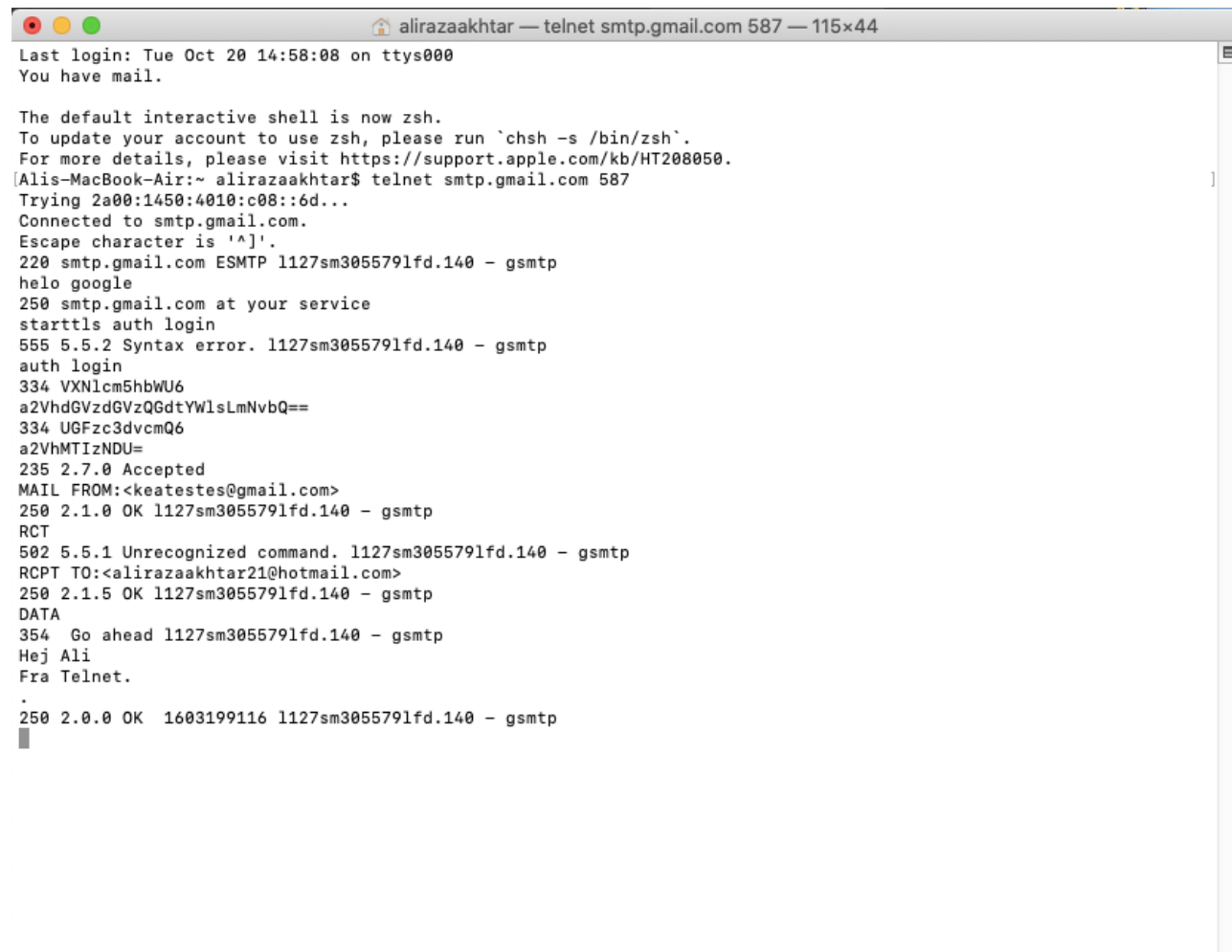


Ali Raza Akhtar
Tech2
DAT19A
Mandatory I

1: Deploy et Spring project på Heroku and del linket:

- <http://helloali123.herokuapp.com/>

2: Send en mail over Telnet:



keatestes@gmail.com

(Intet emne)

Til: Undisclosed recipients;;

Indbak... - Exchange 15.05

K

Hej Ali
Fra Telnet.

3: En kort beskrivelse af “A day in the life of a web page request”

Når Bob forbinder hans computer til WIFI-router, så får den tildelt en IP-adresse. Dette sker igennem DHCP service, hvor Bob's computer skaber en DHCP-request som bliver sendt med UDP segment. UDP-segment bliver placeret i en IP-datagram, hvor den efterfølgende bliver sat ind i et Ethernet Frame, som har en MAC-adresse som destination. Broadcast Ethernet-rammen, der indeholder DHCP-request, er den første ramme, der sendes af Bobs bærbare computer til Ethernet-switchen.

Routeren modtager Broadcast Ethernet-rammen indeholdende DHCP-request. DHCP-serveren behandler nu DHCP-anmodningsmeddelelsen.

DHCP-meddelelsen placeres i et UDP-segment, der placeres i et IP-datagram, der placeres inde i en Ethernet-ramme. Ethernet-rammen har en kilde-MAC-adresse på routerens interface til hjemmenetværket (00: 22: 6B: 45: 1F: 1B) og en destinations-MAC-adresse på Bobs bærbare computer (00: 16: D3: 23: 68: 8A).

Ethernet-rammen, der indeholder DHCP ACK, sendes af routeren til kontakten.

Bobs bærbare computer modtager Ethernet-rammen indeholdende DHCP ACK, udtrækker IP-datagrammet fra Ethernet-rammen, udtrækker UDP-segmentet fra IP-datagrammet og udtrækker DHCP ACK-meddelelsen fra UDP-segmentet. Bobs DHCP-klient registrerer derefter sin IP-adresse og IP-adressen på dens DNS-server. Bobs bærbare computer er klar til at begynde at behandle hentningen af websiden.

For at komme på google.dk, så skal Bobs webbrowser starte processen med at oprette et TCP-socket, der bruges til at sende HTTP-anmodningen til www.google.dk. For at oprette socketen skal Bobs bærbare computer kende IP-adressen på www.google.com.

Operativsystemet på Bobs bærbare computer opretter således en DNS-forespørgselsmeddelelse.

Denne DNS-meddelelse placeres derefter i et UDP-segment med en destinationsport på 53 (DNS-server). UDP-segmentet placeres derefter i et IP-datagram

Bobs bærbare computer placerer derefter datagrammet, der indeholder DNS-

forespørgselsmeddelelsen, i en Ethernet-ramme. Denne ramme sendes til gatewayrouteren i Bobs skoles netværk. For at få MAC-adressen på gateway-routeren skal Bobs bærbare computer bruge ARP-protokollen

Bobs bærbare computer kan nu adressere Ethernet-rammen, der indeholder DNS-forespørgslen, til gateway-routerens MAC-adresse.

Gateway-routeren modtager rammen og udtrækker IP-datagrammet, der indeholder DNS-forespørgslen. Routeren ser destinationsadressen på dette datagram og bestemmer at datagrammet skal sendes til den venstre router i Comcast-netværket i.

Til sidst ankommer IP-datagrammet, der indeholder DNS-forespørgslen, til DNS-serveren. DNS-serveren søger på navnet www.google.com i sin DNS-database og finder IP-adressen til www.google.com. DNS-serveren danner en DNS-response, der indeholder denne IP-adresse, og placerer DNS-response i et UDP-segment og segmentet i et IP-datagram adresseret til Bobs bærbare computer. Dette datagram videresendes tilbage via Comcast-netværket til skolens router og derfra via Ethernet-switchen til Bobs bærbare computer.

Bobs bærbare computer udtrækker IP-adressen på serveren www.google.com fra DNS-response.

Nu hvor Bobs bærbare computer har IP-adressen www.google.com, kan den oprette TCP-socket, der bruges til at sende HTTP GET-request til www.google.com. Når Bob opretter TCP-socketet, skal TCP i Bobs bærbare computer først udføre et 3. vejshåndtryk med TCP på www.google.com. Bobs bærbare computer opretter således først et TCP SYN-segment og placerer TCP-segmentet inde i et IP-datagram med en destinations-IP-adresse på www.google.com.

TCP SYN-meddelelsen ekstraheres fra datagrammet. Der oprettes et forbindelse socket til TCP-forbindelsen mellem Google HTTP-serveren og Bobs bærbare computer. Et TCP SYNACK-afsnit genereres, placeres inde i et datagram adresseret til Bobs bærbare computer og placeres til sidst inde i en linklagsramme, der passer til det link, der forbinder www.google.com til sin første-hop-router.

Nu er Bobs bærbare computer klar til at oprette browser HTTP GET- request, der indeholder den URL, der skal hentes. HTTP GET- request skrives derefter ind i soklen, hvor GET- request bliver sat ind i en TCP-segment. TCP-segmentet placeres i et datagram og sendes og leveres til www.google.com.

HTTP-serveren på www.google.com læser HTTP GET- request fra TCP-sokkel, opretter en HTTP-response placerer det ønskede websideindhold i selve HTTP-response og sender meddelelsen til TCP-soklen.

Datagrammet, der indeholder HTTP-response, videresendes via Google-, Comcast- og skolens netværk og ankommer til Bobs bærbare computer. Bobs webbrowserprogram læser HTTP-svaret fra soklen, udtrækker html til websiden fra selve HTTP-svaret og viser endelig websiden

4: hvad er internettet til ikke teknisk person

- Internettet er at et system, hvor man kan komme på forskellige hjemmesider ved at indtaste deres domain adresse. Siderne indeholder forskellige funktioner og information fra hele verden alt afhængigt af hvad brugeren søger. Siderne bliver vist ved at brugeren sender et request til domain adresse, hvor den efterfølgende returnerer den ønsket side. Internet gør det muligt at sende mail til hinanden over hele verden på få sekunder.

5: hvad er internettet til en teknisk person

- Internet er et computernetværk, hvor flere computer kommunikerer med hinanden ved hjælp fra 2 protokoller TCP og IP. Når vi skal besøge en hjemmeside, så taster vi dens DNS-navn ind på browseren som i virkeligheden er dens IP-adresse, hvor hjemmesiden bliver kørt af en computer. For at få forbindelse til computeren bruger vi TCP som opretter forbindelse til IP-adressen, som så sender noget data tilbage der bliver vist på skærmen

6: DHCP

- Dynamic host configuration protocol har til formål at tildele IP-adresser til hosts. Processen foregår således at host ser efter DHCP server, som tilbyder en adresse. Hosten anmoder om at leje adressen, hvor DHCP-serveren sender en IP-adresse tilbage til hosten

7: NAT

- Network address translation oversætter public IP-adresser til private IP-adresser og omvendt. Dette skyldes begrænset public IP-adresser, hvilket betyder at alle enheder ikke kan få hvert sit public IP-adresse. Derfor er det mere optimalt at f.eks. en virksomhed har en router med en public IP-adresse, hvor alle enheder i virksomheden får tildelt en privat IP-adresse af router. Når disse privat IP-adresser skal kommunikere med andre computer,

så skal til de igennem router, hvor de bliver oversat til en public IP-adresse som routeren besidder og sender videre. Når nogen prøver at kommunikere med virksomheden private IP-adresse, så skal public IP-adressen igennem routeren, hvor den bliver oversat af NAT til en privat IP-adresse som henviser til den pågældende enhed som der skal kommunikere med.

8: Udvikling af internettet

- For blot 60 år siden bestod computer af tunge kropsdele som vejede op til flere ton. En computer havde endnu ikke kommunikation til andre computer, men da behovet for at dele information opstod, så blev den første forbindelse udviklet mellem 2 computer. Her blev verdens første besked sendt fra en computer til en anden computer, hvilket var en begyndelse på internettet. Man begyndte stille og roligt i 1970'erne at flere computer kunne kommunikere sammen på et netværk. I 1983 kom der et gennembrud, hvor flere computer og netværker kunne kommunikere med hinanden ved hjælp fra TCP/IP-protokol. De første små hjemmesider begyndte at dukke op i 1990'erne, hvor internettet begyndte at blive mere allemandsseje. Internettet har nu udviklet sig utrolig stærkt og det kan man se på af resultat i dag.

Ali Raza Akhtar

Tech2

DAT19A

Wireshark [NAT]

1.What is the IP address of the client?

192.168.43.75

2.The client actually communicates with several different Google servers in order to implement “safe browsing.” (See extra credit section at the end of this lab). The main Google server that will serve up the main Google web page has IP address 64.233.169.104. In order to display only those frames containing HTTP messages that are sent to/from this Google, server, enter the expression “http && ip.addr == 64.233.169.104” (without quotes) into the Filter: field in Wireshark

The image displays two screenshots of the Wireshark network protocol analyzer. Both screenshots show a packet capture file named 'NAT_home_side.pcap'. The top screenshot shows the initial capture with a filter applied to show only HTTP traffic to and from 64.233.169.104. The bottom screenshot shows the same capture with the filter applied, displaying the same list of frames.

Top Screenshot:

- Filter: `http && ip.addr == 64.233.169.104`
- Packet List (Frames 56-112):

No.	Time	Source	Destination	Protocol	Length	Info
56	7.109267	192.168.1.100	64.233.169.104	HTTP	689	GET / HTTP/1.1
60	7.158797	64.233.169.104	192.168.1.100	HTTP	814	HTTP/1.1 200 OK (text/html)
62	7.281399	192.168.1.100	64.233.169.104	HTTP	719	GET /intl/en_ALL/images/logo.gif HTTP/1.1
73	7.349451	64.233.169.104	192.168.1.100	HTTP	226	HTTP/1.1 200 OK (GIF89a)
75	7.370185	192.168.1.100	64.233.169.104	HTTP	809	GET /extern_js/f/CgJlbhICdXMrMao4NUAILCswDjgHLCswFjgQLCswFzgDLCswGDgELCswGTgJLCswHTgJLCswJTjJiAEsKzAmOAU...
92	7.448649	64.233.169.104	192.168.1.100	HTTP	648	HTTP/1.1 200 OK (text/javascript)
94	7.492324	192.168.1.100	64.233.169.104	HTTP	695	GET /extern_chrome/ee36edbd3c16a1c5.js HTTP/1.1
100	7.537353	64.233.169.104	192.168.1.100	HTTP	870	HTTP/1.1 200 OK (text/html)
107	7.652836	192.168.1.100	64.233.169.104	HTTP	712	GET /images/nav_logo7.png HTTP/1.1
112	7.682361	192.168.1.100	64.233.169.104	HTTP	806	GET /csi?v=3&s=webhp&action=&tran=undefined&e=17259,21588,21766,21920&ei=2502Ssb1G4_CeJvxxaM0&rt=prt.128...

Bottom Screenshot:

- Filter: `http && ip.addr == 64.233.169.104`
- Packet List (Frames 56-112):

No.	Time	Source	Destination	Protocol	Length	Info
56	7.109267	192.168.1.100	64.233.169.104	HTTP	689	GET / HTTP/1.1
60	7.158797	64.233.169.104	192.168.1.100	HTTP	814	HTTP/1.1 200 OK (text/html)
62	7.281399	192.168.1.100	64.233.169.104	HTTP	719	GET /intl/en_ALL/images/logo.gif HTTP/1.1
73	7.349451	64.233.169.104	192.168.1.100	HTTP	226	HTTP/1.1 200 OK (GIF89a)
75	7.370185	192.168.1.100	64.233.169.104	HTTP	809	GET /extern_js/f/CgJlbhICdXMrMao4NUAILCswDjgHLCswFjgQLCswFzgDLCswGDgELCswGTgJLCswHTgJLCswJTjJiAEsKzAmOAU...
92	7.448649	64.233.169.104	192.168.1.100	HTTP	648	HTTP/1.1 200 OK (text/javascript)
94	7.492324	192.168.1.100	64.233.169.104	HTTP	695	GET /extern_chrome/ee36edbd3c16a1c5.js HTTP/1.1
100	7.537353	64.233.169.104	192.168.1.100	HTTP	870	HTTP/1.1 200 OK (text/html)
107	7.652836	192.168.1.100	64.233.169.104	HTTP	712	GET /images/nav_logo7.png HTTP/1.1
112	7.682361	192.168.1.100	64.233.169.104	HTTP	806	GET /csi?v=3&s=webhp&action=&tran=undefined&e=17259,21588,21766,21920&ei=2502Ssb1G4_CeJvxxaM0&rt=prt.128...

Ali Raza Akhtar

Tech2

DAT19A

Der bliver retuneret samme resultat.

3. Consider now the HTTP GET sent from the client to the Google server (whose IP address is IP address 64.233.169.104) at time 7.109267. What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET?

The image shows two screenshots of the Wireshark network protocol analyzer. The top screenshot displays a list of captured packets, with packet 56 selected. The bottom screenshot shows the detailed view of packet 56, which is an HTTP GET request.

Packet List (Top Screenshot):

No.	Time	Source	Destination	Protocol	Length	Info
56	7.109267	192.168.1.100	64.233.169.104	HTTP	689	GET / HTTP/1.1
60	7.158797	64.233.169.104	192.168.1.100	HTTP	814	HTTP/1.1 200 OK (text/html)
62	7.281399	192.168.1.100	64.233.169.104	HTTP	719	GET /intl/en_ALL/images/logo.gif HTTP/1.1
73	7.349451	64.233.169.104	192.168.1.100	HTTP	226	HTTP/1.1 200 OK (GIF89a)
75	7.370185	192.168.1.100	64.233.169.104	HTTP	809	GET /extern_js/f/CgJlbnhldXMrMAo4NUAILCswDjgHLCswFjgQLCswFzgDLCSwGDgELCswGTgJLCswHTgZLCswJTjJiAEsKzAmOAU...
92	7.448649	64.233.169.104	192.168.1.100	HTTP	648	HTTP/1.1 200 OK (text/javascript)
94	7.492324	192.168.1.100	64.233.169.104	HTTP	695	GET /extern_chrome/ee36edbd3c16a1c5.js HTTP/1.1
100	7.537353	64.233.169.104	192.168.1.100	HTTP	870	HTTP/1.1 200 OK (text/html)
107	7.652836	192.168.1.100	64.233.169.104	HTTP	712	GET /images/nav_logo7.png HTTP/1.1
112	7.682361	192.168.1.100	64.233.169.104	HTTP	806	GET /csi?v=3&s=webhp&action=&tran=undefined&e=17259,21588,21766,21920&ei=2502Ssb1G4_CeJvxxaMO&rt=prt.128...

Packet Details (Bottom Screenshot - Packet 56):

- Fragment offset: 0
- Time to live: 128
- Protocol: TCP (6)
- Header checksum: 0xa94a [validation disabled]
- [Header checksum status: Unverified]
- Source: 192.168.1.100
- Destination: 64.233.169.104
- > Transmission Control Protocol, Src Port: 4335, Dst Port: 80, Seq: 1, Ack: 1, Len: 635
- > Hypertext Transfer Protocol

Packet Bytes (Bottom Screenshot):

0000 00 22 6b 45 1f 1b 00 22 68 0d ca 8f 08 00 45 00 ... "kE..." h.....E.
0010 02 a3 a2 ac 40 00 80 06 a9 4a c0 a8 01 64 40 e9@... J...de.
0020 a9 68 10 ef 00 50 f8 32 36 e5 e9 4f 38 95 50 18h...P.2 6...08-P.
0030 fe 14 ae f3 00 00 47 45 54 20 2f 20 48 54 54 50GE T / HTTP
0040 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 77 77 2e ... /1.1..Ho st: www.
0050 67 6f 6f 67 6c 65 2e 63 6f 6d 0d 0a 55 73 65 72 ... google.c om..User
0060 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f ... -Agent: Mozilla/
0070 35 2e 30 20 28 57 69 6e 64 6f 77 73 3b 20 55 3b ... 5.0 (win dows); U;
0080 20 57 69 6e 64 6f 77 73 20 4e 54 20 35 2e 31 3b ... Windows NT 5.1;
0090 20 65 6e 2d 55 53 3b 20 72 76 3a 31 2e 39 2e 30 ... en-US; rv:1.9.0

Ali Raza Akhtar

Tech2

DAT19A

Source IP Address: 192.168.43.75

Port: 4335 Destination

IP Address: 64.233.169.104,

Port: 80

4. At what time is the corresponding 200 OK HTTP message received from the Google server? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message?

Time corresponding 200 OK: 7.158797

Src: 64.233.169.104

Dst: 192.168.43.75

TCP src: 80

TCP Destination: 59909

5. Recall that before a GET command can be sent to an HTTP server, TCP must first set up a connection using the three-way SYN/ACK handshake. At what time is the client-to-server TCP SYN segment sent that sets up the connection used by the GET sent at time 7.109267?

The image shows a Wireshark packet capture of a network session. The top pane displays a list of packets. Packet 53, at time 7.109267, is a TCP SYN segment from 192.168.1.100 to 64.233.169.104. Packet 60, at time 7.158797, is an HTTP 200 OK message from 64.233.169.104 to 192.168.1.100. The middle pane shows the details of the selected packet (Frame 53), including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol fields. The bottom pane shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
46	2.064877	74.125.106.31	192.168.1.100	HTTP	1089	HTTP/1.1 200 OK (application/vnd.google.safebrowsing-chunk)
47	2.178596	192.168.1.100	74.125.106.31	TCP	54	4331 → 80 [ACK] Seq=2876 Ack=20452 Win=260176 Len=0
53	7.075657	192.168.1.100	64.233.169.104	TCP	66	4335 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 SACK_PERM=1
54	7.108986	64.233.169.104	192.168.1.100	TCP	66	80 → 4335 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 SACK_PERM=1 WS=64
55	7.109053	192.168.1.100	64.233.169.104	TCP	54	4335 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
56	7.109267	192.168.1.100	64.233.169.104	HTTP	689	GET / HTTP/1.1
57	7.140728	64.233.169.104	192.168.1.100	TCP	60	80 → 4335 [ACK] Seq=1 Ack=636 Win=7040 Len=0
58	7.158432	64.233.169.104	192.168.1.100	TCP	1484	80 → 4335 [ACK] Seq=1 Ack=636 Win=7040 Len=1430 [TCP segment of a reassembled PDU]
59	7.158761	64.233.169.104	192.168.1.100	TCP	1484	80 → 4335 [ACK] Seq=1431 Ack=636 Win=7040 Len=1430 [TCP segment of a reassembled PDU]
60	7.158797	64.233.169.104	192.168.1.100	HTTP	814	HTTP/1.1 200 OK (text/html)

> Frame 53: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
> Ethernet II, Src: HonHaiPr_0d:ca:8f (00:22:68:0d:ca:8f), Dst: Cisco-Li_45:1f:1b (00:22:6b:45:1f:1b)
> Internet Protocol Version 4, Src: 192.168.1.100, Dst: 64.233.169.104
> Transmission Control Protocol, Src Port: 4335, Dst Port: 80, Seq: 0, Len: 0
> TRANSMISSION RTE Data

0000 00 22 6b 45 1f 1b 00 22 68 0d ca 8f 00 00 45 00 - "kE..." h.....E-
0010 00 34 a2 aa 40 00 80 06 ab bb c0 a8 01 64 40 e9 - 4..@.....d@-
0020 a9 68 10 ef 00 50 f8 32 36 e4 00 00 00 00 80 02 - h...P 2 6
0030 ff ff 82 62 00 00 02 04 05 b4 01 03 03 02 01 01 - ...b.....
0040 64 02 ..

SYN-TIME: 7.07567 sekunder.

SYN:SRC: 192.168.43.75 SYN SRC:PORT: 4335

SYN:DST:IP: 64.233.169.104 SYN:DST:PORT 80

ACK-TIME: 7.108986

ACK:SRC 64.233.169.104 ACK: SRC:PORT: 80

ACK:DST:IP: 192.168.1.100 ACK:SRC:PORT: 4335

6. In the NAT_ISP_sidedtrace file, find the HTTP GET message was sent from the client to the Google server at time 7.109267 (where t=7.109267 is time at which this was sent as recorded in the NAT_home_side trace file). At what time does this message appear in the NAT_ISP_sidedtrace file? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET (as recording in the NAT_ISP_sidedtrace file)? Which of these fields are the same, and which are different, than in your answer to question 3 above?

The image shows a Wireshark packet capture of NAT_ISP_side.pcap. The packet list pane shows several packets, with packet 85 highlighted. Packet 85 is an HTTP GET request from source IP 71.192.34.104 to destination IP 64.233.169.104 on port 80. The packet details pane shows the Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol layers. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
83	6.067775	64.233.169.104	71.192.34.104	TCP	66	80 → 4335 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 SACK_PERM=1 WS=64
84	6.068754	71.192.34.104	64.233.169.104	TCP	60	4335 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
85	6.069168	71.192.34.104	64.233.169.104	HTTP	689	GET / HTTP/1.1
86	6.092755	Cisco_bf:6c:01	Broadcast	ARP	60	Who has 71.192.35.144? Tell 71.192.32.1
87	6.099637	64.233.169.104	71.192.34.104	TCP	60	80 → 4335 [ACK] Seq=1 Ack=636 Win=7040 Len=0
88	6.117078	64.233.169.104	71.192.34.104	TCP	1484	80 → 4335 [ACK] Seq=1 Ack=636 Win=7040 Len=1430 [TCP segment of a reassembled PDU]
89	6.117407	64.233.169.104	71.192.34.104	TCP	1484	80 → 4335 [ACK] Seq=1431 Ack=636 Win=7040 Len=1430 [TCP segment of a reassembled PDU]
90	6.117570	64.233.169.104	71.192.34.104	HTTP	814	HTTP/1.1 200 OK (text/html)
91	6.118515	71.192.34.104	64.233.169.104	TCP	60	4335 → 80 [ACK] Seq=636 Ack=3621 Win=260176 Len=0
92	6.162091	169.254.247.1...	169.254.255.2...	NBNS	92	Name query NB HPAB9D4C<00>

Frame 85: 689 bytes on wire (5512 bits), 689 bytes captured (5512 bits)
> Ethernet II, Src: Dell_4f:36:23 (00:08:74:4f:36:23), Dst: Cisco_bf:6c:01 (00:0e:d6:bf:6c:01)
> Internet Protocol Version 4, Src: 71.192.34.104, Dst: 64.233.169.104
> Transmission Control Protocol, Src Port: 4335, Dst Port: 80, Seq: 1, Ack: 1, Len: 635
> Hypertext Transfer Protocol
> TRANSM RTE Data

0000 00 0e d6 bf 6c 01 00 08 74 4f 36 23 08 00 45 00l... t06#..E-
0010 02 a3 a2 ac 40 00 7f 06 02 2f 47 c0 22 68 40 e5g.... /G:"he-
0020 a9 68 10 ef 00 50 f8 32 36 e5 e9 4f 38 95 50 18 ...h...P.2 6...08-P-
0030 fe 14 38 6d 00 00 47 45 54 20 2f 20 48 54 54 50 ...8m...GE T / HTTP
0040 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 77 72 e0 /1.1..Ho st: www.
0050 67 6f 6f 67 6c 65 2e 63 6f 6d 0d 0a 55 73 65 72 google.c om..User-
0060 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f -Agent: Mozilla/
0070 35 2e 30 20 28 57 69 6e 64 6f 77 73 3b 20 55 3b 5.0 (Win dows; U;
0080 20 57 69 6e 64 6f 77 73 20 4e 54 20 35 2e 31 3b Windows NT 5.1;
0090 20 65 6e 2d 55 53 3b 20 72 76 3a 31 2e 39 2e 3e en-US; rv:1.9.6

Tid:6.069168 sekunder // SRC:71.192.34.104 SRC:PORT: 4335 // DST:64.233.169.104 DST:PORT: 80
Destination og port er forskellige.

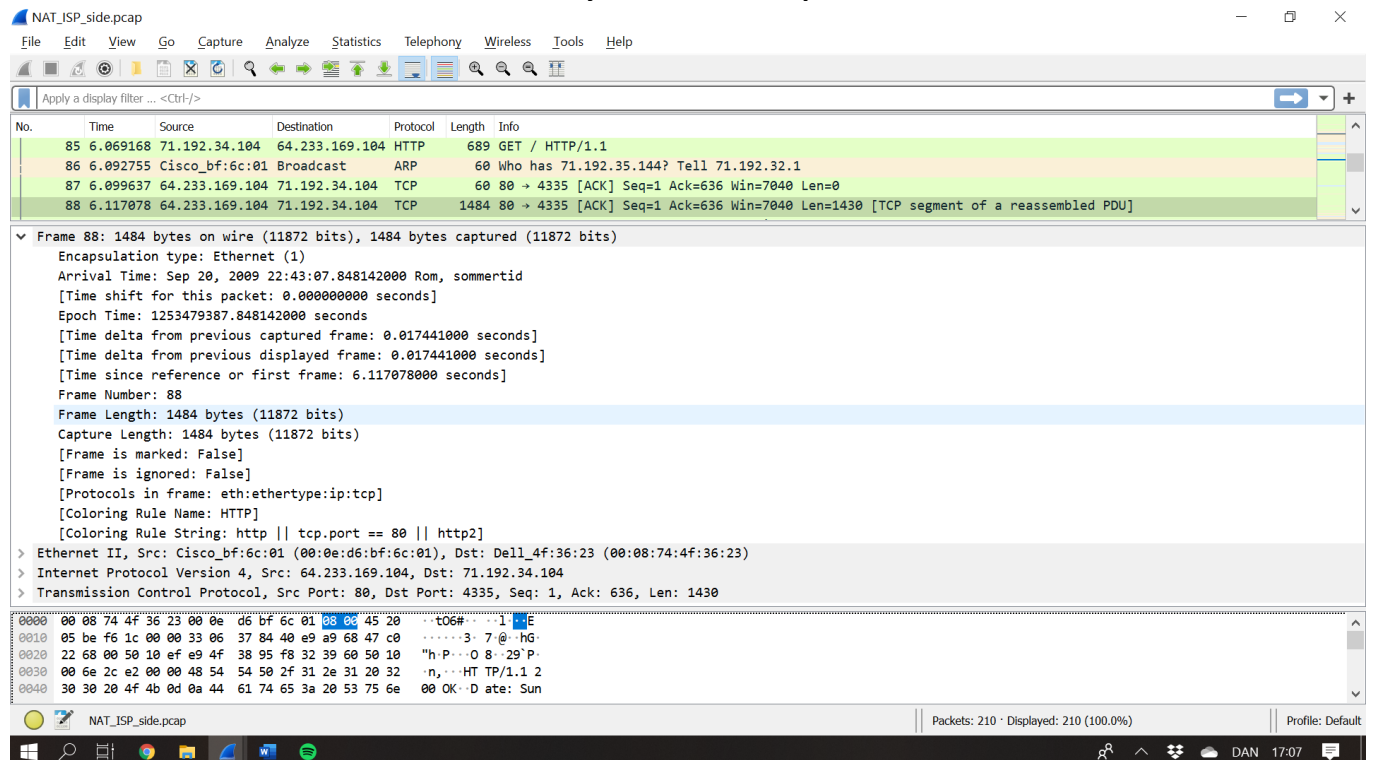
7. Are any fields in the HTTP GET message changed? Which of the following fields in the IP datagram carrying the HTTP GET are changed: Version, Header Length, Flags, Checksum. If any of these fields have changed, give a reason (in one sentence) stating why this field needed to change.

The image displays two Wireshark capture windows. The top window, titled 'NAT_home_side.pcap', shows an HTTP GET request from 192.168.1.100 to 64.233.169.104. The bottom window, titled 'NAT_ISP_side.pcap', shows the corresponding response from 64.233.169.104 to 71.192.34.104. The response is an HTTP 200 OK with a Content-Type of application/vnd.google.safebrowsing-chunk. The packet details for the response show the following fields: Version: 4, Header Length: 20 bytes (5), Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT), Total Length: 675, Identification: 0xa2ac (41644), Flags: 0x4000, Don't fragment, Fragment offset: 0, Time to live: 128, Protocol: TCP (6), Header checksum: 0xa94a [validation disabled], [Header checksum status: Unverified], Source: 192.168.1.100, Destination: 64.233.169.104. The packet bytes show the HTTP response structure, including the status line '200 OK (application/vnd.google.safebrowsing-chunk)' and the 'Content-Type: application/vnd.google.safebrowsing-chunk' header.

Sourceport er ændret fra 192.168.1.100 til 71.192.34.104
Header checksum er ændret fra 0xa94a til 0x022f
Header checksum er ændret pga. IP-adressen ændrede sig fra 192.168.43.75 til 71.192.34.104

Ali Raza Akhtar
Tech2
DAT19A

8. In the NAT_ISP_sidetrace file, at what time is the first 200 OK HTTP message received from the Google server? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message? Which of these fields are the same, and which are different than your answer to question 4 above?



HTTP 200 OK TID: 6.117078 sekunder
SRC: 64.233.169.104 SRC:PORT: 80
DST: 71.192.34.104 DST:PORT: 4335

Det er samme version.
Flag ændrer sig ikke.
Time to live er ændret.
Header checksum ændrer sig.

9. In the NAT_ISP_sidetracefile, at what time were the client-to-server TCP SYN segment and the server-to-client TCP ACK segment corresponding to the segments in question 5 above captured? What are the source and destination IP addresses and source and destination ports for these two segments? Which of these fields are the same, and which are different than your answer to question 5 above?

SYN-TIME: 6.035475 sekunder.
SYN:SRC: 71.192.34.104
SYN:DST:IP: 64.233.169.104
Time to live ændret.
ACK-TIME: 6.067775 sekunder.
ACK:SRC: 64.233.169.104

Ali Raza Akhtar

Tech2

DAT19A

ACK:DST:IP: 71.192.34.104

Ændret: Identification, Time to live, Flags, Source & Destination IP.

10.Using your answers to 1-8 above, fill in the NAT translation table entries for HTTP connection considered in questions 1-8 above

WANr

IP: 71.192.34.104

PORT: 4335

LAN

IP: 192.168.43.75

PORT: 4335