

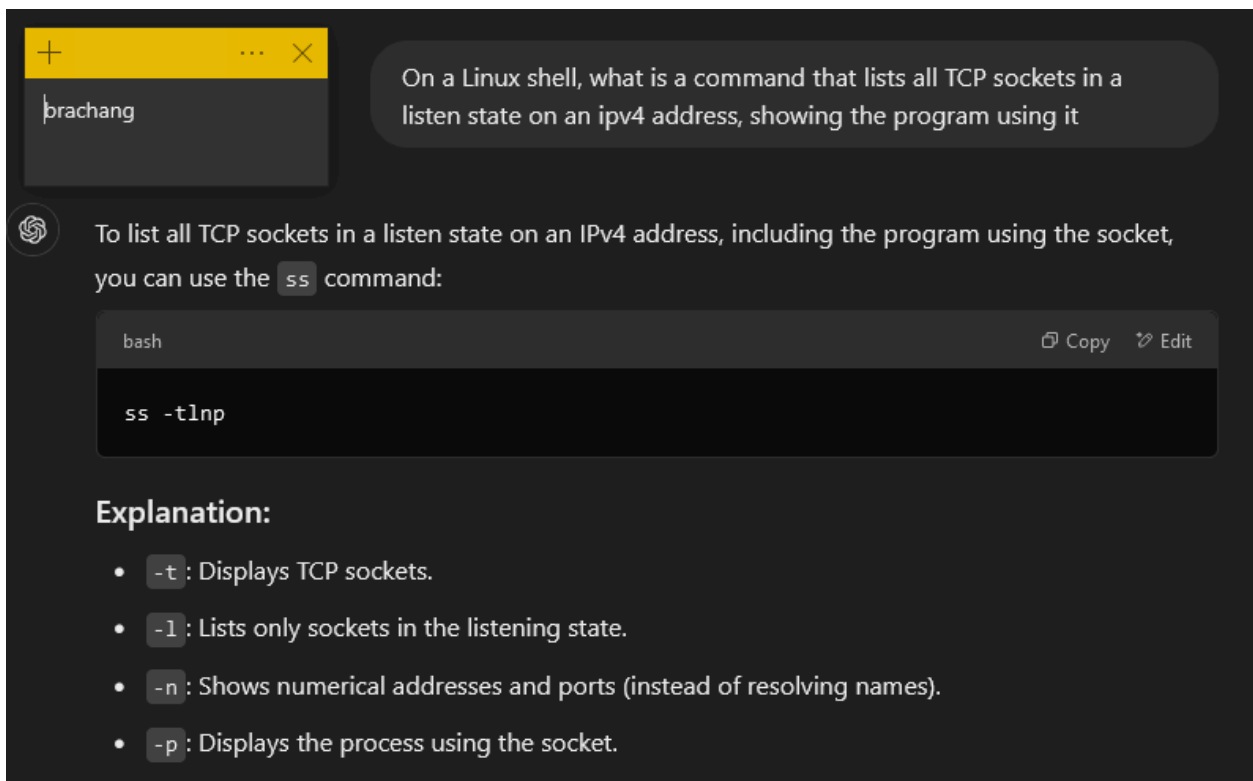
Bradley Chang

<b>2.1: TCP, HTTP</b>	<b>2</b>
1. TCP #1: Sockets	2
2. -	3
3. TCP #2: Throughput	5
4. -	5
<b>5. HTTP #3: Requests</b>	<b>5</b>
<b>02.2: DNS, Recap</b>	<b>9</b>
1. DNS reconnaissance #1 (dig)	9
2. Iterative DNS lookups	12
3. Reverse DNS lookups	13
4. Host enumeration	15
5. Geographic DNS #2	15
6. Wireshark Lab #3	18
7. -	18
8. Network Recap Lab #4	19

## 2.1: TCP, HTTP

### 1. TCP #1: Sockets

Take a screenshot of the prompt and the command that ChatGPT generates



The screenshot shows a ChatGPT window with a yellow title bar. The prompt is: "On a Linux shell, what is a command that lists all TCP sockets in a listen state on an ipv4 address, showing the program using it". The response is: "To list all TCP sockets in a listen state on an IPv4 address, including the program using the socket, you can use the `ss` command:". Below the response is a code block containing the command `ss -tlnp`. Below the code block is an "Explanation:" section with a bulleted list of options: `-t` (Displays TCP sockets), `-l` (Lists only sockets in the listening state), `-n` (Shows numerical addresses and ports (instead of resolving names)), and `-p` (Displays the process using the socket).

brachang

On a Linux shell, what is a command that lists all TCP sockets in a listen state on an ipv4 address, showing the program using it

To list all TCP sockets in a listen state on an IPv4 address, including the program using the socket, you can use the `ss` command:

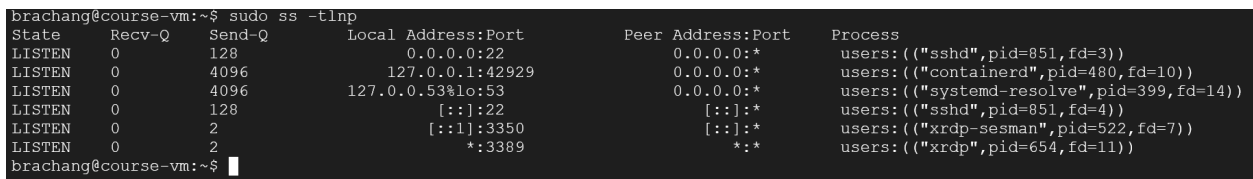
```
bash
```

```
ss -tlnp
```

**Explanation:**

- `-t` : Displays TCP sockets.
- `-l` : Lists only sockets in the listening state.
- `-n` : Shows numerical addresses and ports (instead of resolving names).
- `-p` : Displays the process using the socket.

Run the command using `sudo` and take a screenshot of the output to include in your lab notebook.



The screenshot shows a terminal window with the command `brachang@course-vm:~$ sudo ss -tlnp` and its output. The output is a table with columns: State, Recv-Q, Send-Q, Local Address:Port, Peer Address:Port, and Process.

```
brachang@course-vm:~$ sudo ss -tlnp
```

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
LISTEN	0	128	0.0.0.0:22	0.0.0.0:*	users: (("sshd",pid=851,fd=3))
LISTEN	0	4096	127.0.0.1:42929	0.0.0.0:*	users: (("containerd",pid=480,fd=10))
LISTEN	0	4096	127.0.0.53%lo:53	0.0.0.0:*	users: (("systemd-resolve",pid=399,fd=14))
LISTEN	0	128	:::22	:::*	users: (("sshd",pid=851,fd=4))
LISTEN	0	2	:::1:3350	:::*	users: (("xrdp-sesman",pid=522,fd=7))
LISTEN	0	2	:::3389	:::*	users: (("xrdp",pid=654,fd=11))

```
brachang@course-vm:~$
```

List a service that can be contacted from any interface on the machine. List a service that can only be contacted by local processes.

0.0.0.0:22 can be contacted from any interface.

127.0.0.1:42929 can only be contacted by local processes.

Run the command again, but do not use sudo as this is a machine managed by CAT.

Include a screenshot of the output.

```
brachang@ada:~$ ss -tlnp
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port      Process
LISTEN     0            100         127.0.0.1:25             0.0.0.0:*                sshd
LISTEN     0            4096         127.0.0.1:631           0.0.0.0:*                sshd
LISTEN     0            128         127.0.0.1:6102          0.0.0.0:*                sshd
LISTEN     0            128         127.0.0.1:6100          0.0.0.0:*                sshd
LISTEN     0            128         127.0.0.1:6101          0.0.0.0:*                sshd
LISTEN     0            4096         127.0.0.53%lo:53        0.0.0.0:*                sshd
LISTEN     0            1024         127.0.0.1:41555         0.0.0.0:*                sshd
LISTEN     0            1024         127.0.0.1:40927         0.0.0.0:*                sshd
LISTEN     0            1024         127.0.0.1:38019         0.0.0.0:*                sshd
LISTEN     0            4096         127.0.0.54:53           0.0.0.0:*                sshd
LISTEN     0            4096         *:22                    *:22                      sshd
LISTEN     0            2           *:3389                   *:3389                    sshd
LISTEN     0            100         [::]:25                 [::]:25                   sshd
LISTEN     0            4096         [::]:631                 [::]:631                  sshd
LISTEN     0            2           [::]:3350                 [::]:3350                  sshd
LISTEN     0            128         [::]:6102                 [::]:6102                  sshd
LISTEN     0            128         [::]:6100                 [::]:6100                  sshd
LISTEN     0            128         [::]:6101                 [::]:6101                  sshd
brachang@ada:~$
```

List the services that this machine provides for external access

All services hear appear to only be used for local access

2. -

Take a screenshot of the prompt and the command that ChatGPT generates

brachang

Find a `lsof` command and its command line flags that lists all TCP sockets in a listen state on an IPv4 address, showing the program using it.

To achieve the same result using the `lsof` command, you can run:

bash

Copy Edit

```
lsof -nP -iTCP -sTCP:LISTEN
```

### Explanation of Flags:

- `-n` : Prevents hostname resolution, showing numerical IP addresses.
- `-P` : Prevents port name resolution, showing port numbers instead of service names.
- `-iTCP` : Filters for TCP sockets.
- `-sTCP:LISTEN` : Filters to show only sockets in the `LISTEN` state.

Run the command using `sudo` and take a screenshot of the output to include in your lab notebook.

```
brachang@course-vm:~$ sudo lsof -nP -iTCP -sTCP:LISTEN
COMMAND  PID    USER      FD  TYPE  DEVICE  SIZE/OFF  NODE NAME
systemd-r 399    systemd-resolve 14u  IPv4  3804      0t0      TCP 127.0.0.53:53 (LISTEN)
container 480     root      10u  IPv4  5967      0t0      TCP 127.0.0.1:42929 (LISTEN)
xrdp-sesm 522     root       7u   IPv6  5298      0t0      TCP [::1]:3350 (LISTEN)
xrdp      654     xrdp      11u  IPv6  5545      0t0      TCP *:3389 (LISTEN)
sshd      851     root       3u   IPv4  5055      0t0      TCP *:22 (LISTEN)
sshd      851     root       4u   IPv6  5057      0t0      TCP *:22 (LISTEN)
brachang@course-vm:~$
```

### 3. TCP #2: Throughput

4. -

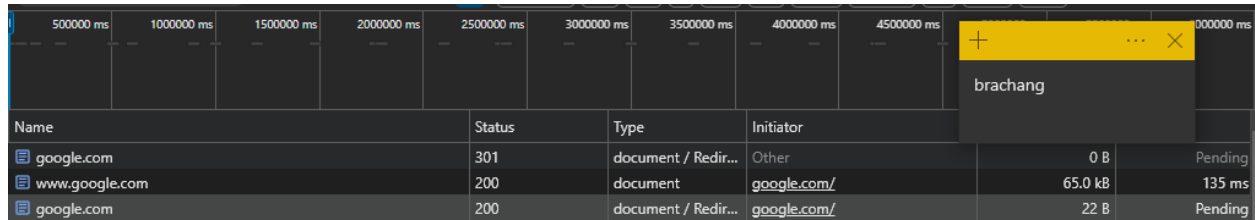
Show a screenshot of the measured bandwidth available between your us-west1-b VM and each of the other Compute Engine VMs. Explain the relative differences (or lack thereof) in your results.

```
brachang@vm-us-west1-b:~$ iperf -c 10.152.0.2 -p 80
-----
Client connecting to 10.152.0.2, TCP port 80
TCP window size: 85.0 KByte (default)
-----
[  1] local 10.138.0.7 port 54920 connected with 10.152.0.2 port 80
[ ID] Interval      Transfer    Bandwidth
[  1] 0.0000-10.1819 sec   151 MBytes  125 Mbits/sec
brachang@vm-us-west1-b:~$ iperf -c 10.132.0.2 -p 80
-----
Client connecting to 10.132.0.2, TCP port 80
TCP window size: 85.0 KByte (default)
-----
[  1] local 10.138.0.7 port 60182 connected with 10.132.0.2 port 80
[ ID] Interval      Transfer    Bandwidth
[  1] 0.0000-10.1713 sec   186 MBytes  154 Mbits/sec
brachang@vm-us-west1-b:~$ iperf -c 10.142.0.3 -p 80
-----
Client connecting to 10.142.0.3, TCP port 80
TCP window size: 85.0 KByte (default)
-----
[  1] local 10.138.0.7 port 55484 connected with 10.142.0.3 port 80
[ ID] Interval      Transfer    Bandwidth
[  1] 0.0000-10.1100 sec   392 MBytes  326 Mbits/sec
brachang@vm-us-west1-b:~$
```

It appears that the closer a VM is to us-west1-b, the more bandwidth there is.

### 5. HTTP #3: Requests

Take a screenshot of the initial requests for your lab notebook.



Name	Status	Type	Initiator	Size	Time
google.com	301	document / Redir...	Other	0 B	Pending
www.google.com	200	document	google.com/	65.0 kB	135 ms
google.com	200	document / Redir...	google.com/	22 B	Pending

What is the URL being requested?

1. <http://google.com/>
2. <https://www.google.com/>
3. <https://google.com/>

Explain the HTTP status code that is returned and what the code indicates

1. 301 Temporary Redirect
2. 200 OK
3. 200 OK

Take a screenshot indicating the version of the HTTP protocol that is used for each request. (Hint: look at the response status line and alt-svc: HTTP response headers indicating HTTP/2 or HTTP/3).



1. **General**

Request URL:	<a href="http://google.com/">http://google.com/</a>
Request Method:	GET
Status Code:	301 Temporary Redirect
Referrer Policy:	strict-origin-when-cross-origin

**Response Headers**

Alt-Svc:	h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
----------	--

2.

Request URL:	https://www.google.com/
Request Method:	GET
Status Code:	200 OK
Remote Address:	172.217.14.196:443
Referrer Policy:	strict-origin-when-cross-origin
▼ Response Headers	
Accept-Ch:	Sec-CH-Prefers-Color-Scheme
Accept-Ch:	Sec-CH-UA-Form-Factors
Accept-Ch:	Sec-CH-UA-Platform
Accept-Ch:	Sec-CH-UA-Platform-Version
Accept-Ch:	Sec-CH-UA-Full-Version
Accept-Ch:	Sec-CH-UA-Arch
Accept-Ch:	Sec-CH-UA-Model
Accept-Ch:	Sec-CH-UA-Bitness
Accept-Ch:	Sec-CH-UA-Full-Version-List
Accept-Ch:	Sec-CH-UA-WoW64
Alt-Svc:	h3=":443"; ma=2592000,h3-29=":443"; ma=2592000

3.

▼ General	
Request URL:	https://google.com/
Request Method:	GET
Status Code:	200 OK
Remote Address:	172.217.14.206:443
Referrer Policy:	strict-origin-when-cross-origin
▼ Response Headers	
Accept-Ch:	Sec-CH-Prefers-Color-Scheme
Accept-Ch:	Sec-CH-UA-Form-Factors
Accept-Ch:	Sec-CH-UA-Platform
Accept-Ch:	Sec-CH-UA-Platform-Version
Accept-Ch:	Sec-CH-UA-Full-Version
Accept-Ch:	Sec-CH-UA-Arch
Accept-Ch:	Sec-CH-UA-Model
Accept-Ch:	Sec-CH-UA-Bitness
Accept-Ch:	Sec-CH-UA-Full-Version-List
Accept-Ch:	Sec-CH-UA-WoW64
Alt-Svc:	h3=":443"; ma=2592000,h3-29=":443"; ma=2592000

Show the URLs the browser is redirected to via this header.

<https://www.google.com/>

▼ General	
Request URL:	http://google.com/
Request Method:	GET
Status Code:	301 Temporary Redirect
Referrer Policy:	strict-origin-when-cross-origin
▼ Response Headers	
Alt-Svc:	h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
Cache-Control:	public, max-age=2592000
Content-Length:	220
Content-Security-Policy-Report-Only:	object-src 'none';base-uri 'self';script-src 'nonce-oX
Content-Type:	text/html; charset=UTF-8
Cross-Origin-Opener-Policy:	same-origin-allow-popups; re
Date:	Tue, 21 Jan 2025 01:42:32 GM
Expires:	Thu, 20 Feb 2025 01:42:32 GM
Location:	https://www.google.com/

Take a screenshot of when cookies are set via Set-Cookie:

Expires:	-1
P3p:	CP="This is not a PSP pol"
Permissions-Policy:	unload=()
Report-To:	["group","gws","max_age"
Server:	gws
Set-Cookie:	AEC=AZ6Zc-UBOQbQfIMB1z1-rTN9boy5h2zeV0UDZH_a26g6g6jowXdLJMSMXFE; expires=Sun, 20-Jul-2025 01:42:32 GMT; path=/; domain=.google.com; Secure; HttpOnly; SameSite=lax
Set-Cookie:	NID=520=hnTMhPSIdEil7_ogFuqRvCL6JtRM_bb9_877PPZU0yzR4Z2biL2dXr4ERJHFmXqgQ7i8NWIIKfAeU_Ln8ET_2dQ3PxXLwPkYoWTetFQ31CEnt58PmJhGHT4swkMOiRQ98VE9g56NfuMO31968xg5kBG1DiBHWJ_E0t5D7xlqao2Y-lqkfuTEZlid4PUZyPYPKu6D47zTCdA; expires=Wed, 23-Jul-2025 01:42:32 GMT; path=/; domain=.google.com; Secure; HttpOnly; SameSite=none

Take a screenshot of when cookies are attached via Cookie:

Cookie:	AEC=AZ6Zc-UBOQbQfIMB1z1-rTN9boy5h2zeV0UDZH_a26g6g6jowXdLJMSMXFE; NID=520=hnTMhPSIdEil7_ogFuqRvCL6JtRM_bb9_877PPZU0yzR4Z2biL2dXr4ERJHFmXqgQ7i8NWIIKfAeU_Ln8ET_2dQ3PxXLwPkYoWTetFQ31CEnt58PmJhGHT4swkMOiRQ98VE9g56NfuMO31968xg5kBG1DiBHWJ_E0t5D7xlqao2Y-lqkfuTEZlid4PUZyPYPKu6D47zTCdA u=0 https://www.google.com/
---------	---



Show the requests and responses in the listing. Click on the last request sent, then click on the response to see that its payload has returned the data that is then rendered on the search page similar to what is shown below for "rabbid"

Request	Response	Size	Time
log?format=json&hasfast=true	other	0 B	1 ms
log?format=json&hasfast=true	other	0 B	2 ms
log?format=json&hasfast=true	other	0 B	1 ms
log?format=json&hasfast=true	other	0 B	1 ms
log?format=json&hasfast=true	other	0 B	1 ms
search?q=p&cp=1&client=gws-wiz&xssi=t&gs_pcr=unde...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	941 B	264 ms
search?q=p&ocp=2&client=gws-wiz&xssi=t&gs_pcr=und...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	830 B	173 ms
search?q=p&port&cp=3&client=gws-wiz&xssi=t&gs_pcr=und...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	1.0 kB	61 ms
search?q=p&port&cp=4&client=gws-wiz&xssi=t&gs_pcr=u...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	1.0 kB	58 ms
search?q=p&port&cp=5&client=gws-wiz&xssi=t&gs_pcr=...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	1.1 kB	70 ms
search?q=p&port&cp=6&client=gws-wiz&xssi=t&gs_pcr...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	1.1 kB	65 ms
search?q=p&port&cp=7&client=gws-wiz&xssi=t&gs_pcr...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	1.1 kB	74 ms
search?q=p&port&cp=8&client=gws-wiz&xssi=t&gs_pcr...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	950 B	74 ms
search?q=p&port&cp=9&client=gws-wiz&xssi=t&gs_pcr...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	961 B	63 ms
search?q=p&port&cp=10&client=gws-wiz&xssi=t&gs_pcr...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	1.3 kB	164 ms
search?q=p&port&cp=11&client=gws-wiz&xssi=t...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	1.0 kB	66 ms
search?q=p&port&cp=12&client=gws-wiz&xssi=...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	1.1 kB	71 ms
search?q=p&port&cp=13&client=gws-wiz&xssi=...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	1.1 kB	60 ms
search?q=p&port&cp=14&client=gws-wiz&xssi=...0&psi=iP...	m=cdo, hsm, jsa, mb4ZUb, cE90b, SNU	1.1 kB	64 ms

The screenshot shows a terminal window with a dark background. At the top, a search command is entered: `logformat=json&hasfast=L`. Below it, a list of search results is displayed, each starting with a magnifying glass icon and followed by a search query. The first result is highlighted in yellow. A small pop-up window is visible over the first result, displaying the word "branchang".

```
logformat=json&hasfast=L
logformat=json&hasfast=L
logformat=json&hasfast=L
logformat=json&hasfast=L
logformat=json&hasfast=L
logformat=json&hasfast=L
logformat=json&hasfast=L
search?q=p&cp=1&client=...
search?q=p&cp=2&client=...
search?q=p&cp=3&client=...
search?q=p&cp=4&client=...
search?q=p&cp=5&client=...
search?q=p&cp=6&client=...
search?q=p&cp=7&client=...
search?q=p&cp=8&client=...
search?q=p&cp=9&client=...
search?q=p&cp=10&client=...
search?q=p&cp=11&client=...
search?q=p&cp=12&client=...
search?q=p&cp=13&client=...
```

branchang

## 02.2: DNS, Recap

## 1. DNS reconnaissance #1 (dig)

**Take a screenshot of the prompt and the dig command produced**

+

...

×

brachang

Produce a dig command that can query a local DNS server at 131.252.208.53 for the A record of www.pdx.edu using TCP

To query a local DNS server at 131.252.208.53 for the A record of www.pdx.edu using TCP, you can use the following dig command:

bash

Copy

Edit

dig @131.252.208.53 www.pdx.edu A +tcp

**Explanation:**

- @131.252.208.53 : Specifies the DNS server to query (in this case, 131.252.208.53 ).
- www.pdx.edu : The domain name for which to query the A record.
- A : Specifies the query type (A record, which maps a domain to an IPv4 address).
- +tcp : Forces the query to use TCP instead of the default UDP.

Take a screenshot of the records returned for your lab notebook.

```

brachang@ada:~$ dig @131.252.208.53 www.pdx.edu A +tcp
; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @131.252.208.53 www.pdx.edu A +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 22239
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 9f531d269b0700e501000000678f0c4722c3c771a9cdf2bc (good)
;; QUESTION SECTION:
;www.pdx.edu.                IN      A

;; ANSWER SECTION:
www.pdx.edu.                 60      IN      A      18.161.6.112
www.pdx.edu.                 60      IN      A      18.161.6.84
www.pdx.edu.                 60      IN      A      18.161.6.96
www.pdx.edu.                 60      IN      A      18.161.6.120

;; Query time: 22 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (TCP)
;; WHEN: Mon Jan 20 18:54:00 PST 2025
;; MSG SIZE rcvd: 132

brachang@ada:~$ dig @131.252.208.53 www.pdx.edu MX +tcp
; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @131.252.208.53 www.pdx.edu MX +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 26834
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: cee2b6fb5eca6dc201000000678f0cb398180d5ecaeb1d03 (good)
;; QUESTION SECTION:
;www.pdx.edu.                IN      MX

;; AUTHORITY SECTION:
www.pdx.edu.                 545     IN      SOA     ns-988.awsdns-59.net. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400

;; Query time: 0 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (TCP)
;; WHEN: Mon Jan 20 18:55:48 PST 2025
;; MSG SIZE rcvd: 152

brachang@ada:~$

```

What cloud provider hosts the web site for [www.pdx.edu](http://www.pdx.edu)?

Amazon.com

What cloud provider handles mail for pdx.edu?

Gmail

Take a screenshot of the results for both records for your lab notebook.

```

brachang@ada:~$ dig mashimaro.cs.pdx.edu NS
; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> mashimaro.cs.pdx.edu NS
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 40979
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;mashimaro.cs.pdx.edu.          IN      NS

;; AUTHORITY SECTION:
cs.pdx.edu.                    297     IN      SOA     walt.ee.pdx.edu. support.cat.pdx.edu. 2025011704 600 300 1209600 300

;; Query time: 1 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Jan 20 20:26:25 PST 2025
;; MSG SIZE rcvd: 105

brachang@ada:~$ dig @walt.ee.pdx.edu mashimaro.cs.pdx.edu A
; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @walt.ee.pdx.edu mashimaro.cs.pdx.edu A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 24950
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 432db8b0dca7ff8801000000678f225ff547c151a95cf9d2 (good)
;; QUESTION SECTION:
;mashimaro.cs.pdx.edu.          IN      A

;; ANSWER SECTION:
mashimaro.cs.pdx.edu.          14400   IN      A        131.252.220.66

;; Query time: 1 msec
;; SERVER: 131.252.208.38#53(walt.ee.pdx.edu) (UDP)
;; WHEN: Mon Jan 20 20:28:16 PST 2025
;; MSG SIZE rcvd: 93

brachang@ada:~$ █

```

## 2. Iterative DNS lookups

List all of the iterative dig commands performed for the lookup

dig

dig @198.41.0.4 console.cloud.google.com A +norecurse +tcp

dig @192.5.6.30 console.cloud.google.com A +norecurse +tcp

dig @ns1.google.com console.cloud.google.com A +norecurse +tcp

Take a screenshot of the results of the final query for your lab notebook.

```
brachang@ada:~$ dig @ns1.google.com console.cloud.google.com A +norecurse +tcp

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @ns1.google.com console.cloud.google.com A +norecurse +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 54478
;; 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: 26 msec
;; SERVER: 216.239.32.10#53(ns1.google.com) (TCP)
;; WHEN: Tue Jan 21 12:13:13 PST 2025
;; MSG SIZE rcvd: 90

brachang@ada:~$
```

### 3. Reverse DNS lookups

Take a screenshot of the prompt and the command produced

+

...

×

brachang|

Produce a single command line with commands `dig`, `egrep`, and `awk`, to list all IPv4 addresses that `espn.go.com` points to

Here's a single command-line that combines `dig`, `egrep`, and `awk` to list all IPv4 addresses (A records) that `espn.go.com` points to:

bash

Copy Edit

```
dig espn.go.com A +short | egrep -o '^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+'
```

### Explanation:

- `dig espn.go.com A +short` :
  - Queries the A records for `espn.go.com`.
  - The `+short` option ensures the output is simplified, showing only the results (IP addresses and CNAMEs).
- `egrep -o '^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+'` :
  - Filters the output to include only IPv4 addresses by matching the pattern of an IPv4 address.
- Output:**
  - Displays only the list of IPv4 addresses.

Take a screenshot of its results for your lab notebook

```
brachang@ada:~$ dig espn.go.com A +short | egrep -o '^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+'
99.84.66.98
99.84.66.108
99.84.66.17
99.84.66.55
brachang@ada:~$
```

Take a screenshot of the command and its results for your lab notebook

```
brachang@ada:~$ for ip in $(dig -x $ip +short | egrep -o '^[a-zA-Z0-9+-]+' | awk '{print $1}'); do
server-99-84-66-108.hio50.r.cloudfront.net.
server-99-84-66-98.hio50.r.cloudfront.net.
server-99-84-66-17.hio50.r.cloudfront.net.
server-99-84-66-55.hio50.r.cloudfront.net.
brachang@ada:~$
```

## 4. Host enumeration

Take a screenshot of the results in your lab notebook

```
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.  
brachang@ada:~/CS430$ cat 220hosts.txt | head -185 | tail -158
```

## 5. Geographic DNS #2

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

The address 131.252.208.53 returned Portland, Oregon

The address 198.82.247.66 returned Blacksburg, Virginia

Record one address for [www.google.com](http://www.google.com) from each result for your lab notebook.

```

brachang@ada:~/CS430$ dig @131.252.208.53 www.google.com

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @131.252.208.53 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 10786
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 9283fd23154b72c00100000067902caf3b0d1b5c0c57d70d (good)
;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.                207     IN      A      142.250.217.68

;; Query time: 0 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (UDP)
;; WHEN: Tue Jan 21 15:24:32 PST 2025
;; MSG SIZE rcvd: 87

brachang@ada:~/CS430$ dig @198.82.247.66 www.google.com

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @198.82.247.66 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 47900
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 226edcf48edf68900100000067902d4635f5670c7e765517 (good)
;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.                103     IN      A      172.253.62.104
www.google.com.                103     IN      A      172.253.62.103
www.google.com.                103     IN      A      172.253.62.99
www.google.com.                103     IN      A      172.253.62.147
www.google.com.                103     IN      A      172.253.62.106
www.google.com.                103     IN      A      172.253.62.105

;; Query time: 79 msec
;; SERVER: 198.82.247.66#53(198.82.247.66) (UDP)
;; WHEN: Tue Jan 21 15:27:02 PST 2025
;; MSG SIZE rcvd: 167

brachang@ada:~/CS430$ █

```

What are the geographic coordinates of each DNS server and the IP address it resolves for www.google.com?



Address 142.250.217.68 returned Seattle, Washington. Lat: 47.6043. Long: -122.3298

Address 172.253.62.104 returned Mountain View, California. Lat: 37.4060. Long: -122.0785

Take a screenshot of the results for your lab notebook.

```
brachang@ada:~/CS430$ traceroute 131.252.208.53
traceroute to 131.252.208.53 (131.252.208.53), 30 hops max, 60 byte packets
 1 rdns.cat.pdx.edu (131.252.208.53) 1.141 ms 1.061 ms 0.994 ms
brachang@ada:~/CS430$
```

```
brachang@ada:~/CS430$ traceroute 198.82.247.66
traceroute to 198.82.247.66 (198.82.247.66), 30 hops max, 60 byte packets
 1 glados.cat.pdx.edu (131.252.208.21) 6.246 ms 6.150 ms 6.080 ms
 2 0015-opsense.cat.pdx.edu (10.208.91.1) 0.133 ms 0.125 ms 0.126 ms
 3 CORE1.net.pdx.edu (131.252.5.142) 1.077 ms 1.026 ms 0.936 ms
 4 131.252.5.213 (131.252.5.213) 0.589 ms 0.492 ms 1.284 ms
 5 e0-28.switch4.pdx1.he.net (216.218.230.89) 5.127 ms 5.015 ms 5.098 ms
 6 100ge0-28.core1.pdx3.he.net (184.104.188.77) 1.321 ms 100ge0-36.core1.pdx2.he.net (184.104.195.66) 1.844 ms 2.092 ms
 7 ae11.bar4.por1.sp.lumen.tech (4.68.38.101) 19.869 ms 19.781 ms 19.717 ms
 8 * ae11.bar4.por1.sp.lumen.tech (4.68.38.101) 19.482 ms *
 9 RADWARE-LTD.edge1.SanJose1.Level3.net (4.35.71.202) 16.480 ms * *
10 * RADWARE-LTD.edge1.SanJose1.Level3.net (4.35.71.202) 16.837 ms *
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 128.173.0.214 (128.173.0.214) 79.788 ms 79.692 ms cas-core.lo0.2000.cns.vt.edu (198.82.1.143) 80.012 ms
24 jeru.cns.vt.edu (198.82.247.66) 79.486 ms 79.409 ms cas-core.lo0.2000.cns.vt.edu (198.82.1.143) 79.804 ms
brachang@ada:~/CS430$
```

```
brachang@ada:~/CS430$ traceroute 142.250.217.68
traceroute to 142.250.217.68 (142.250.217.68), 30 hops max, 60 byte packets
 1 glados.cat.pdx.edu (131.252.208.21) 5.868 ms * 5.710 ms
 2 router.seas.pdx.edu (10.208.91.1) 0.225 ms 0.153 ms 0.133 ms
 3 CORE1.net.pdx.edu (131.252.5.142) 3.918 ms 3.850 ms 3.754 ms
 4 131.252.5.213 (131.252.5.213) 0.509 ms 0.530 ms 0.480 ms
 5 google.nwax.net (198.32.195.34) 8.484 ms 4.477 ms 4.484 ms
 6 192.178.105.35 (192.178.105.35) 4.651 ms 4.515 ms 108.170.255.123 (108.170.255.123) 5.461 ms
 7 142.251.55.199 (142.251.55.199) 4.036 ms 142.251.55.197 (142.251.55.197) 4.725 ms 142.251.55.199 (142.251.55.199) 3.860 ms
 8 sea09s29-in-f4.1e100.net (142.250.217.68) 4.281 ms 4.633 ms 4.645 ms
brachang@ada:~/CS430$
```

```
brachang@ada:~/CS430$ traceroute 172.253.62.104
traceroute to 172.253.62.104 (172.253.62.104), 30 hops max, 60 byte packets
 1 glados.cat.pdx.edu (131.252.208.21) 3.633 ms 8.133 ms 8.076 ms
 2 router.seas.pdx.edu (10.208.91.1) 0.188 ms 0.120 ms 0.116 ms
 3 CORE1.net.pdx.edu (131.252.5.142) 7.676 ms 7.613 ms 7.548 ms
 4 131.252.5.213 (131.252.5.213) 0.819 ms 0.752 ms 0.690 ms
 5 google.nwax.net (198.32.195.34) 4.591 ms 4.526 ms 4.725 ms
 6 108.170.255.123 (108.170.255.123) 5.018 ms 192.178.105.35 (192.178.105.35) 4.412 ms 192.178.105.129 (192.178.105.129) 5.019 ms
 7 108.170.255.132 (108.170.255.132) 13.089 ms 108.170.255.186 (108.170.255.186) 4.753 ms 192.178.105.46 (192.178.105.46) 4.846 ms
 8 216.239.57.194 (216.239.57.194) 12.213 ms 142.251.64.18 (142.251.64.18) 10.262 ms 216.239.41.34 (216.239.41.34) 10.864 ms
 9 142.251.226.157 (142.251.226.157) 52.528 ms 142.251.226.161 (142.251.226.161) 53.481 ms 142.250.213.71 (142.250.213.71) 52.066 ms
10 192.178.81.224 (192.178.81.224) 65.599 ms 192.178.81.228 (192.178.81.228) 66.354 ms 192.178.81.238 (192.178.81.238) 70.923 ms
11 142.251.244.140 (142.251.244.140) 65.721 ms 142.250.210.246 (142.250.210.246) 66.869 ms 142.251.244.158 (142.251.244.158) 64.636 ms
12 72.14.237.105 (72.14.237.105) 66.353 ms 142.251.245.105 (142.251.245.105) 66.751 ms 209.85.247.109 (209.85.247.109) 66.945 ms
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 bc-in-f104.1e100.net (172.253.62.104) 64.046 ms 66.335 ms *
brachang@ada:~/CS430$
```

## 6. Wireshark Lab #3

In a terminal, using commands from prior labs, find the addresses and interfaces on the VM.

Make a note of:

The IP address of the VM

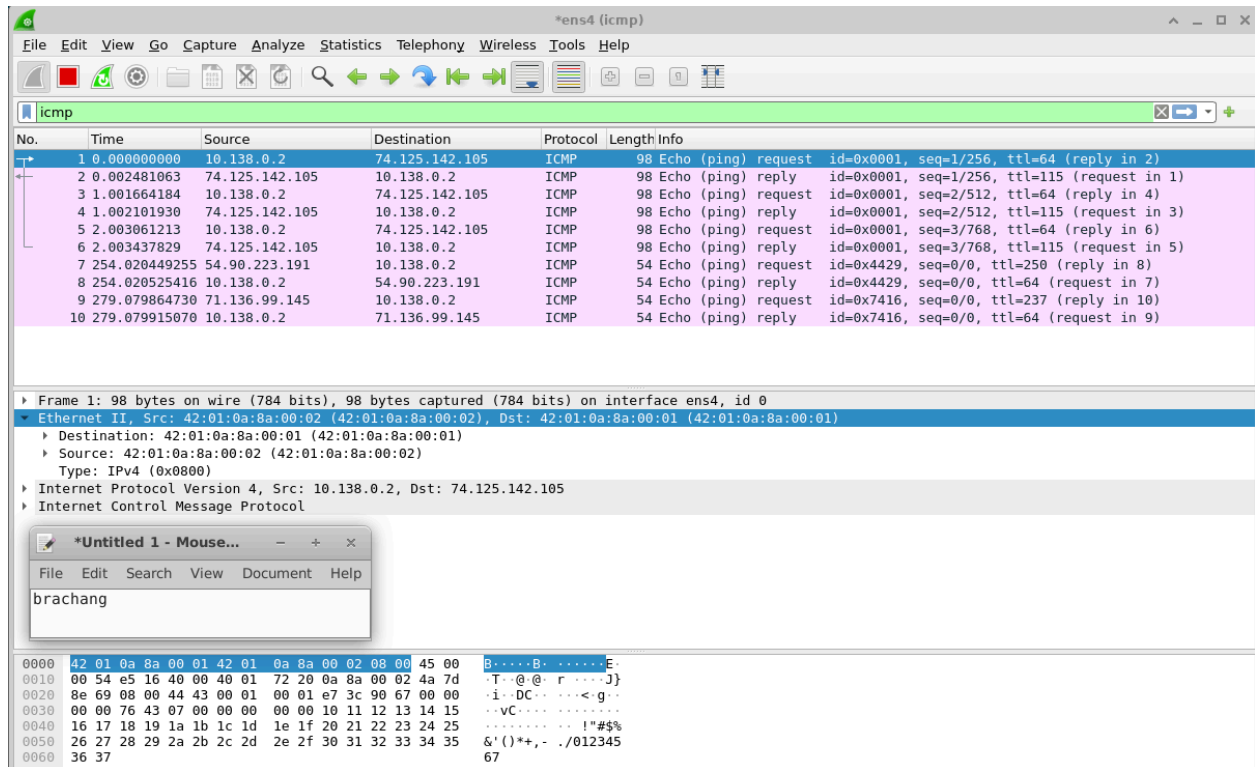
The name of the local virtual ethernet interface

The IP address of the default router

```
brachang@course-vm:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1460 qdisc mq state UP group default qlen 1000
    link/ether 42:01:0a:8a:00:02 brd ff:ff:ff:ff:ff:ff
    inet 10.138.0.2/32 metric 100 scope global dynamic ens4
        valid_lft 85179sec preferred_lft 85179sec
    inet6 fe80::4001:aff:fe8a:2/64 scope link
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:7b:22:ae:6c brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
brachang@course-vm:~$ ip -o link show | awk -F: ' '{print $2}'
lo
ens4
docker0
brachang@course-vm:~$ ip route show default
default via 10.138.0.1 dev ens4 proto dhcp src 10.138.0.2 metric 100
brachang@course-vm:~$
```

## 7. -

Take a screenshot of the bytes in the packet dump window as shown below



**Does the destination MAC address correspond to an interface on the VM, an interface on the default router or an interface on Google's web site?**

An interface on the default router

**Does the destination MAC address correspond to an interface on the VM, an interface on the default router or an interface on Google's web site?**

Also an interface on the default route

## 8. Network Recap Lab #4

**Find the IP address of <OdinId>.oregonctf.org, replacing <OdinId> with your OdinId**

```
brachang@course-vm:~$ dig brachang.oregonctf.org

; <<>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <<>> brachang.oregonctf.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47055
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
brachang.oregonctf.org. 3600     IN      A      35.233.233.233

;; Query time: 189 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Wed Jan 22 01:25:41 UTC 2025
;; MSG SIZE rcvd: 67

brachang@course-vm:~$ S█
```

35.233.233.233

**Take a screenshot of the all of the packets returned within Wireshark that includes their packet numbers**

arp or dns or http or ip.addr==35.233.233.233						
No.	Time	Source	Destination	Protocol	Length Info	
376	1.840661	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
377	1.840689	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
308	7.835294	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
399	7.835317	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
419	13.834712	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
420	13.834741	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
1171	19.836280	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
1172	19.836299	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
2307	25.836675	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
2308	25.836697	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
2481	31.835185	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
2482	31.835216	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
2558	37.838008	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
2559	37.838028	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
2706	43.840034	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
2707	43.840055	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
2728	49.808906	169.254.169.254	10.138.0.2	HTTP/J...	1962	[TCP ACKed unseen segment] HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
2730	49.809594	10.138.0.2	169.254.169.254	HTTP	281	GET /computeMetadata/v1/?recursive=true&alt=json&wait_for_change=true&last_etag=16c570cd6f6bf436&timeout_sec=60 HTTP/1.1
2733	49.837905	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
2734	49.837907	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
2740	51.362256	169.254.169.254	10.138.0.2	HTTP/J...	1962	[TCP ACKed unseen segment] HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
2742	51.363108	10.138.0.2	169.254.169.254	HTTP	281	GET /computeMetadata/v1/?alt=json&last_etag=16c570cd6f6bf436&recursive=true&timeout_sec=60&wait_for_change=true HTTP/1.1
2769	55.835119	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
2770	55.835145	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
2819	61.836679	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
2820	61.836694	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
2914	67.837408	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
2915	67.837430	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
3232	73.838412	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
3233	73.838436	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
3309	79.837767	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
3390	79.837798	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
3633	85.837271	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
3634	85.837291	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
4130	91.835165	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
4131	91.835187	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
4208	97.834020	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
4209	97.834051	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
4226	100.013203	10.138.0.2	169.254.169.254	DNS	112	Standard query 0xa9be A course-vm.c.cloud-chang-brachang.internal OPT
4227	100.013382	10.138.0.2	169.254.169.254	DNS	112	Standard query 0xc111 AAAA course-vm.c.cloud-chang-brachang.internal OPT
4228	100.017569	169.254.169.254	10.138.0.2	DNS	128	Standard query response 0xa9be A course-vm.c.cloud-chang-brachang.internal A 10.138.0.2 OPT
4229	100.017611	169.254.169.254	10.138.0.2	DNS	201	Standard query response 0xc111 AAAA course-vm.c.cloud-chang-brachang.internal SOA ns.global.gcedns-prod.internal OPT
4230	100.030435	42:01:0a:8a:00:02	Broadcast	ARP	42	Who has 10.138.0.1? Tell 10.138.0.2
4231	100.032724	42:01:0a:8a:00:01	42:01:0a:8a:00:02	ARP	42	10.138.0.1 is at 42:01:0a:8a:00:01
4233	100.035727	10.138.0.2	169.254.169.254	DNS	93	Standard query response 0x68b7 AAAA brachang.oregonctf.org OPT
4238	100.120857	169.254.169.254	10.138.0.2	DNS	175	Standard query response 0x68b7 AAAA brachang.oregonctf.org SOA ns-cloud-d1.googledomains.com OPT
4242	100.121716	10.138.0.2	35.233.233.233	TCP	74	80 - 58352 [SYN] Seq=0 Win=65320 Len=0 MSS=1420 SACK_PERM=1 TSval=1664931287 TSecr=0 WS=128
4246	100.127548	35.233.233.233	10.138.0.2	TCP	74	80 - 58352 [FIN, ACK] Seq=7774 Ack=139 Win=64640 Len=0 TSval=663611590 TSecr=1664931295
4247	100.127600	10.138.0.2	35.233.233.233	TCP	66	58352 - 80 [ACK] Seq=139 Ack=7775 Win=81024 Len=0 TSval=1664931296 TSecr=663611590
4248	100.127719	10.138.0.2	35.233.233.233	HTTP	203	GET / HTTP/1.1
4248	100.127719	10.138.0.2	35.233.233.233	HTTP	203	GET / HTTP/1.1
4249	100.128263	35.233.233.233	10.138.0.2	TCP	66	80 - 58352 [ACK] Seq=1 Ack=138 Win=64640 Len=0 TSval=663611588 TSecr=1664931293
4250	100.128465	35.233.233.233	10.138.0.2	TCP	7106	80 - 58352 [PSH, ACK] Seq=1 Ack=138 Win=64640 Len=7048 TSval=663611588 TSecr=1664931293 [TCP segment of a reassembled PDU]
4251	100.128497	10.138.0.2	35.233.233.233	TCP	66	58352 - 80 [ACK] Seq=138 Ack=7041 Win=78208 Len=0 TSval=1664931294 TSecr=663611588
4252	100.128593	35.233.233.233	10.138.0.2	HTTP	799	HTTP/1.1 200 OK (text/html)
4253	100.128606	10.138.0.2	35.233.233.233	TCP	66	58352 - 80 [ACK] Seq=138 Ack=7774 Win=81024 Len=0 TSval=1664931294 TSecr=663611588
4254	100.129672	10.138.0.2	35.233.233.233	TCP	66	58352 - 80 [FIN, ACK] Seq=138 Ack=7774 Win=81024 Len=0 TSval=1664931295 TSecr=663611588
4255	100.130227	35.233.233.233	10.138.0.2	TCP	66	80 - 58352 [FIN, ACK] Seq=7774 Ack=139 Win=64640 Len=0 TSval=663611590 TSecr=1664931295
4256	100.130260	10.138.0.2	35.233.233.233	TCP	66	58352 - 80 [ACK] Seq=139 Ack=7775 Win=81024 Len=0 TSval=1664931296 TSecr=663611590
4280	103.838669	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
4289	103.838890	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
4310	109.828797	169.254.169.254	10.138.0.2	HTTP/J...	1962	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
4312	109.829337	10.138.0.2	169.254.169.254	HTTP	281	GET /computeMetadata/v1/?recursive=true&alt=json&wait_for_change=true&last_etag=16c570cd6f6bf436&timeout_sec=60 HTTP/1.1
4315	109.838347	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
4316	109.838368	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02
4321	111.381597	169.254.169.254	10.138.0.2	HTTP/J...	1962	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
4323	111.382403	10.138.0.2	169.254.169.254	HTTP	281	GET /computeMetadata/v1/?alt=json&last_etag=16c570cd6f6bf436&recursive=true&timeout_sec=60&wait_for_change=true HTTP/1.1
4400	115.836661	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
4481	115.836679	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02

arp or dns or http or ip.addr==35.233.233.233

ARP

What packet numbers in the trace are the result of the VM attempting to get the hardware address of the default router?

These ones

4480	115.836661	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
4315	109.838347	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
4288	103.838869	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
4208	97.834929	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
4130	91.835165	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
3633	85.837271	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
3389	79.837767	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
3232	73.838412	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
2914	67.837408	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
2819	61.836879	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
2769	55.835119	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
2733	49.837965	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
2706	43.840034	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
2558	37.838008	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
2481	31.835185	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
2307	25.836675	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
1171	19.836280	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
419	13.834712	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
398	7.835294	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
376	1.840661	42:01:0a:8a:00:01	Broadcast	ARP	42 Who has 10.138.0.2? Tell 10.138.0.1
4230	100.030435	42:01:0a:8a:00:02	Broadcast	ARP	42 Who has 10.138.0.1? Tell 10.138.0.2

What is this hardware address?

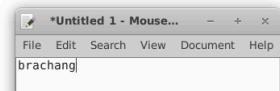
4481	115.836679	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
4316	109.838368	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
4289	103.838890	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
4209	97.834951	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
4131	91.835187	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
3634	85.837291	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
3390	79.837798	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
3233	73.838436	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
2915	67.837430	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
2820	61.836904	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
2770	55.835145	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
2734	49.837986	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
2707	43.840055	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
2559	37.838028	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
2482	31.835216	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
2308	25.836697	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
1172	19.836299	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
420	13.834741	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
399	7.835317	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
377	1.840689	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42 10.138.0.2 is at 42:01:0a:8a:00:02
4231	100.032724	42:01:0a:8a:00:01	42:01:0a:8a:00:02	ARP	42 10.138.0.1 is at 42:01:0a:8a:00:01

Based off of these packets, the hardware address should be 42:01:0a:8a:00:02

DNS

What packet numbers in the trace correspond to the DNS request for the web site?

No.	Time	Source	Destination	Protocol	Length	Info
4226	100.013203	10.138.0.2	169.254.169.254	DNS	112	Standard query 0xa9be A course-vm.c.cloud-chang-brachang.internal OPT
4227	100.013382	10.138.0.2	169.254.169.254	DNS	112	Standard query 0xcff1 AAAA course-vm.c.cloud-chang-brachang.internal OPT
4228	100.017569	169.254.169.254	10.138.0.2	DNS	128	Standard query response 0xa9be A course-vm.c.cloud-chang-brachang.internal A 10.138.0.2 OPT
4229	100.017611	169.254.169.254	10.138.0.2	DNS	201	Standard query response 0xcff1 AAAA course-vm.c.cloud-chang-brachang.internal SOA ns.global.gcedns-prod.internal OPT
4233	100.035727	10.138.0.2	169.254.169.254	DNS	93	Standard query 0x68b7 AAAA brachang.oregonctf.org OPT
4238	100.120857	169.254.169.254	10.138.0.2	DNS	175	Standard query response 0x68b7 AAAA brachang.oregonctf.org SOA ns-cloud-d1.googleddomains.com OPT



Packet numbers 4233 and 4238 correspond to the request to brachang.oregonctf.org

What is the IP address of the local DNS server being queried?



169.254.169.254

TCP

What packet numbers in the trace correspond to the initial TCP handshake for the web request?

No.	Time	Source	Destination	Protocol	Length	Info
4242	100.121716	10.138.0.2	35.233.233.233	TCP	74	58352 → 80 [SYN] Seq=0 Win=65536 Len=0 MSS=1420 SACK_PERM=1 TSval=1664931287 TSecr=0 WS=128
4246	100.127548	35.233.233.233	10.138.0.2	TCP	74	80 → 58352 [SYN, ACK] Seq=0 Ack=1 Win=64768 Len=0 MSS=1420 SACK_PERM=1 TSval=663611586 TSecr=1664931287 WS=128
4247	100.127600	10.138.0.2	35.233.233.233	TCP	66	58352 → 80 [ACK] Seq=1 Ack=1 Win=65408 Len=0 TSval=1664931293 TSecr=663611586
4248	100.127719	10.138.0.2	35.233.233.233	HTTP	203	GET / HTTP/1.1
4249	100.128263	35.233.233.233	10.138.0.2	TCP	66	80 → 58352 [ACK] Seq=1 Ack=138 Win=64640 Len=0 TSval=663611588 TSecr=1664931293
4250	100.128465	35.233.233.233	10.138.0.2	TCP	7106	80 → 58352 [PSH, ACK] Seq=1 Ack=138 Win=64640 Len=7040 TSval=663611588 TSecr=1664931293 [TCP segment of a reassembled PDU]
4251	100.128497	10.138.0.2	35.233.233.233	TCP	66	58352 → 80 [ACK] Seq=138 Ack=7041 Win=78208 Len=0 TSval=1664931294 TSecr=663611588
4252	100.128593	35.233.233.233	10.138.0.2	HTTP	799	HTTP/1.1 200 OK (text/html)
4253	100.128606	10.138.0.2	35.233.233.233	TCP	66	58352 → 80 [ACK] Seq=138 Ack=7774 Win=81824 Len=0 TSval=1664931294 TSecr=663611588
4254	100.129672	10.138.0.2	35.233.233.233	TCP	66	58352 → 80 [FIN, ACK] Seq=138 Ack=7774 Win=91824 Len=0 TSval=1664931295 TSecr=663611588
4255	100.130237	35.233.233.233	10.138.0.2	TCP	66	80 → 58352 [FIN, ACK] Seq=7774 Ack=139 Win=64640 Len=0 TSval=663611590 TSecr=1664931295
4256	100.130260	10.138.0.2	35.233.233.233	TCP	66	58352 → 80 [ACK] Seq=139 Ack=7775 Win=81824 Len=0 TSval=1664931296 TSecr=663611590

Packet numbers 4242 and 4246

How long does it take to perform the initial TCP handshake?

In about .00005 seconds

HTTP

What packet numbers in the trace correspond to the actual HTTP request and response?

No.	Time	Source	Destination	Protocol	Length	Info
2738	49.808986	169.254.169.254	10.138.0.2	HTTP/1.1	1952	1952 [TCP Acked unseen segment] HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
2739	49.809594	10.138.0.2	169.254.169.254	HTTP	281	GET /computeMetadata/v1/?recursive=true&alt=json&wait_for_change=true&last_etag=16c570cd6f6fb436timeout_sec=60 HTTP/1.1
2740	51.362256	169.254.169.254	10.138.0.2	HTTP/1.1	1962	[TCP Acked unseen segment] HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
2742	51.363100	10.138.0.2	169.254.169.254	HTTP	281	GET /computeMetadata/v1/?alt=json&last_etag=16c570cd6f6fb436recursive=true&timeout_sec=60&wait_for_change=true HTTP/1.1
4248	100.127719	10.138.0.2	35.233.233.233	HTTP	203	GET / HTTP/1.1
4252	100.128593	35.233.233.233	10.138.0.2	HTTP	799	HTTP/1.1 200 OK (text/html)
4310	109.828797	169.254.169.254	10.138.0.2	HTTP/1.1	1962	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
4312	109.829337	10.138.0.2	169.254.169.254	HTTP	281	GET /computeMetadata/v1/?recursive=true&alt=json&wait_for_change=true&last_etag=16c570cd6f6fb436timeout_sec=60 HTTP/1.1
4321	111.381597	169.254.169.254	10.138.0.2	HTTP/1.1	1962	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
4323	111.382403	10.138.0.2	169.254.169.254	HTTP	281	GET /computeMetadata/v1/?alt=json&last_etag=16c570cd6f6fb436recursive=true&timeout_sec=60&wait_for_change=true HTTP/1.1

Packet numbers 4248 and 4252

How long does it take to process the HTTP request after the handshake?

9.70 seconds