

ARPSpoofing—report

PB5111662 李双利

• 1. 实验准备

- 在VMware下配置两台虚拟机实验环境，分别为

- Ubuntu 18.04 lts 64位
- Windows 10 32位

- 网络配置

此次ARP攻击是在局域网下进行的，否则无法实施攻击。

所以需要在虚拟机下配置好局域网的网络环境，具体在VMware中需要将虚拟机的网络连接修改为桥接模式（默认为NAT连接），这样才能保证攻击主机和被攻击主机在同一网段。

网络连接

☒ 桥接模式(B): 直接连接物理网络

☒ 复制物理网络连接状态(P)

☐ NAT 模式(N): 用于共享主机的 IP 地址

☐ 仅主机模式(H): 与主机共享的专用网络

- arpspoof源码以及编译环境

在攻击主机（Ubuntu 18.04）中下载源码：

```
sudo apt-get source dsniiff
```

安装编译依赖库

```
sudo apt-get install libnet1
sudo apt-get install libpcap-dev
sudo apt-get install libnet1-dev
```

• 2. 修改并编译arpspoof源码

提取dsniiff源码目录下的 `arp.c`，`arp.h` 和 `arpspoof.c` 三个文件进行整合，增加相关注释和相关标记后方便用gcc进行编译，生成arpspoof可执行文件进行攻击。

阅读过源码并且整合为一个arpspoof文件之后，在打印之处进行修改做标记，修改部分的代码如下所示，将每次打印的前面加注学号（PB15111662）作为标记。

```

fprintf(stderr, "[PB15111662 ARPSpoofing Lab]\n%s ",
           ether_ntoa((struct ether_addr *)me));
/*回显处理*/
if (op == ARPOP_REQUEST) {
    fprintf(stderr, "[-----]%s 0806 42: arp who-has %s tell %s\n",
            ether_ntoa((struct ether_addr *)tha),
            libnet_addr2name4(tpa, LIBNET_DONT_RESOLVE),
            libnet_addr2name4(spa, LIBNET_DONT_RESOLVE));
}
else {
    fprintf(stderr, "[-----]%s 0806 42: arp reply %s is-at ",
            ether_ntoa((struct ether_addr *)tha),
            libnet_addr2name4(spa, LIBNET_DONT_RESOLVE));
    fprintf(stderr, "%s\n",
            ether_ntoa((struct ether_addr *)sha));
}

```

然后用GCC编译器进行编译即可生成可执行文件

```
gcc arpspoof.c -lnet -lpcap -o arpspoof
```

• 3. 进行攻击

开启虚拟机之后，首先查看攻击主机和被攻击主机的IPv4地址：

```
C:\Users\Administrator>arp -a
```

接口: 172.20.10.14 --- 0x2	Internet 地址	物理地址	类型
	172.20.10.1	00-0c-29-b1-fa-78	动态
	172.20.10.13	00-0c-29-b1-fa-78	动态
	172.20.10.15	ff-ff-ff-ff-ff-ff	静态
	224.0.0.22	01-00-5e-00-00-16	静态
	224.0.0.252	01-00-5e-00-00-fc	静态
	239.255.255.250	01-00-5e-7f-ff-fa	静态
	255.255.255.255	ff-ff-ff-ff-ff-ff	静态

这是在被攻击主机上查看arp表的结果，172.20.10.14为被攻击主机的地址，172.20.10.13为攻击主机的地址，且两台主机处于同一网段。

然后在攻击主机尝试fping一下被攻击主机，已检验是否满足实验条件：

```
shuangli@ubuntu:~/Documents/arpspoof/dsniff-2.4b1+debian$ fping 172.20.10.14
172.20.10.14 is alive
```

查看攻击主机的内核IP路由表，从而确定Iface为ens33

```
shuangli@ubuntu:~/Documents/arpspoof/dsniff-2.4b1+debian$ netstat -rn
```

内核 IP 路由表							
Destination	Gateway	Genmask	Flags	MSS	Window	irrt	Iface
0.0.0.0	172.20.10.1	0.0.0.0	UG	0	0	0	ens33
169.254.0.0	0.0.0.0	255.255.0.0	U	0	0	0	ens33
172.20.10.0	0.0.0.0	255.255.255.240	U	0	0	0	ens33

在攻击主机上打开四个终端窗口，分别执行以下命令

在攻击主机终端下的执行过程如下：

可以看到，正确的输出了修改后源码的打印内容。同时再查看被攻击主机的arp表，

发现172.20.10.1的物理地址被定向到了攻击主机的物理地址，被攻击主机误信172.20.10.1的硬件地址是00-0c-29-b1-fa-78，并且动态更新缓存表。这时在被攻击主机上ping的时候发现失败。

