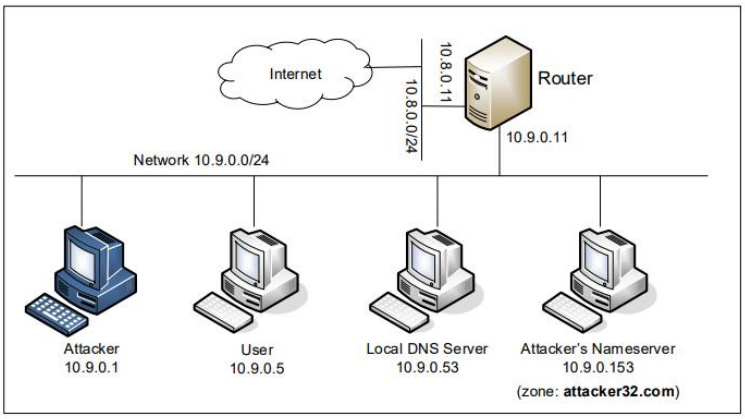
COMPUTER NETWORK SECURITY LABORATORY

NAME : PREM SAGAR J S	SRN : PES1UG20CS825	SEC : H

Local DNS Attack Lab

Lab Environment Setup



Verification of the DNS setup

Getting the IP address of ns.attacker32.com

On the victim terminal run the command:
dig ns.attacker32.com

```
User:PES1UG20CS825:Prem Sagar J S/
#dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25395
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 7a3aafbe2d87bd35010000006345a6e7938350593fab7df0 (good)
;; QUESTION SECTION:
;ns.attacker32.com.             IN      A

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

;; Query time: 44 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Oct 11 17:24:55 UTC 2022
```

Getting the IP address of www.example.com

On the victim terminal run the commands:
dig www.example.com

```
User:PES1UG20CS825:Prem Sagar J S/
#dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62584
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 53256d674614773f010000006345a736b3ef71e657b1eac8 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 2019 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Oct 11 17:26:14 UTC 2022
```

dig @ns.attacker32.com www.example.com

```
User:PES1UG20CS825:Prem Sagar J S/
#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: 46963
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: f701f59cc783736a010000006345a76da8cdee974715b493 (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)
```

Attacks on DNS

Task 1: Directly Spoofing Response to User

when the client sends the DNS request to the local DNS server it accepts a response back, but if the attacker sends a spoofed DNS response to the user before the legitimate attack from
the local DNS server then the attack is successful.

On the local DNS server's terminal run the command:
rndc flush

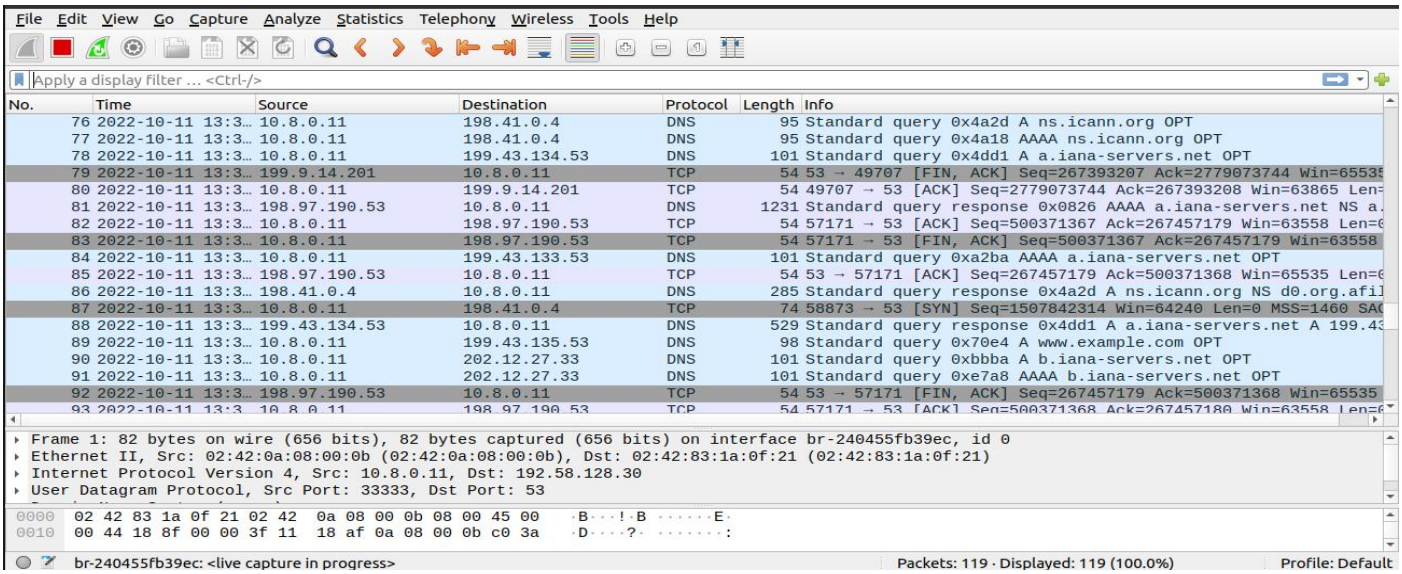
```
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#rndc flush
```

The victim machine sends out a DNS query to the local DNS server, which will eventually send out a
DNS query to the authoritative nameserver of the example.com domain. This is done using the dig
command. Before running the command keep wireshark open to view the packets being sent.

On the victim terminal run the command:
dig www.example.com

```
User:PES1UG20CS825:Prem Sagar J S/
#dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62664
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: fd02911928a6b929010000006345a8a873de5d4b305c6823 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 500 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Oct 11 17:32:24 UTC 2022
```

Wireshark Screenshot of capturing dns packets:



Before launching the attack, making sure that the cache in the local DNS server is cleaned.

On the local DNS server's terminal run the command:
rndc flush

```
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#rndc flush
```

On the attacker terminal run the command:
python3 task1.py

```
Attacker:PES1UG20CS825:Prem Sagar J S/volumes/Code
#python3 task1.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        = 30452
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0xef59
     src       = 10.9.0.5
     dst       = 10.9.0.53
     \options   \
###[ UDP ]###
```

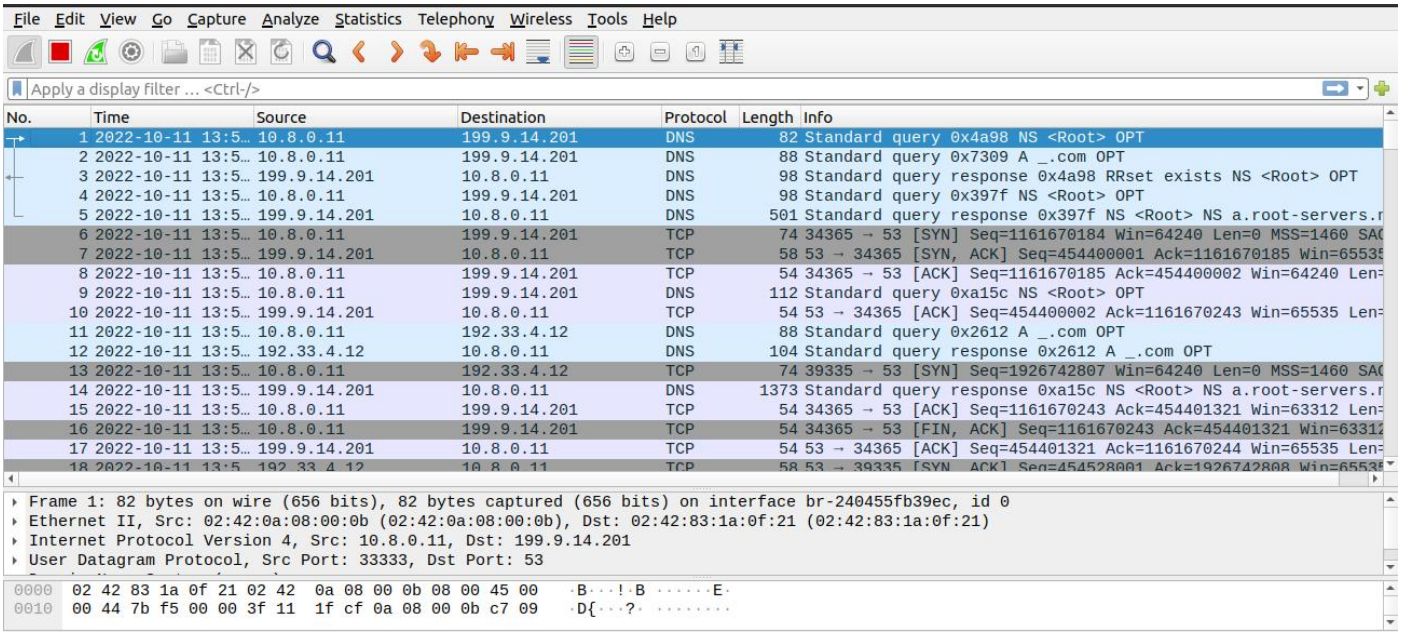
On the victim terminal run the command:
dig www.example.com

```
User:PES1UG20CS825:Prem Sagar J S/
#dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62425
;; 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: 84 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Oct 11 17:57:43 UTC 2022
;; MSG SIZE  rcvd: 64

User:PES1UG20CS825:Prem Sagar J S/
```

The Wireshark on the attacker machine shows the spoofed response which is sent to the victim.

The IP address mapped to www.example.com is 1.1.1.1 which is seen in the above image. We can see that the spoofed response comes before the legitimate response and hence is displayed as such in the victim machine.

Wireshark Screenshot :



the cache on the local DNS server we can use the rndc command to dump the cache and this dump is stored in **/var/cache/bind/dump.db** in our case.

On the local DNS server's terminal run the commands:
rndc dumpdb -cache
cat /var/cache/bind/dump.db | grep example

```
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#rndc dumpdb -cache
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#cat /var/cache/bind/dump.db | grep example
example.com.            691005  NS      a.iana-servers.net.
                                        20221022214625 20221001223409 1686 example.com.
www.example.com.        691005  A       93.184.216.34
                                        20221022040544 20220930163209 1686 example.com.
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#
```

A potential issue

We intentionally slow down the traffic going to the outside, so the authentic replies will not come that fast. This can be done using the following tc command on the router to add some delay to the outgoing network traffic.

```
Router:PES1UG20CS825:Prem Sagar J S/
#tc qdisc add dev eth0 root netem delay 100ms
Router:PES1UG20CS825:Prem Sagar J S/
#tc qdisc del dev eth0 root netem
Router:PES1UG20CS825:Prem Sagar J S/
#tc qdisc show dev eth0
qdisc noqueue 0: root refcnt 2
Router:PES1UG20CS825:Prem Sagar J S/
#
```

Task 2: DNS Cache Poisoning Attack – Spoofing Answers

A better way to conduct attacks by targeting the DNS server, instead of the user's machine. When a local DNS server receives a query, it first looks for the answer from its own cache; if the answer is there, the DNS server will simply reply with the information from its cache. If the answer is not in the cache, the DNS server will try to get the answer from other DNS servers. When it gets the answer, it will store the answer in the cache, so next time, there is no need to ask another DNS server.

On the local DNS server's terminal run the command:
rndc flush

```
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#rndc flush
```

On the attacker terminal run the command:
python3 task2.py

```
Attacker:PES1UG20CS825:Prem Sagar J S/volumes/Code
#python3 task2.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        = 8997
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0xd6
     src       = 10.9.0.53
     dst       = 199.43.133.53
     \options   \
###[ UDP ]###
```

The victim machine sends out a DNS query to the local DNS server using the dig command.

On the victim terminal run the command:
dig www.example.com

```
User:PES1UG20CS825:Prem Sagar J S/
#dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30961
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 41cb9863b3af9073010000006345b297411c86eee23977fc (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 2918 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Oct 11 18:14:47 UTC 2022
```

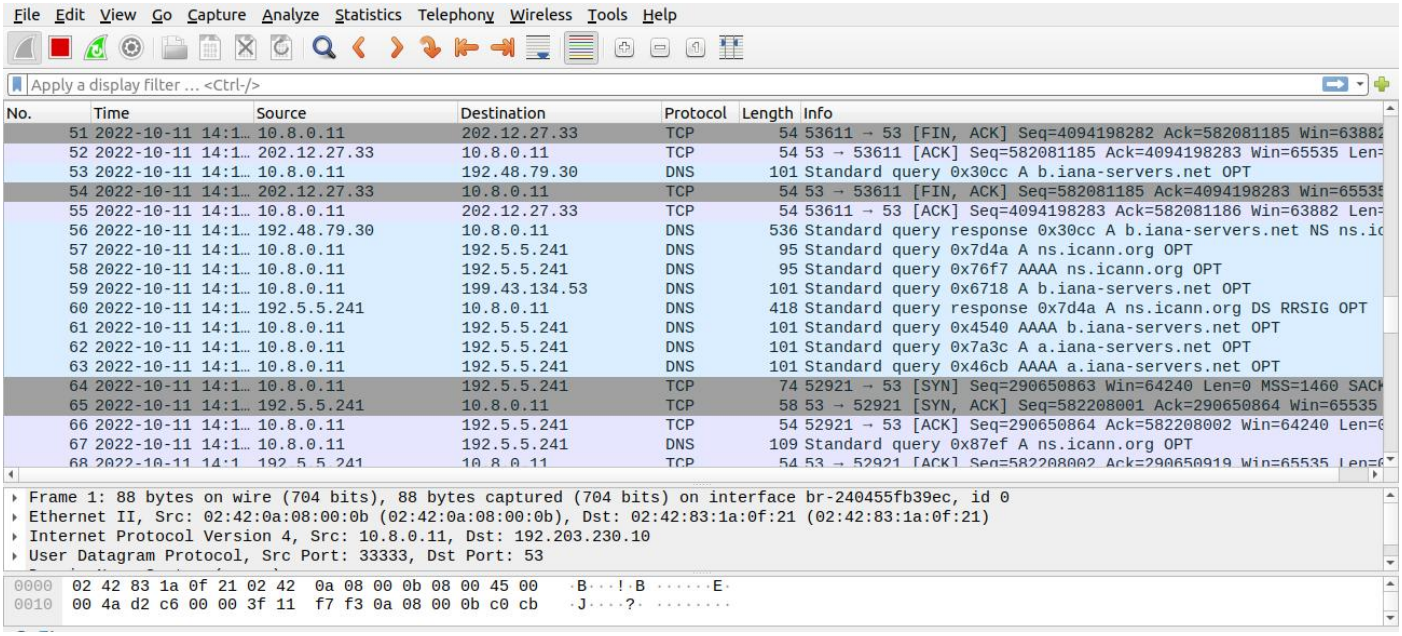
the spoofed packet captured on wireshark and the cache of the local DNS server

On the local DNS server's terminal run the commands:
rndc dumpdb -cache

cat /var/cache/bind/dump.db | grep example

```
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#rndc dumpdb -cache
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#cat /var/cache/bind/dump.db | grep example
example.com.            777575  NS      a.iana-servers.net.
www.example.com.        863976  A       1.1.1.1
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#
```

Wireshark screenshot :



Task 3: Spoofing NS Records

The idea is to use the Authority section in DNS replies. Basically, when we spoofed a reply, in addition to spoofing the answer (in the Answer section), we add the following in the Authority section. When this entry is cached by the local DNS server, ns.attacker32.com will be used as the nameserver for future queries of any hostname in the example.com domain. Since ns.attacker32.com is controlled by attackers, it can provide a forged answer for any query.

```
;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.attacker32.com.
```

On the local DNS server's terminal run the command:
rndc flush

```
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#rndc flush
```

On the attacker terminal run the command:
python3 task3.py

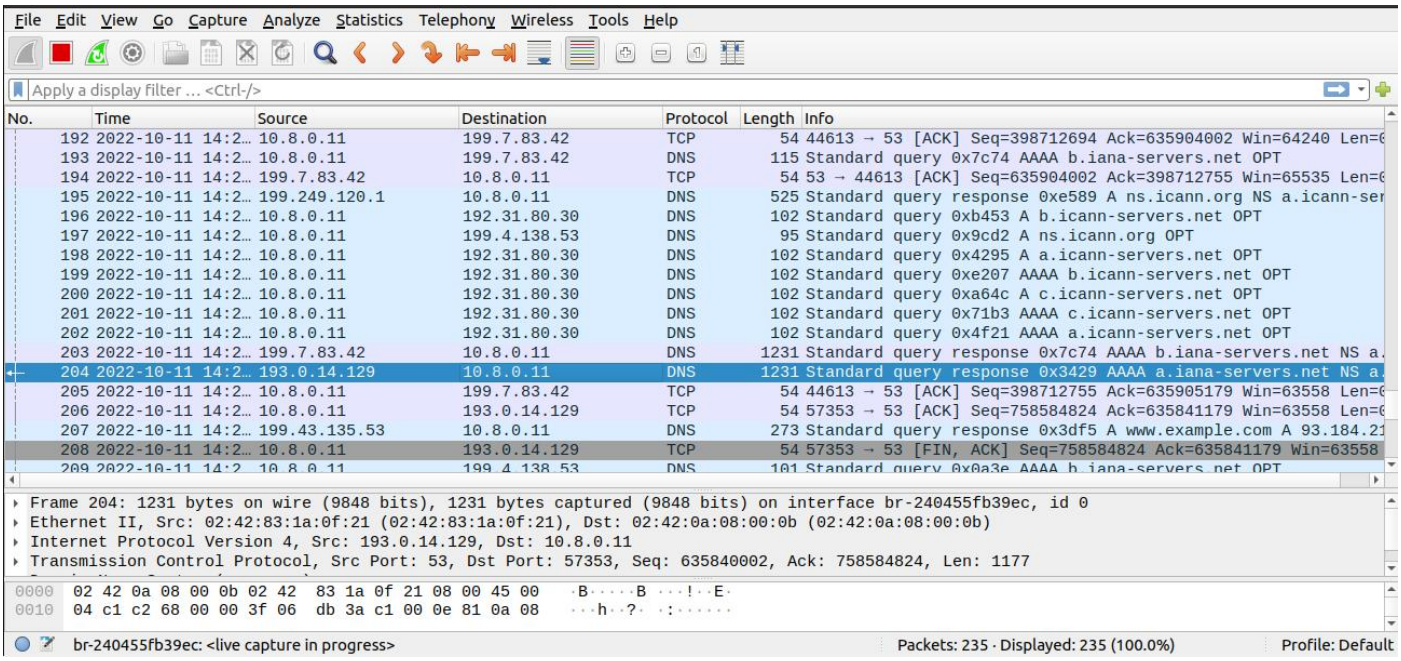
```
Attacker:PES1UG20CS825:Prem Sagar J S/volumes/Code
#python3 task3.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:0b
  src       = 02:42:0a:09:00:35
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 40068
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0x8576
     src       = 10.9.0.53
     dst       = 199.43.135.53
     \options   \
###[ UDP ]###
```

On the victim terminal run the command:
dig www.example.com

```
User:PES1UG20CS825:Prem Sagar J S/
#dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32197
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 92c71113514cf543010000006345b444c388985d7ecd07b4 (good)
```

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

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

;; Query time: 3999 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Oct 11 18:21:56 UTC 2022
```

Wireshark Screenshot :



If my attack is successful, when i run the dig command on the user machine for any hostname in the example.com domain, you will get the fake IP address provided by ns.attacker32.com.

On the victim terminal run the command:
dig www.example.com

```
User:PES1UG20CS825:Prem Sagar J S/
#dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27285
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: e313b3a2df8751ea010000006345b54e93e210a161ec24e0 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        258934  IN      A       1.1.1.1

;; Query time: 3 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Oct 11 18:26:22 UTC 2022
```

dig ftp.example.com

```
User:PES1UG20CS825:Prem Sagar J S/
#dig ftp.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> ftp.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9785
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: c44d5d3815016d2c010000006345b57a65c604fcf94734ac (good)
;; QUESTION SECTION:
;ftp.example.com.               IN      A

;; ANSWER SECTION:
ftp.example.com.        259200  IN      A       1.2.3.6

;; Query time: 31 msec
```

On the local DNS server's terminal run the commands:
rndc dumpdb -cache
cat /var/cache/bind/dump.db | grep example

```
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#rndc dumpdb -cache
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#cat /var/cache/bind/dump.db | grep example
example.com.            777211  NS      ns.attacker32.com.
ftp.example.com.        863923  A       1.2.3.6
www.example.com.        863613  A       1.1.1.1
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#
```

Task 4: Spoofing NS Records for Another Domain

the spoofed response triggered by a query for www.example.com, we would like to add additional entry in the Authority section (see the following),
so ns.attacker32.com is also used as the nameserver for google.com. The goal of this task is to see whether the entries we provide in the authority section are cached on the local DNS server or not and explain your results.

```
;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.attacker32.com.
google.com.             259200  IN      NS      ns.attacker32.com.
```

On the local DNS server's terminal run the command:
rndc flush

```
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#rndc flush
```

On the attacker terminal run the command:
python3 task4.py

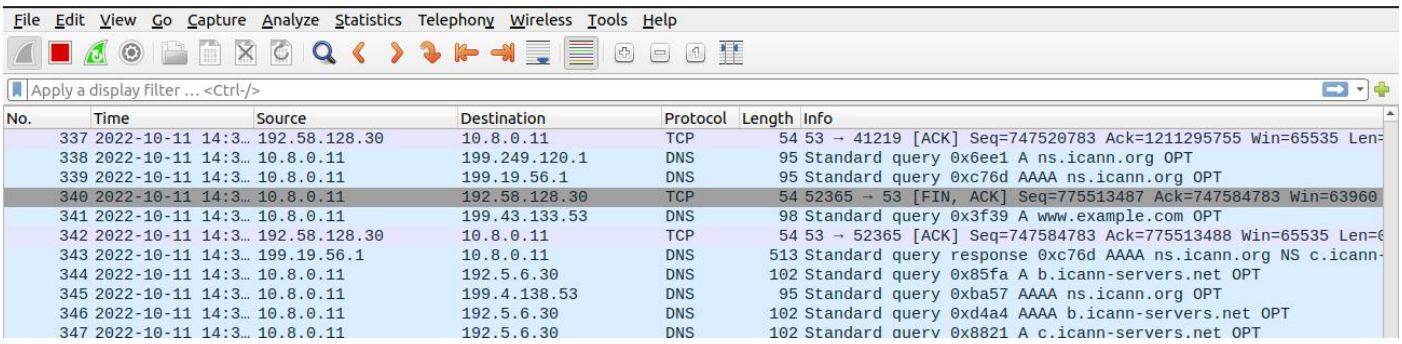
```
Attacker:PES1UG20CS825:Prem Sagar J S/volumes/Code
#python3 task4.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:0b
  src       = 02:42:0a:09:00:35
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 2636
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0x19af
     src       = 10.9.0.53
     dst       = 199.43.133.53
     \options   \
###[ UDP ]###
```

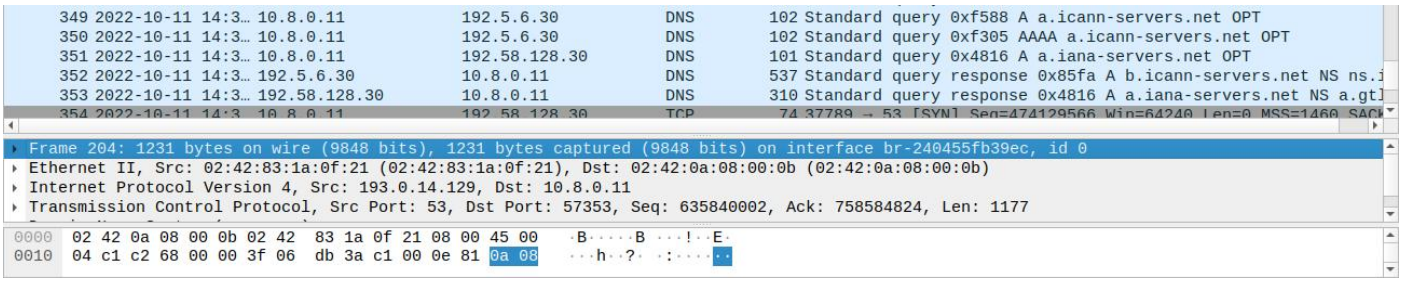
On the victim terminal run the command:
dig www.example.com

```
User:PES1UG20CS825:Prem Sagar J S/
#dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61087
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: ad5c2ad19b521167010000006345b7c3f8260d9d66cd354a (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 3452 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Oct 11 18:36:51 UTC 2022
```

the **spoofed packet captured on wireshark** and the cache of the local DNS server

Wireshark screenshot :





On the local DNS server's terminal run the commands:
rndc dumpdb -cache
cat /var/cache/bind/dump.db | grep example

```
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#rndc dumpdb -cache
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#cat /var/cache/bind/dump.db | grep example
example.com.            777446  NS      ns.attacker32.com.
www.example.com.        863848  A       1.1.1.1
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#
```

Task 5: Spoofing Records in the Additional Section

In DNS replies, there is a section called Additional Section, which is used to provide additional information. In practice, it is mainly used to provide IP addresses for some hostnames, especially for those appearing in the Authority section. In particular, when responding to the query for www.example.com, we add the following entries in the spoofed reply, in addition to the entries in the Answer section. The goal of this task is to spoof some entries in this section and see whether they will
be successfully cached by the target local DNS server.

```
;; AUTHORITY SECTION:
example.com.            259200  IN   NS   ns.attacker32.com.
example.com.            259200  IN   NS   ns.example.com.

;; ADDITIONAL SECTION:
ns.attacker32.com.      259200  IN   A    1.2.3.4   ①
ns.example.net.         259200  IN   A    5.6.7.8   ②
www.facebook.com.       259200  IN   A    3.4.5.6   ③
```

clearing the cache on the local DNS server first

On the local DNS server's terminal run the command:
rndc flush

```
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#rndc flush
```

On the attacker terminal run the command:
python3 task5.py

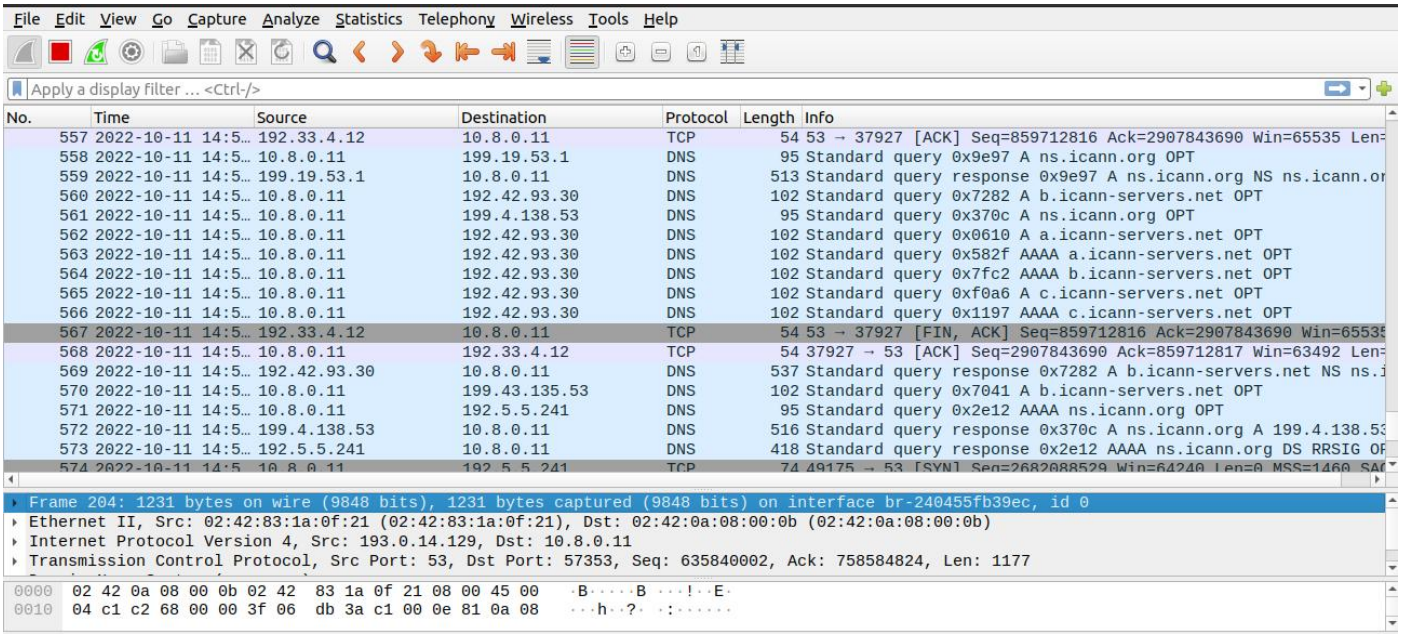
```
Attacker:PES1UG20CS825:Prem Sagar J S/volumes/Code
#python3 task5.py
.
Sent 1 packets.
.
Sent 1 packets.
```

The victim machine sends out a DNS query to the local DNS server using the dig command

On the victim terminal run the command:
dig www.example.com

```
User:PES1UG20CS825:Prem Sagar J S/
#dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1021
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

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

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

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.attacker32.com.
example.com.            259200  IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.attacker32.com.      259200  IN      A       1.2.3.4
```

Wireshark screenshot :



check the cache on the local DNS server and see whether the spoofed NS record is in the cache or not.

On the local DNS server's terminal run the commands:
rndc dumpdb -cache
cat /var/cache/bind/dump.db | grep example

```
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#rndc dumpdb -cache
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#cat /var/cache/bind/dump.db | grep example
example.com.            777595  NS      ns.example.com.
www.example.com.        863996  A       1.1.1.1
Local-DNS:PES1UG20CS825:Prem Sagar J S/
#
```