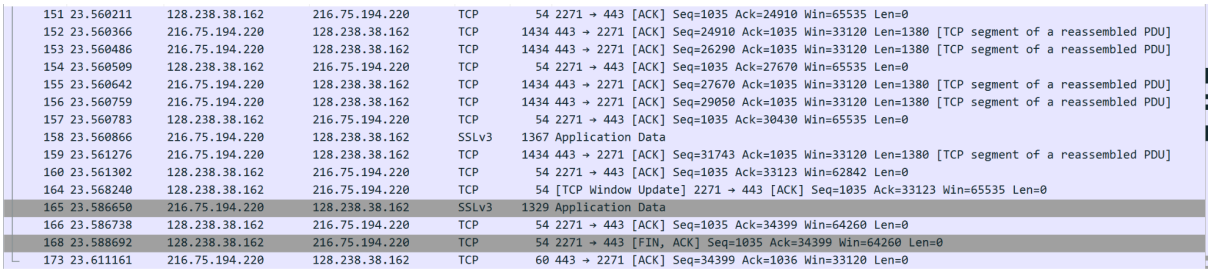
ZADANIE 1 SLOVENSKE

1) Vyfiltrujte všetky TCP správy, ktoré obsahujú TCP správu s príznakmi FIN a PUSH. Následne pre prvú identifikovanú správu zobrazte celé TCP spojenie a vypíšte ukončenie spojenia.

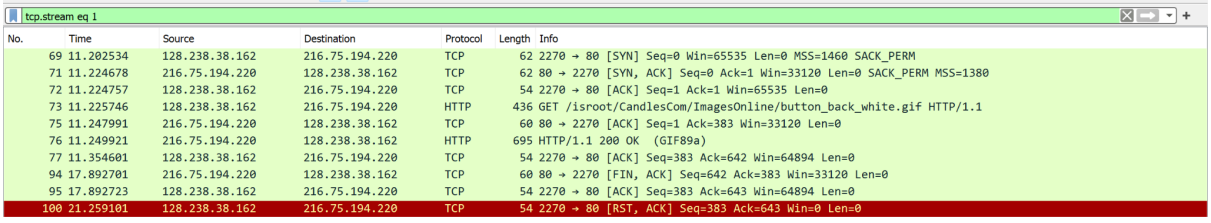
odpoved: FIN a PUSH som nasiel pomocou hladanie s filtrom: tcp.flags.fin == 1 && tcp.flags.push == 1
Ukoncenie spojenia som nasiel na packete číšlo 168, keďže obsahuje flagy FIN a ACK
toto spojenie moze byt najdene s filtrom: tcp.stream eq 2



151	23.560211	128.238.38.162	216.75.194.220	TCP	54 2271 → 443 [ACK] Seq=1035 Ack=24910 Win=65535 Len=0
152	23.560366	216.75.194.220	128.238.38.162	TCP	1434 443 → 2271 [ACK] Seq=24910 Ack=1035 Win=33120 Len=1380 [TCP segment of a reassembled PDU]
153	23.560486	216.75.194.220	128.238.38.162	TCP	1434 443 → 2271 [ACK] Seq=26290 Ack=1035 Win=33120 Len=1380 [TCP segment of a reassembled PDU]
154	23.560509	128.238.38.162	216.75.194.220	TCP	54 2271 → 443 [ACK] Seq=1035 Ack=27670 Win=65535 Len=0
155	23.560642	216.75.194.220	128.238.38.162	TCP	1434 443 → 2271 [ACK] Seq=27670 Ack=1035 Win=33120 Len=1380 [TCP segment of a reassembled PDU]
156	23.560759	216.75.194.220	128.238.38.162	TCP	1434 443 → 2271 [ACK] Seq=29050 Ack=1035 Win=33120 Len=1380 [TCP segment of a reassembled PDU]
157	23.560783	128.238.38.162	216.75.194.220	TCP	54 2271 → 443 [ACK] Seq=1035 Ack=30430 Win=65535 Len=0
158	23.560866	216.75.194.220	128.238.38.162	SSLv3	1367 Application Data
159	23.561276	216.75.194.220	128.238.38.162	TCP	1434 443 → 2271 [ACK] Seq=31743 Ack=1035 Win=33120 Len=1380 [TCP segment of a reassembled PDU]
160	23.561302	128.238.38.162	216.75.194.220	TCP	54 2271 → 443 [ACK] Seq=1035 Ack=33123 Win=62842 Len=0
164	23.568240	128.238.38.162	216.75.194.220	TCP	54 [TCP Window Update] 2271 → 443 [ACK] Seq=1035 Ack=33123 Win=65535 Len=0
165	23.586650	216.75.194.220	128.238.38.162	SSLv3	1329 Application Data
166	23.586738	128.238.38.162	216.75.194.220	TCP	54 2271 → 443 [ACK] Seq=1035 Ack=34399 Win=64260 Len=0
168	23.588692	128.238.38.162	216.75.194.220	TCP	54 2271 → 443 [FIN, ACK] Seq=1035 Ack=34399 Win=64260 Len=0
173	23.611161	216.75.194.220	128.238.38.162	TCP	60 443 → 2271 [ACK] Seq=34399 Ack=1036 Win=33120 Len=0

2)Vyšetrite TCP komunikáciu číslo 1 (stream) a identifikujte ukončenie spojenia:

odpoved: komunikaciu č. 1 som našiel pomocou filtra: tcp.stream eq 1
Ukončenie sa nachádza na packete č. 100, keďže obsahuje flagy RST a ACK, co oznacuje ukoncenie a obnovenie povodnej komunikacie



tcp.stream eq 1

No.	Time	Source	Destination	Protocol	Length	Info
69	11.202534	128.238.38.162	216.75.194.220	TCP	62	2270 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
71	11.224678	216.75.194.220	128.238.38.162	TCP	62	80 → 2270 [SYN, ACK] Seq=0 Ack=1 Win=33120 Len=0 SACK_PERM MSS=1380
72	11.224757	128.238.38.162	216.75.194.220	TCP	54	2270 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
73	11.225746	128.238.38.162	216.75.194.220	HTTP	436	GET /isroot/CandlesCom/ImagesOnline/button_back_white.gif HTTP/1.1
75	11.247991	216.75.194.220	128.238.38.162	TCP	60	80 → 2270 [ACK] Seq=1 Ack=383 Win=33120 Len=0
76	11.249921	216.75.194.220	128.238.38.162	HTTP	695	HTTP/1.1 200 OK (GIF89a)
77	11.354601	128.238.38.162	216.75.194.220	TCP	54	2270 → 80 [ACK] Seq=383 Ack=642 Win=64894 Len=0
94	17.892701	216.75.194.220	128.238.38.162	TCP	60	80 → 2270 [FIN, ACK] Seq=642 Ack=383 Win=33120 Len=0
95	17.892723	128.238.38.162	216.75.194.220	TCP	54	2270 → 80 [ACK] Seq=383 Ack=643 Win=64894 Len=0
100	21.259101	128.238.38.162	216.75.194.220	TCP	54	2270 → 80 [RST, ACK] Seq=383 Ack=643 Win=0 Len=0

3. Určte ukončenie posledného TCP spojenia (4-way handshake). Ako sa odlišuje od štandardného 4-way handshaku?

odpoved: 4-way handshake sa odlistuje od 3-way handshaku tak, ze 3-way je (SYN, SYN-ACK, ACK), pricom 4-way je (FIN, ACK, FIN, ACK)

Pri poslednej komunikacii (tcp.stream eq 14) si mozeme vsimnut, ze packet cislo 343 a 344 obsahuje FIN, ACK a 345 a 346 ACK, teda by sa to dalo napisat ako (FIN, FIN, ACK, ACK)

```
337 177219548.43… 192.168.1.33    147.175.1.12    TCP      62 1552 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
338 177219548.45… 147.175.1.12    192.168.1.33    TCP      62 80 → 1552 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SACK_PERM
339 177219548.45… 192.168.1.33    147.175.1.12    TCP      54 1552 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
340 177219548.45… 192.168.1.33    147.175.1.12    HTTP    703 GET /new/stu_upgrade/js/lightbox.js HTTP/1.1
341 177219548.47… 147.175.1.12    192.168.1.33    TCP      60 80 → 1552 [ACK] Seq=1 Ack=650 Win=6490 Len=0
342 177219548.47… 147.175.1.12    192.168.1.33    HTTP    205 HTTP/1.1 304 Not Modified
343 177219548.47… 192.168.1.33    147.175.1.12    TCP      54 1552 → 80 [FIN, ACK] Seq=650 Ack=152 Win=65384 Len=0
344 177219548.47… 147.175.1.12    192.168.1.33    TCP      60 80 → 1552 [FIN, ACK] Seq=152 Ack=650 Win=6490 Len=0
345 177219548.47… 192.168.1.33    147.175.1.12    TCP      54 1552 → 80 [ACK] Seq=651 Ack=153 Win=65384 Len=0
346 177219548.49… 147.175.1.12    192.168.1.33    TCP      60 80 → 1552 [ACK] Seq=153 Ack=651 Win=6490 Len=0
```

4. Vyfiltrujte všetky HTTPS spojenia a overte či všetky otvorenia spojení nastali pomocou 3-way handshake

odpoved: ano
pouzil som filter: tcp.port == 443
kedze tento port je znamy pre HTTP
nasledne som pozeral na packety ktore obsahuju SYN a pozrel som na ich ip adresu, a nasledne hladat SYN,ACK a ACK ktore boli pre tuto istu ip adresu a nasledovali v packete po nom.

ZADANIE ANGLICKE:

1) What is the IP address of your computer?

```
Source Address: 147.175.163.154
```

2) Within the IP packet header, what is the value in the upper layer protocol field?

```
Protocol: ICMP (1)
Header Checksum: 0xa0cb [validation disabled]
```

3)How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.

IP header: 20 bytes

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

Payload:
40 bytes

odpoved: Total lenght je suma payload a IP header, ak je teda IP header 20 bytov, tak payload je 60-20=40

```
Total Length: 60
```

4)Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented

No it wasnt fragmented, as MF flag is set to 0 (if it was fragmented, would be 1)

```
.0.. .... = Don't fragment: Not set
..0. .... = More fragments: Not set
```

5) . Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer?

```
Identification: 0x2df6 (11766)
```

6) Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?

Mali by zostat rovnake:
IP version field (IPV4, IPV6…)
Type of service (ICMP)
IHL (Internet header length)
Total length
Flags and fragment offset (ak nie su fragmentovane)
Header checksum

MUSIA ROVNAKE:
Protocol

musia sa menit:
Identification
Time to live
Checksum
Payload data
Source/destination IP address (vymienaju sa)

7) Describe the pattern you see in the values in the Identification field of the IP datagram
Kazdy poslany packet z mojho PC mal identification field o 1 vacsi nez predosli (7911,7912,7913,7914) ale packety prijate mojim PC mali cisla rozne, kedze sender z danej adresy neposielal iba na moj PC v tom istom case.

8)What is the value in the Identification field and the TTL field?
ID: 7911
TTL: 128

9) Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Why?

nemam ziadne TTL - exceeded replies.

10, 11) Nemam ziadne fragmentovane packety