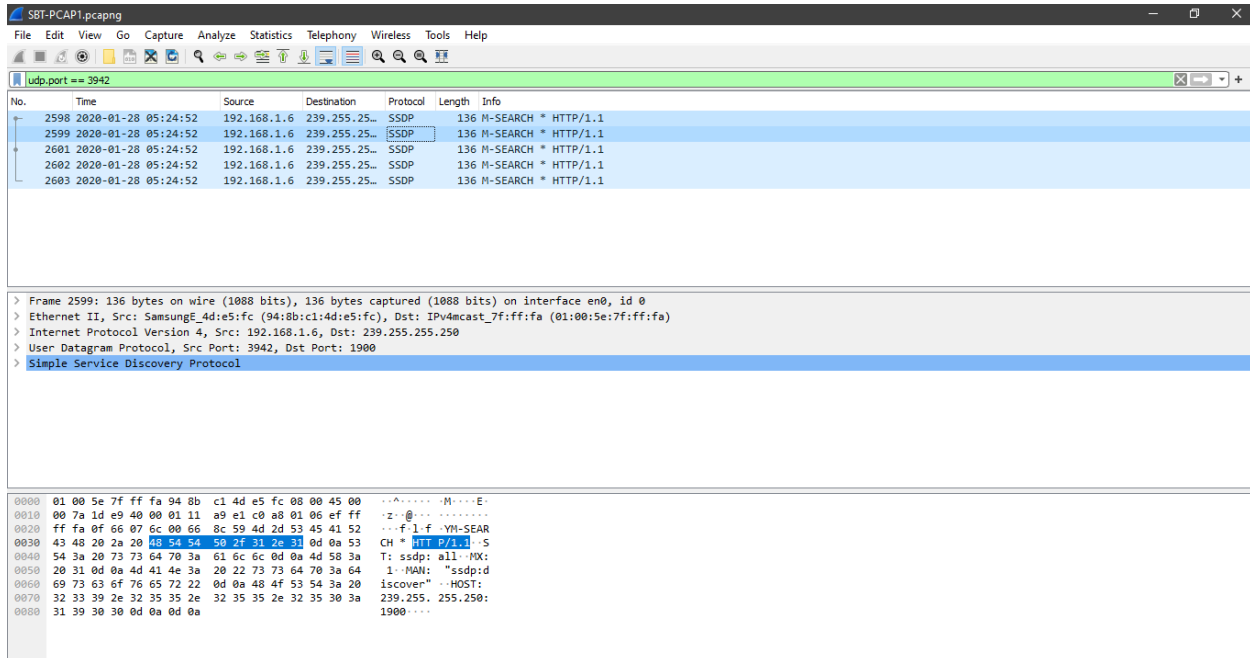


Investigando con Wireshark - PCAP01

PCAP 1

¿Qué protocolo se utilizó en el puerto 3942?



No.	Time	Source	Destination	Protocol	Length	Info
2598	2020-01-28 05:24:52	192.168.1.6	239.255.25.250	SSDP	136	M-SEARCH * HTTP/1.1
2599	2020-01-28 05:24:52	192.168.1.6	239.255.25.250	SSDP	136	M-SEARCH * HTTP/1.1
2601	2020-01-28 05:24:52	192.168.1.6	239.255.25.250	SSDP	136	M-SEARCH * HTTP/1.1
2602	2020-01-28 05:24:52	192.168.1.6	239.255.25.250	SSDP	136	M-SEARCH * HTTP/1.1
2603	2020-01-28 05:24:52	192.168.1.6	239.255.25.250	SSDP	136	M-SEARCH * HTTP/1.1

> Frame 2599: 136 bytes on wire (1088 bits), 136 bytes captured (1088 bits) on interface en0, id 0
> Ethernet II, Src: SamsungE_4d:e5:fc (94:8b:c1:4d:e5:fc), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
> Internet Protocol Version 4, Src: 192.168.1.6, Dst: 239.255.25.250
> User Datagram Protocol, Src Port: 3942, Dst Port: 1900
> Simple Service Discovery Protocol

```
0000  01 00 5e 7f ff fa 8b c1 4d e5 fc 08 00 45 00  ..@...M...E...
0010  00 7a 1d e9 40 01 11 a9 e1 c0 a8 01 06 ef ff  -z...@...
0020  ff fa 0f 66 07 6c 00 66 8c 59 4d 2d 53 45 41 52  -f...l...YH-SEAR
0030  43 48 20 2a 20 10 05 45 50 2f fa 7f ff fa 0d 0a 53  CH * 192.255.25.250
0040  54 3a 20 73 73 64 70 3a 61 6c 6c 0d 0a 4d 58 3a  T: ssdp: all: (MX)
0050  20 31 0d 0a 4d 41 4e 3a 20 22 73 73 64 70 3a 64  1..MAN: "ssdp:d
0060  69 73 63 6f 76 65 72 22 0d 0a 48 4f 53 54 3a 20  iscover" ..HOST:
0070  32 33 39 2e 32 35 35 2e 32 35 35 2e 32 35 30 3a  239.255. 255.250:
0080  31 39 30 30 0d 0a 0d 0a 1900....
```

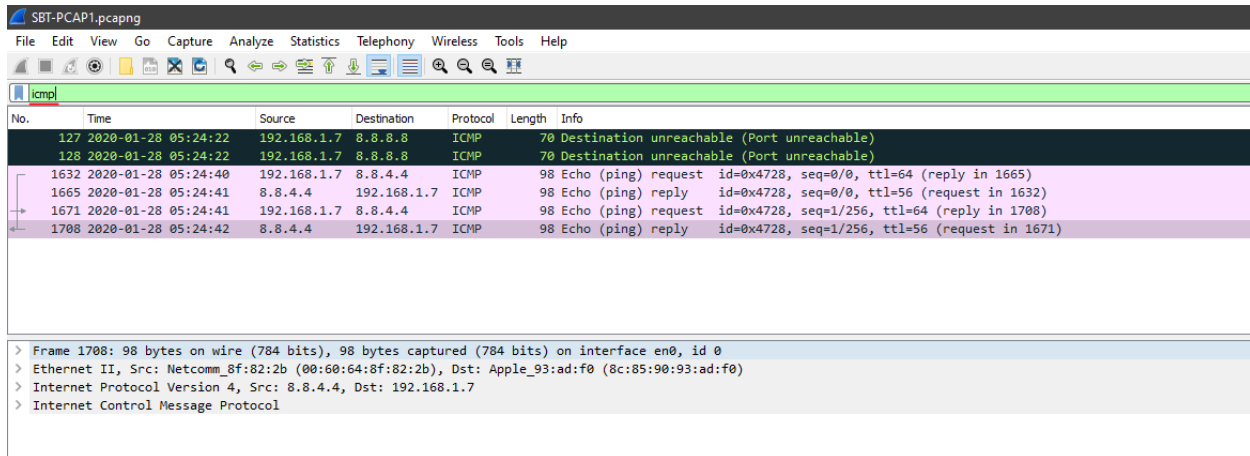
Para filtrar un puerto necesitamos saber que los puertos se utilizan en dos protocolos: UDP y TCP.

Esta vez vamos a utilizar el filtro "**udp.port == 3942**" para filtrar el puerto 3942 sobre UDP.

Observamos que el protocolo que se utilizó sobre el puerto 3942 es **SSDP** (Simple Service Discovery Protocol).

¿Cuál es la dirección IP del host al que se le hizo ping dos veces?

Investigando con Wireshark - PCAP01



SBT-PCAP1.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

icmp

No.	Time	Source	Destination	Protocol	Length	Info
127	2020-01-28 05:24:22	192.168.1.7	8.8.8.8	ICMP	70	Destination unreachable (Port unreachable)
128	2020-01-28 05:24:22	192.168.1.7	8.8.8.8	ICMP	70	Destination unreachable (Port unreachable)
1632	2020-01-28 05:24:40	192.168.1.7	8.8.4.4	ICMP	98	Echo (ping) request id=0x4728, seq=0/0, ttl=64 (reply in 1665)
1665	2020-01-28 05:24:41	8.8.4.4	192.168.1.7	ICMP	98	Echo (ping) reply id=0x4728, seq=0/0, ttl=56 (request in 1632)
1671	2020-01-28 05:24:41	192.168.1.7	8.8.4.4	ICMP	98	Echo (ping) request id=0x4728, seq=1/256, ttl=64 (reply in 1708)
1708	2020-01-28 05:24:42	8.8.4.4	192.168.1.7	ICMP	98	Echo (ping) reply id=0x4728, seq=1/256, ttl=56 (request in 1671)

> Frame 1708: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0
> Ethernet II, Src: Netcomm_8f:82:2b (00:60:64:8f:82:2b), Dst: Apple_93:ad:f0 (8c:85:90:93:ad:f0)
> Internet Protocol Version 4, Src: 8.8.4.4, Dst: 192.168.1.7
> Internet Control Message Protocol

Para filtrar el ping, tenemos que aplicar filtro por el protocolo **ICMP**.

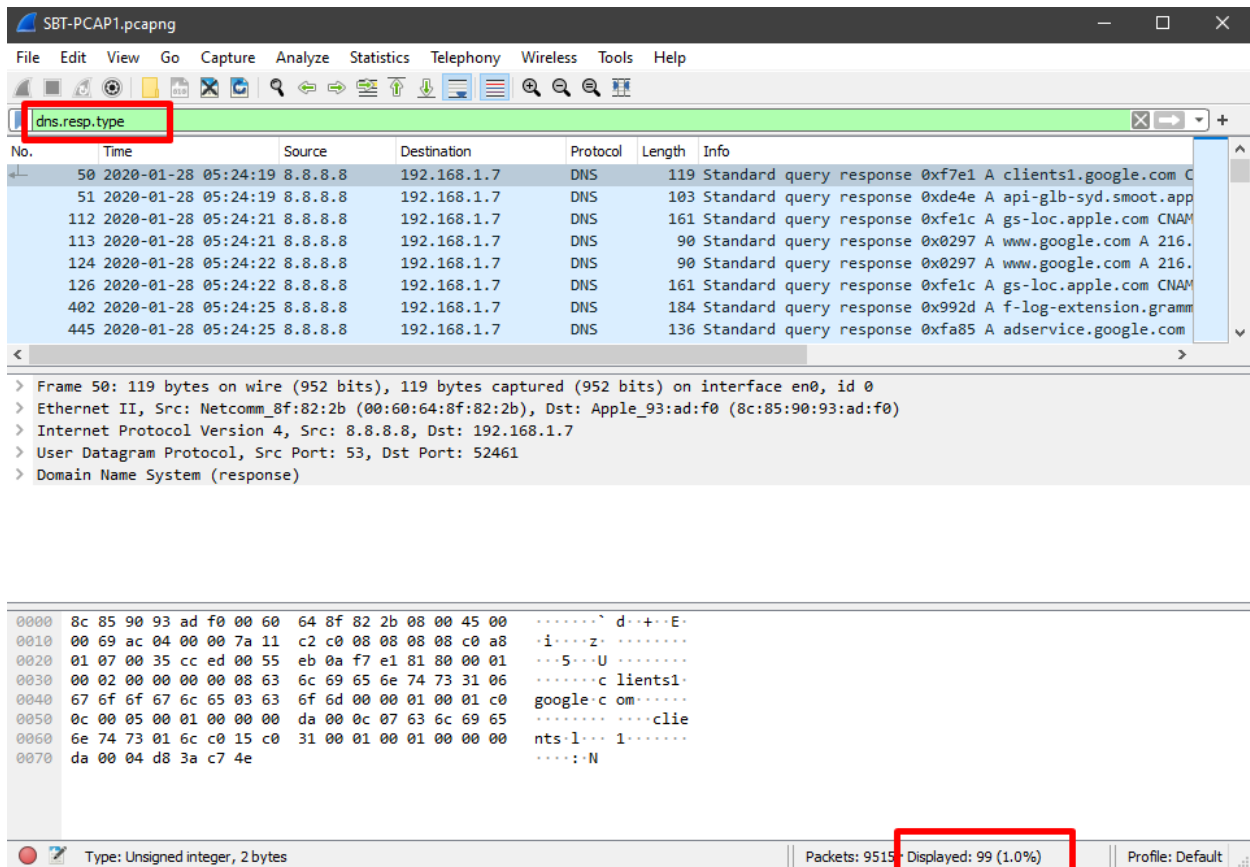
En Wireshark utilizamos el filtro "**icmp**".

Observamos que hay dos peticiones de ping y 2 respuestas.

Vemos que el host **192.168.1.7** ha realizado **2** peticiones de ping al DNS **8.8.4.4**.

¿Cuántos paquetes de respuesta a la consulta DNS se han capturado?

Investigando con Wireshark - PCAP01



The screenshot shows the Wireshark interface with the packet capture filter `dns.resp.type` applied. The packet list shows several DNS response packets from 192.168.1.7 to 8.8.8.8. The packet details pane shows the structure of a DNS response packet, including Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Domain Name System (response). The packet bytes pane shows the raw data of the selected packet.

No.	Time	Source	Destination	Protocol	Length	Info
50	2020-01-28 05:24:19	8.8.8.8	192.168.1.7	DNS	119	Standard query response 0xf7e1 A clients1.google.com C
51	2020-01-28 05:24:19	8.8.8.8	192.168.1.7	DNS	103	Standard query response 0xde4e A api-glb-syd.smoot.app
112	2020-01-28 05:24:21	8.8.8.8	192.168.1.7	DNS	161	Standard query response 0xfe1c A gs-loc.apple.com CNAM
113	2020-01-28 05:24:21	8.8.8.8	192.168.1.7	DNS	90	Standard query response 0x0297 A www.google.com A 216.
124	2020-01-28 05:24:22	8.8.8.8	192.168.1.7	DNS	90	Standard query response 0x0297 A www.google.com A 216.
126	2020-01-28 05:24:22	8.8.8.8	192.168.1.7	DNS	161	Standard query response 0xfe1c A gs-loc.apple.com CNAM
402	2020-01-28 05:24:25	8.8.8.8	192.168.1.7	DNS	184	Standard query response 0x992d A f-log-extension.gramm
445	2020-01-28 05:24:25	8.8.8.8	192.168.1.7	DNS	136	Standard query response 0xfa85 A adservice.google.com

Frame 50: 119 bytes on wire (952 bits), 119 bytes captured (952 bits) on interface en0, id 0
> Ethernet II, Src: Netcomm_8f:82:2b (00:60:64:8f:82:2b), Dst: Apple_93:ad:f0 (8c:85:90:93:ad:f0)
> Internet Protocol Version 4, Src: 8.8.8.8, Dst: 192.168.1.7
> User Datagram Protocol, Src Port: 53, Dst Port: 52461
> Domain Name System (response)

Type: Unsigned integer, 2 bytes | Packets: 9515 | Displayed: 99 (1.0%) | Profile: Default

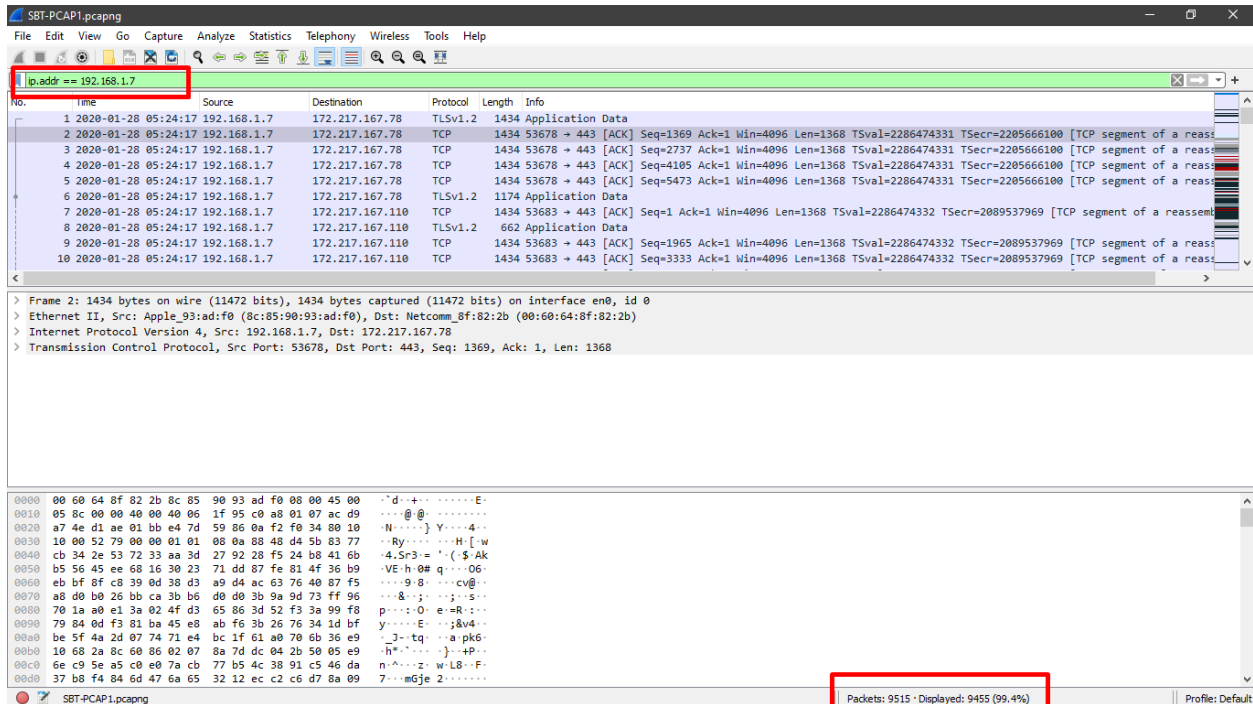
Para filtrar la respuesta a la consulta DNS utilizamos el filtro "**dns.resp.type**" y vamos a la parte inferior de Wireshark donde se muestran los paquetes desplegados.

En este caso vemos que hay **99 paquetes** de respuesta a la consulta DNS capturados.

RESPUESTA: 99 paquetes.

¿Cuál es la dirección IP del host que ha enviado el mayor número de bytes?

Investigando con Wireshark - PCAP01



SBT-PCAP01.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 192.168.1.7

No.	Time	Source	Destination	Protocol	Length	Info
1	2020-01-28 05:24:17	192.168.1.7	172.217.167.78	TLSv1.2	1434	Application Data
2	2020-01-28 05:24:17	192.168.1.7	172.217.167.78	TCP	1434	53678 → 443 [ACK] Seq=1369 Ack=1 Win=4096 Len=1368 TSval=2286474331 TSecr=2205666100 [TCP segment of a reassembled data stream]
3	2020-01-28 05:24:17	192.168.1.7	172.217.167.78	TCP	1434	53678 → 443 [ACK] Seq=2737 Ack=1 Win=4096 Len=1368 TSval=2286474331 TSecr=2205666100 [TCP segment of a reassembled data stream]
4	2020-01-28 05:24:17	192.168.1.7	172.217.167.78	TCP	1434	53678 → 443 [ACK] Seq=4105 Ack=1 Win=4096 Len=1368 TSval=2286474331 TSecr=2205666100 [TCP segment of a reassembled data stream]
5	2020-01-28 05:24:17	192.168.1.7	172.217.167.78	TCP	1434	53678 → 443 [ACK] Seq=5473 Ack=1 Win=4096 Len=1368 TSval=2286474331 TSecr=2205666100 [TCP segment of a reassembled data stream]
6	2020-01-28 05:24:17	192.168.1.7	172.217.167.78	TLSv1.2	1174	Application Data
7	2020-01-28 05:24:17	192.168.1.7	172.217.167.110	TCP	1434	53683 → 443 [ACK] Seq=1 Ack=1 Win=4096 Len=1368 TSval=2286474332 TSecr=2089537969 [TCP segment of a reassembled data stream]
8	2020-01-28 05:24:17	192.168.1.7	172.217.167.110	TLSv1.2	662	Application Data
9	2020-01-28 05:24:17	192.168.1.7	172.217.167.110	TCP	1434	53683 → 443 [ACK] Seq=1965 Ack=1 Win=4096 Len=1368 TSval=2286474332 TSecr=2089537969 [TCP segment of a reassembled data stream]
10	2020-01-28 05:24:17	192.168.1.7	172.217.167.110	TCP	1434	53683 → 443 [ACK] Seq=3333 Ack=1 Win=4096 Len=1368 TSval=2286474332 TSecr=2089537969 [TCP segment of a reassembled data stream]

> Frame 2: 1434 bytes on wire (11472 bits), 1434 bytes captured (11472 bits) on interface en0, id 0
> Ethernet II, Src: Apple_93:ad:f0 (8c:85:90:93:ad:f0), Dst: Netcomm_8f:82:2b (00:60:64:8f:82:2b)
> Internet Protocol Version 4, Src: 192.168.1.7, Dst: 172.217.167.78
> Transmission Control Protocol, Src Port: 53678, Dst Port: 443, Seq: 1369, Ack: 1, Len: 1368

0000 00 60 64 8f 82 2b 8c 85 90 93 ad f0 08 00 45 00 ...d...+...E...
0010 05 8c 00 00 40 00 40 06 1f 95 c0 a8 01 07 ac d9 ...@...Y...
0020 a7 4e d1 ae 01 bb e4 7d 59 06 0a f2 f0 34 80 10 ...N...}Y...4...
0030 10 00 52 79 00 00 01 01 08 0a 88 48 d4 5b 83 77 ...Ry...H...w...
0040 cb 34 2e 53 72 33 aa 3d 27 92 28 f5 24 b8 41 6b ...4.Sr3...(-\$Ak...
0050 b5 56 45 ee 68 16 30 23 71 dd 87 fe 81 4f 36 b9 ...VE-h 0# q...06...
0060 eb bf 8f c8 39 0d 38 d3 a9 d4 ac 63 76 40 87 f5 ...9.8...cv...
0070 a8 d0 b0 26 bb ca 3b b6 d0 d0 3b 9a 9d 73 ff 96 ...&...;...s...
0080 70 1a a0 e1 3a 02 4f d3 65 86 3d 52 f3 3a 99 f8 ...p...:O e=R...
0090 79 84 0d f3 81 ba 45 e0 ab f6 3b 26 76 34 1d bf ...y...E...&v4...
00a0 be 5f 4a 2d 07 71 e4 bc 1f 61 a0 70 6b 36 e9 ...J-...-a pk6...
00b0 10 68 2a 8c 60 86 02 07 8a 7d dc 04 2b 50 05 e9 ...h*...z...+P...
00c0 6e c9 5e a5 c0 e0 7a cb 77 b5 4c 38 91 c5 46 da ...n^...z...w-LB...F...
00d0 37 b8 f4 84 6d 47 6a 65 32 12 ec c2 c6 d7 8a 09 7...m6je 2...
SBT-PCAP01.pcapng

Packets: 9515 · Displayed: 9455 (99.4%)

Profile: Default

Si vemos los paquetes, la mayoría provienen del host **192.168.1.7**, por lo que filtraremos el host con el filtro "**ip.addr == 192.168.1.7**". Aquí vemos que efectivamente el 99,4% de los paquetes de este pcapng provienen de este host, por lo que podemos decir que la mayoría de los bytes son enviados por este host.

RESPUESTA: **192.168.1.7**