CS356-Lab02:2-Branden-Codd-940428984

1. DNS #1 (dig)

- DNS reconnaissance

- Use dig to query the local DNS server for the A record of www.sou.edu using TCP. Then, use dig to do the same for the MX record of sou.edu.

```
branden@branden-VirtualBox:~$ dig www.sou.edu A tcp

; <<>> DiG 9.16.1-Ubuntu <<>> www.sou.edu A tcp
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56937
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.sou.edu.                   IN      A

;; ANSWER SECTION:
www.sou.edu.            1076    IN      CNAME   sou.edu.
sou.edu.                41      IN      A       198.199.109.37

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Wed Jan 20 20:54:00 PST 2021
;; MSG SIZE  rcvd: 70

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 3326
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;tcp.                           IN      A

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Wed Jan 20 20:54:00 PST 2021
;; MSG SIZE  rcvd: 32
```

```
branden@branden-VirtualBox:~$ dig sou.edu MX

; <<>> DiG 9.16.1-Ubuntu <<>> sou.edu MX
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41748
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;sou.edu.                       IN      MX

;; ANSWER SECTION:
sou.edu.                1611    IN      MX      30 ALT4.ASPMX.L.GOOGLE.COM.
sou.edu.                1611    IN      MX      10 ASPMX.L.GOOGLE.COM.
sou.edu.                1611    IN      MX      30 ALT3.ASPMX.L.GOOGLE.COM.
sou.edu.                1611    IN      MX      20 ALT1.ASPMX.L.GOOGLE.COM.
sou.edu.                1611    IN      MX      20 ALT2.ASPMX.L.GOOGLE.COM.

;; Query time: 12 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Wed Jan 20 20:46:04 PST 2021
;; MSG SIZE  rcvd: 154

branden@branden-VirtualBox:~$
```

Running commands

- dig www.sou.edu A tcp
- Dig sou.edu MX

- When a web request hits port 80 of 131.252.220.66, how does the server know which site to serve from? (i.e. what protocol header)
    - Port 80 is a known port for http, this in conjunction with DNS tells the server which site to serve from.

- DNS iterative lookups
    - On your VM simulate the operation of a local DNS server. Choose a DNS name containing at least 4 parts (e.g) www.inside.sou.edu, console.cloud.google.com, www.unsw.edu.au, www.amazon.co.uk)
    - Dig to get ip

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ dig f.root-servers.net

; <<>> DiG 9.16.1-Ubuntu <<>> f.root-servers.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58714
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;f.root-servers.net.            IN      A

;; ANSWER SECTION:
f.root-servers.net.     3585868 IN      A       192.5.5.241

;; Query time: 11 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Thu Jan 21 11:13:36 PST 2021
;; MSG SIZE  rcvd: 63
```

- Then: dig @192.5.5.241 console.cloud.google.com +tcp +norecurse

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ dig @192.5.5.241 console.cloud.google.com +tcp +nore
curse

; <<>> DiG 9.16.1-Ubuntu <<>> @192.5.5.241 console.cloud.google.com +tcp +norecurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19912
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 27

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65535
;; QUESTION SECTION:
;console.cloud.google.com.      IN      A

;; AUTHORITY SECTION:
com.                    172800  IN      NS      l.gtld-servers.net.
com.                    172800  IN      NS      b.gtld-servers.net.
com.                    172800  IN      NS      c.gtld-servers.net.
com.                    172800  IN      NS      d.gtld-servers.net.
com.                    172800  IN      NS      e.gtld-servers.net.
com.                    172800  IN      NS      f.gtld-servers.net.
com.                    172800  IN      NS      g.gtld-servers.net.
com.                    172800  IN      NS      a.gtld-servers.net.
com.                    172800  IN      NS      h.gtld-servers.net.
com.                    172800  IN      NS      i.gtld-servers.net.
com.                    172800  IN      NS      j.gtld-servers.net.
com.                    172800  IN      NS      k.gtld-servers.net.
com.                    172800  IN      NS      m.gtld-servers.net.

;; ADDITIONAL SECTION:
l.gtld-servers.net.     172800  IN      A       192.41.162.30
l.gtld-servers.net.     172800  IN      AAAA    2001:500:d937::30
b.gtld-servers.net.     172800  IN      A       192.33.14.30
b.gtld-servers.net.     172800  IN      AAAA    2001:503:231d::2:30
c.gtld-servers.net.     172800  IN      A       192.26.92.30
c.gtld-servers.net.     172800  IN      AAAA    2001:503:83eb::30
d.gtld-servers.net.     172800  IN      A       192.31.80.30
```

- Now use this as our new ip to dig on
- dig @192.41.162.30 console.cloud.google.com +tcp +norecurse

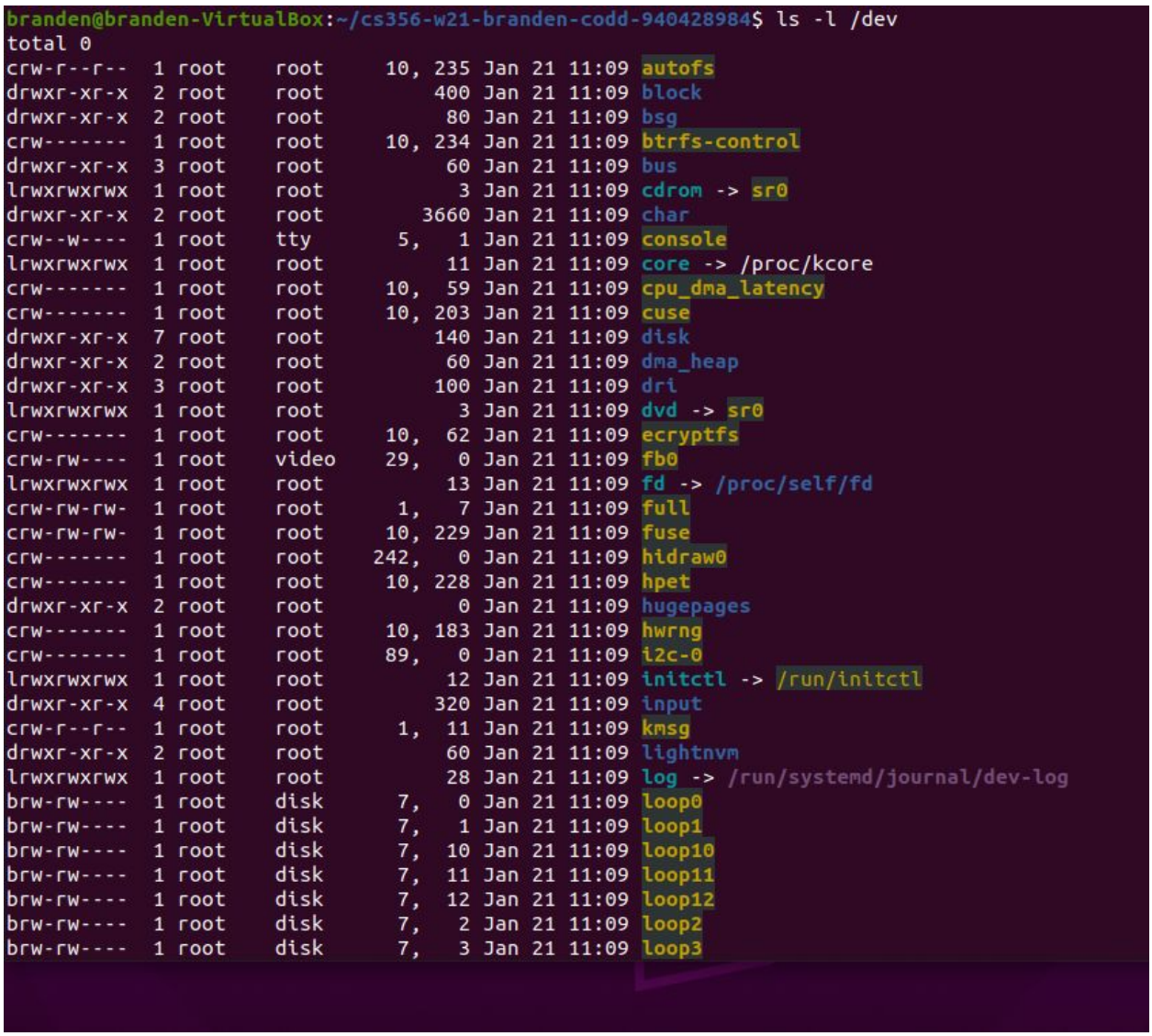
```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ dig @192.41.162.30 console.cloud.google.com +tcp +no
recurse

; <<>> DiG 9.16.1-Ubuntu <<>> @192.41.162.30 console.cloud.google.com +tcp +norecurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55767
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;console.cloud.google.com.      IN      A

;; AUTHORITY SECTION:
google.com.             172800  IN      NS      ns2.google.com.
google.com.             172800  IN      NS      ns1.google.com.
google.com.             172800  IN      NS      ns3.google.com.
google.com.             172800  IN      NS      ns4.google.com.

;; ADDITIONAL SECTION:
ns2.google.com.         172800  IN      AAAA    2001:4860:4802:34::a
ns2.google.com.         172800  IN      A       216.239.34.10
ns1.google.com.         172800  IN      AAAA    2001:4860:4802:32::a
ns1.google.com.         172800  IN      A       216.239.32.10
ns3.google.com.         172800  IN      AAAA    2001:4860:4802:36::a
ns3.google.com.         172800  IN      A       216.239.36.10
ns4.google.com.         172800  IN      AAAA    2001:4860:4802:38::a
ns4.google.com.         172800  IN      A       216.239.38.10

;; Query time: 75 msec
;; SERVER: 192.41.162.30#53(192.41.162.30)
;; WHEN: Thu Jan 21 11:26:01 PST 2021
;; MSG SIZE  rcvd: 301
```

- dig @216.239.34.10 console.cloud.google.com +tcp +norecurse

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ dig @216.239.34.10 console.cloud.google.com +tcp +no
recurse

; <<>> DiG 9.16.1-Ubuntu <<>> @216.239.34.10 console.cloud.google.com +tcp +norecurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59651
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;console.cloud.google.com.      IN      A

;; ANSWER SECTION:
console.cloud.google.com. 300   IN      CNAME   www3.l.google.com.
www3.l.google.com.      300     IN      A       172.217.14.206

;; Query time: 43 msec
;; SERVER: 216.239.34.10#53(216.239.34.10)
;; WHEN: Thu Jan 21 11:29:41 PST 2021
;; MSG SIZE  rcvd: 90
```

2. Reverse DNS lookups

- First, perform the following commands and examine the output to understand what `egrep`, `awk`, and the pipes (`|`) are doing:



- Large output that spans past terminal screen

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ ls -l /dev | egrep vcs
crw-rw----  1 root    tty       7,   0 Jan 21 11:09 vcs
crw-rw----  1 root    tty       7,   1 Jan 21 11:09 vcs1
crw-rw----  1 root    tty       7,   2 Jan 21 11:09 vcs2
crw-rw----  1 root    tty       7,   3 Jan 21 11:09 vcs3
crw-rw----  1 root    tty       7,   4 Jan 21 11:09 vcs4
crw-rw----  1 root    tty       7,   5 Jan 21 11:09 vcs5
crw-rw----  1 root    tty       7,   6 Jan 21 11:09 vcs6
crw-rw----  1 root    tty       7, 128 Jan 21 11:09 vcsa
crw-rw----  1 root    tty       7, 129 Jan 21 11:09 vcsa1
crw-rw----  1 root    tty       7, 130 Jan 21 11:09 vcsa2
crw-rw----  1 root    tty       7, 131 Jan 21 11:09 vcsa3
crw-rw----  1 root    tty       7, 132 Jan 21 11:09 vcsa4
crw-rw----  1 root    tty       7, 133 Jan 21 11:09 vcsa5
crw-rw----  1 root    tty       7, 134 Jan 21 11:09 vcsa6
crw-rw----  1 root    tty       7,  64 Jan 21 11:09 vcsu
crw-rw----  1 root    tty       7,  65 Jan 21 11:09 vcsu1
crw-rw----  1 root    tty       7,  66 Jan 21 11:09 vcsu2
crw-rw----  1 root    tty       7,  67 Jan 21 11:09 vcsu3
crw-rw----  1 root    tty       7,  68 Jan 21 11:09 vcsu4
crw-rw----  1 root    tty       7,  69 Jan 21 11:09 vcsu5
crw-rw----  1 root    tty       7,  70 Jan 21 11:09 vcsu6
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$
```

- Now only get results that match vcs

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ ls -l /dev | egrep vcs | awk '{print $10}'
vcs
vcs1
vcs2
vcs3
vcs4
vcs5
vcs6
vcsa
vcsa1
vcsa2
vcsa3
vcsa4
vcsa5
vcsa6
vcsu
vcsu1
vcsu2
vcsu3
vcsu4
vcsu5
vcsu6
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$
```

- Print only the result we want

- You can save the standard output of a command and use it in subsequent command-line arguments. One way is to save it to an environment variable using back-ticks `command` or `$(command)`

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ X=`ls -l /dev | egrep loop | awk '{print $10}'`
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ echo $X
loop0 loop1 loop10 loop11 loop12 loop2 loop3 loop4 loop5 loop6 loop7 loop8 loop9 loop-control
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ X=$(ls -l /dev | egrep loop | awk '{print $10}')
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ echo $X
loop0 loop1 loop10 loop11 loop12 loop2 loop3 loop4 loop5 loop6 loop7 loop8 loop9 loop-control
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ for i in `echo $X`
> do
>   file /dev/$i
> done
/dev/loop0: block special (7/0)
/dev/loop1: block special (7/1)
/dev/loop10: block special (7/10)
/dev/loop11: block special (7/11)
/dev/loop12: block special (7/12)
/dev/loop2: block special (7/2)
/dev/loop3: block special (7/3)
/dev/loop4: block special (7/4)
/dev/loop5: block special (7/5)
/dev/loop6: block special (7/6)
/dev/loop7: block special (7/7)
/dev/loop8: block special (7/8)
/dev/loop9: block special (7/9)
/dev/loop-control: character special (10/237)
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$
```

Examine the man page for dig to find the flag for doing simplified reverse lookups on IPv4 addresses. Given the flag and the shell tutorial above, perform the following and include the results in your lab notebook:

- The flag for simplified reverse lookup is -x, -4 for only ipv4 addresses

- Use a single command line with commands dig, egrep, and awk, to list all IPv4 addresses that espn.go.com points to.
    - dig espn.go.com -x -4 | egrep 99 | awk '{print $5}'

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ dig espn.go.com  | egrep 99 | awk '{print $5}'
99.86.35.90
99.86.35.95
99.86.35.96
99.86.35.52
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$
```

- Take that list and create a single for loop in the shell that iterates over the list and performs a reverse lookup of each IP address to find each address's associated DNS name. As with the previous step, pipe the output of the for loop to egrep and awk so that the output consists only of the DNS names.
    - Set X=`dig espn.go.com -x -4 | egrep 99 | awk '{print $5}'`
    - Then for i in `echo $X`; do dig -x $i; done | egrep server | awk '{print $5}'

3. Host Enumeration

- On your VM, practice some more shell (bash) preliminaries.
- Ranges in the shell can be specified via { } notation. Perform the following two commands to see how it works.

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ echo {0..2}{0..9}
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$ for i in {1..20}
> do
>   echo $i
> done
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984$
```

- Using a for loop, perform a reverse DNS lookup for each IP address on the 131.252.220.0/24 subnet. Note that some addresses on the subnet do not have names bound to them and will not return a record.
- Take the output of the loop and pipe it to egrep and awk to list just the names of the hosts, then redirect the final output to a file called 220hosts.txt output using the > character to perform output redirection to a file.
    - for i in {0..255}; do dig -x 131.252.220.$i; done | egrep -A1 "ANSWER SECTION" | awk '{print $5}' | egrep -v '^$' >> 220hosts.txt

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$ for i in {0..255}; do dig -x 131.252.22
0.$i; done | egrep -A1 "ANSWER SECTION" | awk '{print $5}' | egrep -v '^$' >> 220hosts.txt
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$ wc -l 220hosts.txt
247 220hosts.txt
```

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$ cat 220hosts.txt
colt45.cs.pdx.edu.
kingcobra.cs.pdx.edu.
mickeys.cs.pdx.edu.
magnum.cs.pdx.edu.
phatboy.cs.pdx.edu.
schlitz.cs.pdx.edu.
boar.cs.pdx.edu.
dog.cs.pdx.edu.
dragon.cs.pdx.edu.
horse.cs.pdx.edu.
monkey.cs.pdx.edu.
ox.cs.pdx.edu.
rabbit.cs.pdx.edu.
rat.cs.pdx.edu.
rooster.cs.pdx.edu.
sheep.cs.pdx.edu.
snake.cs.pdx.edu.
tiger.cs.pdx.edu.
assault.cs.pdx.edu.
aztec.cs.pdx.edu.
backalley.cs.pdx.edu.
cbble.cs.pdx.edu.
dust.cs.pdx.edu.
estate.cs.pdx.edu.
havana.cs.pdx.edu.
inferno.cs.pdx.edu.
italy.cs.pdx.edu.
militia.cs.pdx.edu.
nuke.cs.pdx.edu.
office.cs.pdx.edu.
oilrig.cs.pdx.edu.
piranesi.cs.pdx.edu.
prodigy.cs.pdx.edu.
siege.cs.pdx.edu.
survivor.cs.pdx.edu.
torn.cs.pdx.edu.
train.cs.pdx.edu.
vertigo.cs.pdx.edu.
ak47.cs.pdx.edu.
aug.cs.pdx.edu.
```

- Some of the output

- Within the range of hosts is a set of car manufacturer names. Using the head and tail commands, craft a command in the format below that returns their names.
    - The car manufacturer names are located from 158 - 185

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$ cat 220hosts.txt | head -185 | tail -28
audi.cs.pdx.edu.
bentley.cs.pdx.edu.
bmw.cs.pdx.edu.
cadillac.cs.pdx.edu.
ferrari.cs.pdx.edu.
fiat.cs.pdx.edu.
ford.cs.pdx.edu.
honda.cs.pdx.edu.
hummer.cs.pdx.edu.
jaguar.cs.pdx.edu.
jeep.cs.pdx.edu.
lamborghini.cs.pdx.edu.
landrover.cs.pdx.edu.
lexus.cs.pdx.edu.
lotus.cs.pdx.edu.
maserati.cs.pdx.edu.
mazda.cs.pdx.edu.
mclaren.cs.pdx.edu.
mercedes.cs.pdx.edu.
nissan.cs.pdx.edu.
panoz.cs.pdx.edu.
porsche.cs.pdx.edu.
subaru.cs.pdx.edu.
toyota.cs.pdx.edu.
tvr.cs.pdx.edu.
ultima.cs.pdx.edu.
volvo.cs.pdx.edu.
vw.cs.pdx.edu.
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$
```

4. DNS #2 (Geographic DNS)

- What geographic locations do ipinfo.io and DB-IP return?
    - 131.252.208.53

Geolocation data from ipinfo.io (Product: API, real-time)

IP Address	Country	Region	City
131.252.208.53	United States	Oregon	Portland
ISP	**Organization**	**Latitude**	**Longitude**
Portland State University	Portland State University (pdx.edu)	45.5234	-122.6762

Geolocation data from DB-IP (Product: Full, 2021-1-1)

IP Address	Country	Region	City
131.252.208.53	United States	Oregon	Portland
ISP	**Organization**	**Latitude**	**Longitude**
Portland State University	Portland State University	45.5231	-122.676

- 198.82.247.66

Geolocation data from ipinfo.io (Product: API, real-time)

IP Address	Country	Region	City
198.82.247.66	United States	Virginia	Blacksburg
ISP	**Organization**	**Latitude**	**Longitude**
Virginia Polytechnic Institute and State Univ.	Virginia Polytechnic Institute and State Univ. (vt.edu)	37.2296	-80.4139

Geolocation data from DB-IP (Product: Full, 2021-1-1)

IP Address	Country	Region	City
198.82.247.66	United States	Virginia	Blacksburg (Farmview - Ramble)
ISP	**Organization**	**Latitude**	**Longitude**
Virginia Polytechnic Institute and State Univ.	Virginia Polytechnic Institute and State Univ.	37.2037	-80.4143

Then, using dig, resolve www.google.com from each of the DNS servers (dig @<DNS_server_IP> www.google.com).

- Record each result for your lab notebook.
    - Issue with first dig, refused
    - Second dig no issues

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$ dig @131.252.208.53 www.google.com

; <<>> DiG 9.16.1-Ubuntu <<>> @131.252.208.53 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 47016
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

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

;; Query time: 24 msec
;; SERVER: 131.252.208.53#53(131.252.208.53)
;; WHEN: Thu Jan 21 16:43:16 PST 2021
;; MSG SIZE  rcvd: 43

branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$ dig @198.82.247.66 www.google.com

; <<>> DiG 9.16.1-Ubuntu <<>> @198.82.247.66 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18810
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: c560301b8bba3b6e1c4d4e6b600a1fd39c771e1d826ab6f4 (good)
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         153     IN      A       172.253.62.147
www.google.com.         153     IN      A       172.253.62.106
www.google.com.         153     IN      A       172.253.62.103
www.google.com.         153     IN      A       172.253.62.104
www.google.com.         153     IN      A       172.253.62.99
www.google.com.         153     IN      A       172.253.62.105

;; Query time: 92 msec
;; SERVER: 198.82.247.66#53(198.82.247.66)
;; WHEN: Thu Jan 21 16:44:03 PST 2021
;; MSG SIZE  rcvd: 167

branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$
```

Go back to https://www.iplocation.net/ and lookup the geographical location of each IP address returned. What geographic locations do ipinfo.io and DB-IP return?

- 172.253.62.147

Geolocation data from ipinfo.io (Product: API, real-time)

IP Address	Country	Region	City
172.253.62.147	United States	California	Mountain View
ISP	**Organization**	**Latitude**	**Longitude**
Google LLC	Google LLC (google.com)	38.0088	-122.1175

Geolocation data from DB-IP (Product: Full, 2021-1-1)

IP Address	Country	Region	City
172.253.62.147	United States	California	Mountain View
ISP	**Organization**	**Latitude**	**Longitude**
Google LLC	Google LLC	37.422	-122.084

- 172.253.62.106

Geolocation data from ipinfo.io (Product: API, real-time)

IP Address	Country	Region	City
172.253.62.106	United States	California	Mountain View
ISP	**Organization**	**Latitude**	**Longitude**
Google LLC	Google LLC (google.com)	38.0088	-122.1175

Geolocation data from DB-IP (Product: Full, 2021-1-1)

IP Address	Country	Region	City
172.253.62.106	United States	California	Mountain View
ISP	**Organization**	**Latitude**	**Longitude**
Google LLC	Google LLC	37.422	-122.084

- 172.253.62.103

Geolocation data from ipinfo.io (Product: API, real-time)

IP Address	Country	Region	City
172.253.62.103	United States	California	Mountain View
ISP	**Organization**	**Latitude**	**Longitude**
Google LLC	Google LLC (google.com)	38.0088	-122.1175

Geolocation data from DB-IP (Product: Full, 2021-1-1)

IP Address	Country	Region	City
172.253.62.103	United States	California	Mountain View
ISP	**Organization**	**Latitude**	**Longitude**
Google LLC	Google LLC	37.422	-122.084

- 172.253.62.104

Geolocation data from ipinfo.io (Product: API, real-time)

IP Address	Country	Region	City
172.253.62.104	United States	California	Mountain View
ISP	**Organization**	**Latitude**	**Longitude**
Google LLC	Google LLC (google.com)	38.0088	-122.1175

Geolocation data from DB-IP (Product: Full, 2021-1-1)

IP Address	Country	Region	City
172.253.62.104	United States	California	Mountain View
ISP	**Organization**	**Latitude**	**Longitude**
Google LLC	Google LLC	37.422	-122.084

- 172.253.62.99

Geolocation data from ipinfo.io (Product: API, real-time)

IP Address	Country	Region	City
172.253.62.99	United States	California	Mountain View
ISP	**Organization**	**Latitude**	**Longitude**
Google LLC	Google LLC (google.com)	38.0088	-122.1175

Geolocation data from DB-IP (Product: Full, 2021-1-1)

IP Address	Country	Region	City
172.253.62.99	United States	California	Mountain View
ISP	**Organization**	**Latitude**	**Longitude**
Google LLC	Google LLC	37.422	-122.084

- 172.253.62.105

Geolocation data from ipinfo.io (Product: API, real-time)

IP Address	Country	Region	City
172.253.62.105	United States	California	Mountain View
ISP	**Organization**	**Latitude**	**Longitude**
Google LLC	Google LLC (google.com)	38.0088	-122.1175

Geolocation data from DB-IP (Product: Full, 2021-1-1)

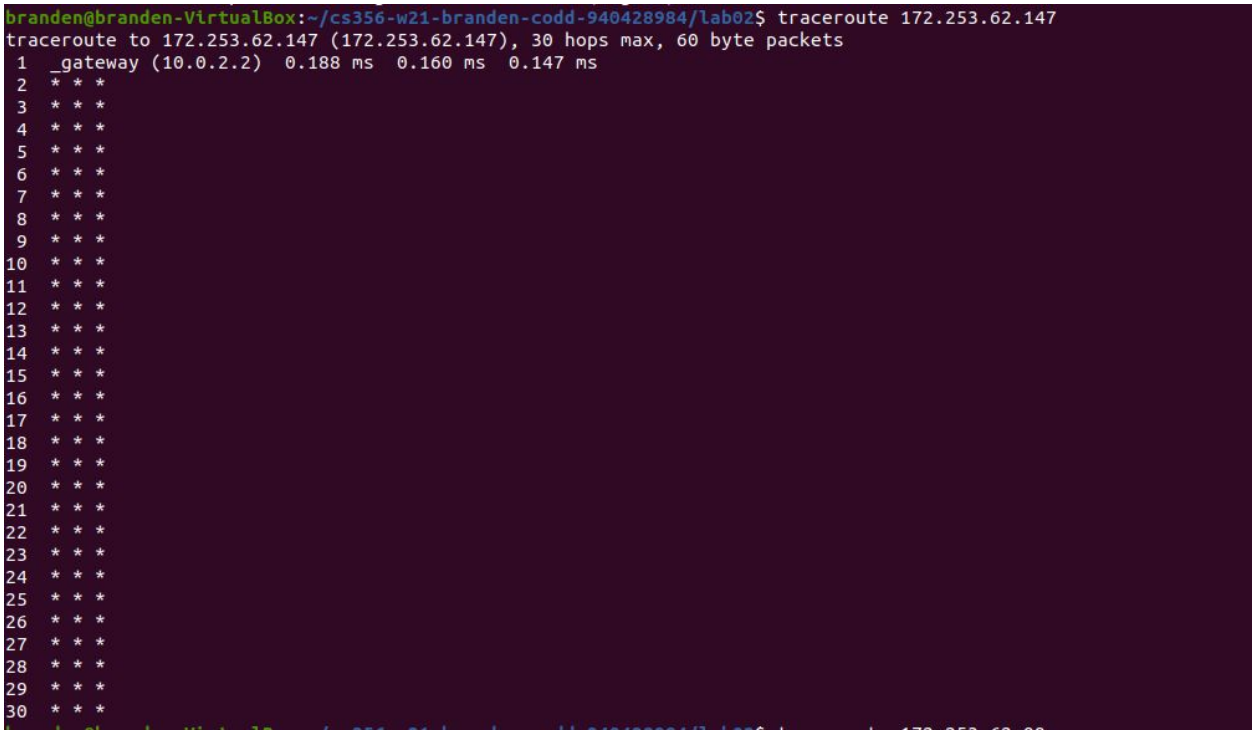
IP Address	Country	Region	City
172.253.62.105	United States	California	Mountain View
ISP	**Organization**	**Latitude**	**Longitude**
Google LLC	Google LLC	37.422	-122.084

- All had the same location

- What is the geographic distance between each pair of DNS server and web server?
    - Portland Oregon to Mountain view California is 667.2 miles
    - Blacksburg Virginia to Mountain view California is 2685.5 miles

Perform a traceroute to all 4 IP addresses from your network.

- Do the routes reveal any information on the accuracy of the geographic locations given? (Answer might be no)
    - The lab suggests 4 ips, however I had 6. Also when performing traceroute on any of these 6 ips gives me no information. Possibly another issue here.



5. Network Recap Lab #3

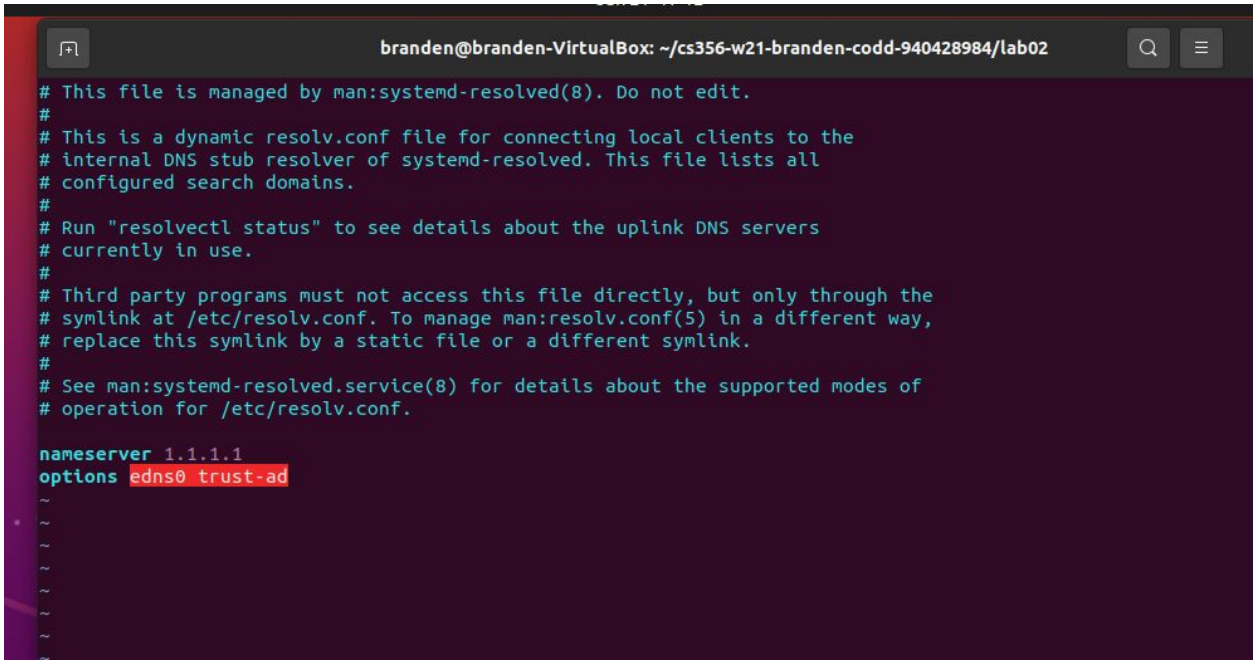
- Use ifconfig to find the IP address of the VM and the name of the local virtual ethernet interface.

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::b8ee:720f:dfbc:b26a  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:2b:d1:39  txqueuelen 1000  (Ethernet)
        RX packets 6440  bytes 4293682 (4.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4023  bytes 303055 (303.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 10868  bytes 789783 (789.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10868  bytes 789783 (789.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$
```

- Use `netstat` to find the IP address of the default router

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
link-local      0.0.0.0         255.255.0.0     U         0 0          0 enp0s3
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$
```

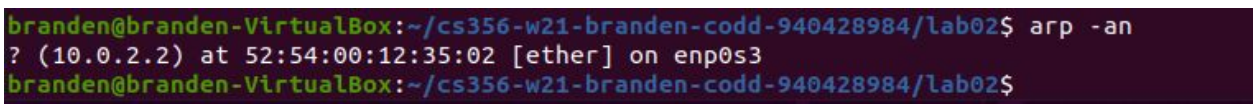
- Temporarily change the default DNS server by performing `sudo vim /etc/resolv.conf` and changing the IP address of the nameserver to `1.1.1.1`. Note that this will be overwritten upon next DHCP renew



- Perform a reverse DNS lookup on the DNS server to find its name

Dump ARP table

- To begin with, we will use the shell to delete all entries in the ARP table for the VM. Examine the output of the command below to see all of the entries in the table and their numeric IP addresses.



- Use a single command-line to create a file that contains each IP address that appears in the machine's ARP table and places the results in a file called arp_entries.
    - arp -an | awk -F '[()]' '{print $2}' > arp_entries

```
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$ arp -an | awk -F '[()]' '{print $2}' > arp_entrie
s
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$ cat arp_entries
10.0.2.2
branden@branden-VirtualBox:~/cs356-w21-branden-codd-940428984/lab02$
```

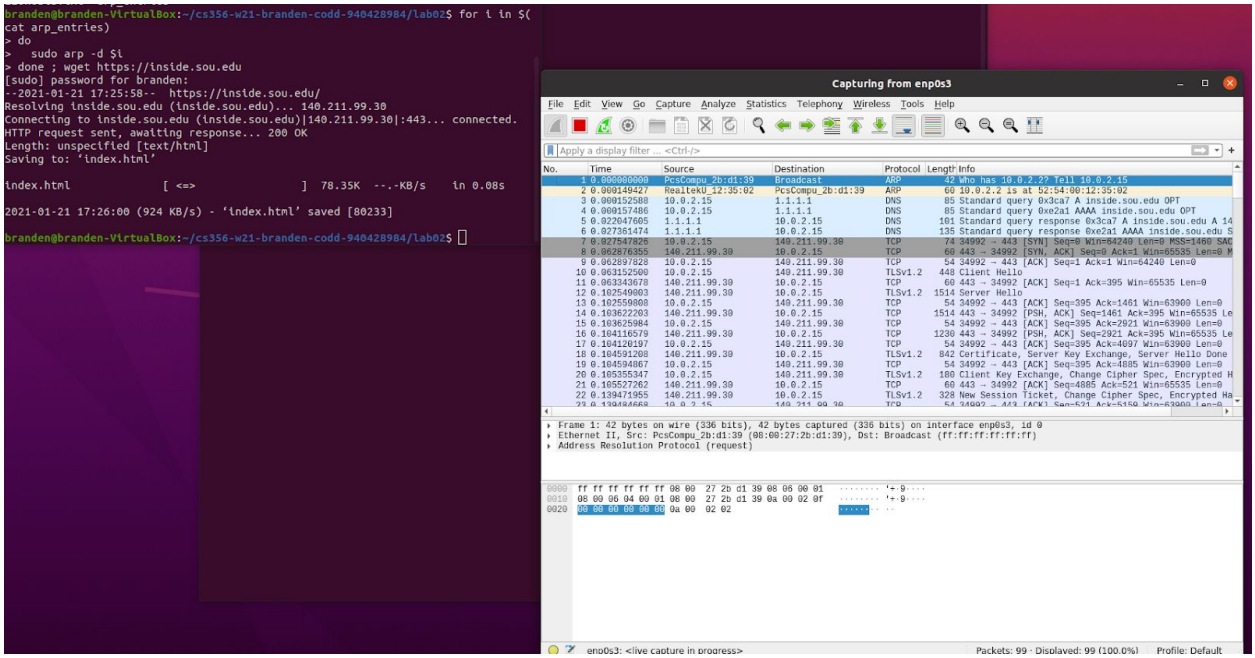
6. Collect and analyze the network trace of a connection

- Clear ARP table and retrieve site

Analyze trace

Stop the packet capture and inspect it. Ensure the ARP packets and DNS request and response packets for the request show up in the trace. If not, you are using a cached lookup.

- Take a screenshot of the trace within Wireshark and include an annotation of the packets in the trace to explain the purpose of each of the packets being exchanged.



- Little confused here on what is being asked. Slack says to answer for a ping but we only captured packets through wireshark
- 1 and 2 are an ARP request and an ARP response
- 3-6 are DNS queries to https://inside.sou.edu and then a response
- After that we have a TCP 3 way handshake
- Then a bunch of TCP and TLSv1.2 for https connection and encryption key exchange.

- How many DNS requests are made?
    - There are 2 DNS requests made
- How many TCP connections does the browser initiate simultaneously to the site?
    - There are 2
- How many HTTP GET requests are there for embedded objects?
    - Need to redo wireshark capture. Per original instructions we went through https://inside.sou.edu which is encrypted.
    - After re running with http://inside.sou.edu we see one get, but it's just attempting to go to https://inside.sou.edu.