问题5：是否要主动进行IP头的校验和计算？

不必要，在本程序运行中就没有添加计算功能，仍然能正常发送。

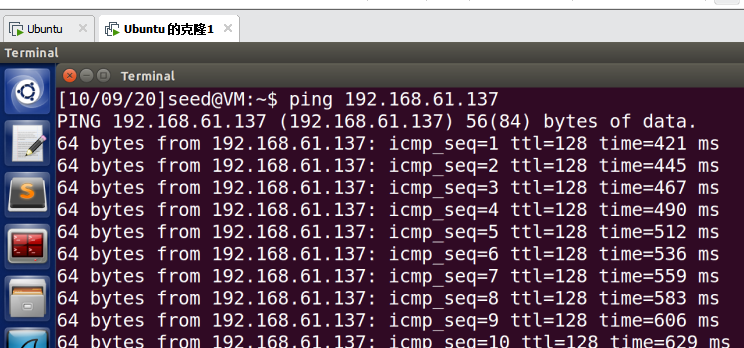
问题6：为什么需要特权模式来运行程序？如果不使用特权模式会发生什么？

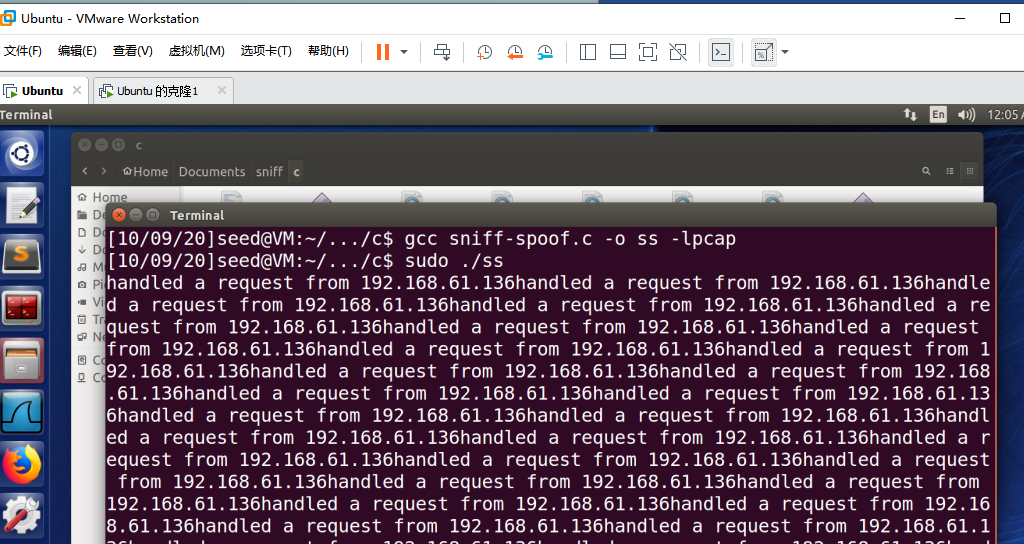


如果不使用特权模式运行程序，程序会在调用socket层的时候出错，因为raw socket是系统级命令，必须由具有系统级权限的操作者调用。

**3、Task2.3：Sniff and then Spoof**

在192.168.61.136上向未开机的192.168.61.137发送icmp请求，并且在192.168.61.135上运行伪造程序。





**三、实验讨论：**

1、实现嗅探的基础：

①网卡能够运行在混杂模式，也即网卡将会捕获子网内所有发送到自身的数据包，尽管该数据包并不是发送给自身的。

②在我们的网络中，大量数据包是采用广播发送的，即使并非目标主机也会收到包。

2、实现欺骗的基础：

①我们的TCP/IP协议族并不对网络数据包中的信息进行校验，无条件信任其目标和来源信息。

②我们可以控制socket层甚至更底层的数据包构造，实现对数据包的自定义。

3、可行的防御措施：

①对数据包的来源进行真伪性检验。

②传输数据时进行加密传输而非明文传输。

③使用路由器进行网络隔离，防止广播数据包的产生。

4、icmp报文类型

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **类型** | **代码** | **状态** | **描述** | **查询** | **差错** |
| 0 - Echo Reply | 0 |  | echo响应 (被程序ping使用） | ● |  |
| 1 and 2 |  | 未分配 | 保留 |  | ● |
| 3 - 目的不可达 | 0 |  | 目标网络不可达 |  | ● |
| 1 |  | 目标主机不可达 |  | ● |
| 2 |  | 目标协议不可达 |  | ● |
| 3 |  | 目标端口不可达 |  | ● |
| 4 |  | 要求分段并设置DF flag标志 |  | ● |
| 5 |  | 源路由失败 |  | ● |
| 6 |  | 未知的目标网络 |  | ● |
| 7 |  | 未知的目标主机 |  | ● |
| 8 |  | 源主机隔离（作废不用） |  | ● |
| 9 |  | 禁止访问的网络 |  | ● |
| 10 |  | 禁止访问的主机 |  | ● |
| 11 |  | 对特定的TOS 网络不可达 |  | ● |
| 12 |  | 对特定的TOS 主机不可达 |  | ● |
| 13 |  | 由于过滤 网络流量被禁止 |  | ● |
| 14 |  | 主机越权 |  | ● |
| 15 |  | 优先权终止生效 |  | ● |
| 5 - 重定向 | 0 |  | 重定向网络 |  | ● |
| 1 |  | 重定向主机 |  | ● |
| 2 |  | 基于TOS 的网络重定向 |  | ● |
| 3 |  | 基于TOS 的主机重定向 |  | ● |
| 8 - 请求回显 | 0 |  | Echo请求 | ● |  |
| 9 - 路由器通告 | 0 |  | 路由通告 | ● |  |
| 10 - 路由器请求 | 0 |  | 路由器的发现/选择/请求 | ● |  |
| 11 - ICMP 超时 | 0 |  | TTL 超时 |  | ● |
| 1 |  | 分片重组超时 |  | ● |
| 12 - 参数问题：错误IP头部 | 0 |  | IP 报首部参数错误 |  | ● |
| 1 |  | 丢失必要选项 |  | ● |
| 2 |  | 不支持的长度 |  |  |
| 13 - 时间戳请求 | 0 |  | 时间戳请求 | ● |  |
| 14 - 时间戳应答 | 0 |  | 时间戳应答 | ● |  |

5、icmp校验和计算

Icmp校验和计算范围是icmp包头加其数据部分