

# NETWORKING AND SECURITY

## LAB - 2

Submitted by – **Kartik Attri**

Submitted To – **Dr. Shaoquan Jiang**

Student Number – **110091738**

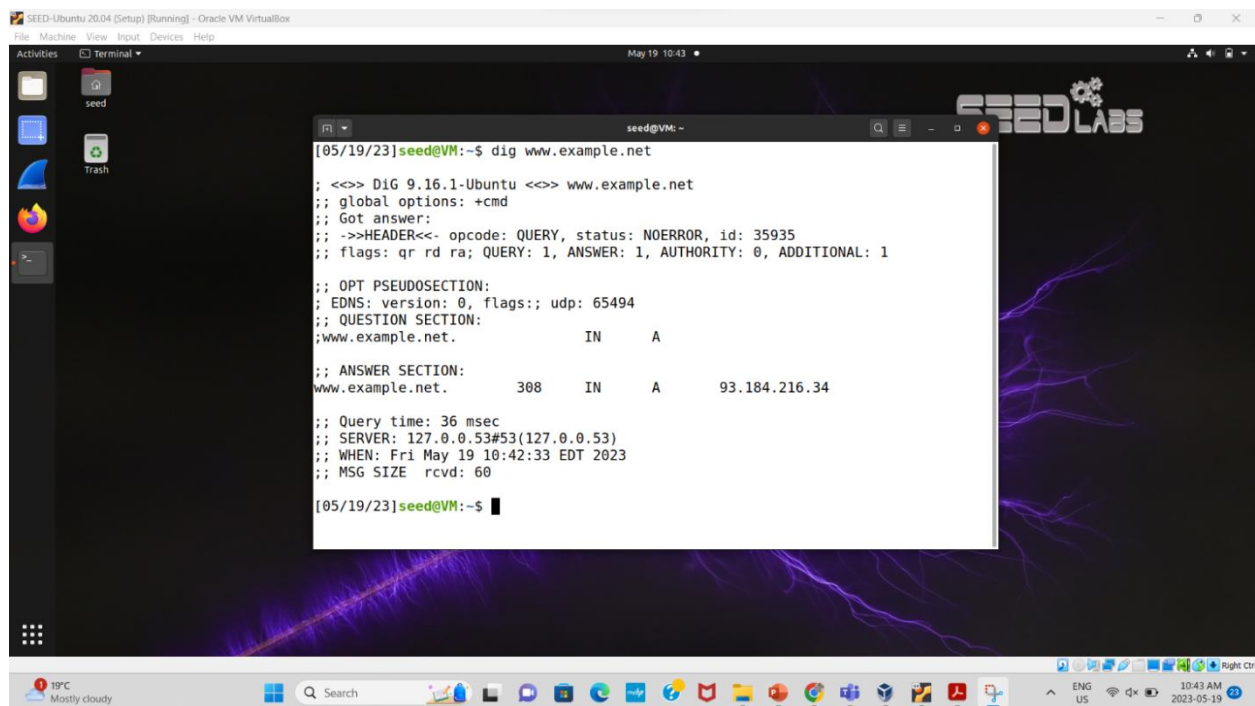
## Question 1-

### Part A

Command dig is used to find the IP address of a hostname such as [www.google.com](http://www.google.com). To do this, you can simply run `dig www.google.com`. This is the result returned by your local DNS server when you run dig command to request it to find out the IP address of [www.google.com](http://www.google.com). Question section is to repeat your question and A stands for the record of IP address. We can see that [www.google.com](http://www.google.com) has an IP address 172.217.1.164.

### Answer-

Ip address of [www.example.net](http://www.example.net) = 93.184.216.34



```
[05/19/23]seed@VM:~$ dig www.example.net

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35935
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.net.                308     IN      A      93.184.216.34

;; Query time: 36 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Fri May 19 10:42:33 EDT 2023
;; MSG SIZE rcvd: 60

[05/19/23]seed@VM:~$
```

### Part B

run Wireshark on your VM, then `$ dig www.example.net` and stop wireshark. Look at the DNS request packet (using filter DNS to find it easily), confirm that the transport layer protocol is UDP. What are the values of this UDP header (you need to first check the header fields learned in class)?

Note: You might need to clear the DNS cache, using `$ sudo systemd-resolve - - flush-caches`

## Answer-

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.211.55.3	10.211.55.1	DNS	97	Standard query 0x54c2 A www.github.com OPT
2	0.817661490	10.211.55.1	10.211.55.3	DNS	115	Standard query response 0x54c2 A www.github.com CNAME
3	7.037119607	10.211.55.3	34.117.65.55	TLSv1.2	84	Application Data
4	7.037851667	34.117.65.55	10.211.55.3	TCP	54	443 → 35674 [ACK] Seq=1 Ack=31 Win=16384 Len=0
5	7.110853678	34.117.65.55	10.211.55.3	TLSv1.2	80	Application Data
6	7.152840616	10.211.55.3	34.117.65.55	TCP	54	35674 → 443 [ACK] Seq=31 Ack=27 Win=501 Len=0
7	7.215184417	34.117.65.55	10.211.55.3	TLSv1.2	78	Application Data
8	7.215239207	10.211.55.3	34.117.65.55	TCP	54	35674 → 443 [ACK] Seq=31 Ack=51 Win=501 Len=0
9	7.215760649	10.211.55.3	34.117.65.55	TLSv1.2	82	Application Data
10	7.216019600	34.117.65.55	10.211.55.3	TCP	54	443 → 35674 [ACK] Seq=51 Ack=59 Win=16384 Len=0

Frame 2: 115 bytes on wire (920 bits), 115 bytes captured (920 bits)	0000	00 1c 42 07 a5 87 00 1c 42 00 00 18 08 00 45 00
Ethernet II, Src: Parallel_00:00:18 (00:1c:42:00:00:18), Dst: Parallel_00:00:18 (00:1c:42:00:00:18)	0010	00 65 66 c5 00 00 80 11 50 19 0a d3 37 01 0a d3
Internet Protocol Version 4, Src: 10.211.55.1, Dst: 10.211.55.3	0020	37 03 00 35 b5 ea 00 51 fa b2 54 c2 81 80 00 01
User Datagram Protocol, Src Port: 53, Dst Port: 46570	0030	00 02 00 00 00 01 03 77 77 77 06 67 69 74 68 75
Source Port: 53	0040	62 03 63 6f 6d 00 00 01 00 01 c0 0c 00 05 00 01
Destination Port: 46570	0050	00 00 09 2d 00 02 c0 10 c0 10 00 01 00 01 00 00
Length: 81	0060	00 05 00 04 8c 52 70 03 00 00 29 10 00 00 00 00
Checksum: 0xfab2 [unverified]	0070	00 00 00
[Checksum Status: Unverified]		

### Values in UDP header:-

Source port: 53

Destination Port: 46570

Length: 81

### Part C

In the DNS request packet in step b, the destination IP is your local DNS server's IP. What is this value? As said, DNS is serviced by UDP and has no connection setup before sending DNS request. You can confirm this by checking that there is no any packet in Wireshark exchanged between your VM and local DNS server, prior to the DNS request packet (show the screen shot of the window of Wireshark for the list of packets).

**Answer-** As in below screenshot we can see that there is no exchange of packets which confirm that's it's UDP ,

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.211.55.3	10.211.55.1	DNS	97	Standard query 0x54c2 A www.github.com OPT
2	0.017661490	10.211.55.1	10.211.55.3	DNS	115	Standard query response 0x54c2 A www.github.com CN
3	7.037119607	10.211.55.3	34.117.65.55	TLSv1.2	84	Application Data
4	7.037851667	34.117.65.55	10.211.55.3	TCP	54	443 → 35674 [ACK] Seq=1 Ack=31 Win=16384 Len=0
5	7.110853678	34.117.65.55	10.211.55.3	TLSv1.2	80	Application Data
6	7.152840616	10.211.55.3	34.117.65.55	TCP	54	35674 → 443 [ACK] Seq=31 Ack=27 Win=501 Len=0
7	7.215184417	34.117.65.55	10.211.55.3	TLSv1.2	78	Application Data
8	7.215239207	10.211.55.3	34.117.65.55	TCP	54	35674 → 443 [ACK] Seq=31 Ack=51 Win=501 Len=0
9	7.215760649	10.211.55.3	34.117.65.55	TLSv1.2	82	Application Data
10	7.216019600	34.117.65.55	10.211.55.3	TCP	54	443 → 35674 [ACK] Seq=51 Ack=59 Win=16384 Len=0

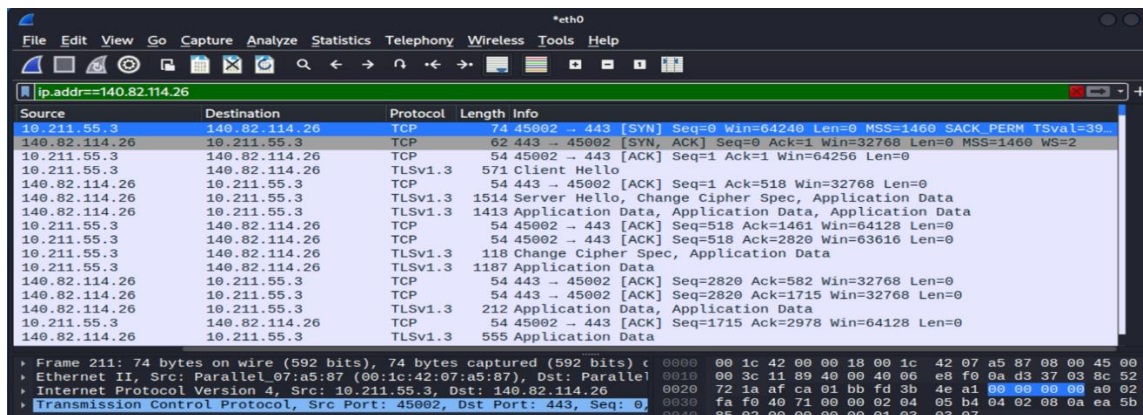
  

▶ Frame 2: 115 bytes on wire (920 bits), 115 bytes captured (920 bits) ▶ Ethernet II, Src: Parallel_00:00:18 (00:1c:42:00:00:18), Dst: Paralle ▶ Internet Protocol Version 4, Src: 10.211.55.1, Dst: 10.211.55.3 ▶ User Datagram Protocol, Src Port: 53, Dst Port: 46570 Source Port: 53 Destination Port: 46570 Length: 81 Checksum: 0xfab2 [unverified] [Checksum Status: Unverified]	0000 00 1c 42 07 a5 87 00 1c 42 00 00 18 08 00 45 00 0010 00 65 06 c5 00 00 80 11 50 19 0a d3 37 01 0a d3 0020 37 03 00 35 b5 ea 00 51 fa b2 54 c2 81 80 00 01 0030 00 02 00 00 00 01 03 77 77 77 06 67 69 74 68 75 0040 62 03 63 6f 6d 00 00 01 00 01 c8 0c 00 05 00 01 0050 00 00 09 2d 00 02 c0 10 c0 10 00 01 00 01 00 00 0060 00 05 00 04 8c 52 70 03 00 00 29 10 00 00 00 00 0070 00 00 00
---	---

## Question 2-

Run Wireshark and then access [www.example.net](http://www.example.net) using Firefox (you might need to clear the browser history). Then stop the Wireshark. Check your list of packets in Wireshark window, filtered by the ip address of [www.example.net](http://www.example.net). You can see that before the HTTP request to [www.example.net](http://www.example.net), there is a connection stage with three packets: SYN packet, SYN-ACK packet and ACK packet. This is to provide the connection setup between your VM and [www.example.net](http://www.example.net). Confirm this. Also, confirm that the transport layer protocol in these packets (check one of them is good enough) is TCP. When the message exchange starts, you can see ACK packet. This is to confirm the receipt of a packet. Find out such a packet. This is to find a packet with flags bit A=1. This provides an evidence that TCP is a reliable protocol. This is different from the UDP protocol. ACK packet might or might not contain the application data. Verify the ACK packet you consider (any of them is ok) to see if it contains application data.

## Answer 2-



### Question 3-

Run Wireshark and access [www.example.net](http://www.example.net) and then close your webpage and stop your

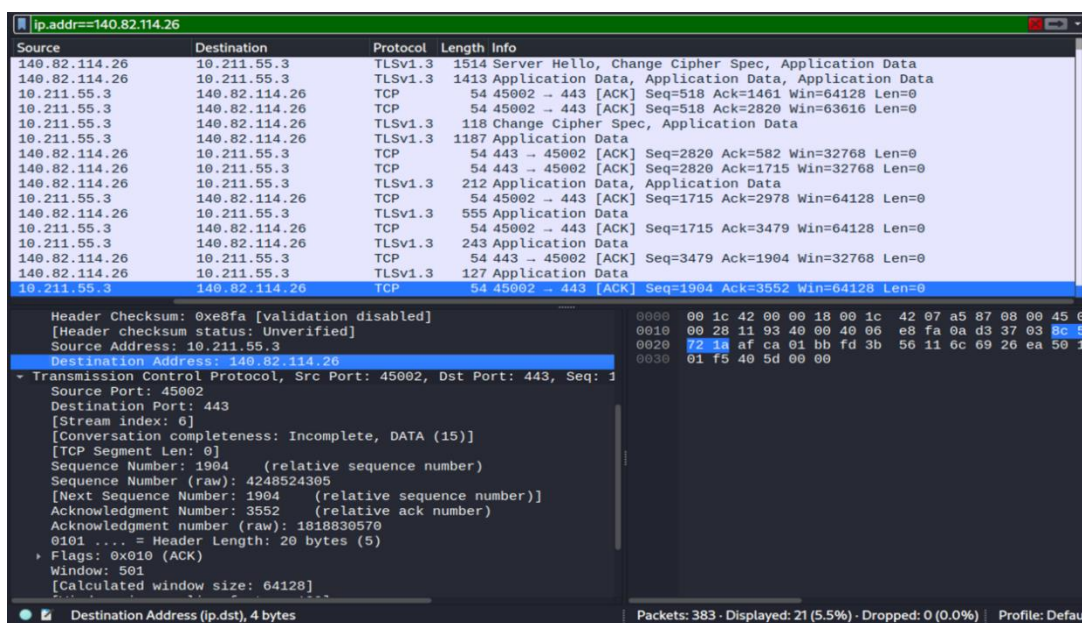
Wireshark. Answer the following questions: -

#### Part A

Find out the first packet from your VM to [www.example.net](http://www.example.net) (you should know the ip address of [www.exampnenet](http://www.exampnenet) now). This should be the SYN-packet (i.e., the first packet of the 3-way handshake protocol). What is source port # and destination port #? Confirm that they are in the TCP header in the Wireshark packet window. What is source IP and destination IP?

Confirm that they are in the ip header in the Wireshark packet window?

#### Answer –

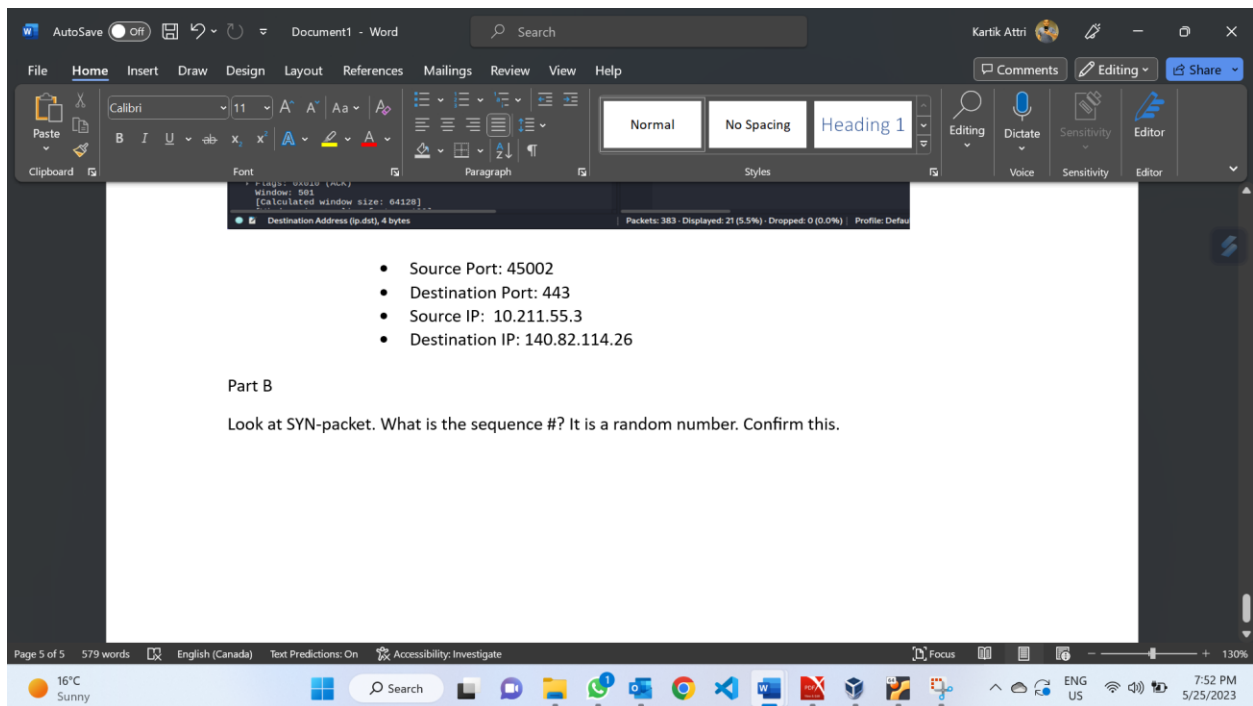


- Source Port: 45002
- Destination Port: 443
- Source IP: 10.211.55.3
- Destination IP: 140.82.114.26

## Part B

Look at SYN-packet. What is the sequence #? It is a random number. Confirm this?

**Answer** - Sequence #: 4248522402



## Part C

Find out in the TCP header the flag bits U|A|P|R|S|F in the SYN-ACK packet

**Answer –**

Flag A = 1



```

ip.addr==140.82.114.26
Source      Destination      Protocol Length Info
10.211.55.3 140.82.114.26   TCP      74 45002 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=39...
140.82.114.26 10.211.55.3     TCP      62 443 → 45002 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460 WS=2
10.211.55.3 140.82.114.26   TCP      54 45002 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0
10.211.55.3 140.82.114.26   TLSv1.3  571 Client Hello
140.82.114.26 10.211.55.3     TCP      54 443 → 45002 [ACK] Seq=1 Ack=518 Win=32768 Len=0
140.82.114.26 10.211.55.3     TLSv1.3  1514 Server Hello, Change Cipher Spec, Application Data
140.82.114.26 10.211.55.3     TLSv1.3  1413 Application Data, Application Data, Application Data
10.211.55.3 140.82.114.26   TCP      54 45002 → 443 [ACK] Seq=518 Ack=1461 Win=64128 Len=0
10.211.55.3 140.82.114.26   TCP      54 45002 → 443 [ACK] Seq=518 Ack=2820 Win=63616 Len=0
10.211.55.3 140.82.114.26   TLSv1.3  118 Change Cipher Spec, Application Data
10.211.55.3 140.82.114.26   TLSv1.3  1187 Application Data
140.82.114.26 10.211.55.3     TCP      54 443 → 45002 [ACK] Seq=2820 Ack=582 Win=32768 Len=0
140.82.114.26 10.211.55.3     TCP      54 443 → 45002 [ACK] Seq=2820 Ack=1715 Win=32768 Len=0
140.82.114.26 10.211.55.3     TLSv1.3  212 Application Data, Application Data
10.211.55.3 140.82.114.26   TCP      54 45002 → 443 [ACK] Seq=1715 Ack=2978 Win=64128 Len=0
140.82.114.26 10.211.55.3     TLSv1.3  555 Application Data

Acknowledgment number (raw): 1818827019
0101 ... = Header Length: 20 bytes (5)
  Flags: 0x010 (ACK)
    000. .... = Reserved: Not set
    ...0 .... = Accurate ECN: Not set
    ....0... = Congestion Window Reduced: Not set
    ....0... = ECN-Echo: Not set
    ....0... = Urgent: Not set
    ....1... = Acknowledgment: Set
    ....0... = Push: Not set
    ....0... = Reset: Not set
    ....0... = Syn: Not set
    ....0... = Fin: Not set
    [TCP Flags: .....A....]
    Window: 502
0000 00 1c 42 00 00 18 00 1c 42 07 a5 87 08 00 45 00
0010 00 28 11 8a 40 00 40 06 e9 03 0a d3 37 03 8c 52
0020 72 1a af ca 01 bb fd 3b 4e a2 6c 69 19 0b 50 10
0030 01 f6 40 5d 00 00

```

## Part D

The receive window field is to tell its partner the current **receive-buffer** size it has. Find out the window size of SYN-ACK packet and that of http response packet. Are they equal?

Answer –

Window Size: 64256

Http Window size: 64256

```

ip.addr==140.82.114.26
Source      Destination      Protocol Length Info
10.211.55.3 140.82.114.26   TCP      74 45002 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=39...
140.82.114.26 10.211.55.3     TCP      62 443 → 45002 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460 WS=2
10.211.55.3 140.82.114.26   TCP      54 45002 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0
10.211.55.3 140.82.114.26   TLSv1.3  571 Client Hello
140.82.114.26 10.211.55.3     TCP      54 443 → 45002 [ACK] Seq=1 Ack=518 Win=32768 Len=0
140.82.114.26 10.211.55.3     TLSv1.3  1514 Server Hello, Change Cipher Spec, Application Data
140.82.114.26 10.211.55.3     TLSv1.3  1413 Application Data, Application Data, Application Data
10.211.55.3 140.82.114.26   TCP      54 45002 → 443 [ACK] Seq=518 Ack=1461 Win=64128 Len=0
10.211.55.3 140.82.114.26   TCP      54 45002 → 443 [ACK] Seq=518 Ack=2820 Win=63616 Len=0
10.211.55.3 140.82.114.26   TLSv1.3  118 Change Cipher Spec, Application Data
10.211.55.3 140.82.114.26   TLSv1.3  1187 Application Data
140.82.114.26 10.211.55.3     TCP      54 443 → 45002 [ACK] Seq=2820 Ack=582 Win=32768 Len=0
140.82.114.26 10.211.55.3     TCP      54 443 → 45002 [ACK] Seq=2820 Ack=1715 Win=32768 Len=0
140.82.114.26 10.211.55.3     TLSv1.3  212 Application Data, Application Data
10.211.55.3 140.82.114.26   TCP      54 45002 → 443 [ACK] Seq=1715 Ack=2978 Win=64128 Len=0
140.82.114.26 10.211.55.3     TLSv1.3  555 Application Data

000. .... = Reserved: Not set
...0 .... = Accurate ECN: Not set
....0... = Congestion Window Reduced: Not set
....0... = ECN-Echo: Not set
....0... = Urgent: Not set
....1... = Acknowledgment: Set
....0... = Push: Not set
....0... = Reset: Not set
....0... = Syn: Not set
....0... = Fin: Not set
    [TCP Flags: .....A....]
    Window: 502
[Calculated window size: 64256]
0000 00 1c 42 00 00 18 00 1c 42 07 a5 87 08 00 00
0010 00 28 11 8a 40 00 40 06 e9 03 0a d3 37 03 8c 52
0020 72 1a af ca 01 bb fd 3b 4e a2 6c 69 19 0b 50
0030 01 f6 40 5d 00 00

```

## Part E

Find out the sequence # of http request packet and its payload size (the **segment len** is the payload size). The next sequence # is the sum of these two numbers. Verify that this is indeed the sequence # of the next packet sent by your VM (I used Different MACHine)?

**Answer** – Sequence Number: + TCP Segment len = next Sequence number

227236893 + 296  
227237189

1069	2023-01-24 14:2..	10.0.2.15	151.101.1.69	TCP	54 38252 → 443 [ACK] Seq=245096432 Ack=9472002 Win=6
1070	2023-01-24 14:2..	10.0.2.15	151.101.1.69	TLSv1.2	571 Client Hello
1071	2023-01-24 14:2..	151.101.1.69	10.0.2.15	TCP	60 443 → 38252 [ACK] Seq=9472002 Ack=245096949 Win=6
1072	2023-01-24 14:2..	151.101.1.69	10.0.2.15	TLSv1.2	1506 Server Hello
1073	2023-01-24 14:2..	10.0.2.15	151.101.1.69	TCP	54 38252 → 443 [ACK] Seq=245096949 Ack=9473454 Win=6
1074	2023-01-24 14:2..	151.101.1.69	10.0.2.15	TCP	1514 443 → 38252 [ACK] Seq=9473454 Ack=245096949 Win=6

Acknowledgment number: 9472002

0101 .... = Header Length: 20 bytes (5)

Flags: 0x010 (ACK)

- 000. .... = Reserved: Not set
- ...0 .... = Nonce: Not set
- ....0... .... = Congestion Window Reduced (CWR): Not set
- ....0... .... = ECN-Echo: Not set
- ....0... .... = Urgent: Not set
- ....1... .... = Acknowledgment: Set
- ....0... .... = Push: Not set
- ....0... .... = Reset: Not set
- ....0... .... = Syn: Not set
- ....0... .... = Fin: Not set

[TCP Flags: .....A....]

Window size value: 64240

[Calculated window size: 64240]

[Window size scaling factor: -2 (no window scaling used)]

Checksum: 0xa4d3 [unverified]

## Part F

Find out the acknowledgement # in http response packet. Is this the same as the next sequence # you calculated above for the request packet? Explain why?

**Answer** -

2023-01-24 16:4..	34.107.221.82	10.0.2.15	HTTP	270 HTTP/1.1 200 OK (text/plain)
2023-01-24 16:4..	10.0.2.15	34.107.221.82	TCP	54 59804 → 80 [ACK] Seq=227237189 Ack=2
2023-01-24 16:4..	192.168.2.1	10.0.2.15	DNS	218 Standard query response 0xbca5 AAAA
2023-01-24 16:4..	192.168.2.1	10.0.2.15	DNS	206 Standard query response 0x14fc A det
2023-01-24 16:4..	10.0.2.15	192.168.2.1	DNS	82 Standard query 0x7981 A example.org
2023-01-24 16:4..	10.0.2.15	192.168.2.1	DNS	82 Standard query 0x5087 AAAA example.o

ation: 10.0.2.15

ssion Control Protocol, Src Port: 80, Dst Port: 59804, Seq: 2880218, Ack: 227237189, Len: 216

e Port: 80

nation Port: 59804

am index: 1]

Segment Len: 216]

nce number: 2880218

sequence number: 2880434]

wledgment number: 227237189

.... = Header Length: 20 bytes (5)

## Part G

What is the flags bits U|A|P|R|S|F in the http response packet?

**Answer** -



740	2023-01-24 16:4	34.107.221.82	10.0.2.15	HTTP	270 HTTP/1.1: 200 OK (text/plain)
741	2023-01-24 16:4	10.0.2.15	34.107.221.82	TCP	54 59804 → 80 [ACK] Seq=227237189 Ack=2880434 Win=64024 Len=0
742	2023-01-24 16:4	192.168.2.1	10.0.2.15	DNS	218 Standard query response 0xbca5 AAAA detectportal.firefox.com
743	2023-01-24 16:4	192.168.2.1	10.0.2.15	DNS	206 Standard query response 0x14fc A detectportal.firefox.com CNA
744	2023-01-24 16:4	10.0.2.15	192.168.2.1	DNS	82 Standard query 0x7981 A example.org OPT
745	2023-01-24 16:4	10.0.2.15	192.168.2.1	DNS	82 Standard query 0x5087 AAAA example.org OPT
Destination: 10.0.2.15					
* Transmission Control Protocol, Src Port: 80, Dst Port: 59804, Seq: 2880218, Ack: 227237189, Len: 216					
Source Port: 80					
Destination Port: 59804					
[Stream index: 1]					
[TCP Segment Len: 216]					
Sequence number: 2880218					
[Next sequence number: 2880434]					
Acknowledgment number: 227237189					
0101 .... = Header Length: 20 bytes (5)					
* Flags: 0x018 (PSH, ACK)					
000. .... = Reserved: Not set					
...0 .... = Nonce: Not set					
....0... .... = Congestion Window Reduced (CWR): Not set					
....0... .... = ECN-Echo: Not set					
....0... .... = Urgent: Not set					
....1... .... = Acknowledgment: Set					
....1... .... = Push: Set					
....0... .... = Reset: Not set					
....0... .... = Syn: Not set					
....0... .... = Fin: Not set					
[TCP Flags: .....AP...]					

## Part H

Find out the packet your VM requests to terminate the TCP connection. This packet will be sent when you close the webpage. What is the flags bit U|A|P|R|S|F in this packet?

Answer -

740	2023-01-24 16:4	34.107.221.82	10.0.2.15	HTTP	270 HTTP/1.1: 200 OK (text/plain)
741	2023-01-24 16:4	10.0.2.15	34.107.221.82	TCP	54 59804 → 80 [ACK] Seq=227237189 Ack=2880434 Win=64024 Len=0
742	2023-01-24 16:4	192.168.2.1	10.0.2.15	DNS	218 Standard query response 0xbca5 AAAA detectportal.firefox.com
743	2023-01-24 16:4	192.168.2.1	10.0.2.15	DNS	206 Standard query response 0x14fc A detectportal.firefox.com CNA
744	2023-01-24 16:4	10.0.2.15	192.168.2.1	DNS	82 Standard query 0x7981 A example.org OPT
745	2023-01-24 16:4	10.0.2.15	192.168.2.1	DNS	82 Standard query 0x5087 AAAA example.org OPT
Destination: 10.0.2.15					
* Transmission Control Protocol, Src Port: 80, Dst Port: 59804, Seq: 2880218, Ack: 227237189, Len: 216					
Source Port: 80					
Destination Port: 59804					
[Stream index: 1]					
[TCP Segment Len: 216]					
Sequence number: 2880218					
[Next sequence number: 2880434]					
Acknowledgment number: 227237189					
0101 .... = Header Length: 20 bytes (5)					
* Flags: 0x018 (PSH, ACK)					
000. .... = Reserved: Not set					
...0 .... = Nonce: Not set					
....0... .... = Congestion Window Reduced (CWR): Not set					
....0... .... = ECN-Echo: Not set					
....0... .... = Urgent: Not set					
....1... .... = Acknowledgment: Set					
....1... .... = Push: Set					
....0... .... = Reset: Not set					
....0... .... = Syn: Not set					
....0... .... = Fin: Not set					
[TCP Flags: .....AP...]					