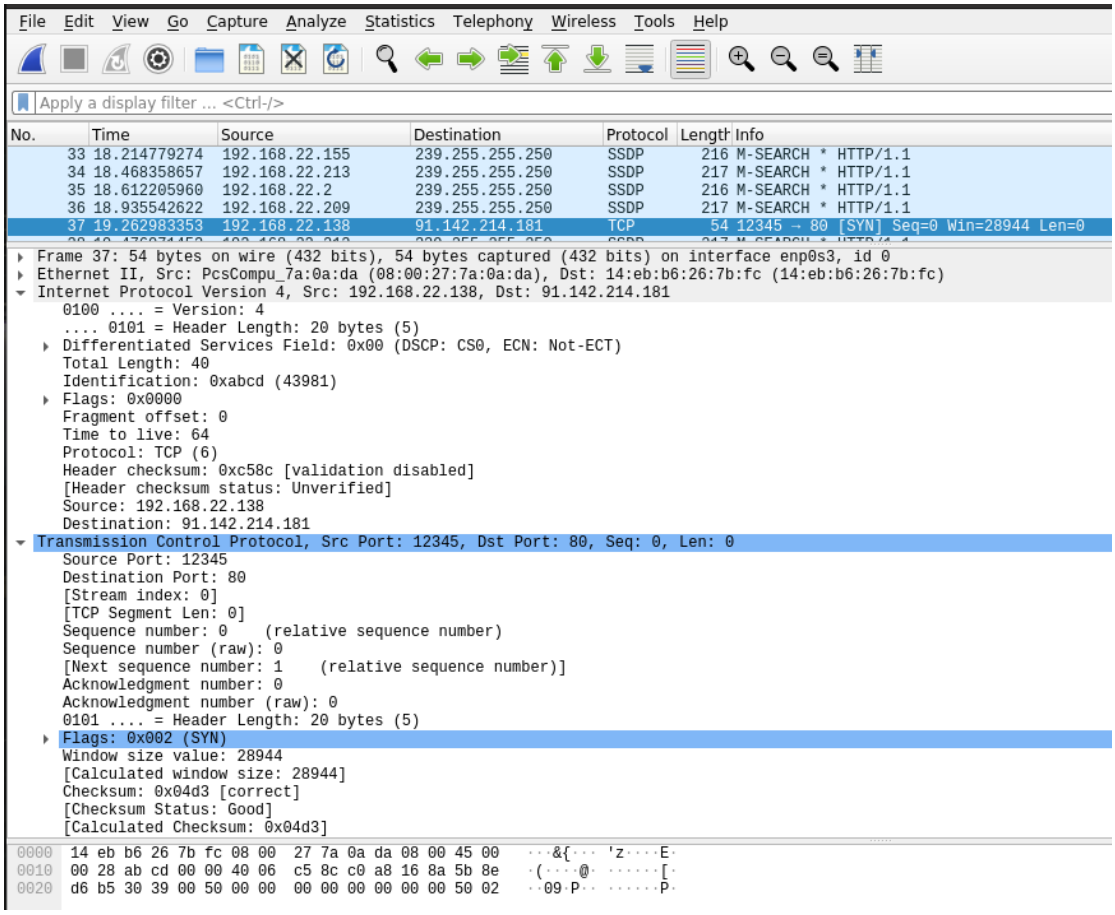
Crea un paquete TCP SYN que vaya a 91.142.214.181 , escucha con Wireshark y observa si obtienes la respuesta.

1-Crea un pantallazo de lo mostrado en Wireshark

Este es el protocolo TCP/IP.

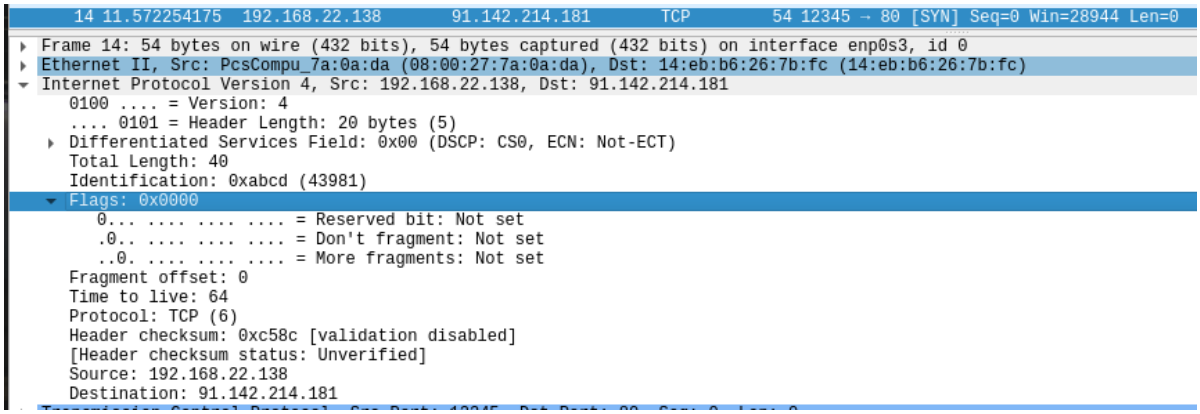
```python
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_TCP)
s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)

ip_header  = b'\x45\x00\x00\x28'  # Version, IHL, Type of Service | Total Length
ip_header += b'\xab\xcd\x00\x00'  # Identification | Flags, Fragment Offset
ip_header += b'\x40\x06\xa6\xec'  # TTL, Protocol | Header Checksum
ip_header += b'\xc0\xa8\x16\x8a'  # Source Address
ip_header += b'\x5b\x8e\xd6\xb5'  # Destination Address

tcp_header  = b'\x30\x39\x00\x50' # Source Port | Destination Port
tcp_header += b'\x00\x00\x00\x00' # Sequence Number
tcp_header += b'\x00\x00\x00\x00' # Acknowledgement Number
tcp_header += b'\x50\x02\x71\x10' # Data Offset, Reserved, Flags | Window Size
tcp_header += b'\x04\xd3\x00\x00' # Checksum | Urgent Pointer

packet = ip_header + tcp_header
s.sendto(packet, ('91.142.214.181', 0))
```

Observamos en wireshark nuestro paquete realizado anteriormente.

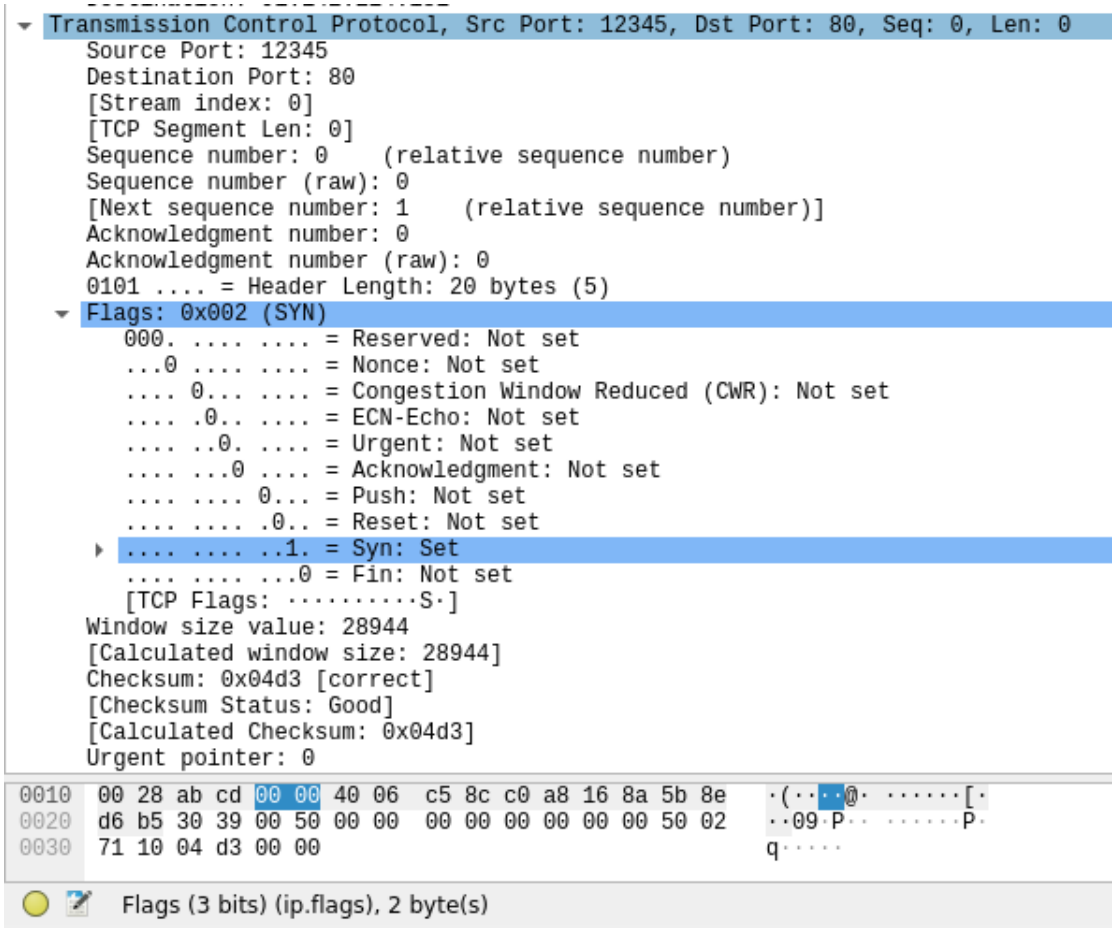


2-¿Qué flags tiene "encendidos" tu paquete?, ¿y el de vuelta?

No tiene encendido ninguna flag.



Y el de vuelta tiene encendido la flag de SYN.

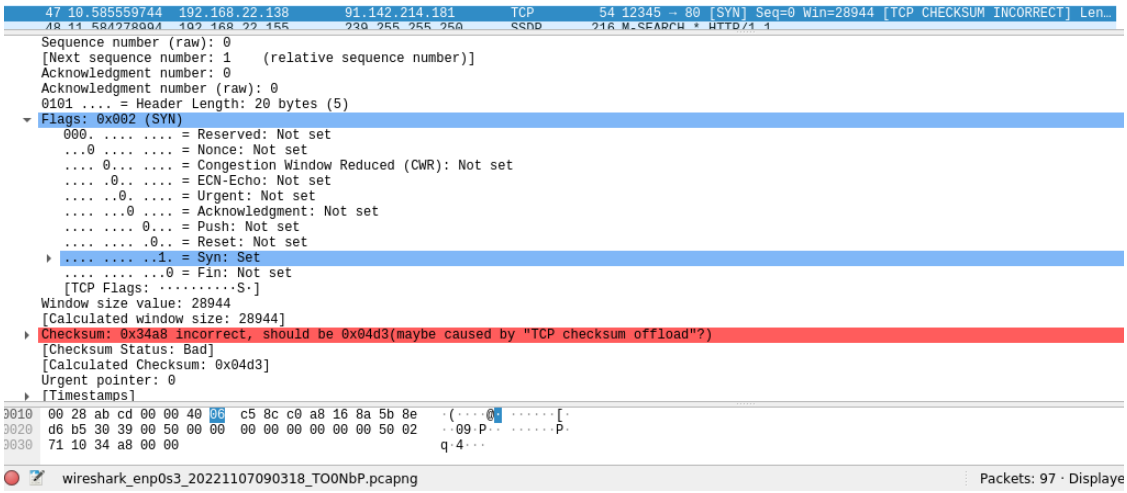


3-Pon mal el checksum y observa qué pasa

```
1 import socket
2
3 s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_TCP)
4 s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)
5
6 ip_header  = b'\x45\x00\x00\x28'  # Version, IHL, Type of Service | Total Length
7 ip_header += b'\xab\xcd\x00\x00'  # Identification | Flags, Fragment Offset
8 ip_header += b'\x40\x06\xa6\xec'  # TTL, Protocol | Header Checksum
9 ip_header += b'\xc0\xa8\x16\x8a'  # Source Address
0 ip_header += b'\x5b\x8e\xd6\xb5'  # Destination Address
1
2 tcp_header  = b'\x30\x39\x00\x50' # Source Port | Destination Port
3 tcp_header += b'\x00\x00\x00\x00' # Sequence Number
4 tcp_header += b'\x00\x00\x00\x00' # Acknowledgement Number
5 tcp_header += b'\x50\x02\x71\x10' # Data Offset, Reserved, Flags | Window Size
6 tcp_header += b'\x34\xa8\x00\x00' # Checksum | Urgent Pointer
7
8 packet = ip_header + tcp_header
9 s.sendto(packet, ('91.142.214.181', 0))
```

Al poner mal el checksum el wireshark inmediatamente nos dice que está mal y que lo cambiemos al

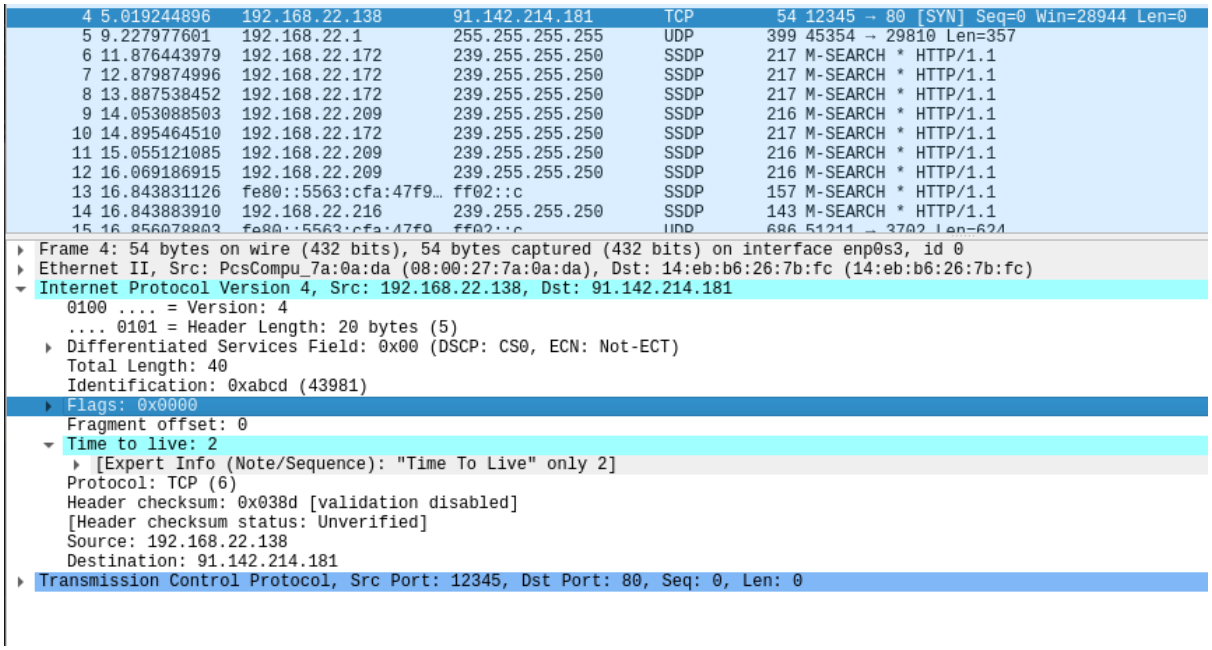
correcto, que es el 0x04d3.



4-Pon un TTL=2 y observa qué pasa

```python
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_TCP)
s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)

ip_header  = b'\x45\x00\x00\x28'  # Version, IHL, Type of Service | Total Length
ip_header += b'\xab\xcd\x00\x00'  # Identification | Flags, Fragment Offset
ip_header += b'\x02\x06\xa6\xec'  # TTL, Protocol | Header Checksum
ip_header += b'\xc0\xa8\x16\x8a'  # Source Address
ip_header += b'\x5b\x8e\xd6\xb5'  # Destination Address

tcp_header  = b'\x30\x39\x00\x50' # Source Port | Destination Port
tcp_header += b'\x00\x00\x00\x00' # Sequence Number
tcp_header += b'\x00\x00\x00\x00' # Acknowledgement Number
tcp_header += b'\x50\x02\x71\x10' # Data Offset, Reserved, Flags | Window Size
tcp_header += b'\x04\xd3\x00\x00' # Checksum | Urgent Pointer

packet = ip_header + tcp_header
s.sendto(packet, ('91.142.214.181', 0))
```

Hemos cambiado el TTL de 40 a 2 y nos ha dicho que no puede ser solo 2.



Alberto Heras Herrera

Juan Sánchez Balastegui