NOV 03, 2025

LAB 3

TIAN HUANG

Exercise 3:

Question 1:

CODE USED: dig www.amazon.com.au

RESULT:

In ANSWER SECTION< we found the IP address is 3.175.109.140. The DNS request is A type, searching for IPv4 address.

Question 2:

CODE USED: dig www.amazon.com.au

RESULT:

The canonical name is tp.04f01a85e-frontier.amazon.com.au

The reason of using canonical name is to flexible server switch and could make domain name management easier

Question 3:

```
z5641211@vx21: $ dig www.amazon.com.au

; <<>> DiG 9.18.41-1~deb12u1-Debian <<>> www.amazon.com.au
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1227
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 1ac2223ea007be5f010000006907c03185d3abf595da579a (good)
;; QUESTION SECTION:
;www.amazon.com.au.             IN      A

;; ANSWER SECTION:
www.amazon.com.au.      998     IN      CNAME   tp.04f01a85e-frontier.amazon.com.au.
tp.04f01a85e-frontier.amazon.com.au. 60 IN CNAME cf.04f01a85e-frontier.amazon.com.au.
cf.04f01a85e-frontier.amazon.com.au. 60 IN A    3.175.109.140

;; Query time: 12 msec
;; SERVER: 129.94.242.2#53(129.94.242.2) (UDP)
;; WHEN: Mon Nov 03 07:33:53 AEDT 2025
;; MSG SIZE  rcvd: 156
```

RESULT:

The EDNS part is for extended informing, for security communication and better data capacity

COOKIE is used to prevent spoofed request

Question 4:

CODE USED: cat /etc/resolv.conf

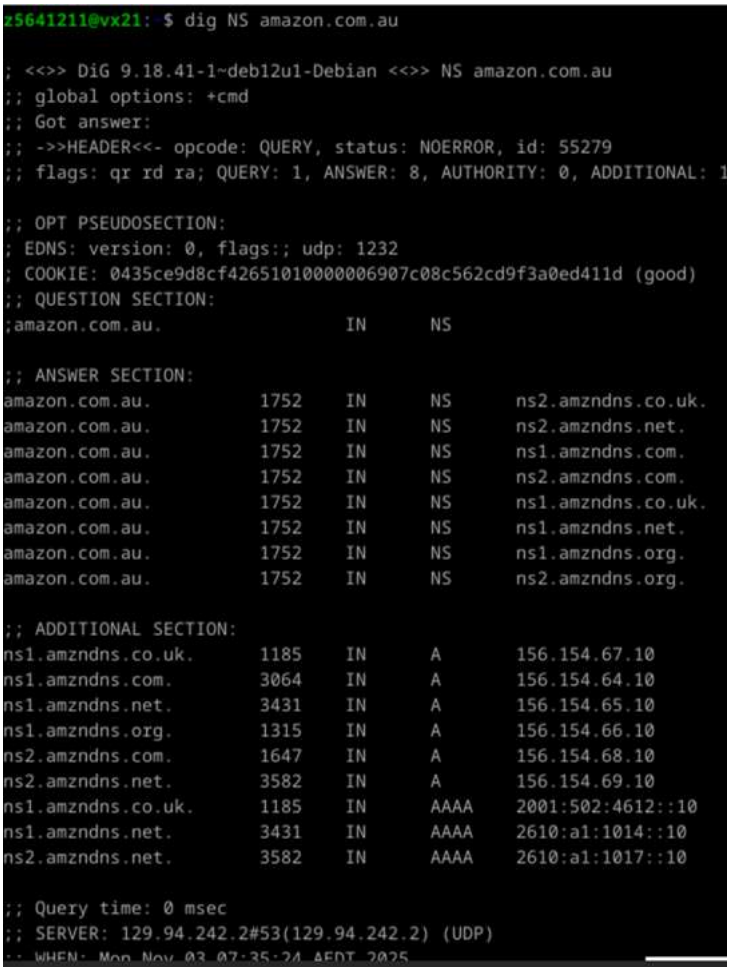
RESULT:

```
z5641211@vx21: $ cat /etc/resolv.conf
domain orchestra.cse.unsw.EDU.AU
search orchestra.cse.unsw.EDU.AU cse.unsw.edu.au.
nameserver 129.94.242.2
nameserver 129.94.242.45
nameserver 129.94.242.33
```

Question 5:

CODE USED: dig NS amazon.com.au

RESULT:

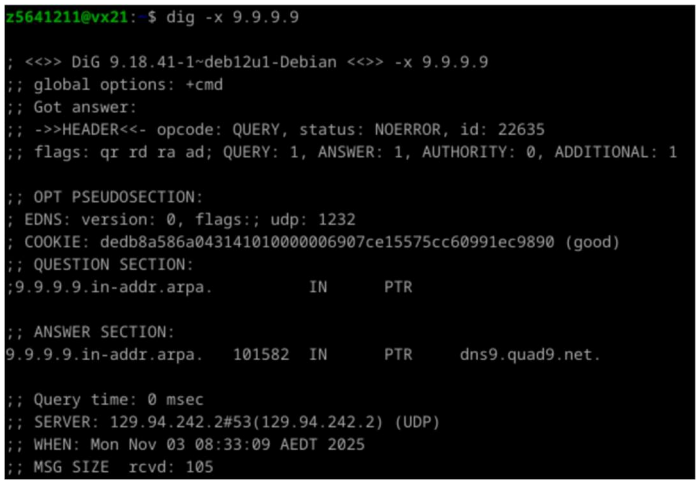


ANSWER:

ns1.amzdns.co.uk	156.154.67.10
ns2.amzdns.net	156.154.69.10
ns1.amzdns.org	156.154.66.10
ns2.amzdns.com	156.154.68.10

Question 6:

CODE USED: dig -x 9.9.9.9



ANSWER:

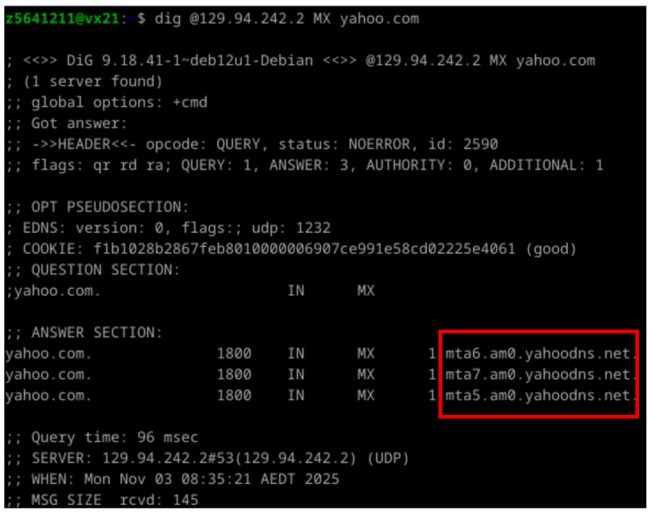
Domain: dns9.quad9.net

Request type: PTR

Question 7:

CODE USED: dig @129.94.242.2 MX yahoo.com

RESULT:



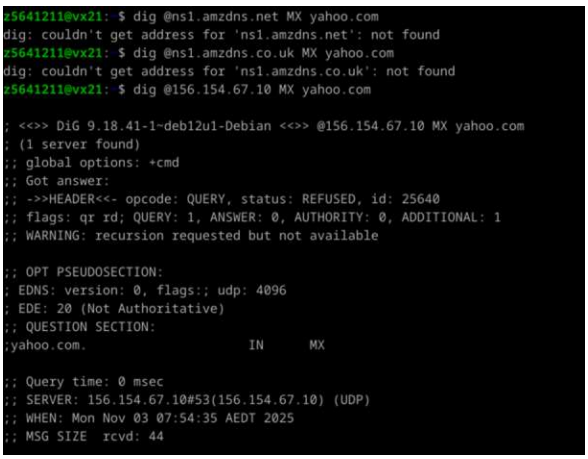
ANSWER:

For lags, there is no aa(authoritative answer), it comes from CSE DNS

Question 8:

CODE USED: dig @156.154.67.10 MX yahoo.com

RESULT:



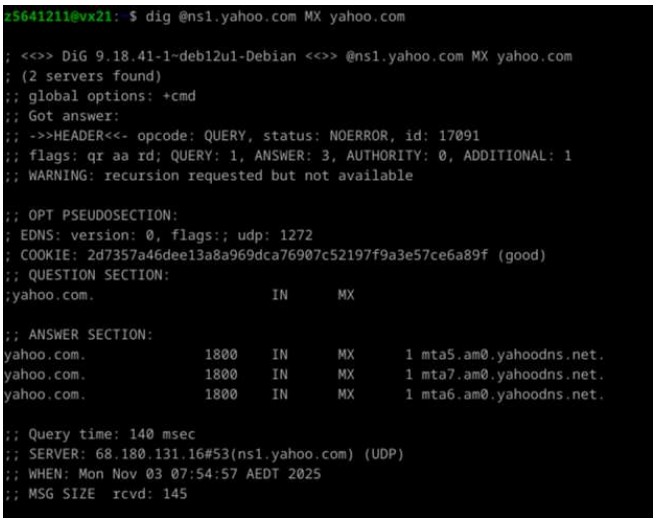
ANSWER:

I tried Amazon DNS server (ns1.amzdns.co.uk, 156.154.67.10) and got a REFUSED and Not Authoritative

Question 9:

CODE USED: dig @ns1.yahoo.com MX yahoo.com

RESULT:



ANSWER:

We got status NOERROR, the DNS query used is MX

Question 10:

STRATEGY:

www.lab.cse.unsw.edu.au

Go root first, then au, then edu, then unsw, then cse

CODE USED:

dig NS .

Search for root servers

dig @a.root-severs.net NS au.

Search for au servers

Got a.au address as 58.65.254.1

dig @58.65.254.1 NS edu.au

Search for edu servers

At this point, we found that au is not a sub domain under au, it is at the same level with au, so we can directly go from au to unse.edu.au, no need to edu.au

dig @58.65.254.1 NS unsw.edu.au

Search for unsw servers

Got ns1-ext unsw.edu.au at 54.79.80.189

dig @54.79.80.189 NS cse.unsw.edu.au

search for cse severs

Got Beethoven at 129.94.242.2

dig @129.94.242.2 A lyre00.cse.unsw.edu.au

Got lyre00.cse.unsw.edu.au at 129.94.210.20

ANSWER:

There are 5 layers required:

. -> au -> unsw.edu.au -> cse.unsw.edu.au -> target

And 6 has been achieved, the one more is edu.au(not required)

The query type include NS and A

Question 11:

ANSWER:

Yes, one machine can have more than one domain names and more than on IP address.

Sometimes there are Virtual Hosting or CNAME using one physical server hosting multiple websites, which come with different domain name for each website. Just like apple.com.us and apple.com might go to same address

Sometimes it could have more than one network interfaces or assigned IP address. Or have one IPv4 and one IPv6 at the same time, like Amazon DNS sever in formal questions

Exercise 4:

See code

As a backup if the file is missing:

https://github.com/tanglehunter00/UNSW-COMP-9331.git

```
import socket, os, urllib.parse

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("", 10000))
s.listen(1)
print("http://127.0.0.1:10000/index.html")
print("http://127.0.0.1:10000/myimage.jpg")
print("http://127.0.0.1:10000/myimage2.jpg")
print("http://127.0.0.1:10000/bio.html")

while True:
    c, a = s.accept()
    while True:
        data = b""
        while b"\r\n\r\n" not in data:
            chunk = c.recv(4096)
            if not chunk:
                break
            data += chunk
        if not data:
            break
        req_line = data.split(b"\r\n", 1)[0].decode()
        m, t, v = req_line.split(" ")
        path = urllib.parse.urlsplit(t).path or "/"
        if path == "/":
            path = "/index.html"
        target = os.path.join(os.getcwd(), path.lstrip("/"))
        ext = os.path.splitext(target.lower())[1]
        if ext not in (".html", ".jpg") or not os.path.isfile(target):
            body = b"<h1>404 Not Found</h1>"
            resp = (
                "HTTP/1.1 404 Not Found\r\n"
                "Content-Type: text/html\r\n"
                f"Content-Length: {len(body)}\r\n\r\n"
            ).encode() + body
            c.sendall(resp)
            continue
```

```
            with open(target, "rb") as f:
                body = f.read()
            ctype = "text/html" if ext == ".html" else "image/jpeg"
            resp = (
                "HTTP/1.1 200 OK\r\n"
                f"Content-Type: {ctype}\r\n"
                f"Content-Length: {len(body)}\r\n\r\n"
            ).encode() + body
            c.sendall(resp)
```