

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.

```
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
10 ip_header += b'\x5b\x8e\xd6\xb5' # Destination Address
11
12 tcp_header = b'\x30\x39\x00\x50' # Source Port | Destination Port
13 tcp_header += b'\x00\x00\x00\x00' # Sequence Number
14 tcp_header += b'\x00\x00\x00\x00' # Acknowledgement Number
15 tcp_header += b'\x50\x02\x71\x10' # Data Offset, Reserved, Flags | Window Size
16 tcp_header += b'\x04\xd3\x00\x00' # Checksum | Urgent Pointer
17
18 packet = ip_header + tcp_header
19 s.sendto(packet, ('91.142.214.181', 0))
```

Observamos en wireshark nuestro paquete realizado anteriormente.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
33	18.214779274	192.168.22.155	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
34	18.468358657	192.168.22.213	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
35	18.612205960	192.168.22.2	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
36	18.935542622	192.168.22.209	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
37	19.262933353	192.168.22.138	91.142.214.181	TCP	54	12345 → 80 [SYN] Seq=0 Win=28944 Len=0

Frame 37: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface enp0s3, id 0

Ethernet II, Src: PcsCompu_7a:0a:da (08:00:27:7a:0a:da), Dst: 14:eb:b6:26:7b:fc (14:eb:b6:26:7b:fc)

Internet Protocol Version 4, Src: 192.168.22.138, Dst: 91.142.214.181

0100 = Version: 4

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

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 40

Identification: 0xabcd (43981)

Flags: 0x0000

Fragment offset: 0

Time to live: 64

Protocol: TCP (6)

Header checksum: 0xc58c [validation disabled]

[Header checksum status: Unverified]

Source: 192.168.22.138

Destination: 91.142.214.181

Transmission Control Protocol, Src Port: 12345, Dst Port: 80, Seq: 0, Len: 0

Source Port: 12345

Destination Port: 80

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 0 (relative sequence number)

Sequence number (raw): 0

[Next sequence number: 1 (relative sequence number)]

Acknowledgment number: 0

Acknowledgment number (raw): 0

0101 = Header Length: 20 bytes (5)

Flags: 0x002 (SYN)

Window size value: 28944

[Calculated window size: 28944]

Checksum: 0x04d3 [correct]

[Checksum Status: Good]

[Calculated Checksum: 0x04d3]

0000 14 eb b6 26 7b fc 08 00 27 7a 0a da 08 00 45 00 ..&{... 'z....E.

0010 00 28 ab cd 00 00 40 06 c5 8c c0 a8 16 8a 5b 8e .(....@.....[.

0020 d6 b5 30 39 00 50 00 00 00 00 00 00 00 50 02 ..09.P.....P.

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

No tiene encendido ninguna flag.

14 11.572254175 192.168.22.138 91.142.214.181 TCP 54 12345 → 80 [SYN] Seq=0 Win=28944 Len=0

Frame 14: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface enp0s3, id 0

Ethernet II, Src: PcsCompu_7a:0a:da (08:00:27:7a:0a:da), Dst: 14:eb:b6:26:7b:fc (14:eb:b6:26:7b:fc)

Internet Protocol Version 4, Src: 192.168.22.138, Dst: 91.142.214.181

0100 = Version: 4

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

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 40

Identification: 0xabcd (43981)

Flags: 0x0000

0... .. = Reserved bit: Not set

.0.. .. = Don't fragment: Not set

..0. = More fragments: Not set

Fragment offset: 0

Time to live: 64

Protocol: TCP (6)

Header checksum: 0xc58c [validation disabled]

[Header checksum status: Unverified]

Source: 192.168.22.138

Destination: 91.142.214.181

Y el de vuelta tiene encendido la flag de SYN.

Transmission Control Protocol, Src Port: 12345, Dst Port: 80, Seq: 0, Len: 0

Source Port: 12345
Destination Port: 80
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
Sequence number (raw): 0
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 0
Acknowledgment number (raw): 0
0101 = Header Length: 20 bytes (5)

Flags: 0x002 (SYN)

000. = Reserved: Not set
...0 = Nonce: Not set
.... 0... = Congestion Window Reduced (CWR): Not set
.... .0.. = ECN-Echo: Not set
.... ..0. = Urgent: Not set
.... ...0 = Acknowledgment: Not set
....0... = Push: Not set
....0.. = Reset: Not set
▶1. = Syn: Set
....0 = Fin: Not set
[TCP Flags:S.]
Window size value: 28944
[Calculated window size: 28944]
Checksum: 0x04d3 [correct]
[Checksum Status: Good]
[Calculated Checksum: 0x04d3]
Urgent pointer: 0

0010	00 28 ab cd 00 00 40 06 c5 8c c0 a8 16 8a 5b 8e	.(...@.[.
0020	d6 b5 30 39 00 50 00 00 00 00 00 00 00 50 02	..09.P.P.
0030	71 10 04 d3 00 00	q.....

Flags (3 bits) (ip.flags), 2 byte(s)

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
10 ip_header += b'\x5b\x8e\xd6\xb5' # Destination Address
11
12 tcp_header = b'\x30\x39\x00\x50' # Source Port | Destination Port
13 tcp_header += b'\x00\x00\x00\x00' # Sequence Number
14 tcp_header += b'\x00\x00\x00\x00' # Acknowledgement Number
15 tcp_header += b'\x50\x02\x71\x10' # Data Offset, Reserved, Flags | Window Size
16 tcp_header += b'\x34\xa8\x00\x00' # Checksum | Urgent Pointer
17
18 packet = ip_header + tcp_header
19 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.

47	10.585559744	192.168.22.138	91.142.214.181	TCP	54	12345	->	80	[SYN]	Seq=0	Win=28944	[TCP CHECKSUM INCORRECT]	Len...
48	11.582778004	102.168.22.155	220.255.255.255	SSHD	216	M-SEARCH	*	HTTP/1.1					

Sequence number (raw): 0
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 0
Acknowledgment number (raw): 0
0101 = Header Length: 20 bytes (5)
▼ Flags: 0x002 (SYN)
 000. = Reserved: Not set
 ...0. = Nonce: Not set
 0... = Congestion Window Reduced (CWR): Not set
 0... = ECN-Echo: Not set
 0... = Urgent: Not set
 0... = Acknowledgment: Not set
 0... = Push: Not set
 0... = Reset: Not set
 ...1... = Syn: Set
 0... = Fin: Not set
 [TCP Flags:S-]
Window size value: 28944
[Calculated window size: 28944]
► Checksum: 0x34a8 incorrect, should be 0x04d3(maybe caused by "TCP checksum offload"?)
[Checksum Status: Bad]
[Calculated Checksum: 0x04d3]
Urgent pointer: 0
► [Timestamps]

3010	00 28 ab cd 00 00 40 00	c5 8c c0 a8 16 8a 5b 8e	..(....0).....[.
3020	d6 b5 30 39 00 50 00 00	00 00 00 00 00 50 02	..09.P.....P.
3030	71 10 34 a8 00 00		q.4...

wireshark_enp0s3_20221107090318_TO0NbP.pcapng Packets: 97 · Display

4-Pon un TTL=2 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'\x02\x06\xa6\xec' # TTL, Protocol | Header Checksum
9 ip_header += b'\xc0\xa8\x16\x8a' # Source Address
10 ip_header += b'\x5b\x8e\xd6\xb5' # Destination Address
11
12 tcp_header = b'\x30\x39\x00\x50' # Source Port | Destination Port
13 tcp_header += b'\x00\x00\x00\x00' # Sequence Number
14 tcp_header += b'\x00\x00\x00\x00' # Acknowledgement Number
15 tcp_header += b'\x50\x02\x71\x10' # Data Offset, Reserved, Flags | Window Size
16 tcp_header += b'\x04\xd3\x00\x00' # Checksum | Urgent Pointer
17
18 packet = ip_header + tcp_header
19 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.

4	5.019244896	192.168.22.138	91.142.214.181	TCP	54 12345 → 80 [SYN] Seq=0 Win=28944 Len=0
5	9.227977601	192.168.22.1	255.255.255.255	UDP	399 45354 → 29810 Len=357
6	11.876443979	192.168.22.172	239.255.255.250	SSDP	217 M-SEARCH * HTTP/1.1
7	12.879874996	192.168.22.172	239.255.255.250	SSDP	217 M-SEARCH * HTTP/1.1
8	13.887538452	192.168.22.172	239.255.255.250	SSDP	217 M-SEARCH * HTTP/1.1
9	14.053088503	192.168.22.209	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1
10	14.895464510	192.168.22.172	239.255.255.250	SSDP	217 M-SEARCH * HTTP/1.1
11	15.055121085	192.168.22.209	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1
12	16.069186915	192.168.22.209	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1
13	16.843831126	fe80::5563:cfa:47f9...	ff02::c	SSDP	157 M-SEARCH * HTTP/1.1
14	16.843883910	192.168.22.216	239.255.255.250	SSDP	143 M-SEARCH * HTTP/1.1
15	16.856078803	fe80::5563:cfa:47f9...	ff02::c	SSDP	686 51211 → 3702 Len=624
▶ Frame 4: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface enp0s3, id 0 ▶ Ethernet II, Src: PcsCompu_7a:0a:da (08:00:27:7a:0a:da), Dst: 14:eb:b6:26:7b:fc (14:eb:b6:26:7b:fc) ▼ Internet Protocol Version 4, Src: 192.168.22.138, Dst: 91.142.214.181					
0100 = Version: 4 0101 = Header Length: 20 bytes (5) ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) Total Length: 40 Identification: 0xabcd (43981)					
▶ Flags: 0x0000 Fragment offset: 0					
▼ Time to live: 2 ▶ [Expert Info (Note/Sequence): "Time To Live" only 2] Protocol: TCP (6) Header checksum: 0x038d [validation disabled] [Header checksum status: Unverified] Source: 192.168.22.138 Destination: 91.142.214.181					
▶ Transmission Control Protocol, Src Port: 12345, Dst Port: 80, Seq: 0, Len: 0					

Alberto Heras Herrera

Juan Sánchez Balastegui