

1.

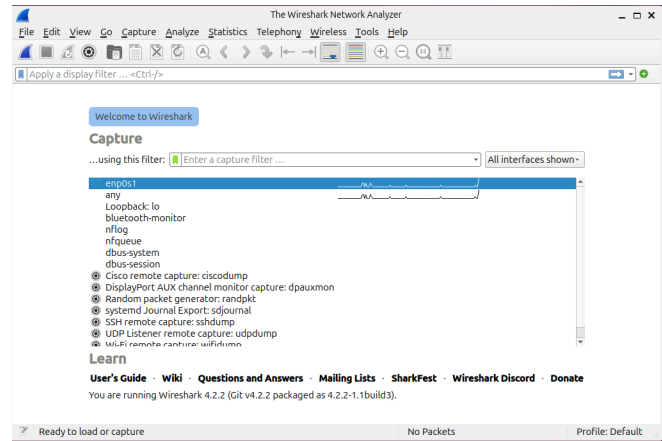
```

) ...
Setting up libqt6waylandeglclientintegration6:arm64 (6.4.2-5build3) ...
Setting up qt6-wayland:arm64 (6.4.2-5build3) ...
Processing triggers for ncolur-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for shared-mime-info (2.4-4) ...
Processing triggers for desktop-file-utils (0.27-2build1) ...
gabriell@gabriell-QEMU-Virtual-Machine:~/hell$ sudo usermod -aG wireshark $USER
newgrp wireshark
gabriell@gabriell-QEMU-Virtual-Machine:~/hell$ wireshark --version
Wireshark 4.2.2 (Git v4.2.2 packaged as 4.2.2-1.1build3).

Copyright 1998-2024 Gerald Combs <gerald@wireshark.org> and contributors.
Licensed under the terms of the GNU General Public License (version 2 or later).
This is free software; see the file named COPYING in the distribution. There is
NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Compiled (64-bit) using GCC 13.2.0, with Glib 2.80.0, with Qt 6.4.2, with
libpcap, with POSIX capabilities (Linux), with libnl 3, with zlib 1.3, with
PCRE2, with Lua 5.2.4, with GnuTLS 3.8.3 and PKCS #11 support, with Gcrypt
1.10.3, with Kerberos (MIT), with MaxMind, with nghtcp 1.59.0, with nghtcp3
0.8.0, with brotli, with LZ4, with Zstandard, with snappy, with libbrotli 2.9.14,
with libsnm 0.4.8, with QMulltimedia, without automatic updates, with Minizip,
with binary plugins.

Running on Linux 6.14.0-33-generic, with 7917 MB of physical memory, with Glib
2.80.0, with Qt 6.4.2, with libpcap 1.10.4 (with TPACKET_V3), with zlib 1.3,
with PCRE2 10.42.2022-12-11, with c-ares 1.27.0, with GnuTLS 3.8.3, with Gcrypt
1.10.3, with nghtcp2 1.59.0, with nghtcp3 0.8.0, with brotli 1.1.0, with LZ4
1.9.4, with Zstandard 1.5.5, with libsnm 0.4.8, with LC_TYPE=en_US.UTF-8, binary
plugins supported.
gabriell@gabriell-QEMU-Virtual-Machine:~/hell$
    
```

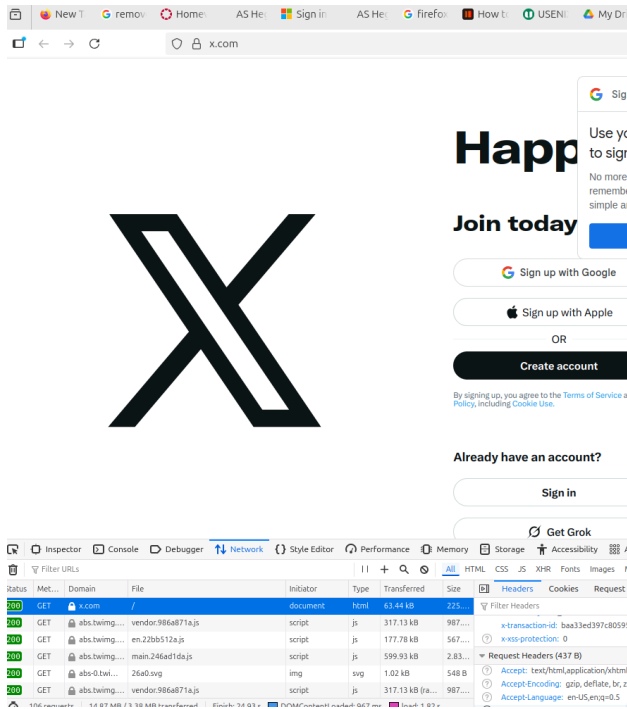


2.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.64.1	192.168.64.255	DB-LSP...	270	Dropbox LAN sync Discovery Protoc
2	2.141236486	192.168.64.5	192.168.64.1	DNS	100	Standard query 0x419a AAAA connec
3	2.165062471	86:2f:57:14:a9:64	6a:23:79:b0:8a:fc	ARP	42	Who has 192.168.64.5? Tell 192.16
4	2.165087138	6a:23:79:b0:8a:fc	86:2f:57:14:a9:64	ARP	42	192.168.64.5 is at 6a:23:79:b0:8a
5	2.165063096	192.168.64.1	192.168.64.5	DNS	436	Standard query response 0x419a AA
6	4.674055571	fe80::842f:57ff:fe1...	ff02::1	ICMPv6	142	Router Advertisement from 86:2f:5
7	7.403350791	6a:23:79:b0:8a:fc	86:2f:57:14:a9:64	ARP	42	Who has 192.168.64.1? Tell 192.16
8	7.403890493	86:2f:57:14:a9:64	6a:23:79:b0:8a:fc	ARP	42	192.168.64.1 is at 86:2f:57:14:a9
9	30.065956629	192.168.64.1	192.168.64.255	DB-LSP...	270	Dropbox LAN sync Discovery Protoc

Frame 1: 270 bytes on wire (2160 bits), 270 bytes captured on interface enp0s1, 270 bytes from 192.168.64.1 to 192.168.64.255 on interface enp0s1
 Ethernet II, Src: 86:2f:57:14:a9:64 (86:2f:57:14:a9:64), Dst: 01:00:5e:00:00:00 (01:00:5e:00:00:00)
 Internet Protocol Version 4, Src: 192.168.64.1, Dst: 192.168.64.255
 User Datagram Protocol, Src Port: 17500, Dst Port: 17500
 Dropbox LAN sync Discovery Protocol

3.
1(a) Accept-Language: en-US,en;q=0.5



b-g are found the exact same way
 1(b) Accept-Language: en-US,en;q=0.5
 1(c) Accept-Language: en-US,en;q=0.5
 1(d) Accept-Language: en-US,en;q=0.5
 1(e) Accept-Language: en-US,en;q=0.5
 1(f) Accept-Language: en-US,en;q=0.5
 1(g) Accept-Language: en-US,en;q=0.5

Question 2 is with 4 and 5 for each one, as they can be answered from individual Wireshark screenshots

- 3(a) 200
- 3(b) 200
- 3(c) 200
- 3(d) 200
- 3(e) 200
- 3(f) 200

a-f are found the exact same way as (g)

The screenshot shows a web browser displaying a PDF document. The browser's developer tools are open to the Network tab, showing a request for 'romain_sigcomm2.pdf' with a status of '304 Not Modified'. The document title is 'AS Hegemony: A Robust Metric for AS Centrality'.

3(g) 304 Not Modified

The screenshot shows a network traffic capture tool displaying a sequence of events for an HTTPS session. The capture shows packets 481 through 503, including the complete setup for x.com (SNI: twitter.com, IP 172.66.0.227).

No.	Time	Source	Destination	Protocol	Length	Info
480	11.599715966	192.168.64.5	172.66.0.227	TCP	74	47844 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3293111328 TSecr=0 WS=128
481	11.599777293	192.168.64.5	172.66.0.227	TCP	74	47858 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3293111328 TSecr=0 WS=128
482	11.631967922	172.66.0.227	192.168.64.5	TCP	74	443 → 47858 [ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1490 SACK_PERM TSval=3315707245 TSecr=3...
483	11.631994838	192.168.64.5	172.66.0.227	TCP	66	47858 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3293111360 TSecr=3315707245
484	11.632398903	192.168.64.5	172.66.0.227	TCP	1454	47858 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=1388 TSval=3293111361 TSecr=3315707245 [TCP segmen...
485	11.632399044	192.168.64.5	172.66.0.227	TLSv1.3	574	Client Hello (SNI=twitter.com)
486	11.632799541	172.66.0.227	192.168.64.5	TCP	74	443 → 47844 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1490 SACK_PERM TSval=3283399582 TSecr=3...
487	11.632801457	192.168.64.5	172.66.0.227	TCP	66	47844 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3293111361 TSecr=3283399582
488	11.633047664	192.168.64.5	172.66.0.227	TCP	1454	47844 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=1388 TSval=3293111361 TSecr=3283399582 [TCP segmen...
489	11.633948705	192.168.64.5	172.66.0.227	TLSv1.3	574	Client Hello (SNI=twitter.com)
490	11.667798022	172.66.0.227	192.168.64.5	TCP	66	443 → 47858 [ACK] Seq=1 Ack=1389 Win=131072 Len=0 TSval=3315707280 TSecr=3293111361
491	11.667798064	172.66.0.227	192.168.64.5	TCP	66	443 → 47858 [ACK] Seq=1 Ack=1897 Win=131072 Len=0 TSval=3315707280 TSecr=3293111361
492	11.670052797	172.66.0.227	192.168.64.5	TLSv1.3	3261	Server Hello, Change Cipher Spec, Application Data
493	11.670065172	192.168.64.5	172.66.0.227	TCP	66	47858 → 443 [ACK] Seq=1897 Ack=3190 Win=70656 Len=0 TSval=3293111398 TSecr=3315707283
494	11.670396544	172.66.0.227	192.168.64.5	TCP	66	443 → 47844 [ACK] Seq=1 Ack=1389 Win=131072 Len=0 TSval=3283399619 TSecr=3293111361
495	11.670396586	172.66.0.227	192.168.64.5	TCP	66	443 → 47844 [ACK] Seq=1 Ack=1897 Win=131072 Len=0 TSval=3283399620 TSecr=3293111361
496	11.6709774123	192.168.64.5	172.66.0.227	TLSv1.3	139	Change Cipher Spec, Application Data
497	11.671176663	192.168.64.5	172.66.0.227	TLSv1.3	158	Application Data
498	11.671192246	192.168.64.5	172.66.0.227	TLSv1.3	390	Application Data
499	11.672971775	172.66.0.227	192.168.64.5	TLSv1.3	3262	Server Hello, Change Cipher Spec, Application Data
500	11.672980858	192.168.64.5	172.66.0.227	TCP	66	47844 → 443 [ACK] Seq=1897 Ack=3197 Win=70656 Len=0 TSval=3293111481 TSecr=3283399621
501	11.673528062	192.168.64.5	172.66.0.227	TLSv1.3	139	Change Cipher Spec, Application Data
502	11.673688394	192.168.64.5	172.66.0.227	TLSv1.3	175	Application Data
503	11.703286834	172.66.0.227	192.168.64.5	TLSv1.3	578	Application Data, Application Data

2(a) My IP: 192.168.64.5, URL IP: 172.66.0.227

4(a) – x.com (Twitter) – Sequence of Events

Packets 481 – 503 show the complete HTTPS session setup for x.com (SNI: twitter.com, IP 172.66.0.227).

1. TCP 3-way handshake – Packet 481 is the SYN from my host (192.168.64.5 → 172.66.0.227, port 443).
Packet 486 is the SYN-ACK from the server, and the next packet is my ACK, completing the handshake.
2. TLS 1.3 handshake – Packet 485 contains the Client Hello, with the Server Name Indication field showing twitter.com.
Packets 490 – 493 show the Server Hello, key exchange, and cipher negotiation.
3. Change Cipher Spec / Finished – Packets 494 – 496 indicate encryption is established.
4. Application Data exchange – Packets 497 – 503 carry the encrypted HTTPS traffic for the X.com home page request and its response.

5(a) Protocols: TCP, TLSv1.3 Ports: 47866 (client ephemeral port), 443 (HTTPS server port)

No.	Time	Source	Destination	Protocol	Length	Info
2446	29.877579296	192.168.64.5	129.24.172.124	TCP	74	48514 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=4140802649 TSecr=0 WS=128
2447	29.877625629	192.168.64.5	129.24.172.124	TCP	74	48518 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=4140802649 TSecr=0 WS=128
2448	29.905528628	129.24.172.124	192.168.64.5	TCP	78	443 → 48514 [SYN, ACK] Seq=0 Ack=1 Win=4380 Len=0 MSS=1460 SACK_PERM TSval=3906052286 TSecr=4140802649
2449	29.905528753	129.24.172.124	192.168.64.5	TCP	78	443 → 48518 [SYN, ACK] Seq=0 Ack=1 Win=4380 Len=0 MSS=1460 SACK_PERM TSval=3906052287 TSecr=4140802649
2450	29.905557502	192.168.64.5	129.24.172.124	TCP	66	48514 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0 TSval=4140802677 TSecr=3906052286
2451	29.905570336	192.168.64.5	129.24.172.124	TCP	66	48518 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0 TSval=4140802677 TSecr=3906052287
2452	29.905893625	192.168.64.5	129.24.172.124	TCP	1514	48518 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=1448 TSval=4140802677 TSecr=3906052287 [TCP segment...]
2453	29.905950458	192.168.64.5	129.24.172.124	TLSv1.2	514	Client Hello (SNI=www.unm.edu)
2454	29.906184581	192.168.64.5	129.24.172.124	TCP	1514	48514 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=1448 TSval=4140802677 TSecr=3906052287 [TCP segment...]
2455	29.906234414	192.168.64.5	129.24.172.124	TLSv1.2	514	Client Hello (SNI=www.unm.edu)
2456	29.936732977	129.24.172.124	192.168.64.5	TCP	66	443 → 48518 [ACK] Seq=1 Ack=1897 Win=6276 Len=0 TSval=3906052318 TSecr=4140802677
2457	29.938662129	129.24.172.124	192.168.64.5	TLSv1.2	1514	Server Hello
2458	29.938679045	192.168.64.5	129.24.172.124	TCP	66	48518 → 443 [ACK] Seq=1897 Ack=1449 Win=65535 Len=0 TSval=4140802710 TSecr=3906052319
2459	29.939140792	129.24.172.124	192.168.64.5	TCP	1514	443 → 48518 [ACK] Seq=1449 Ack=1897 Win=6276 Len=1448 TSval=3906052319 TSecr=4140802677 [TCP s...]
2460	29.939149560	192.168.64.5	129.24.172.124	TCP	66	48518 → 443 [ACK] Seq=1897 Ack=2897 Win=65535 Len=0 TSval=4140802710 TSecr=3906052319
2461	29.939520539	129.24.172.124	192.168.64.5	TCP	1514	443 → 48518 [PSH, ACK] Seq=2897 Ack=1897 Win=6276 Len=1448 TSval=3906052319 TSecr=4140802677 [...]
2462	29.939520581	129.24.172.124	192.168.64.5	TCP	66	443 → 48514 [ACK] Seq=1 Ack=1897 Win=6276 Len=0 TSval=3906052320 TSecr=4140802677
2463	29.939529997	192.168.64.5	129.24.172.124	TCP	66	48518 → 443 [ACK] Seq=1897 Ack=4345 Win=65535 Len=0 TSval=4140802711 TSecr=3906052319
2464	29.940390866	129.24.172.124	192.168.64.5	TLSv1.2	1514	Server Hello
2465	29.940313616	192.168.64.5	129.24.172.124	TCP	66	48514 → 443 [ACK] Seq=1897 Ack=1449 Win=65535 Len=0 TSval=4140802712 TSecr=3906052320
2466	29.940941362	129.24.172.124	192.168.64.5	TCP	2962	443 → 48514 [PSH, ACK] Seq=1449 Ack=1897 Win=6276 Len=2896 TSval=3906052320 TSecr=4140802677 [...]
2467	29.940949986	192.168.64.5	129.24.172.124	TCP	66	48514 → 443 [ACK] Seq=1897 Ack=4345 Win=65535 Len=0 TSval=4140802712 TSecr=3906052320
2468	29.964244186	129.24.172.124	192.168.64.5	TCP	1514	443 → 48518 [PSH, ACK] Seq=4345 Ack=1897 Win=6276 Len=1448 TSval=3906052345 TSecr=4140802710 [...]
2470	29.964254645	192.168.64.5	129.24.172.124	TCP	66	48518 → 443 [ACK] Seq=1897 Ack=5793 Win=65535 Len=0 TSval=4140802735 TSecr=3906052345

2(b) My IP: 192.168.64.5, URL IP: 129.24.172.124
4(b) – cs.unm.edu — Sequence of Events

Packets 2446 – 2469 show the HTTPS session setup for cs.unm.edu (129.24.172.124).

1. TCP 3-way handshake – Packet 2446 is the SYN from my host (192.168.64.5 → 129.24.172.124, port 443).
Packet 2449 is the SYN-ACK reply from the server, and the next ACK completes the handshake.
2. TLS 1.2 handshake – Packet 2451 contains the Client Hello, with SNI www.unm.edu.
Packets 2456 – 2458 show the Server Hello, certificate exchange, and cipher suite negotiation.
3. Change Cipher Spec / Finished – Packets 2459 – 2461 indicate the start of encrypted communication.
4. Application Data exchange – Packets 2462 – 2469 carry encrypted HTTPS payloads for the page request and response.

5(b) Protocols: TCP, TLSv1.2 Ports: 59634 (client ephemeral port), 443 (HTTPS server port)

No.	Time	Source	Destination	Protocol	Length	Info
7426	62.438317139	192.168.64.5	64.106.20.76	TCP	74	60364 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1778112680 TSecr=0 WS=128
7427	62.438362972	192.168.64.5	64.106.20.76	TCP	74	60380 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1778112680 TSecr=0 WS=128
7428	62.466340268	64.106.20.76	192.168.64.5	TCP	74	443 → 60364 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=14160089157 TSecr=1
7429	62.466340476	64.106.20.76	192.168.64.5	TCP	74	443 → 60380 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=14160089157 TSecr=1
7430	62.466376864	192.168.64.5	64.106.20.76	TCP	66	60364 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1778112708 TSecr=4160089157
7431	62.466396351	192.168.64.5	64.106.20.76	TCP	66	60380 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1778112708 TSecr=4160089157
7432	62.466749807	192.168.64.5	64.106.20.76	TCP	1514	60380 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=1448 TSval=1778112708 TSecr=4160089157 [TCP segmen..
7433	62.466751057	192.168.64.5	64.106.20.76	TLSv1.3	517	Client Hello (SNI=www.cs.unm.edu)
7434	62.467698929	192.168.64.5	64.106.20.76	TCP	1514	60364 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=1448 TSval=1778112708 TSecr=4160089157 [TCP segmen..
7435	62.467899637	192.168.64.5	64.106.20.76	TLSv1.3	517	Client Hello (SNI=www.cs.unm.edu)
7436	62.495188558	64.106.20.76	192.168.64.5	TCP	66	443 → 60380 [ACK] Seq=1 Ack=1900 Win=45056 Len=0 TSval=4160089188 TSecr=1778112708
7437	62.496386716	64.106.20.76	192.168.64.5	TLSv1.3	1514	Server Hello, Change Cipher Spec, Application Data
7438	62.496401641	192.168.64.5	64.106.20.76	TCP	66	60380 → 443 [ACK] Seq=1900 Ack=1449 Win=67200 Len=0 TSval=1778112738 TSecr=4160089187
7439	62.496912987	64.106.20.76	192.168.64.5	TLSv1.3	1486	Application Data, Application Data, Application Data
7440	62.496912171	64.106.20.76	192.168.64.5	TCP	66	443 → 60364 [ACK] Seq=1 Ack=1900 Win=45056 Len=0 TSval=4160089188 TSecr=1778112708
7441	62.496923920	192.168.64.5	64.106.20.76	TCP	66	60380 → 443 [ACK] Seq=1900 Ack=2869 Win=78144 Len=0 TSval=1778112738 TSecr=4160089187
7442	62.497759748	64.106.20.76	192.168.64.5	TLSv1.3	1514	Server Hello, Change Cipher Spec, Application Data
7443	62.497763789	192.168.64.5	64.106.20.76	TCP	66	60364 → 443 [ACK] Seq=1900 Ack=1449 Win=67200 Len=0 TSval=1778112739 TSecr=4160089188
7444	62.497794931	192.168.64.5	64.106.20.76	TLSv1.3	139	Change Cipher Spec, Application Data
7445	62.498051662	192.168.64.5	64.106.20.76	TLSv1.3	800	Application Data
7446	62.498399785	64.106.20.76	192.168.64.5	TLSv1.3	1485	Application Data, Application Data, Application Data
7447	62.498403576	192.168.64.5	64.106.20.76	TCP	66	60364 → 443 [ACK] Seq=1900 Ack=2868 Win=78144 Len=0 TSval=1778112740 TSecr=4160089188
7448	62.498786907	192.168.64.5	64.106.20.76	TLSv1.3	139	Change Cipher Spec, Application Data
7450	62.523604664	64.106.20.76	192.168.64.5	TCP	66	443 → 60380 [ACK] Seq=2869 Ack=1964 Win=45056 Len=0 TSval=4160089215 TSecr=1778112739

2(c) My IP: 192.168.64.5, URL IP: 64.106.20.76

4(c) – www.unm.edu – Sequence of Events

Packets 7426 – 7450 show the HTTPS session setup for www.unm.edu (64.106.20.76).

1. TCP 3-way handshake – Packet 7426 is the SYN from my host (192.168.64.5 → 64.106.20.76, port 443).

Packet 7429 is the SYN-ACK from the server, and the following packet completes the handshake with an ACK.

2. TLS 1.3 handshake – Packet 7432 contains the Client Hello, listing supported cipher suites and SNI www.cs.unm.edu.

Packets 7435 – 7438 include the Server Hello, certificate exchange, and cipher negotiation.

3. Change Cipher Spec / Finished – Packets 7439 – 7441 establish encryption for the TLS session.

4. Application Data exchange – Packets 7442 – 7450 carry the encrypted HTTPS content, corresponding to the web page request and server responses.

5(c) Protocols: TCP, TLSv1.3

Ports: 60398 (client ephemeral port), 443 (HTTPS server port)

No.	Time	Source	Destination	Protocol	Length	Info
12269	93.922879360	192.168.64.5	40.126.29.9	TLSv1.3	711	Client Hello (SNI=login.microsoftonline.com)
12393	94.199515432	192.168.64.5	23.62.33.24	TLSv1.3	598	Client Hello (SNI=aadcdn.msftauth.net)
12400	94.192734833	192.168.64.5	13.107.246.51	TLSv1.3	548	Client Hello (SNI=aadcdn.msftauthimages.net)
12408	94.226026973	192.168.64.5	23.62.33.37	TLSv1.3	1233	Client Hello (SNI=aadcdn.msftauth.net)
12423	94.228941536	192.168.64.5	40.126.28.21	TLSv1.2	513	Client Hello (SNI=login.live.com)
12573	94.467911810	192.168.64.5	23.62.33.37	TLSv1.3	598	Client Hello (SNI=aadcdn.msftauth.net)
12598	94.566706738	192.168.64.5	23.62.33.37	TLSv1.3	598	Client Hello (SNI=aadcdn.msftauth.net)
12617	94.618991427	192.168.64.5	13.107.246.51	TLSv1.3	1183	Client Hello (SNI=aadcdn.msftauthimages.net)
14462	134.211784329	192.168.64.5	13.107.246.51	TLSv1.3	1183	Client Hello (SNI=aadcdn.msftauthimages.net)
14464	134.212163076	192.168.64.5	23.62.33.24	TLSv1.3	1233	Client Hello (SNI=aadcdn.msftauth.net)
14674	134.618244363	192.168.64.5	20.190.157.3	TLSv1.2	513	Client Hello (SNI=login.live.com)
18180	155.596166910	192.168.64.5	23.62.33.24	TLSv1.3	1233	Client Hello (SNI=aadcdn.msftauth.net)
18268	155.947145769	192.168.64.5	75.2.14.209	TLSv1.3	520	Client Hello (SNI=sso.canvaslms.com)
19157	157.076529136	192.168.64.5	99.83.186.201	TLSv1.3	520	Client Hello (SNI=sso.canvaslms.com)
20571	193.113089316	192.168.64.5	13.107.246.51	TLSv1.3	1183	Client Hello (SNI=aadcdn.msftauthimages.net)

Above, Microsoft login, Below, Canvas

No.	Time	Source	Destination	Protocol	Length	Info
11194	91.080651349	192.168.64.5	64.106.65.134	TCP	60	35084 → 443 [ACK] Seq=2967 Ack=166697 Win=246912 Len=0 TSval=2176985848 TSecr=1817540959
11195	91.080651432	192.168.64.5	64.106.65.134	TCP	60	48950 → 443 [ACK] Seq=4537 Ack=15963 Win=72320 Len=0 TSval=2658441622 TSecr=1793313883
11196	91.080660140	192.168.64.5	64.106.65.134	TCP	60	443 → 443 [ACK] Seq=14114 Ack=2464 Win=131072 Len=0 TSval=3542832207 TSecr=3578924176
11197	91.080677998	192.168.64.5	64.106.65.134	TCP	60	443 → 443 [ACK] Seq=2464 Ack=14153 Win=79360 Len=0 TSval=3578924250 TSecr=3542832207
11198	91.081886133	192.168.64.5	64.106.65.134	TCP	60	443 → 443 [ACK] Seq=2423 Ack=5265 Win=74752 Len=0 TSval=3211561040 TSecr=48...
11199	91.081886175	192.168.64.5	64.106.65.134	TCP	66	[TCP Keep-Alive] 47532 → 443 [ACK] Seq=2423 Ack=5265 Win=74752 Len=0 TSval=3211561040 TSecr=48...
11200	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11201	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11202	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11203	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11204	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11205	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11206	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11207	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11208	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11209	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11210	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11211	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11212	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11213	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11214	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11215	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11216	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11217	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11218	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11219	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11220	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11221	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11222	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11223	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11224	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11225	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11226	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074
11227	92.090218137	64.106.65.134	192.168.64.5	TCP	66	[TCP Keep-Alive ACK] 443 → 47532 [ACK] Seq=5265 Ack=2424 Win=72832 Len=0 TSval=1781059074 TSecr=1781059074

- 2(d) My IP: My IP 192.168.64.5
- Canvas (initial): canvasinfo.unm.edu → Dest IP: 64.106.65.134
- Microsoft login: login.microsoftonline.com → Dest IP: 40.126.29.9 (row 12269)
- Microsoft auth CDN: aadcdn.msauth.net → Dest IP: 23.60.163.24 (row 12393)
- Microsoft auth images CDN: aadcdn.msauthimages.net → Dest IP: 23.60.163.24 (row 12400)
- Live ID federation: login.live.com → Dest IPs: 13.107.246.51, 40.126.28.37 (rows 12423, 12425)
- Canvas.lms → Dest IP: 75.2.14.209

4(d) – Canvas (canvas.unm.edu) – Sequence of Events (Microsoft SSO Login)

Packets 11119 – 11148 and 12269 – 20571 together show the full HTTPS login process for Canvas at the University of New Mexico, which uses Microsoft 365 (Azure AD) single sign-on authentication.

1. TCP 3-way handshake – Packet 11119 is the SYN from my host (192.168.64.5 → 64.106.65.134, port 443). Packet 11120 is the SYN-ACK reply from the server, and 11122 is my ACK, completing the handshake.

2. TLS 1.3 handshake to Canvas – Packet 11124 contains the Client Hello, with SNI canvasinfo.unm.edu. Packets 11133 – 11134 show the Server Hello, certificate exchange, and cipher negotiation.

3. Change Cipher Spec / Finished – Packets 11135 – 11137 mark the transition to encrypted communication.

4. Initial Application Data (exchange with Canvas) – Packets 11138 – 11148 carry encrypted HTTPS payloads representing the Canvas login page request. Canvas then redirects the browser to Microsoft’s authentication endpoint for single sign-on.

5. Redirect to Microsoft Login Service (Azure AD handshake) – Starting at packet 12269, the browser initiates new TLS 1.3 Client Hello sessions to:

- login.microsoftonline.com (12269) – initial authentication endpoint.
- aadcdn.msauth.net and aadcdn.msauthimages.net (12393 – 12433) – content delivery hosts for Microsoft login resources.
- login.live.com (12423, 12464) – Live ID federation service.

Each of these hosts responds with TLS 1.3 Server Hello and encrypted Application Data packets, completing Microsoft’s secure sign-in sequence.

6. Credential and Token Exchange (Encrypted) – Within the following Application Data frames, the browser submits user credentials and receives OAuth tokens and redirect instructions from Microsoft 365 (Azure AD). These packets are fully encrypted inside TLS and therefore not readable in the capture.

7. Return to Canvas Session (SSO redirect) – Near the end of the chain (packets ≈ 20500 – 20571), new Client Hello messages target sso.canvaslms.com and canvaslms.com, marking the final redirect back to Canvas with the authenticated session. Canvas then establishes a fresh TLS 1.3 connection to deliver the logged-in dashboard.

5(d) Protocols: TCP, TLSv1.3 , TLSv1.2

Ports: 47438 (canvasinfo) (44726 (microsoftonline), 46154, 53756, 46766, 46780, 46788, 53768, 36358, 33874, 60882, 43600 (msftauth), 55036, 35208(login.live), 60594, 35048(canvaslms) (client ephemeral ports), 443 (HTTPS server port)

The screenshot shows a Wireshark capture of network traffic. The top pane displays a list of packets with columns for No., Time, Source, Destination, Protocol, and Length. The bottom pane shows the details of a selected packet (No. 5900), which is a TLSv1.3 Client Hello. The details pane includes sections for 'Frame 5900: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp0s8', 'Ethernet II, Src: 6a:23:79:bb:8a:fc (6a:23:79:bb:8a:fc), Dst: 86:2f:57:14:a9:64 (86:2f:57:14:a9:64)', and 'Internet Protocol Version 4, Src: 192.168.0.5, Dst: 57.144.174.128'. The 'Transmission Control Protocol' section shows 'Src Port: 51376, Dst Port: 443, Seq: 0, Len: 0'. The 'Hypertext Transfer Protocol' section shows 'Content-Type: application/javascript'. The 'Application Data' section shows 'Content-Type: application/javascript'. The 'TLSv1.3' section shows 'Client Hello' with fields for 'Version: 3.1', 'Random: 86 2f 57 14 a9 64 6a 23 79 b0 8a fc 08 06 45 00', 'Session ID: 0000', 'Compression Methods: 0000', 'Supported Versions: 0020', 'Supported Groups: 0020', 'Supported Extensions: 0030', and 'Supported Groups: 0040'.

2(e) My IP: 192.168.64.5 URL IP: 57.144.174.128

4(e) – facebook.com – Sequence of Events

Packets 5900 – 5926 show the HTTPS session setup for facebook.com

1. TCP 3-way handshake – Packet 5900 is the SYN from my host (192.168.64.5 → 57.144.174.128, port 443).

Packet 5912 is the SYN-ACK from the server, and the following packet is my ACK, completing the TCP handshake.

2. TLS 1.3 handshake – Packet 5914 contains the Client Hello, including the SNI connect.facebook.net.

Packets 5916 – 5917 show the Server Hello, cipher negotiation, and certificate exchange.

3. Change Cipher Spec / Finished – Packets 5918 – 5920 complete encryption setup.

4. Application Data exchange – Packets 5921 – 5926 carry the encrypted HTTPS payloads that correspond to Facebook’s dynamic content and resources (scripts, media, etc.).

5(e) Protocols: TCP, TLSv1.3. Ports: 51376 (client ephemeral port), 443 (HTTPS server port)

No.	Time	Source	Destination	Protocol	Length	Info
21699	237.956892372	192.168.64.5	104.21.9.67	TCP	74	47158 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1545981384 TSecr=0 WS=128
21700	237.989989334	104.21.9.67	192.168.64.5	TCP	74	443 → 47158 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PERM TSval=1764324832 TSecr=1
21701	237.989948293	192.168.64.5	104.21.9.67	TCP	66	47158 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1545981337 TSecr=1764324832
21702	237.998444498	192.168.64.5	104.21.9.67	TLSv1.3	672	Client Hello (SNI=cloudflare-ech.com)
21703	238.025328657	104.21.9.67	192.168.64.5	TCP	66	443 → 47158 [ACK] Seq=1 Ack=697 Win=131072 Len=0 TSval=1764324868 TSecr=1545981338
21704	238.029642255	104.21.9.67	192.168.64.5	TLSv1.3	1514	Server Hello, Change Cipher Spec
21705	238.029656380	192.168.64.5	104.21.9.67	TCP	66	47158 → 443 [ACK] Seq=697 Ack=1449 Win=67200 Len=0 TSval=1545981377 TSecr=1764324872
21706	238.030183460	104.21.9.67	192.168.64.5	TLSv1.3	1433	Application Data
21707	238.030195752	192.168.64.5	104.21.9.67	TCP	66	47158 → 443 [ACK] Seq=697 Ack=2816 Win=70144 Len=0 TSval=1545981377 TSecr=1764324872
21708	238.031462494	192.168.64.5	104.21.9.67	TLSv1.3	130	Change Cipher Spec, Application Data
21709	238.031729489	192.168.64.5	104.21.9.67	TLSv1.3	158	Application Data
21710	238.031887366	192.168.64.5	104.21.9.67	TLSv1.3	393	Application Data
21711	238.065206327	104.21.9.67	192.168.64.5	TLSv1.3	578	Application Data, Application Data
21712	238.065206368	104.21.9.67	192.168.64.5	TLSv1.3	97	Application Data
21713	238.065692973	192.168.64.5	104.21.9.67	TLSv1.3	97	Application Data
21714	238.098946499	104.21.9.67	192.168.64.5	TCP	66	443 → 47158 [ACK] Seq=3359 Ack=1121 Win=131072 Len=0 TSval=1764324941 TSecr=1545981379
21715	238.097114721	104.21.9.67	192.168.64.5	TLSv1.3	835	Application Data
21716	238.097114930	104.21.9.67	192.168.64.5	TLSv1.3	1457	Application Data
21717	238.097114971	104.21.9.67	192.168.64.5	TLSv1.3	2848	Application Data, Application Data
21718	238.097114971	104.21.9.67	192.168.64.5	TLSv1.3	3152	Application Data, Application Data, Application Data
21719	238.097115913	104.21.9.67	192.168.64.5	TLSv1.3	97	Application Data
21720	238.097157679	192.168.64.5	104.21.9.67	TCP	66	47158 → 443 [ACK] Seq=1121 Ack=5519 Win=73856 Len=0 TSval=1545981954 TSecr=1764325486
21721	238.097239262	192.168.64.5	104.21.9.67	TCP	66	47158 → 443 [ACK] Seq=1121 Ack=11418 Win=68864 Len=0 TSval=1545981954 TSecr=1764325486
21722	238.031978276	192.168.64.5	104.21.9.67	TLSv1.3	214	Application Data

2(f) My IP 192.168.64.6, URL IP: 104.21.9.67

4(f) – usenix.org – Sequence of Events

Packets 21699 – 21725 show the complete HTTPS session setup for usenix.org (served via Cloudflare at 104.21.9.67).

1. TCP 3-way handshake – Packet 21699 is the SYN from my host (192.168.64.5 → 104.21.9.67, port 443).

Packet 21700 is the SYN-ACK reply from the server, and 21701 is my ACK, completing the handshake.

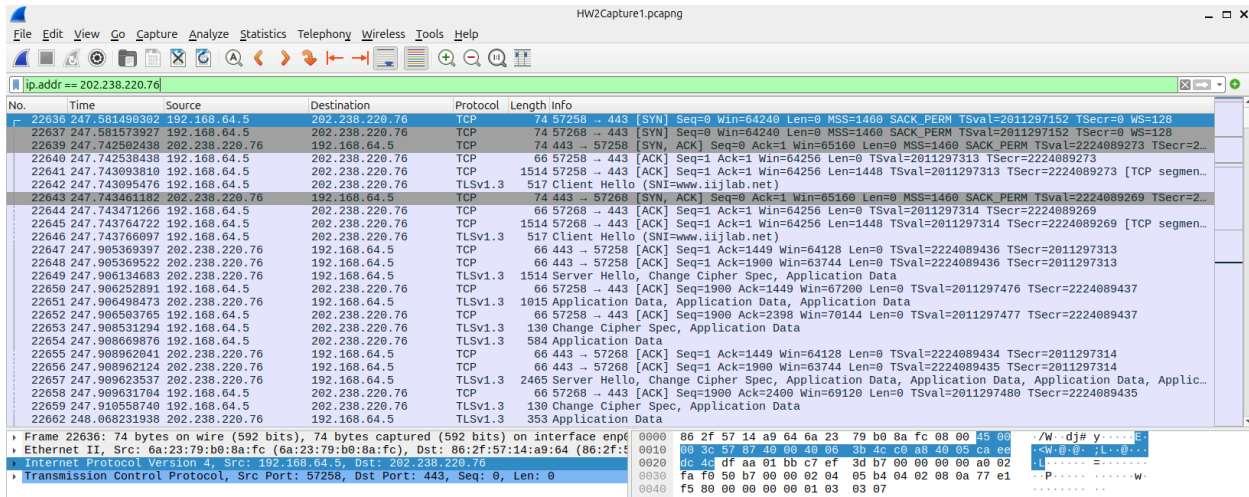
2. TLS 1.3 handshake – Packet 21703 contains the Client Hello, including SNI cloudflare-ech.com.

Packets 21704 and 21705 show the Server Hello and certificate exchange.

3. Change Cipher Spec / Finished – Packets 21706 – 21708 transition to encrypted communication.

4. Application Data exchange – Packets 21709 through 21725 carry encrypted HTTPS payloads representing my GET request and the server’s responses.

5(f) Protocols: TCP, TLSv1.3, Ports: 47158 (client ephemeral port), 443 (HTTPS server port)



2(g) My IP 192.168.64.5, URL IP: 202.238.220.76

4(g) – iijlab.net (PDF) – Sequence of Events

Packets 22636 – 22668 show the complete HTTPS session setup for iijlab.net (202.238.220.76).

1. TCP 3-way handshake – Packet 22636 is the SYN from my host (192.168.64.5 → 202.238.220.76, port 443).

Packet 22637 is the SYN-ACK from the server, and 22638 is my ACK, completing the handshake.

2. TLS 1.3 handshake – Packet 22641 contains the Client Hello, showing the SNI www.iijlab.net.

Packets 22644 and 22645 show the Server Hello, certificate exchange, and cipher suite negotiation.

3. Change Cipher Spec / Finished – Packets 22646 – 22648 transition to encrypted communication.

4. Application Data exchange – Packets 22649 through 22668 carry encrypted HTTPS payloads representing the PDF download request and the server’s response data.

5(g) Protocols: TCP, TLSv1.3 Ports: 57258 (client ephemeral port), 443 (HTTPS server port)

Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A
192.168.64.5	47844	172.66.0.227	443	17	7 kB	8	17	100.00%	10	3 kB	7	4 kB
192.168.64.5	47858	172.66.0.227	443	38	18 kB	9	38	100.00%	22	4 kB	16	14 kB
192.168.64.5	47866	172.66.0.227	443	31	8 kB	16	31	100.00%	17	3 kB	14	5 kB

6(a) $17 + 38 + 31 = 86$ packets

Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Star
192.168.64.5	48514	129.24.172.124	443	34	17 kB	34	34	100.00%	19	4 kB	15	13 kB	29.877571
192.168.64.5	48518	129.24.172.124	443	32	20 kB	35	32	100.00%	14	3 kB	18	17 kB	29.877621
192.168.64.5	48532	129.24.172.124	443	39	25 kB	36	39	100.00%	18	4 kB	21	21 kB	30.046061
192.168.64.5	48544	129.24.172.124	443	26	13 kB	37	26	100.00%	14	4 kB	12	9 kB	30.048121
192.168.64.5	48548	129.24.172.124	443	24	14 kB	38	24	100.00%	13	4 kB	11	10 kB	30.048461
192.168.64.5	48552	129.24.172.124	443	22	14 kB	39	22	100.00%	12	3 kB	10	10 kB	30.048591
192.168.64.5	48566	129.24.172.124	443	22	11 kB	40	22	100.00%	12	3 kB	10	8 kB	30.048921
192.168.64.5	48578	129.24.172.124	443	26	14 kB	49	26	100.00%	13	4 kB	13	11 kB	31.802301
192.168.64.5	48580	129.24.172.124	443	45	64 kB	50	45	100.00%	22	4 kB	23	60 kB	31.802371
192.168.64.5	48582	129.24.172.124	443	26	14 kB	51	26	100.00%	13	4 kB	13	10 kB	31.802401
192.168.64.5	48594	129.24.172.124	443	26	14 kB	52	26	100.00%	13	4 kB	13	10 kB	31.802421
192.168.64.5	48610	129.24.172.124	443	94	198 kB	53	94	100.00%	42	6 kB	52	193 kB	31.802491
192.168.64.5	48620	129.24.172.124	443	92	150 kB	55	92	100.00%	42	6 kB	50	144 kB	31.824401
192.168.64.5	48624	129.24.172.124	443	24	11 kB	64	24	100.00%	13	4 kB	11	8 kB	31.947811
192.168.64.5	48636	129.24.172.124	443	39	24 kB	67	39	100.00%	18	4 kB	21	20 kB	31.975801
192.168.64.5	48650	129.24.172.124	443	66	115 kB	68	66	100.00%	29	5 kB	37	111 kB	31.975901
192.168.64.5	48656	129.24.172.124	443	67	120 kB	69	67	100.00%	31	5 kB	36	115 kB	32.033641
192.168.64.5	48662	129.24.172.124	443	375	1 MB	70	375	100.00%	145	12 kB	230	1 MB	32.100951
192.168.64.5	58668	129.24.172.124	443	22	10 kB	95	22	100.00%	13	3 kB	9	7 kB	48.510411
192.168.64.5	58678	129.24.172.124	443	50	73 kB	96	50	100.00%	19	4 kB	31	69 kB	48.510631
192.168.64.5	58694	129.24.172.124	443	553	2 MB	97	553	100.00%	213	17 kB	340	2 MB	55.627811
192.168.64.5	58696	129.24.172.124	443	18	10 kB	98	18	100.00%	11	3 kB	7	7 kB	55.627931
192.168.64.5	59408	129.24.172.124	443	56	134 kB	93	56	100.00%	27	5 kB	29	130 kB	40.791861
192.168.64.5	59414	129.24.172.124	443	20	10 kB	94	20	100.00%	12	3 kB	8	7 kB	40.791921
192.168.64.5	59506	129.24.172.124	443	22	20 kB	167	22	100.00%	12	3 kB	10	16 kB	135.154511
192.168.64.5	59520	129.24.172.124	443	25	17 kB	169	25	100.00%	14	3 kB	11	13 kB	135.287321
192.168.64.5	59532	129.24.172.124	443	24	14 kB	170	24	100.00%	13	3 kB	11	10 kB	135.287391
192.168.64.5	59548	129.24.172.124	443	24	24 kB	171	24	100.00%	13	3 kB	11	20 kB	135.287791
192.168.64.5	59564	129.24.172.124	443	23	14 kB	172	23	100.00%	13	3 kB	10	11 kB	135.287961
192.168.64.5	59572	129.24.172.124	443	21	12 kB	173	21	100.00%	12	3 kB	9	8 kB	135.288111
192.168.64.5	59578	129.24.172.124	443	23	13 kB	174	23	100.00%	13	3 kB	10	9 kB	135.288351
192.168.64.5	59586	129.24.172.124	443	22	14 kB	179	22	100.00%	11	3 kB	11	11 kB	135.425181
192.168.64.5	59598	129.24.172.124	443	26	14 kB	180	26	100.00%	13	3 kB	13	11 kB	135.425221
192.168.64.5	59602	129.24.172.124	443	24	14 kB	181	24	100.00%	12	3 kB	12	11 kB	135.425241
192.168.64.5	59618	129.24.172.124	443	90	198 kB	182	90	100.00%	38	5 kB	52	193 kB	135.426921
192.168.64.5	59628	129.24.172.124	443	45	64 kB	183	45	100.00%	19	4 kB	26	60 kB	135.426971
192.168.64.5	59634	129.24.172.124	443	79	149 kB	184	79	100.00%	33	5 kB	46	144 kB	135.461661
192.168.64.5	59650	129.24.172.124	443	33	24 kB	187	33	100.00%	14	4 kB	19	20 kB	135.553421
192.168.64.5	59662	129.24.172.124	443	53	114 kB	188	53	100.00%	25	4 kB	28	110 kB	135.553451
192.168.64.5	59676	129.24.172.124	443	410	1 MB	189	410	100.00%	164	14 kB	246	1 MB	135.582391
192.168.64.5	59690	129.24.172.124	443	25	12 kB	192	25	100.00%	14	3 kB	11	8 kB	135.630471
192.168.64.5	59696	129.24.172.124	443	64	120 kB	195	64	100.00%	28	4 kB	36	115 kB	135.682901

6(b) $34+32+39+26+24+22+22+26+45+26+26+94+92+24+39+66+67+375+22+50+553+18+56+20+22+25+24+24+23+21+23+22+26+24+90+45+79+33+53+410+25+64 = 2,831$ packets

Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A
192.168.64.5	60364	64.106.20.76	443	532	1 MB	100	532	100.00%	169	19 kB	363	1 MB
192.168.64.5	60380	64.106.20.76	443	223	296 kB	101	223	100.00%	104	12 kB	119	284 kB
192.168.64.5	60396	64.106.20.76	443	167	215 kB	102	167	100.00%	78	10 kB	89	205 kB
192.168.64.5	60398	64.106.20.76	443	1,824	3 MB	103	1,824	100.00%	722	53 kB	1,102	3 MB
192.168.64.5	60404	64.106.20.76	443	213	316 kB	104	213	100.00%	86	10 kB	127	306 kB
192.168.64.5	60420	64.106.20.76	443	127	169 kB	106	127	100.00%	54	7 kB	73	162 kB

6(c) $532 + 223 + 167 + 1824 + 127 = 2,873$ packets

Wireshark - Conversations - HW2Capture1.pcapng																
Ethernet - 1	IPv4 - 8	IPv6	TCP - 15	UDP												
Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s	
192.168.64.5	36358	13.107.246.51	443	1	1 kB	163	28	3.57%	1	1 kB	0	0 bytes	134.177907	5.1271	1,84	
192.168.64.5	43600	13.107.246.51	443	1	1 kB	228	25	4.00%	1	1 kB	0	0 bytes	193.080418	5.1000	1,85	
192.168.64.5	53756	13.107.246.51	443	1	548 bytes	134	39	2.56%	1	548 bytes	0	0 bytes	94.1599524	5.2147	84	
192.168.64.5	53768	13.107.246.51	443	1	1 kB	140	27	3.70%	1	1 kB	0	0 bytes	94.576924	5.1897	1,82	
192.168.64.5	35208	20.190.157.3	443	1	513 bytes	166	13	7.69%	1	513 bytes	0	0 bytes	134.564878	0.2156		
192.168.64.5	33874	23.62.33.24	443	1	1 kB	162	25	4.00%	1	1 kB	0	0 bytes	134.177770	20.1845	46	
192.168.64.5	46514	23.62.33.24	443	1	598 bytes	133	32	3.13%	1	598 bytes	0	0 bytes	94.156769	20.2699	23	
192.168.64.5	60882	23.62.33.24	443	1	1 kB	199	25	4.00%	1	1 kB	0	0 bytes	155.468614	20.1953	46	
192.168.64.5	46766	23.62.33.37	443	1	1 kB	135	279	0.36%	1	1 kB	0	0 bytes	94.191874	157.3669	6	
192.168.64.5	46780	23.62.33.37	443	1	598 bytes	138	15	6.67%	1	598 bytes	0	0 bytes	94.434580	0.0832		
192.168.64.5	46788	23.62.33.37	443	1	598 bytes	139	227	0.44%	1	598 bytes	0	0 bytes	94.534044	159.0833		
192.168.64.5	55036	40.126.28.21	443	1	513 bytes	136	59	1.69%	1	513 bytes	0	0 bytes	94.192337	182.4907		
192.168.64.5	44726	40.126.29.9	443	1	711 bytes	132	91	1.10%	1	711 bytes	0	0 bytes	93.865687	159.7190		
192.168.64.5	60594	75.2.14.209	443	1	520 bytes	200	36	2.78%	1	520 bytes	0	0 bytes	155.933548	117.7009		

6(d) Count until Canvas.lms → Dest IP:75.2.14.209
 $28+25+39+27+13+25+32+25+279+15+227+59+91+36 = 921$ packets

Wireshark - Conversations - HW2Capture1.pcapng																
Ethernet - 1	IPv4 - 1	IPv6	TCP - 3	UDP												
Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s	
192.168.64.5	51376	57.144.174.128	443	26	8 kB	81	26	100.00%	15	3 kB	11	5 kB	32.904571	0.1602		
192.168.64.5	51378	57.144.174.128	443	96	102 kB	82	96	100.00%	50	6 kB	46	96 kB	32.904596	0.2422		
192.168.64.5	51394	57.144.174.128	443	99	76 kB	83	99	100.00%	54	8 kB	45	68 kB	33.231575	229.8971		

6(e) $26+96+99 = 221$ packets

Wireshark - Conversations - HW2Capture1.pcapng																
Ethernet - 1	IPv4 - 1	IPv6	TCP - 1	UDP												
Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s	
192.168.64.5	47158	104.21.9.67	443	254	511 kB	238	254	100.00%	99	27 kB	155	484 kB	237.956892	1.3249	16	

6(f) 254 packets

Wireshark - Conversations - HW2Capture1.pcapng																
Ethernet - 1	IPv4 - 1	IPv6	TCP - 5	UDP												
Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s	
192.168.64.5	45094	202.238.220.76	443	194	284 kB	248	194	100.00%	82	9 kB	112	274 kB	248.247445	6.6584		
192.168.64.5	45100	202.238.220.76	443	23	7 kB	249	23	100.00%	13	3 kB	10	4 kB	248.283598	5.6489	4,	
192.168.64.5	45106	202.238.220.76	443	103	133 kB	250	103	100.00%	49	6 kB	54	127 kB	248.399552	6.3618	7,	
192.168.64.5	57258	202.238.220.76	443	955	2 MB	246	955	100.00%	413	35 kB	542	2 MB	247.581490	6.4645		
192.168.64.5	57268	202.238.220.76	443	21	6 kB	247	21	100.00%	12	3 kB	9	4 kB	247.581574	6.1098	3,	

6(g) $194+23+103+955+21 = 1,296$ packets

The screenshot shows a browser's developer tools network tab. The left pane displays a list of network requests, including various JavaScript files, images, and a main document. The right pane shows the 'Cookies' tab, which lists several cookies:

- cf_bm**: expires: "2025-10-20T18:32:26.000Z", httpOnly: true, path: "/", secure: true, value: "h.QRkvtfSU2wmB1PujNoJDh_Z1EYKDFib8o8aPqnAW8-1760983346.0595682-1.0.1.1-8WJ0lWj93CzAZyRGLUsqCSJpIn66JAXWTT56uhoqjAE8qVnKQWVxWxiusHv1jGzyowx7AfpXWbd_QBeYHfChfWUEBM6ivadaIBa9i6pT3jgHzvFDLWmWzY3q6QwCq"
- ct0**: expires: "1970-01-01T00:00:01.000Z", path: "/", sameSite: "Lax", secure: true, value: ""
- guest_id**: domain: "x.com", expires: "2027-10-20T18:02:26.000Z", path: "/", sameSite: "None", secure: true, value: "v1:176098334608310815"
- guest_id_ads**: domain: "x.com", expires: "2027-10-20T18:02:26.000Z", path: "/", sameSite: "None", secure: true, value: "v1:176098334608310815"
- guest_id_marketing**: domain: "x.com", expires: "2027-10-20T18:02:26.000Z", path: "/", sameSite: "None", secure: true, value: "v1:176098334608310815"
- personalization_id**: domain: "x.com", expires: "2027-10-20T18:02:26.000Z", path: "/", sameSite: "None", value: ""

7(a) Observed response cookies:

cf_bm – Cloudflare bot-management cookie; helps detect automated traffic.

ct0 – Session token used by Twitter for login state verification.

guest_id – Unique anonymous visitor identifier.

guest_id_ads – Tracking identifier for ad services.

guest_id_marketing – Marketing tracking identifier.

personalization_id – Cookie used for content and ad personalization.

The screenshot shows a browser's developer tools network tab for the website www.unm.edu. The left pane displays a list of network requests, including stylesheets, JavaScript files, and images. The right pane shows the 'Cookies' tab, which lists several cookies:

- Oreo**: httpOnly: true, path: "/", secure: true, value: "1554015498.20480.0000"
- TS0131c641**: domain: "www.unm.edu", path: "/", value: "011889d2d9366bb10a0ec3ee2062260c9395a909439891d7e2e04cabdad1edb415f1b277c69d6ac0fe54e48df143cd45232688c16d5b70e3ebc2070032d5d84bF834cc4"

7(b) Observed response cookies: Oreo and TS0131c641.

These are standard session and load-balancing cookies used by the UNM web server to maintain user sessions and distribute requests across multiple backend servers.

The screenshot shows the Network tab in Chrome DevTools for the URL www.cs.unm.edu. The 'Cookies' sub-tab is active, displaying a list of response cookies. The cookies include session identifiers like `_fbp`, `_ga`, `_gid`, `_gcl_au`, `_gtmcc`, `_tt_enable_cookie`, and `ttcid`, along with a `log_session_id` and `ttcsid`. The cookies are used for analytics and session management.

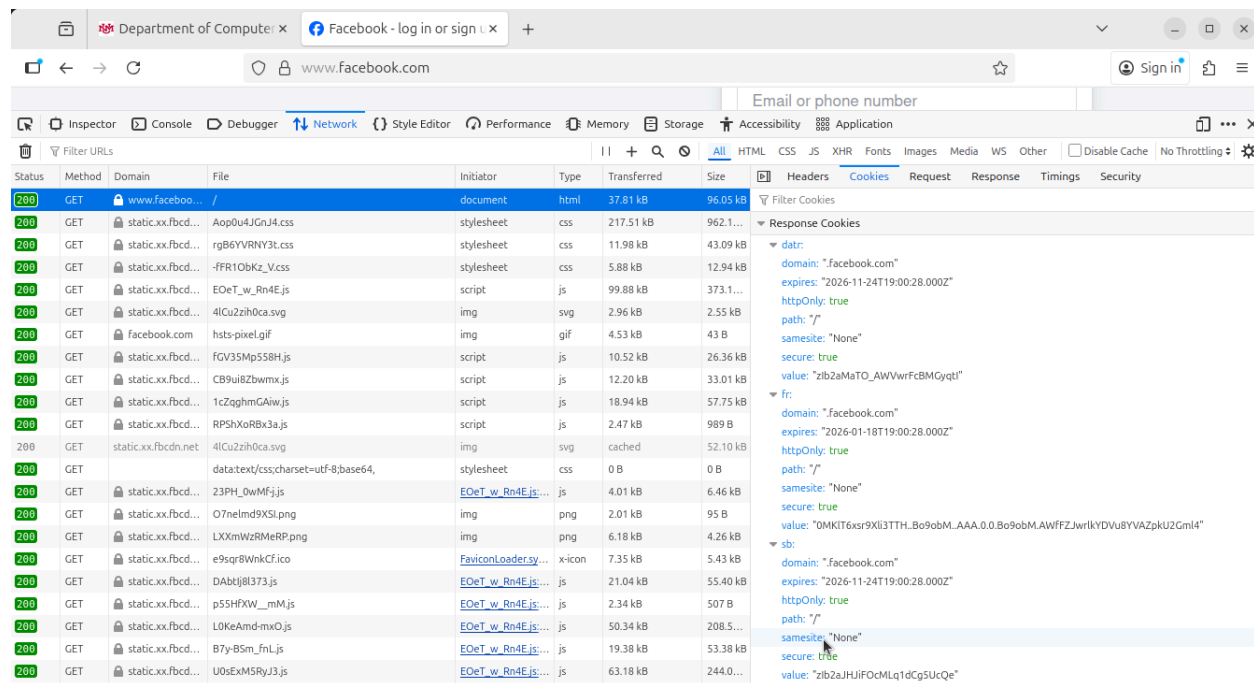
7(c) Observed request cookies: Google Analytics identifiers (_ga, _gid, _gcl_au) and Facebook or ad tracking cookies (_fbp, _tt_enable_cookie, ttcid).

These cookies are used to collect site usage data and integrate analytics and advertising features.

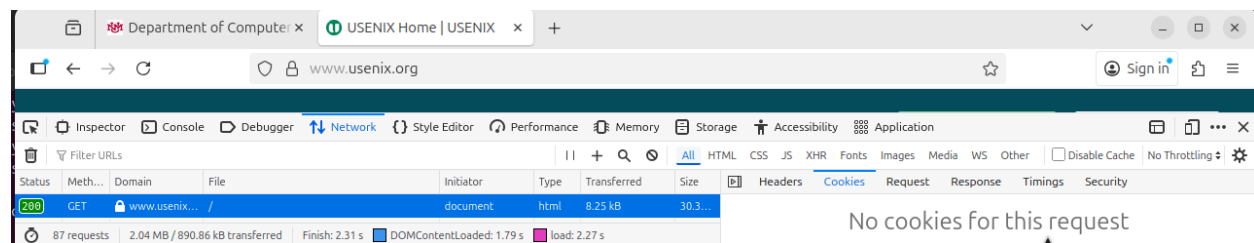
The screenshot shows the Network tab in Chrome DevTools for the URL canvas.unm.edu. The 'Cookies' sub-tab is active, displaying a list of request cookies. The cookies include session identifiers like `_fbp`, `_ga`, `_gid`, `_gcl_au`, `_gtmcc`, `_tt_enable_cookie`, and `ttcid`, along with a `log_session_id` and `ttcsid`. The cookies are used for analytics and session management.

7(d) Response cookies were initially set by the Canvas application server to establish a secure session. These included `_csrf_token`, `canvas_session`, and `log_session_id`. The `_csrf_token` cookie guard against cross-site request forgery, ensuring that any subsequent POST requests made by the user are from legitimate, authenticated sessions. The `canvas_session` cookie creates and maintains the user's active login session; it contains an encrypted session identifier that links the user's browser to their authenticated Canvas account. The `log_session_id` cookie is used internally by Canvas to log session events, correlate user actions, and improve reliability and auditing within the LMS platform.

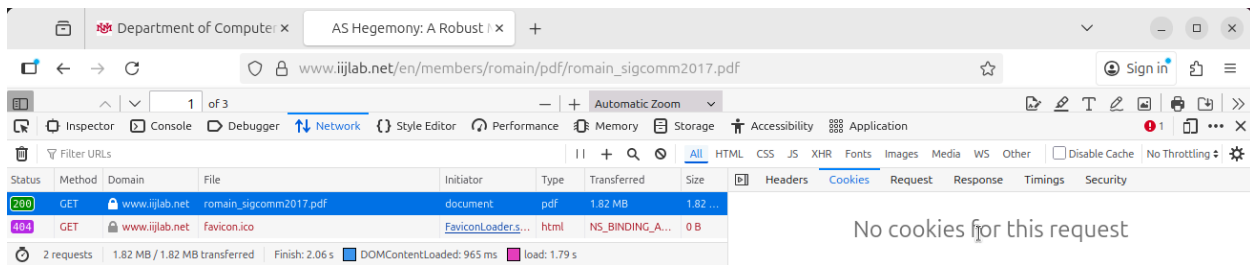
Immediately after these cookies were issued, the browser began including them in its request headers for all subsequent resource requests to Canvas, such as dashboard updates, images, and JavaScript components. The request cookies included `_csrf_token`, `canvas_session`, `log_session_id`, and third-party analytics identifiers like `_ga`, `_gid`, and `ttcid`. This is a typical example where the server sets authentication cookies and the browser returns them automatically with every new request to preserve continuity.



7(e) Response cookies included `datr`, `fr`, and `sb`. These are used to identify the browser, manage login sessions, and support security checks.

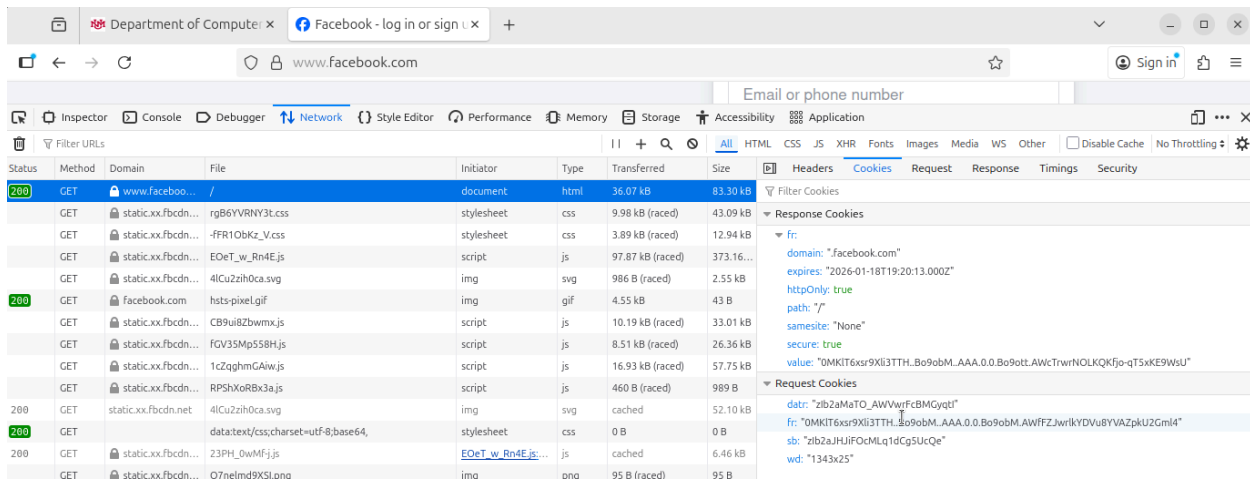


7(f) No cookies were set for this request



The screenshot shows a browser window with the address bar displaying `www.ijlab.net/en/members/romain/pdf/romain_sigcomm2017.pdf`. The developer tools network tab is open, showing a list of requests. The first request is a GET request to `www.ijlab.net/romain_sigcomm2017.pdf` with a status of 200. The 'Cookies' tab is selected, and it displays the text 'No cookies for this request'.

7(g) No cookies were set for this request



The screenshot shows a browser window with the address bar displaying `www.facebook.com`. The developer tools network tab is open, showing a list of requests. The first request is a GET request to `www.facebook.com/` with a status of 200. The 'Cookies' tab is selected, and it displays 'Response Cookies' and 'Request Cookies'.

Method	Domain	File	Initiator	Type	Transferred	Size
GET	www.facebook.com	/	document	html	36.07 kB	83.30 kB
GET	static.xx.fbcdn.net	rgB6YVRNY3t.css	stylesheet	css	9.98 kB (raced)	43.09 kB
GET	static.xx.fbcdn.net	-fFR10bKz_V.css	stylesheet	css	3.89 kB (raced)	12.94 kB
GET	static.xx.fbcdn.net	EOeT_w_Rn4E.js	script	js	97.87 kB (raced)	373.16 kB
GET	static.xx.fbcdn.net	4lCu2zih0ca.svg	img	svg	986 B (raced)	2.55 kB
GET	facebook.com	hsts-pixel.gif	img	gif	4.55 kB	43 B
GET	static.xx.fbcdn.net	CB9ui8Zbwmx.js	script	js	10.19 kB (raced)	33.01 kB
GET	static.xx.fbcdn.net	FGV35Mp558H.js	script	js	8.51 kB (raced)	26.36 kB
GET	static.xx.fbcdn.net	1cZqghmGAiw.js	script	js	16.93 kB (raced)	57.75 kB
GET	static.xx.fbcdn.net	RPSHx0RBx3a.js	script	js	460 B (raced)	989 B
GET	static.xx.fbcdn.net	4lCu2zih0ca.svg	img	svg	cached	52.10 kB
GET	static.xx.fbcdn.net	data:text/css;charset=utf-8;base64,	stylesheet	css	0 B	0 B
GET	static.xx.fbcdn.net	Z3PH_0wMFj.js	EOeT_w_Rn4E.js	js	cached	6.46 kB
GET	static.xx.fbcdn.net	O7neImd9XSLona	img	png	95 B (raced)	95 B

Response Cookies

- fr: domain: "facebook.com", expires: "2026-01-18T19:20:13.000Z", httpOnly: true, path: "/", samesite: "None", secure: true, value: "0MKIT6xr9Xl3TTH...Bo9obM...AAA.0.0.Bo9obM.AWcTrwNOLKQKfjo-qT5xkE9W5U"

Request Cookies

- datr: "zIb2aMaTO_AWVwjFcbMcyqtl"
- fr: "0MKIT6xr9Xl3TTH...Bo9obM...AAA.0.0.Bo9obM.AWfFZJwrlkYDVu8YVAZpkU2GmI4"
- sb: "zIb2aJHJfFOchLq1dCg5UcQe"
- wd: "1343x25"

8(Choose e)

On this second visit to facebook.com, request cookies now appear alongside the response cookies. This happens because after the initial visit, Facebook's cookies (fr, datr, sb) were stored locally by the browser. When the site is revisited or new requests are made, the browser automatically sends these stored cookies back to the server in the HTTP request headers.

Category	Hostname	Family	TRR	Addresses	Expires (Seconds)	Isolation Key	Extra Flags
DNS	flagship-gtm.unm.edu	ipv4	true	142.250.72.51	592	^partitionKey=%28https%2Cunm.edu%29	0 0x12022 2 0
WebSockets	scontent.xx.fbcdn.net	ipv4	true	57.144.104.128	517	^partitionKey=%28https%2Cfacebook.net%29	0 0x12022 2 0
DNS Lookup	prod.ally.ac	ipv4	true	34.199.65.120 34.193.84.96	586	^partitionKey=%28https%2Cally.ac%29	0 0x2 0 0
Logging	mozilla-ohhttp.fastly-edge.com	ipv4	true	151.101.65.91 151.101.193.91 151.101.129.91 151.101.1.91	524		0 0x12022 2 0
Network ID	www.juicer.io	ipv4	true	104.26.12.87 104.26.13.87 172.67.71.67 2606:4700:20::681a:c57 2606:4700:20::681a:d57 2606:4700:20::ac43:4743	397	^partitionKey=%28https%2Cunm.edu%29	0 0x4012 0 0
	sso.canvaslms.com	ipv4	true	99.83.186.201 75.2.14.209	548	^partitionKey=%28https%2Ccanvaslms.com%29	0 0x12022 2 0
	login.microsoftonline.com	ipv4	true	20.190.151.134 20.190.151.8 20.190.151.67 20.190.151.70 20.190.151.132 20.190.151.6 20.190.151.7 20.190.151.131	720	^partitionKey=%28https%2Cmicrosoftonline.com%29	0 0x12022 2 0
	maxcdn.bootstrapcdn.com	ipv4	true	104.18.10.207 104.18.11.207 2606:4700::6812:acf 2606:4700::6812:bcf	674	^partitionKey=%28https%2Cunm.edu%29	0 0x4012 0 0
	du11hcvx0uqb.cloudfront.net	ipv4	true	99.84.117.48 99.84.117.38 99.84.117.219 99.84.117.20	552	^partitionKey=%28https%2Cunm.edu%29	0 0x12022 2 0
	analytics-ipv6.tiktokw.us	ipv4	true	23.219.89.180 23.219.89.178	517	^partitionKey=%28https%2Cunm.edu%29	0 0x12022 2 0

9 It's much larger than the 7-12 pages I visited.
 I did not manually visit these URLs, but my browser requested them automatically while loading the main pages:
 CDNs: Facebook (fbcdn.net), AWS CloudFront, BootstrapCDN.
 Analytics/tracking — Juicer, TikTok analytics, UNM's Google Tag Manager variant.
 Accessibility and browser services — Ally (ally.ac) and Mozilla Fastly Edge.

No.	Time	Source	Destination	Protocol	Length	Info
17	5.496398500	192.168.64.5	192.168.64.1	DNS	97	Standard query 0x27c7 A mozilla.cloudflare-dns.com OPT
44	5.533579709	192.168.64.1	192.168.64.5	DNS	129	Standard query response 0x27c7 A mozilla.cloudflare-dns.com A 162.159.61.4 A 172.64.41.4 OPT
7393	57.762229956	192.168.64.5	192.168.64.1	DNS	160	Standard query 0x35a8 A connectivity-check.ubuntu.com OPT
7394	57.725892833	192.168.64.1	192.168.64.5	DNS	292	Standard query response 0x35a8 A connectivity-check.ubuntu.com A 185.125.190.18 A 91.189.91.96 A 185.1...
17962	149.896861245	192.168.64.5	192.168.64.1	DNS	97	Standard query 0xebbd AAAA mozilla.cloudflare-dns.com OPT
17968	159.825616178	192.168.64.1	192.168.64.5	DNS	153	Standard query response 0xebbd AAAA mozilla.cloudflare-dns.com AAAA 2803:f800:53::4 AAAA 2a06:98c1:52:...
20971	218.613856586	192.168.64.5	192.168.64.1	DNS	97	Standard query 0xe67f AAAA mozilla.cloudflare-dns.com OPT
20985	218.655409991	192.168.64.1	192.168.64.5	DNS	153	Standard query response 0xe67f AAAA mozilla.cloudflare-dns.com AAAA 2a06:98c1:52::4 AAAA 2803:f800:53:...
24083	267.665843971	192.168.64.5	192.168.64.1	DNS	160	Standard query 0xbf29 AAAA connectivity-check.ubuntu.com OPT
24084	267.665843971	192.168.64.1	192.168.64.5	DNS	430	Standard query response 0xbf29 AAAA connectivity-check.ubuntu.com AAAA 2620:2d:4800:11:96 AAAA 2620:2d...

10 They all have a standard query response pair and all use port 53, but they have various lengths, and some are query type A, others AAAA

4

The screenshot shows a Wireshark capture of network traffic. The packet list pane contains the following entries:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.64.1	192.168.64.255	DB-LSP	271	Dropbox LAN sync Discovery Protocol, JSON
2	18.565769264	192.168.64.5	192.168.64.1	DNS	85	Standard query 0x99b7 A www.iijlab.net OPT
3	18.565966306	192.168.64.5	192.168.64.1	DNS	85	Standard query 0x8fb7 AAAA www.iijlab.net OPT
4	18.942741812	192.168.64.1	192.168.64.5	DNS	119	Standard query response 0x99b7 A www.iijlab.net CNAME sh3.iijlab.net A 202.238.220.76 OPT
5	18.942742270	192.168.64.1	192.168.64.5	DNS	131	Standard query response 0x8fb7 AAAA www.iijlab.net CNAME sh3.iijlab.net AAAA 2001:248:bb82:2706::1:49 OPT
6	18.944367357	192.168.64.5	202.238.220.76	TCP	74	51254 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=89495905 TSecr=0 WS=128
7	19.079251626	202.238.220.76	192.168.64.5	TCP	74	443 → 51254 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=220265807 TSecr=89495905 WS=128
8	19.079341628	192.168.64.5	202.238.220.76	TCP	66	51254 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=89495140 TSecr=220265807
9	19.082089383	192.168.64.5	202.238.220.76	TLSv1.3	474	Client Hello (SNI=www.iijlab.net)

1)

In the capture, the client (192.168.64.5) first issues DNS queries to the local resolver (192.168.64.1) to resolve the hostname www.iijlab.net (frames 2–5). These are sent over UDP, using standard DNS query types: A record for IPv4 (frame 2), AAAA record for IPv6 (frame 3). The resolver responds (frames 4–5) with both IPv4 and IPv6 addresses, confirming that DNS resolution succeeded.

Immediately after DNS resolution, a TCP 3-way handshake occurs between: Source: 192.168.64.5, Destination: 202.238.220.76 and Ports: 51254 → 443

This establishes a reliable connection for HTTPS (HTTP over TLS). The packets: SYN → client initiate connection (frame 6), SYN-ACK → server acknowledgment (frame 7) ACK → client confirmation (frame 8). Following the handshake, we see the TLS Client Hello (frame 9), which begins the encrypted HTTPS session to request the PDF file (romain_sigcomm2017.pdf).

Wireshark - Conversations - enp0s1

Conversation Settings	Ethernet · 1	IPv4 · 1	IPv6	TCP · 1	UDP									
<input type="checkbox"/> Name resolution	Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	
<input type="checkbox"/> Absolute start time	192.168.64.5	51254	202.238.220.76	443	724	2 MB	0	724	100.00%	333	23 kB	391	2 MB	
<input checked="" type="checkbox"/> Limit to display filter														

Wireshark - Conversations - enp0s1

Conversation Settings	Ethernet · 1	IPv4 · 1	IPv6	TCP · 1	UDP									
<input type="checkbox"/> Name resolution	Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	
<input type="checkbox"/> Absolute start time	192.168.64.5	42336	192.168.4.45	22	1,641	2 MB	2	1,641	100.00%	1,308	2 MB	333	28 kB	
<input type="checkbox"/> Limit to display filter														

2) Download B->A = 391 packets
Upload A->B = 1,308 packets

The screenshot displays a network traffic capture in Wireshark. The main pane shows a list of captured packets. The selected packet is a TCP segment from source port 443 to destination port 51254. The packet details pane shows the following information:

- Version: 4
- Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
- Total Length: 339
- Identification: 0xb3ff (46079)
- Flags: 0x0
- Fragment Offset: 0
- Time to Live: 53
- Protocol: TCP (6)
- Header Checksum: 0x28bd [validation disabled]
- [Header checksum status: Unverified]
- Source Address: 202.238.228.76
- Destination Address: 192.168.64.5
- Transmission Control Protocol, Src Port: 443, Dst Port: 51254, Seq: 2401, Ack: 489, Len: 287
- Source Port: 443
- Destination Port: 51254
- [Stream index: 0]
- [Conversation completeness: Complete, WITH_DATA (63)]
- [TCP segment Len: 287]
- Sequence Number: 2401 (relative sequence number)
- Sequence Number (raw): 1308540420
- [Next Sequence Number: 2688 (relative sequence number)]
- Acknowledgment Number: 489 (relative ack number)
- Acknowledgment number (raw): 10171413
- 1000 ... = Header Length: 32 bytes (0)
- Flags: 0x018 (PSH, ACK)
- Window: 586
- [Calculated window size: 64768]
- [Window size scaling factor: 128]
- Checksum: 0x26c7 [unverified]
- [Checksum Status: Unverified]
- Urgent Pointer: 0
- Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
- [Timestamps]

The packet bytes pane shows the raw hex and ASCII data of the segment:

```

0000 6a 23 79 b0 8a fc 8e 2f 57 14 a9 64 08 00 45 00  jby..
0010 01 53 b3 ff 00 09 35 06 28 bd ca ee dc c0 a8  S...
0020 40 05 01 bb c8 36 4d fe be 04 06 10 08 df 00 18  @...
0030 01 fa 26 67 00 01 01 08 0a 0d 20 14 05 05 55  @...
0040 96 f3 17 03 03 01 1a 81 74 73 9b d9 a2 0a c4 82  ....
0050 d6 8a ef 9f ef a8 99 97 57 5f a8 82 35 bb 2c b8  ....
0060 ed 8c fc ee ee 7c 87 0f 3e 1a 27 c7 d2 09 ed 9b  ....
0070 13 69 09 13 0c ef f4 39 e3 2e 58 b3 74 2a d2 fa  ....
0080 c1 22 b5 77 57 0c b6 d8 67 70 80 47 3e 61 f8 9f  ....
0090 f5 41 97 b1 73 c8 c6 33 a9 64 94 96 f7 1a b9 f7  ....
00a0 d9 8c a3 8e 9c 33 0b b3 c1 fd 33 8f 77 fa 20 3b  ....
00b0 b5 08 8a 33 92 7c fc bf da 79 f7 f8 50 43 71 82  ....
00c0 fb d6 8e 2e 87 08 9a 66 8f 1a 32 8d 13 84 b9 0b  ....
00d0 67 b4 68 b9 24 c0 56 cb ab ab c7 54 67 f8 2f c3  gh $
00e0 5d 24 d8 df 1b 47 94 a7 4a 62 cb 1e 93 7a 73 e3  $...
00f0 1a 54 97 27 25 43 17 9a fc ce f9 82 1e ea 04  T...M
0100 13 d1 0e 93 be 7a 2c 83 a9 56 40 09 ab 4c c9 b2  ....
0110 1b 8b 2a 13 0d 99 7b 14 f5 08 0a 4a d8 ac 47 51  ....
0120 b3 af 90 30 7d 1f 78 5c 2e 1b 5b 13 63 ee 10 fa  ....0}
0130 c1 bf e3 e9 01 e9 6e aa f6 be e9 f9 b8 3b ec 9a  ....
0140 4a f7 56 e5 3e 81 44 c6 98 99 08 7a 9d c3 d7 1b  J...>
0150 5e ac c0 64 4d 21 af 8e 10 01 7b 7b 04 1c 9d d3  A...M
0160 94

```

3)

Source Port

443 (HTTPS server port)

Destination Port

51254 (ephemeral client port)

Sequence Number

2401 — identifies the first byte of data in this segment

Acknowledgment Number

489 — acknowledges receipt of all bytes up to 488 from the client

Header Length

32 bytes (indicates TCP header size)

Flags

PSH, ACK (0x18) — data being pushed to the receiver and acknowledged

Window Size Value

64,768 — receiver buffer size before scaling

Window Scaling Factor

128 — actual window = $64,768 \times 128 = 8,388,608$ bytes

Checksum

0x26C7 — used for error detection of header + payload

Urgent Pointer

0 (not used)

TCP Options

Timestamps (TSval = 220206220, TSecr = 89495418) used for RTT measurement

4) Checksum = 0x26c7 [unverified].

5) The checksum is calculated by performing a 1's-complement sum over a the source IP, destination IP, protocol, segment length plus the entire TCP header and data, then taking the 1's-complement of the result. When the receiver recomputes the checksum, any mismatch indicates that bits were altered in transit.

5

Redirection: The RFC talks about “3xx Redirection” codes and how they tell the client to go to a different URI. It includes things like 301 Moved Permanently and 302 Found. The idea is that when a page changes its address, the browser automatically finds the new one instead of showing an error. This is necessary because the web changes all the time, and redirection keeps links from breaking when sites move things around.

Client Error: The “4xx Client Error” section covers problems caused by the request itself, like 400 Bad Request, 403 Forbidden, and 404 Not Found. These tell you that something about your input or permissions is wrong. The RFC says the client shouldn’t repeat the request without fixing it.

This is necessary because it tells users when the problem is on their end instead of wasting the server’s time or leaving them guessing why it failed.

Server Error: The “5xx Server Error” codes mean the request was fine but the server couldn’t complete it. Examples are 500 Internal Server Error and 503 Service Unavailable. It basically tells the client, “we understand what you wanted, but something broke here.”

This is necessary because it separates server failures from client mistakes and helps developers figure out what’s actually wrong in a networked system.

Caching: Caching is about saving responses so they don’t have to be downloaded again. The RFC talks about Cache-Control, ETag, and validation headers like Last-Modified. It’s all about saving time and bandwidth by letting browsers reuse what they already have.

This is necessary because it makes the web faster and reduces load on servers instead of re-downloading the same content constantly.

Expiration: This is how servers tell caches when stored data becomes “stale.” The RFC mentions headers like Expires and max-age to control that. It uses caching to manage freshness. This is necessary because it keeps information up to date without making clients check the server every second.

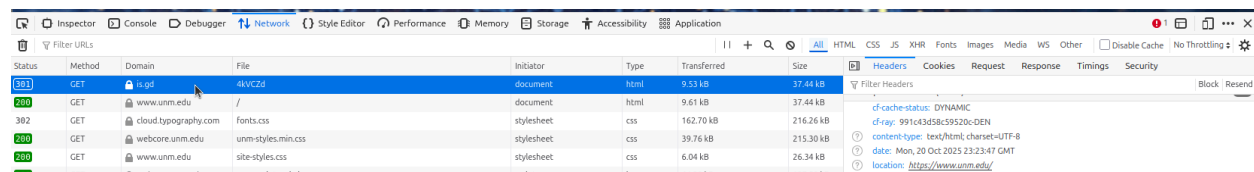
Security Considerations: This section lists HTTP risks, like caching private data, using unencrypted connections, or trusting headers too much. It suggests using TLS (HTTPS) and validating inputs. This is necessary because HTTP by itself isn’t secure, so these guidelines help prevent people from intercepting, changing, or stealing data during communication.

6

1,2,3,4,5

<https://www.unm.edu/> : <https://is.gd/4kVCZd>, tinyurl.com/95rch3ce

<https://www.usenix.org/> : <https://is.gd/T9Pugg>, tinyurl.com/4eh35n9d



Status	Method	Domain	File	Initiator	Type	Transferred	Size	Headers	Cookies	Request	Response	Timings	Security
301	GET	is.gd	4kVCZd	document	html	9.53 kB	37.44 kB						
200	GET	www.unm.edu	/	document	html	9.61 kB	37.44 kB	cf-cache-status: DYNAMIC					
302	GET	cloud.typography.com	fonts.css	stylesheet	css	162.70 kB	216.26 kB	cf-ray: 991c43e58c59120c-DEN					
200	GET	webcore.unm.edu	unm-styles.min.css	stylesheet	css	39.76 kB	215.30 kB	content-type: text/html; charset=UTF-8					
200	GET	www.unm.edu	site-styles.css	stylesheet	css	6.04 kB	26.34 kB	date: Mon, 20 Oct 2025 23:23:47 GMT					

6(a) <https://is.gd/4kVCZd>, Status 301, Location <https://www.unm.edu/>
<https://www.unm.edu/> Status 200, Location reached

Status	Method	Domain	File	Initiator	Type	Transferred	Size
301	GET	is.gd	4kVCZd	document	html	9.53 kB	37.44 kB
200	GET	www.unm.edu	/	document	html	9.53 kB	37.44 kB
302	GET	cloud.typography.com	fonts.css	stylesheet	css	162.70 kB	216.26 kB
200	GET	webcore.unm.edu	unm-styles.min.css	stylesheet	css	39.76 kB	215.30 kB
200	GET	www.unm.edu	site-styles.css	stylesheet	css	6.04 kB	26.34 kB
200	GET	webcore.unm.edu	unm-scripts.min.js	script	js	44.38 kB	137.58 kB
200	GET	www.unm.edu	site.js	script	js	2.03 kB	4.83 kB
200	GET	www.unm.edu	jquery.waypoints.min.js	script	js	3.06 kB	8.84 kB
200	GET	www.unm.edu	velocity.min.js	script	js	12.77 kB	34.86 kB
200	GET	www.unm.edu	velocity.ui.min.js	script	js	3.29 kB	13.26 kB

7(a) Just the shortener

8(a) When I visited the short link <https://is.gd/4kVCZd>, the shortener domain (is.gd) set a cookie. The Set-Cookie header appeared in the response from the shortener before it redirected me to <https://www.unm.edu/>. The cookie from is.gd was not used by UNM's server, it just helped the shortener track that the redirect was successfully followed. It lets the shortener measure link usage and prevent abuse without affecting the target website.

9(a) The first response from is.gd had Status 301 (Moved Permanently) with a Location header pointing to <https://www.unm.edu/>. That tells the browser that the short link has a permanent destination, so it should automatically follow that new URL and possibly cache it for future use. The second response from www.unm.edu was Status 200 (OK), which means the request succeeded and the actual page content was delivered. The Location header specifies exactly where the browser should go next. Without it, the 301 would just tell the browser "this isn't here" with no destination.

Status	Method	Domain	File	Initiator	Type	Transferred	Size
301	GET	tinyurl.com	95rch3ce	document	html	9.98 kB	37.44 kB
200	GET	www.unm.edu	/	document	html	cached	37.44 kB
302	GET	cloud.typography.com	fonts.css	stylesheet	css	162.70 kB	216.26 kB
200	GET	webcore.unm.edu	unm-styles.min.css	stylesheet	css	cached	215.30 kB
200	GET	www.unm.edu	site-styles.css	stylesheet	css	cached	26.34 kB
200	GET	webcore.unm.edu	unm-scripts.min.js	script	js	cached	137.58 kB
200	GET	www.unm.edu	site.js	script	js	cached	4.83 kB
200	GET	www.unm.edu	jquery.waypoints.min.js	script	js	cached	8.84 kB
200	GET	www.unm.edu	velocity.min.js	script	js	cached	34.86 kB
200	GET	www.unm.edu	velocity.ui.min.js	script	js	cached	13.26 kB
200	GET	www.unm.edu	bythenumbers.js	script	js	cached	1.45 kB
200	GET	www.unm.edu	nso-button.jpg	img	jpeg	cached	181.59 kB
200	GET	www.unm.edu	lobograd-unm-edu.jpg	img	jpeg	cached	51.33 kB
200	GET	assets.juicer.io	embed-no-jquery.js	script	js	cached	604.59 kB

Response Headers (957 B)

- age: 43
- alt-svc: h3="443", ma=86400
- cache-control: max-age=0, must-revalidate, no-cache
- cf-cache-status: HIT
- cf-ray: 991ca1bde944e73d-DEN
- content-type: text/html; charset=utf-8
- date: Tue, 21 Oct 2025 00:27:53 GMT
- location: <https://www.unm.edu/>

6(b) tinyurl.com/95rch3ce Status 301, Location <https://www.unm.edu/> <https://www.unm.edu/> Status 200, Location reached

Status	Method	Domain	File	Initiator	Type	Transferred	Size
301	GET	tinyurl.com	95rch3ce	document	html	9.98 kB	37.44 kB
200	GET	www.unm.edu	/	document	html	cached	37.44 kB
302	GET	cloud.typography.com	fonts.css	stylesheet	css	162.70 kB	216.26 kB
200	GET	webcore.unm.edu	unm-styles.min.css	stylesheet	css	cached	215.30 kB
200	GET	www.unm.edu	site-styles.css	stylesheet	css	cached	26.34 kB
200	GET	webcore.unm.edu	unm-scripts.min.js	script	js	cached	137.58 kB
200	GET	www.unm.edu	site.js	script	js	cached	4.83 kB
200	GET	www.unm.edu	jquery.waypoints.min.js	script	js	cached	8.84 kB
200	GET	www.unm.edu	velocity.min.js	script	js	cached	34.86 kB
200	GET	www.unm.edu	velocity.ui.min.js	script	js	cached	13.26 kB
200	GET	www.unm.edu	bythenumbers.js	script	js	cached	1.45 kB
200	GET	www.unm.edu	nso-button.jpg	img	jpeg	cached	181.59 kB
200	GET	www.unm.edu	lobograd-unm-edu.jpg	img	jpeg	cached	51.33 kB
200	GET	assets.juicer.io	embed-no-jquery.js	script	js	cached	604.59 kB
200	CFT	assets.juicer.io	embed.css	stylesheet	css	cached	100.75 kB

Response Headers (957 B)

- age: 43
- alt-svc: h3="443", ma=86400
- cache-control: max-age=0, must-revalidate, no-cache
- cf-cache-status: HIT
- cf-ray: 991ca1bde944e73d-DEN
- content-type: text/html; charset=utf-8
- date: Tue, 21 Oct 2025 00:27:53 GMT
- location: <https://www.unm.edu/>

Status	Method	Domain	File	Initiator	Type	Transferred	Size
301	GET	tinyurl.com	95rch3ce	document	html	9.98 kB	37.44 kB
200	GET	www.unm.edu	/	document	html	cached	37.44 kB
200	GET	cloud.typography.com	fonts.css	stylesheet	css	162.70 kB	216.26 kB
200	GET	webcore.unm.edu	unm-styles.min.css	stylesheet	css	cached	215.30 kB
200	GET	www.unm.edu	site-styles.css	stylesheet	css	cached	26.34 kB
200	GET	webcore.unm.edu	unm-scripts.min.js	script	js	cached	137.58 kB
200	GET	www.unm.edu	site.js	script	js	cached	4.83 kB
200	GET	www.unm.edu	jquery.waypoints.min.js	script	js	cached	8.84 kB
200	GET	www.unm.edu	velocity.min.js	script	js	cached	34.86 kB
200	GET	www.unm.edu	velocity.ui.min.js	script	js	cached	13.26 kB
200	GET	www.unm.edu	bythenumbers.js	script	js	cached	1.45 kB
200	GET	www.unm.edu	nso-button.jpg	img	jpeg	cached	181.59 kB
200	GET	www.unm.edu	lobograd-unm-edu.jpg	img	jpeg	cached	51.33 kB
200	GET	assets.jaice.io	embed-no-jquery.js	script	js	cached	604.59 kB
200	GET	assets.jaice.io	embed.css	stylesheet	css	cached	100.22 kB
200	GET	www.unm.edu	sun.png	img	png	cached	2.70 kB
200	GET	www.unm.edu	snow.png	img	png	cached	3.02 kB

7(b) The Network tab shows that tinyurl.com sent a Set-Cookie header before redirecting to <https://www.unm.edu/>. UNM issued some new cookies but they weren't related to TinyURL, they were from accessing the site a second time.

8(b) When I accessed <https://tinyurl.com/95rch3ce>, the browser received a response from the TinyURL server with a Set-Cookie: header. The cookies here seem to contain encoded session or tracking information, probably used by TinyURL to count link clicks and monitor redirect performance.

9(b) The redirect chain had two main responses: 301 (Moved Permanently) from tinyurl.com with a Location header pointing to <https://www.unm.edu/> and 200 (OK) from the final UNM site, meaning the content was successfully retrieved. The 301 code tells the browser to automatically go to the new URL and treat it as the permanent destination for that short link. The Location header provides the exact address of the target page.

Status	Method	Domain	File	Initiator	Type	Transferred	Size
301	GET	is.gd	T9Pugg	document	html	7.63 kB	30.30 kB
200	GET	www.usenix.org	/	document	html	8.25 kB	30.30 kB
200	GET	www.usenix.org	normalize.css?74g7o1	stylesheet	css	3.08 kB	7.72 kB
200	GET	www.usenix.org	system.css?74g7o1	stylesheet	css	3.28 kB	7.75 kB
200	GET	www.usenix.org	system.theme.css?74g7o1	stylesheet	css	3.57 kB	9.75 kB
200	GET	www.usenix.org	messages.theme.css?74g7o1	stylesheet	css	1.82 kB	3.07 kB
200	GET	www.usenix.org	comment.css?74g7o1	stylesheet	css	1.08 kB	250 B
200	GET	www.usenix.org	date.css?74g7o1	stylesheet	css	1.97 kB	3.29 kB
200	GET	www.usenix.org	field.css?74g7o1	stylesheet	css	1.18 kB	767 B
200	GET	www.usenix.org	og.css?74g7o1	stylesheet	css	1.01 kB	97 B
200	GET	www.usenix.org	paragraphs.css?74g7o1	stylesheet	css	1.93 kB	4.21 kB
200	GET	www.usenix.org	limezone-picker.css?74g7o1	stylesheet	css	1.16 kB	502 B
200	GET	www.usenix.org	usenix_logo_preview.css?74g7o1	stylesheet	css	1.38 kB	1.44 kB
200	GET	www.usenix.org	user.css?74g7o1	stylesheet	css	2.01 kB	3.92 kB
200	GET	www.usenix.org	workflow_admin_ui.css?74g7o1	stylesheet	css	1.11 kB	351 B
200	GET	www.usenix.org	views.css?74g7o1	stylesheet	css	1.53 kB	2.49 kB

Note Status, Location and cookie are all present here. And we saw before that usenix itself doesn't have cookies.

6(c) <https://is.gd/T9Pugg>, Status 301, Location <https://www.usenix.org/>
<https://www.usenix.org/> Status 200, Location reached

7(c) Just the shortener

8(c) When I visited the short link <https://is.gd/T9Pugg>, the shortener domain (is.gd) set a cookie. The Set-Cookie header appeared in the response from the shortener before it redirected me to <https://www.usenix.org/>. The cookie from is.gd was not used by usenix's server, it just helped the shortener track that the redirect was successfully followed. It lets the shortener measure link usage and prevent abuse without affecting the target website.

9(c) The first response from is.gd had Status 301 (Moved Permanently) with a Location header pointing to <https://www.usenix.org/>. That tells the browser that the short link has a permanent destination, so it should automatically follow that new URL and possibly cache it for future use. The second response from www.unm.edu was Status 200 (OK), which means the request succeeded and the actual page content was delivered. The Location header specifies exactly where the browser should go next. Without it, the 301 would just tell the browser “this isn’t here” with no destination.

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Filter Headers
301	GET	tinyurl.com	4eh35n9d	document	html	7.70 kB	29.38 kB	
200	GET	www.usenix.org	/	document	html	7.88 kB	29.38 kB	alt-svc: h3="443"; ma=86400
200	GET	www.usenix.org	normalize.css?4gTo1	stylesheet	css	cached	7.72 kB	cache-control: max-age=0, must-revalidate, no-cache, no-store, private
200	GET	www.usenix.org	system.css?4gTo1	stylesheet	css	cached	7.75 kB	cf-cache-status: MISS
200	GET	www.usenix.org	system.theme.css?4gTo1	stylesheet	css	cached	9.75 kB	cf-ray: 9916c548b92a734c-GEN
200	GET	www.usenix.org	messages.theme.css?4gTo1	stylesheet	css	cached	3.07 kB	content-type: text/html; charset=utf-8
200	GET	www.usenix.org	comment.css?4gTo1	stylesheet	css	cached	250 B	date: Tue, 21 Oct 2025 00:52:14 GMT
								location: https://www.usenix.org/

6(d) tinyurl.com/4eh35n9d Status 301, Location <https://www.usenix.org/>
<https://www.usenix.org/> Status 200, Location reached

Status	Method	Domain	File	Initiator	Type	Transferred	Size
301	GET	tinyurl.com	4eh35n9d	document	html	7.70 kB	29.38 kB
200	GET	www.usenix.org	/	document	html	7.88 kB	29.38 kB
200	GET	www.usenix.org	normalize.css?4gTo1	stylesheet	css	cached	7.72 kB
200	GET	www.usenix.org	system.css?4gTo1	stylesheet	css	cached	7.75 kB
200	GET	www.usenix.org	system.theme.css?4gTo1	stylesheet	css	cached	9.75 kB
200	GET	www.usenix.org	messages.theme.css?4gTo1	stylesheet	css	cached	3.07 kB
200	GET	www.usenix.org	comment.css?4gTo1	stylesheet	css	cached	250 B
200	GET	www.usenix.org	date.css?4gTo1	stylesheet	css	cached	3.29 kB
200	GET	www.usenix.org	field.css?4gTo1	stylesheet	css	cached	767 B
200	GET	www.usenix.org	og.css?4gTo1	stylesheet	css	cached	97 B
200	GET	www.usenix.org	paragaphs.css?4gTo1	stylesheet	css	cached	4.21 kB
200	GET	www.usenix.org	bimstone-pickles.css?4gTo1	stylesheet	css	cached	502 B
200	GET	www.usenix.org	usenix_blog_preview.css?4gTo1	stylesheet	css	cached	1.44 kB
200	GET	www.usenix.org	user.css?4gTo1	stylesheet	css	cached	3.92 kB
200	GET	www.usenix.org	workflow_admin_ui.css?4gTo1	stylesheet	css	cached	351 B
200	GET	www.usenix.org	views.css?4gTo1	stylesheet	css	cached	2.49 kB

And once again, we saw earlier that usenix itself doesnt have cookies.

7(d) The Network tab shows that tinyurl.com sent a Set-Cookie header before redirecting to <https://www.usenix.org/>. We saw earlier that usenix itself doesnt have cookies.

8(d) When I accessed tinyurl.com/4eh35n9d, the browser received a response from the TinyURL server with a Set-Cookie: header. The cookies here seem to contain encoded session or tracking information, probably used by TinyURL to count link clicks and monitor redirect performance.

9(b) The redirect chain had two main responses: 301 (Moved Permanently) from tinyurl.com with a Location header pointing to <https://www.usenix.org/> and 200 (OK) from the final usenix site, meaning the content was successfully retrieved. The 301 code tells the browser to automatically go to the new URL and treat it as the permanent destination for that short link. The Location header provides the exact address of the target page.