ASSIGNMENT - I

Team - Sachin Jalan(21110183) Anushk Bhana (21110031)

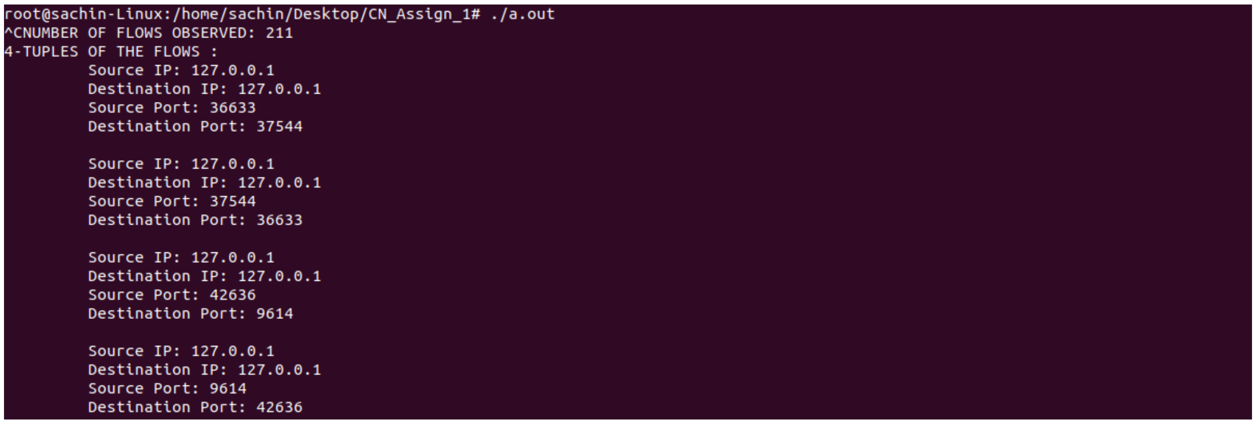
Answer 1:

‘In question 1 we had to open a raw socket and sniff all the packets going through the computer, to solve this problem a raw socket was implemented in C and the 4 tuples client IP, Server IP, Client port, Server Port are being printed in the logs.txt file, also in the question was asked about the number of TCP flows which are the number of distinct 4 tuples which can be found out using set data structure in C++.

The raw socket is opened by using socket.h header file, the program extracts the information from the headers of the packets and prints it in the logsq1.txt.

Some observations recorded were that although the machine was not connected to the internet then also some packets were observed, this is because the machine tries some background process always but as the internet was not there it didn’t get any response.

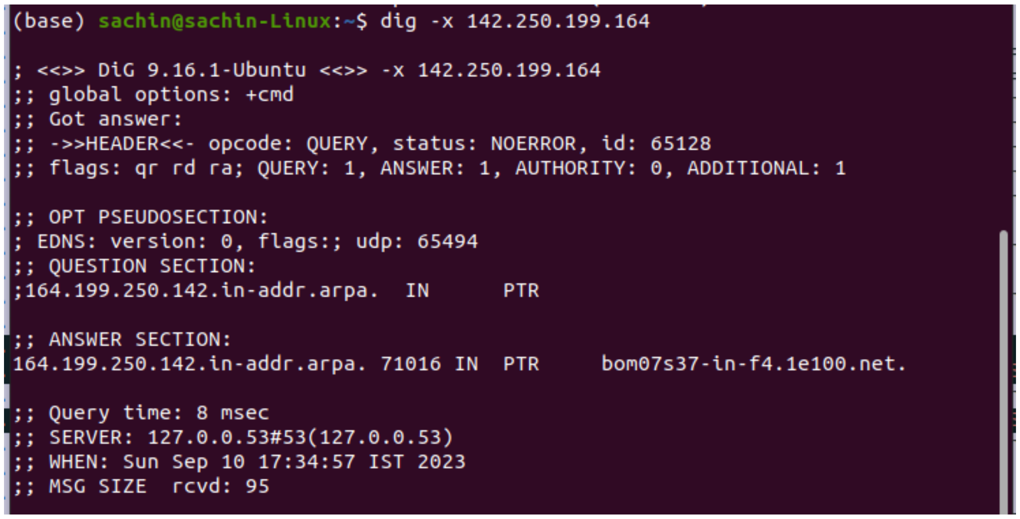
Another observation was that there were multiple logs having same Tcp stream this is because multiple packets of data is being passed through that same stream.



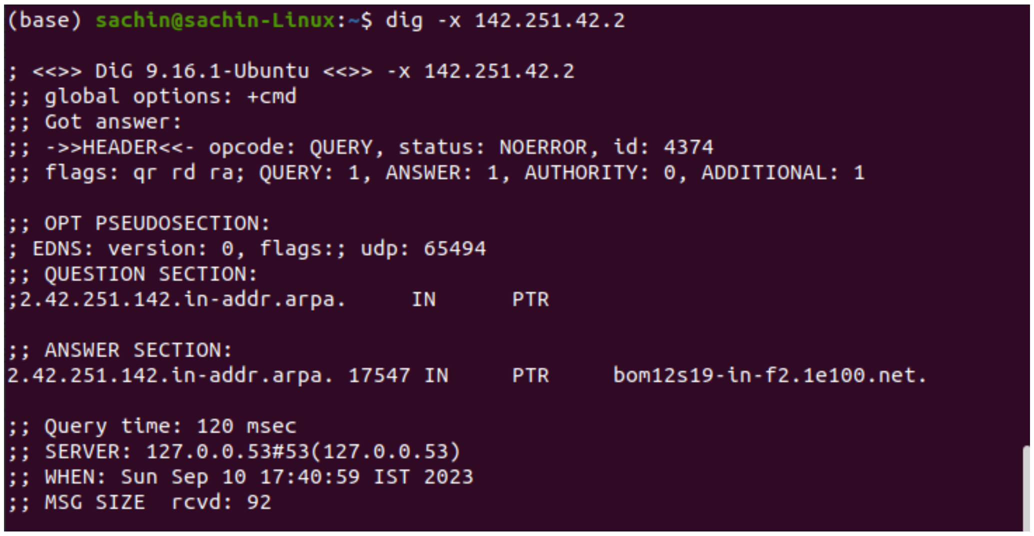
```
root@sachin-Linux:/home/sachin/Desktop/CN_Assign_1# ./a.out
^CNUMBER OF FLOWS OBSERVED: 211
4-TUPLES OF THE FLOWS :
        Source IP: 127.0.0.1
        Destination IP: 127.0.0.1
        Source Port: 36633
        Destination Port: 37544

        Source IP: 127.0.0.1
        Destination IP: 127.0.0.1
        Source Port: 37544
        Destination Port: 36633

        Source IP: 127.0.0.1
        Destination IP: 127.0.0.1
        Source Port: 42636
        Destination Port: 9614

        Source IP: 127.0.0.1
        Destination IP: 127.0.0.1
        Source Port: 9614
        Destination Port: 42636
```

Fig: Internet Off packet observed

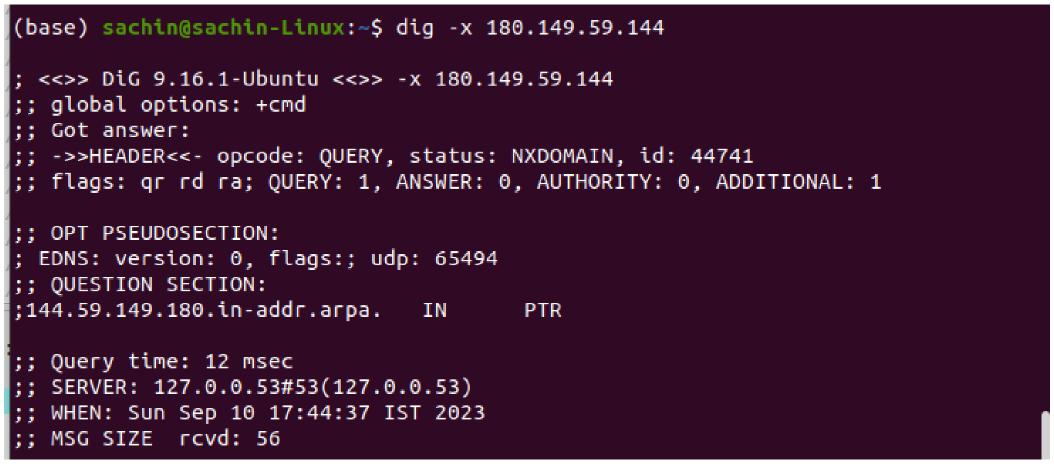
Following are the Reverse DNS Lookup for 5 IPs. For reverse DNS lookup dig command is used which outputs the following where the dns is the following line after PTR.



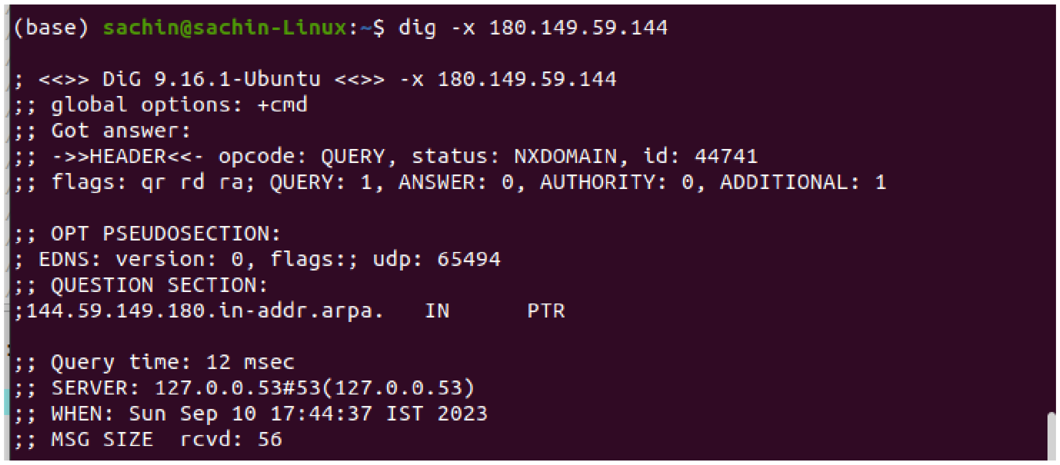
```
(base) sachin@sachin-Linux:~$ dig -x 142.250.199.164

; <<>> DiG 9.16.1-Ubuntu <<>> -x 142.250.199.164
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65128
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;164.199.250.142.in-addr.arpa.  IN      PTR

;; ANSWER SECTION:
164.199.250.142.in-addr.arpa. 71016 IN  PTR     bom07s37-in-f4.1e100.net.

;; Query time: 8 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sun Sep 10 17:34:57 IST 2023
;; MSG SIZE  rcvd: 95
```



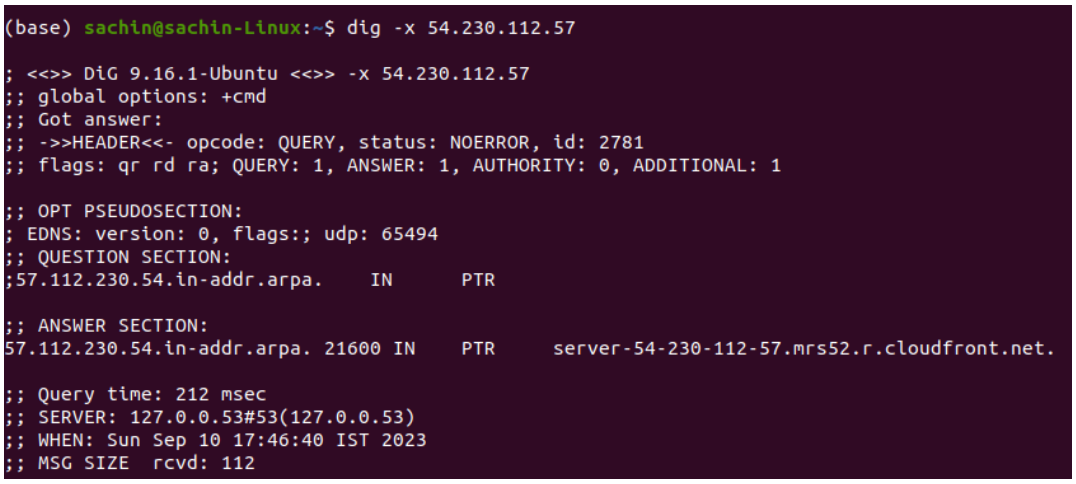
```
(base) sachin@sachin-Linux:~$ dig -x 142.251.42.2

; <<>> DiG 9.16.1-Ubuntu <<>> -x 142.251.42.2
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4374
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;2.42.251.142.in-addr.arpa.     IN      PTR

;; ANSWER SECTION:
2.42.251.142.in-addr.arpa. 17547 IN     PTR     bom12s19-in-f2.1e100.net.

;; Query time: 120 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sun Sep 10 17:40:59 IST 2023
;; MSG SIZE  rcvd: 92
```



```
(base) sachin@sachin-Linux:~$ dig -x 180.149.59.144

; <<>> DiG 9.16.1-Ubuntu <<>> -x 180.149.59.144
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 44741
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;144.59.149.180.in-addr.arpa.   IN      PTR

;; Query time: 12 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sun Sep 10 17:44:37 IST 2023
;; MSG SIZE  rcvd: 56
```



```
(base) sachin@sachin-Linux:~$ dig -x 180.149.59.144

; <<>> DiG 9.16.1-Ubuntu <<>> -x 180.149.59.144
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 44741
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;144.59.149.180.in-addr.arpa.   IN      PTR

;; Query time: 12 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sun Sep 10 17:44:37 IST 2023
;; MSG SIZE  rcvd: 56
```



```
(base) sachin@sachin-Linux:~$ dig -x 54.230.112.57

; <<>> DiG 9.16.1-Ubuntu <<>> -x 54.230.112.57
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2781
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;57.112.230.54.in-addr.arpa.    IN      PTR

;; ANSWER SECTION:
57.112.230.54.in-addr.arpa. 21600 IN    PTR     server-54-230-112-57.mrs52.r.cloudfront.net.

;; Query time: 212 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sun Sep 10 17:46:40 IST 2023
;; MSG SIZE  rcvd: 112
```

Answer 2

In Problem 2 we had to do CTF, there we had to find the flags hidden in the data packets. The approach for this problem is as follows.
First we ran the packet sniffer for the pcap file and printed all the logs in a log.txt file. Then according to each flag code was written in Python to process the logs text file and extract the flags. For matching all the flags I used Regular Expression matching.
For question 1 the hint was keyword Flag. so upon searching for 'Flag:' we got the packet containing the flag.
For question 2 the hint was username, thus we searched for the word 'username' using regex found the packet containing the flag.
For question 3 the hint was TCP checksum and instructions in path, so we inferred that the source from where the packet was coming will be sending some packet with the flag, so first we matched for the TCP Checksum, then we extracted the Source IP of that packet, then we iterated for the entire logs and filtered using regex the packets having this IP, then we filtered with word PASS for password and then found the flag.
For question 4 the hint was some of connection port of a packet with the given IP will lead to a person, we inferred that some source port would be there with that sum. So we first extracted

the source port and destination port using regex, then we summed it up, then we searched for the source port with the calculated value using regex then we found the packet with the flag.
For question 5 hint keyword was milkshake hence we searched for the word milkshake using regex and then found the packet with the flag.

Answer 3

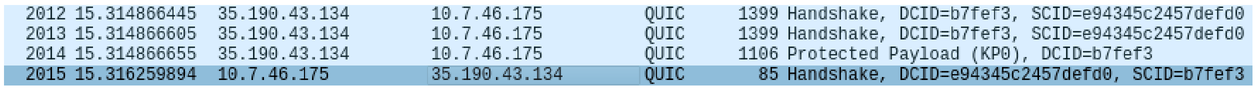
In this question we had to store the process id for the socket which is sniffing and ask the user for the socket and print the pid of the process that was executed by that port.
For this question we used the ss command of linux. We used ss -nap, -n argument specifies to resolve the port id to numerical, -a to display all ports listening and established for TCP, -p argument is to display the process information of that socket. Then we used a series of commands to filter the data, we used grep commands to extract the pid of the port_id which is as argument. We used a pipeopen function popen() to get the output of these commands in a variable in our code. We created the function port_to_pid() and stored the pids of all the process created during the 30 second the sniffer was running. We stored the pids in a map and then the user is prompted to enter a socket number and when the program receives the input socket number it outputs the PID stored . For instructions to run refer README.md.

Answer 4

<u>Five protocols not studied</u>

ARP- Address resolution protocol, connects an IP to a machine address. (RFC 6747)
QUIC-Transport Layer, same as TCP to reduce latency, (RFC 9000)
TLSv1.3-Transport layer security protocol, between application and transport, (RFC 8446)
ICMP-a protocol that devices within a network use to communicate problems with data transmission, Layer 3 in ISO model, (RFC 792)
UDP-Transport layer protocol, does not require handshake to connect, unreliable. (RFC 768)

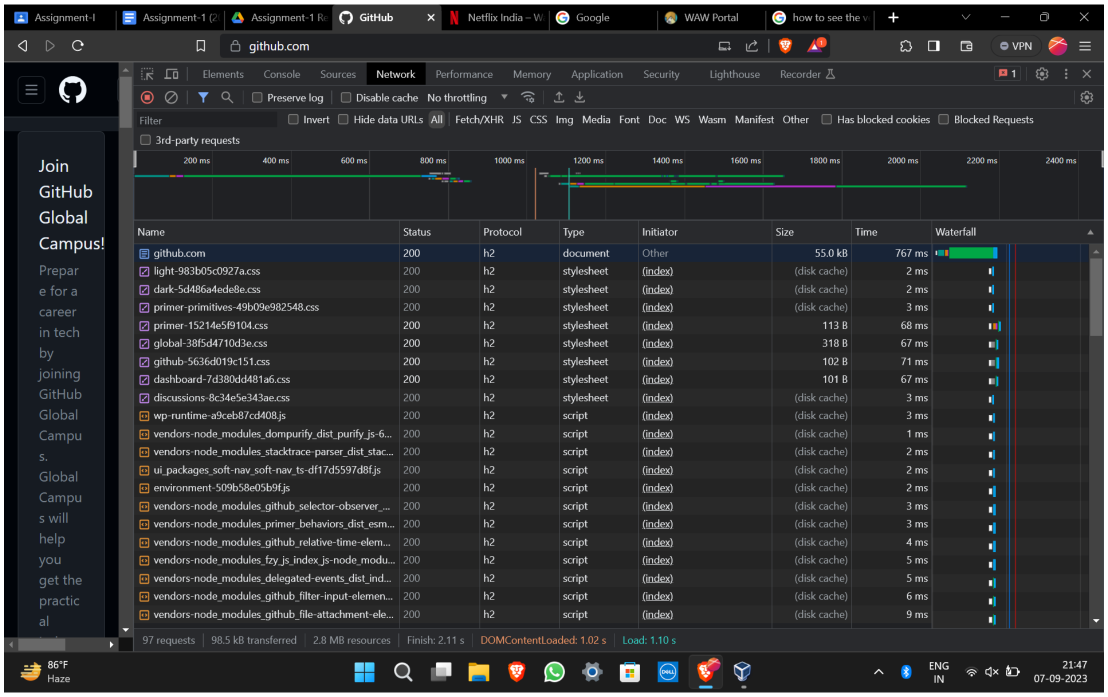
B.<u>Estimating RTT for the packet</u>



```
2012 15.314866445  35.190.43.134   10.7.46.175     QUIC   1399 Handshake, DCID=b7fef3, SCID=e94345c2457defd0
2013 15.314866605  35.190.43.134   10.7.46.175     QUIC   1399 Handshake, DCID=b7fef3, SCID=e94345c2457defd0
2014 15.314866655  35.190.43.134   10.7.46.175     QUIC   1106 Protected Payload (KP0), DCID=b7fef3
2015 15.316259894  10.7.46.175     35.190.43.134   QUIC     85 Handshake, DCID=e94345c2457defd0, SCID=b7fef3
```

Estimated RTT = 15.316259894-15.314866445 = 0.001393449 seconds
2.

github.com → HTTP/2

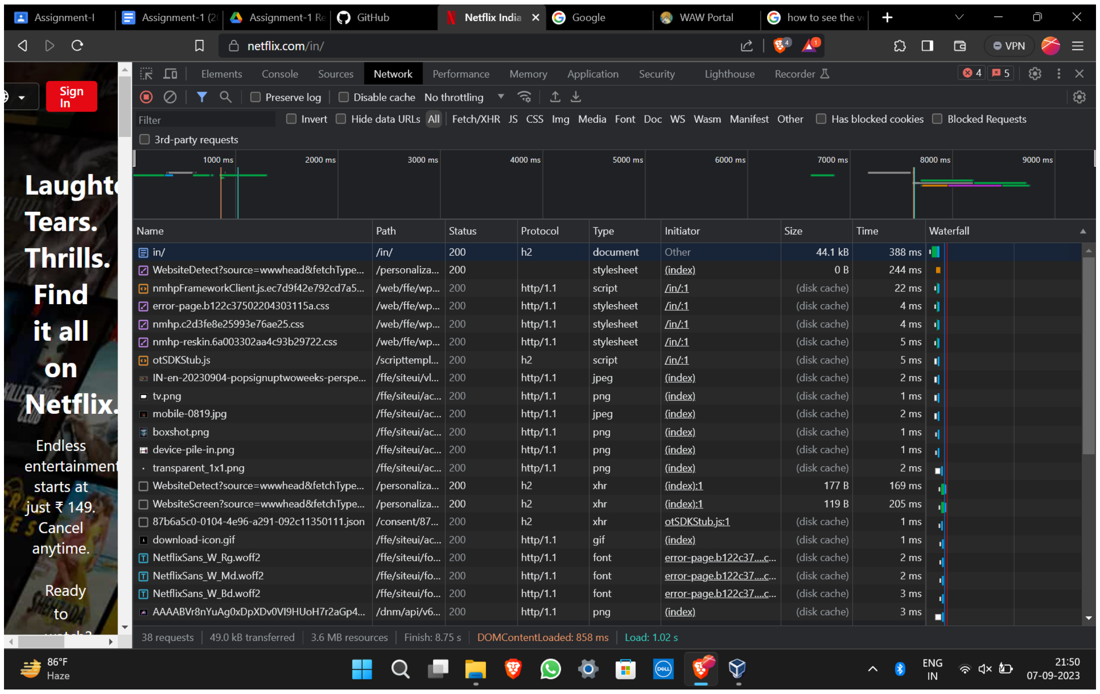


github.com

Name	Status	Protocol	Type	Initiator	Size	Time
github.com	200	h2	document	Other	55.0 kB	767 ms
light-983b05c0927a.css	200	h2	stylesheet	(index)	(disk cache)	2 ms
dark-5d486a4ede8e.css	200	h2	stylesheet	(index)	(disk cache)	2 ms
primer-primitives-49b09e982548.css	200	h2	stylesheet	(index)	(disk cache)	3 ms
primer-15214e5f9104.css	200	h2	stylesheet	(index)	113 B	68 ms
global-38f5d4710d3e.css	200	h2	stylesheet	(index)	318 B	67 ms
github-5636d019c151.css	200	h2	stylesheet	(index)	102 B	71 ms
dashboard-7d380dd481a6.css	200	h2	stylesheet	(index)	101 B	67 ms
discussions-8c34e5e343ae.css	200	h2	stylesheet	(index)	(disk cache)	3 ms
wp-runtime-a9ceb87cd408.js	200	h2	script	(index)	(disk cache)	3 ms
vendors-node_modules_dompurify_dist_purify_js-6...	200	h2	script	(index)	(disk cache)	1 ms
vendors-node_modules_stacktrace-parser_dist_stac...	200	h2	script	(index)	(disk cache)	2 ms
ui_packages_soft-nav_soft-nav_ts-df17d5597d8f.js	200	h2	script	(index)	(disk cache)	2 ms
environment-509b58e05b9f.js	200	h2	script	(index)	(disk cache)	2 ms
vendors-node_modules_github_selector-observer_...	200	h2	script	(index)	(disk cache)	3 ms
vendors-node_modules_primer_behaviors_dist_esm...	200	h2	script	(index)	(disk cache)	3 ms
vendors-node_modules_github_relative-time-elem...	200	h2	script	(index)	(disk cache)	4 ms
vendors-node_modules_fzy_js_index_js-node_modu...	200	h2	script	(index)	(disk cache)	5 ms
vendors-node_modules_delegated-events_dist_ind...	200	h2	script	(index)	(disk cache)	5 ms
vendors-node_modules_github_filter-input-elemen...	200	h2	script	(index)	(disk cache)	6 ms
vendors-node_modules_github_file-attachment-ele...	200	h2	script	(index)	(disk cache)	9 ms

97 requests | 98.5 kB transferred | 2.8 MB resources | Finish: 2.11 s | DOMContentLoaded: 1.02 s | Load: 1.10 s

netflix.com → HTTP/2

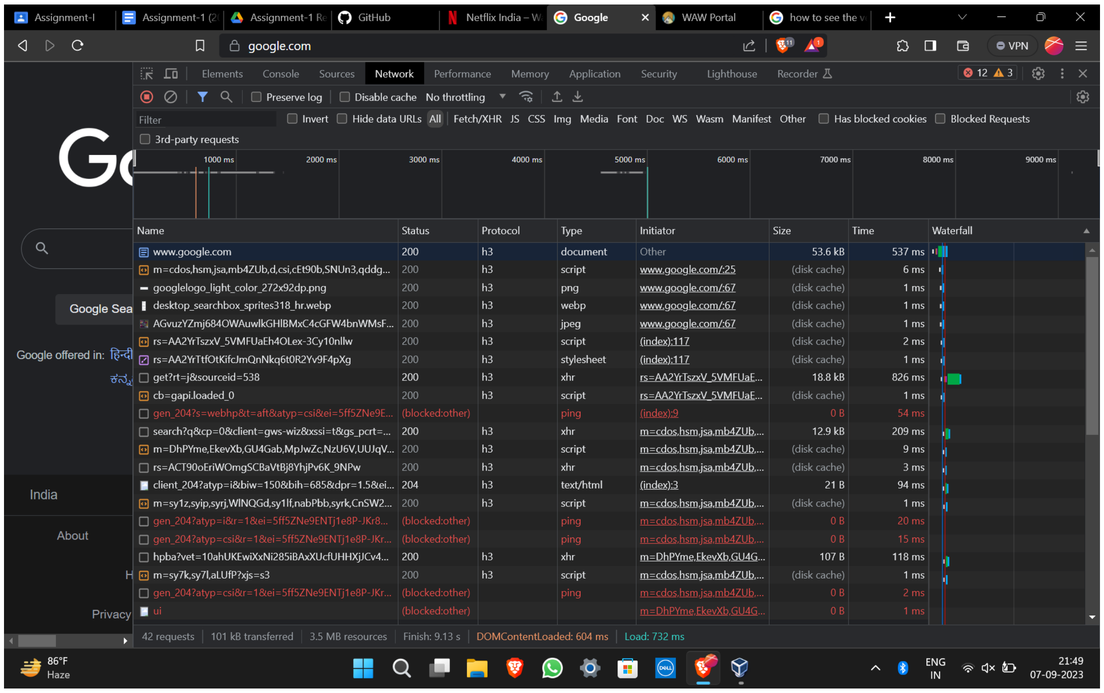


netflix.com/in/

Name	Path	Status	Protocol	Type	Initiator	Size	Time
in/	/in/	200	h2	document	Other	44.1 kB	388 ms
WebsiteDetect?source=wwwhead&fetchType...	/personaliza...	200		stylesheet	(index)	0 B	244 ms
nmhpFrameworkClient.js.ec7d9f42e792cd7a5...	/web/ffe/wp...	200	http/1.1	script	/in/:1	(disk cache)	22 ms
error-page.b122c37502204303115a.css	/web/ffe/wp...	200	http/1.1	stylesheet	/in/:1	(disk cache)	4 ms
nmhp.c2d3fe8e25993e76ae25.css	/web/ffe/wp...	200	http/1.1	stylesheet	/in/:1	(disk cache)	4 ms
nmhp-reskin.6a003302aa4c93b29722.css	/web/ffe/wp...	200	http/1.1	stylesheet	/in/:1	(disk cache)	5 ms
otSDKStub.js	/scripttempl...	200	h2	script	/in/:1	(disk cache)	5 ms
IN-en-20230904-popsignuptwoweeks-perspe...	/ffe/siteui/vl...	200	http/1.1	jpeg	(index)	(disk cache)	2 ms
tv.png	/ffe/siteui/ac...	200	http/1.1	png	(index)	(disk cache)	1 ms
mobile-0819.jpg	/ffe/siteui/ac...	200	http/1.1	jpeg	(index)	(disk cache)	2 ms
boxshot.png	/ffe/siteui/ac...	200	http/1.1	png	(index)	(disk cache)	1 ms
device-pile-in.png	/ffe/siteui/ac...	200	http/1.1	png	(index)	(disk cache)	1 ms
transparent_1x1.png	/ffe/siteui/ac...	200	http/1.1	png	(index)	(disk cache)	2 ms
WebsiteDetect?source=wwwhead&fetchType...	/personaliza...	200	h2	xhr	(index):1	177 B	169 ms
WebsiteScreen?source=wwwhead&fetchType...	/personaliza...	200	h2	xhr	(index):1	119 B	205 ms
87b6a5c0-0104-4e96-a291-092c11350111.json	/consent/87...	200	h2	xhr	otSDKStub.js:1	(disk cache)	1 ms
download-icon.gif	/ffe/siteui/ac...	200	http/1.1	gif	(index)	(disk cache)	1 ms
NetflixSans_W_Rg.woff2	/ffe/siteui/fo...	200	http/1.1	font	error-page.b122c37....c...	(disk cache)	2 ms
NetflixSans_W_Md.woff2	/ffe/siteui/fo...	200	http/1.1	font	error-page.b122c37....c...	(disk cache)	2 ms
NetflixSans_W_Bd.woff2	/ffe/siteui/fo...	200	http/1.1	font	error-page.b122c37....c...	(disk cache)	3 ms
AAAABVr8nYuAg0xDpXDv0VI9HUoH7r2aGp4...	/dnm/api/v6...	200	http/1.1	png	(index)	(disk cache)	3 ms

38 requests | 49.0 kB transferred | 3.6 MB resources | Finish: 8.75 s | DOMContentLoaded: 858 ms | Load: 1.02 s

Google.com → HTTP/3



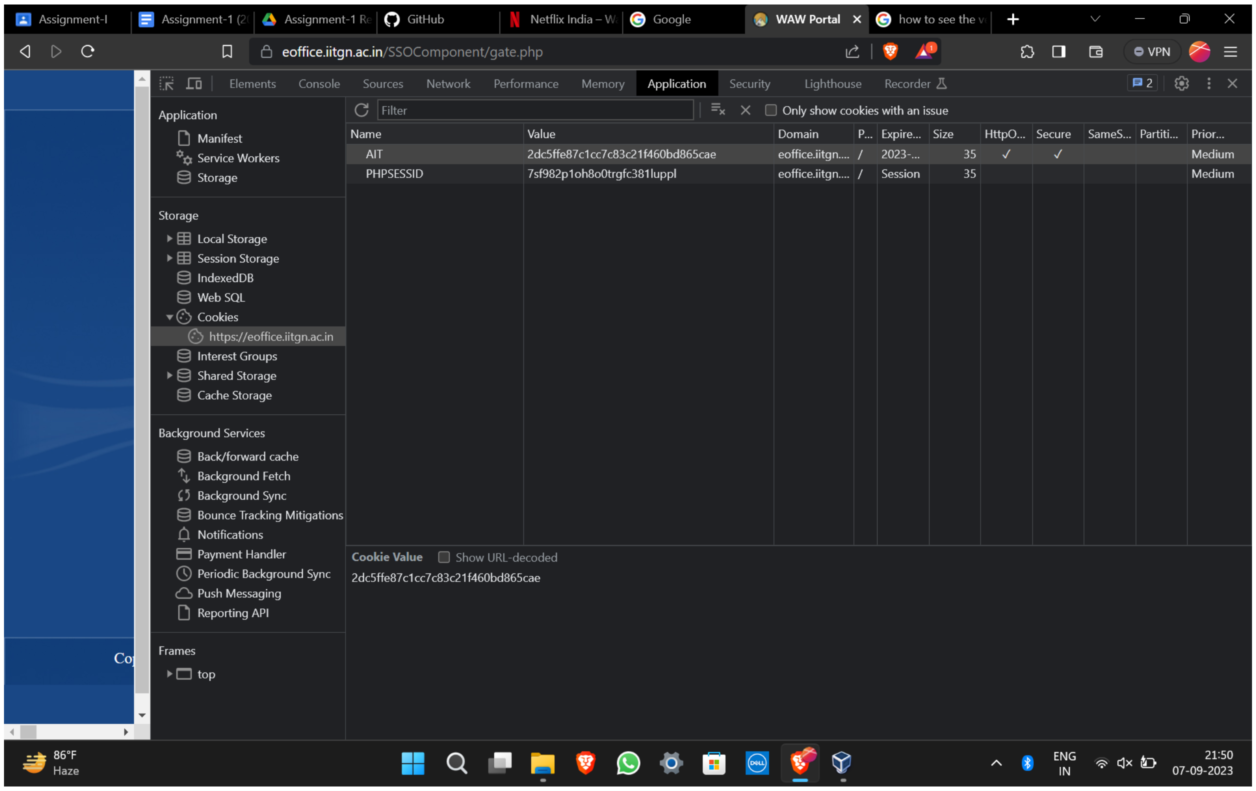
Name	Status	Protocol	Type	Initiator	Size	Time
www.google.com	200	h3	document	Other	53.6 kB	537 ms
m=cdos,hsm,jsa,mb4ZUb,d,csi,cEt90b,SNUn3,qddg...	200	h3	script	www.google.com/:25	(disk cache)	6 ms
googlelogo_light_color_272x92dp.png	200	h3	png	www.google.com/:67	(disk cache)	1 ms
desktop_searchbox_sprites318_hr.webp	200	h3	webp	www.google.com/:67	(disk cache)	1 ms
AGvuzYZmj684OWAuwlkGHlBMxC4cGFW4bnWMsF...	200	h3	jpeg	www.google.com/:67	(disk cache)	1 ms
rs=AA2YrTszxV_5VMFUaEh4OLex-3Cy10nllw	200	h3	script	(index):117	(disk cache)	2 ms
rs=AA2YrTtfOtKifcJmQnNkq6t0R2Yv9F4pXg	200	h3	stylesheet	(index):117	(disk cache)	1 ms
get?rt=j&sourceid=538	200	h3	xhr	rs=AA2YrTszxV_5VMFUaE...	18.8 kB	826 ms
cb=gapi.loaded_0	200	h3	script	rs=AA2YrTszxV_5VMFUaE...	(disk cache)	1 ms
gen_204?s=webhp&t=aft&atyp=csi&ei=5ff5ZNe9E...	(blocked:other)		ping	(index):9	0 B	54 ms
search?q&cp=0&client=gws-wiz&xssi=t&gs_pcrt=...	200	h3	xhr	m=cdos,hsm,jsa,mb4ZUb,...	12.9 kB	209 ms
m=DhPYme,EkevXb,GU4Gab,MpJwZc,NzU6V,UUJqV...	200	h3	script	m=cdos,hsm,jsa,mb4ZUb,...	(disk cache)	9 ms
rs=ACT90oEriWOmgSCBaVtBj8YhjPv6K_9NPw	200	h3	xhr	m=cdos,hsm,jsa,mb4ZUb,...	(disk cache)	3 ms
client_204?atyp=i&biw=150&bih=685&dpr=1.5&ei...	204	h3	text/html	(index):3	21 B	94 ms
m=sy1z,syip,syrj,WlNQGd,sy1lf,nabPbb,syrk,CnSW2...	200	h3	script	m=cdos,hsm,jsa,mb4ZUb,...	(disk cache)	1 ms
gen_204?atyp=i&r=1&ei=5ff5ZNe9ENTj1e8P-JKr8...	(blocked:other)		ping	m=cdos,hsm,jsa,mb4ZUb,...	0 B	20 ms
gen_204?atyp=csi&r=1&ei=5ff5ZNe9ENTj1e8P-JKr...	(blocked:other)		ping	m=cdos,hsm,jsa,mb4ZUb,...	0 B	15 ms
hpba?vet=10ahUKEwiXxNi285iBAxXUcfUHHXjJCv4...	200	h3	xhr	m=DhPYme,EkevXb,GU4G...	107 B	118 ms
m=sy7k,sy7l,aLUfP?xjs=s3	200	h3	script	m=cdos,hsm,jsa,mb4ZUb,...	(disk cache)	1 ms
gen_204?atyp=csi&r=1&ei=5ff5ZNe9ENTj1e8P-JKr...	(blocked:other)		ping	m=cdos,hsm,jsa,mb4ZUb,...	0 B	2 ms
ui	(blocked:other)			m=DhPYme,EkevXb,GU4G...	0 B	1 ms

42 requests | 101 kB transferred | 3.5 MB resources | Finish: 9.13 s | DOMContentLoaded: 604 ms | Load: 732 ms

Differences between HTTP/2 and HTTP/3 →

HTTP/2 and HTTP/3 are two generations of HTTP protocols designed to enhance web performance. HTTP/2 relies on TCP and offers multiplexing for concurrent data transfer but can suffer from head-of-line blocking issues. In contrast, HTTP/3 uses the QUIC transport protocol over UDP, improving performance and reliability, especially on unreliable networks. HTTP/3 introduces better header compression, independent stream error handling, and enforces encryption through TLS. These changes make HTTP/3 a more efficient, secure, and adaptable protocol for modern web communication, addressing many limitations of HTTP/2.
Furthermore, HTTP/3's QUIC-based architecture provides faster connection setup, reducing latency and minimizing performance bottlenecks. This protocol's adaptability to network conditions makes it well-suited for various environments, including high-latency or packet-loss-prone networks. Overall, HTTP/3 represents a significant step forward in optimizing web communication, enhancing the user experience, and bolstering security through its default encryption requirement. As the web continues to evolve, HTTP/3 is poised to play a critical role in shaping the future of online communication and content delivery.

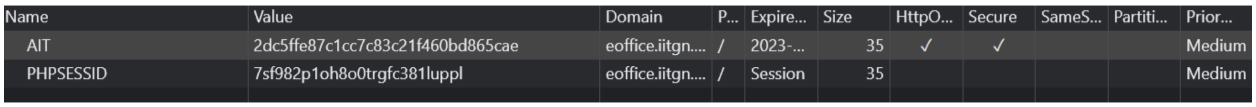
3.



Name	Value	Domain	P...	Expire...	Size	HttpO...	Secure	SameS...	Partiti...	Prior...
AIT	2dc5ffe87c1cc7c83c21f460bd865cae	eoffice.iitgn....	/	2023-...	35	✓	✓			Medium
PHPSESSID	7sf982p1oh8o0trgfc381luppl	eoffice.iitgn....	/	Session	35					Medium

Cookie Value ☐ Show URL-decoded
2dc5ffe87c1cc7c83c21f460bd865cae

The characteristics of these cookies are shown in the following table →



Name	Value	Domain	P...	Expire...	Size	HttpO...	Secure	SameS...	Partiti...	Prior...
AIT	2dc5ffe87c1cc7c83c21f460bd865cae	eoffice.iitgn....	/	2023-...	35	✓	✓			Medium
PHPSESSID	7sf982p1oh8o0trgfc381luppl	eoffice.iitgn....	/	Session	35					Medium

References:

https://www.binarytides.com/packet-sniffer-code-c-libpcap-linux-sockets/ [For the print Data Function]

https://www.opensourceforu.com/2015/03/a-guide-to-using-raw-sockets/ [For the socket opening method]

Geek for Geeks

Various man pages in linux (popen, ss, lsof, netstat etc)