

S11L2

UNIT 3

Marco Falchi

21	7.436387	172.16.0.40	10.0.0.11	TCP	74	80 → 40594 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=1522920915 TSecr=1564391224 WS=512
24	7.436440	172.16.0.40	10.0.0.11	TCP	66	80 → 40594 [ACK] Seq=1 Ack=332 Win=43520 Len=0 TSval=1522920915 TSecr=1564391224
25	7.436625	172.16.0.40	10.0.0.11	TCP	304	80 → 40594 [PSH, ACK] Seq=1 Ack=332 Win=43520 Len=238 TSval=1522920915 TSecr=1564391224 [TCP PDU reassembled in 27]
27	7.436643	172.16.0.40	10.0.0.11	HTTP	681	HTTP/1.1 200 OK (text/html)
38	7.515691	172.16.0.40	10.0.0.11	HTTP	374	HTTP/1.1 404 Not Found (text/html)
20	7.436352	10.0.0.11	172.16.0.40	TCP	74	40594 → 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM TSval=1564391224 TSecr=0 WS=512
22	7.436394	10.0.0.11	172.16.0.40	TCP	66	40594 → 80 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1564391224 TSecr=1522920915
23	7.436436	10.0.0.11	172.16.0.40	HTTP	397	GET / HTTP/1.1
26	7.436628	10.0.0.11	172.16.0.40	TCP	66	40594 → 80 [ACK] Seq=332 Ack=239 Win=42496 Len=0 TSval=1564391224 TSecr=1522920915
28	7.436644	10.0.0.11	172.16.0.40	TCP	66	40594 → 80 [ACK] Seq=332 Ack=854 Win=41984 Len=0 TSval=1564391224 TSecr=1522920915
32	7.515522	10.0.0.11	172.16.0.40	HTTP	411	GET /favicon.ico HTTP/1.1

Frame 21: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, Src: c6:ae:cf:4f:cc:20 (c6:ae:cf:4f:cc:20), Dst: 96:89:28:e2:92:c7 (96:89:28:e2:92:c7) Internet Protocol Version 4, Src: 172.16.0.40, Dst: 10.0.0.11 Transmission Control Protocol, Src Port: 80, Dst Port: 40594, Seq: 0, Ack: 1, Len: 0

Source Port: 80
Destination Port: 40594
[Stream index: 0]
[Stream Packet Number: 2]
[Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 1300498739
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 1268391844
1010 = Header Length: 40 bytes (10)
Flags: 0x012 (SYN, ACK)

0000 96 89 28 e2 92 c7 c6 ae cf 4f cc 20 08 00 45 00 ... (.....)O...E:
0010 00 3c 00 00 40 00 3f 06 85 79 ac 10 00 28 0a 00 ...<..@?..y...(:
0020 00 0b 00 50 9e 92 4d 84 09 33 4b 9a 1f a4 a0 12 ...P..M...3K....
0030 a9 b0 b6 71 00 00 02 04 05 b4 04 02 08 5a c5 ...q.....Z..
0040 ed d3 5d 3e b7 38 01 03 03 09 ...]>8....

Qual è il numero di porta TCP di origine?

Il numero di porta TCP di origine è 40594

Come classifichereesti la porta di origine?

La porta di origine, 40594, è una porta effimera (o dinamica). Le porte effimere sono utilizzate dai sistemi operativi client per connessioni stabilite in uscita.

Non sono porte "ben note" (come la 80 per HTTP o la 443 per HTTPS), né porte registrate (quelle assegnate a servizi specifici ma non comunemente usate).

Qual è il numero di porta TCP di destinazione?

Il numero di porta TCP di destinazione è 80 quindi su un servizio http

Come classifichereesti la porta di destinazione?

La porta di destinazione è la 80 che è una porta nota (well-known port).

Quale flag è impostato?

Il flag SYN (Synchronize) è impostato.

Questo indica che il pacchetto fa parte della prima fase di un **three-way handshake** per stabilire una connessione TCP.

A quale valore è impostato il numero di sequenza relativo?

Il numero di sequenza relativo è impostato a 0.

Un numero di sequenza relativo di 0 è tipico del primo pacchetto di una sessione TCP, come in questo caso, dove il flag SYN è impostato.

21	7.436387	172.16.0.40	10.0.0.11	TCP	74	80 → 40594 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=1522920915 TSecr=1564391224 WS=512
24	7.436440	172.16.0.40	10.0.0.11	TCP	66	80 → 40594 [ACK] Seq=1 Ack=332 Win=43520 Len=0 TSval=1522920915 TSecr=1564391224
25	7.436625	172.16.0.40	10.0.0.11	TCP	304	80 → 40594 [PSH, ACK] Seq=1 Ack=332 Win=43520 Len=238 TSval=1522920915 TSecr=1564391224 [TCP PDU reassembled in 27]
27	7.436643	172.16.0.40	10.0.0.11	HTTP	681	HTTP/1.1 200 OK (text/html)
38	7.515691	172.16.0.40	10.0.0.11	HTTP	374	HTTP/1.1 404 Not Found (text/html)
20	7.436352	10.0.0.11	172.16.0.40	TCP	74	40594 → 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM TSval=1564391224 TSecr=0 WS=512
22	7.436394	10.0.0.11	172.16.0.40	TCP	66	40594 → 80 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1564391224 TSecr=1522920915
23	7.436436	10.0.0.11	172.16.0.40	HTTP	397	GET / HTTP/1.1
26	7.436628	10.0.0.11	172.16.0.40	TCP	66	40594 → 80 [ACK] Seq=332 Ack=239 Win=42496 Len=0 TSval=1564391224 TSecr=1522920915
28	7.436644	10.0.0.11	172.16.0.40	TCP	66	40594 → 80 [ACK] Seq=332 Ack=854 Win=41984 Len=0 TSval=1564391224 TSecr=1522920915
37	7.515522	10.0.0.11	172.16.0.40	HTTP	411	GET /favicon.ico HTTP/1.1

▶ Frame 24: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

▶ Ethernet II, Src: c6:ae:cf:4f:cc:20 (c6:ae:cf:4f:cc:20), Dst: 96:89:28:e2:92:c7 (96:89:28:e2:92:c7)

▶ Internet Protocol Version 4, Src: 172.16.0.40, Dst: 10.0.0.11

▶ Transmission Control Protocol, Src Port: 80, Dst Port: 40594, Seq: 1, Ack: 332, Len: 0

Source Port: 80
Destination Port: 40594
[Stream Index: 0]
[Stream Packet Number: 5]
[Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 0]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 1300498740
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 332 (relative ack number)
Acknowledgment number (raw): 1268392175
1000 = Header Length: 32 bytes (8)
▶ Flags: 0x010 (ACK)

0000 96 89 28 e2 92 c7 c6 ae cf 4f cc 20 08 00 45 00 ..(.....)0...E
0010 00 34 5c ad 40 00 3f 06 28 d4 ac 10 00 28 0a 00 .4\@?:(....(
0020 00 0b 00 50 9e 92 4d 84 09 34 4b 9a 20 ef 80 10 .P.M.4K...
0030 00 55 b6 69 00 00 01 01 08 0a 5a c5 ed d3 5d 3e .U-i....ZZ...>
0040 07 38 8

Quali sono i valori delle porte di origine e destinazione?

I valori delle porte sono invertiti rispetto al primo pacchetto:

Porta di origine: 80 (quella del server web)

Porta di destinazione: 40594 (la porta effimera del client che ha avviato la connessione)

Quali flag sono impostati?

Sono impostati due flag:

SYN (Synchronize): per sincronizzare i numeri di sequenza.

ACK (Acknowledgment): per confermare di aver ricevuto il primo pacchetto SYN del client.

A quali valori sono impostati i numeri relativi di sequenza acknowledgment?

Numero di sequenza relativo: 0. Questo valore è impostato a 0 per la prima volta che il server invia un pacchetto SYN.

Numero di acknowledgment relativo: 1. Questo valore indica che il server sta confermando la ricezione del pacchetto precedente del client e si aspetta il prossimo pacchetto con un numero di sequenza pari a 1.

21	7.436387	172.16.0.40	10.0.0.11	TCP	74	80 → 40594 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=1522920915 TSecr=1564391224 WS=512
24	7.436440	172.16.0.40	10.0.0.11	TCP	66	80 → 40594 [ACK] Seq=1 Ack=332 Win=43520 Len=0 TSval=1522920915 TSecr=1564391224
25	7.436625	172.16.0.40	10.0.0.11	TCP	384	80 → 40594 [PSH, ACK] Seq=1 Ack=332 Win=43520 Len=238 TSval=1522920915 TSecr=1564391224 [TCP PDU reassembled in 27]
27	7.436643	172.16.0.40	10.0.0.11	HTTP	681	HTTP/1.1 200 OK (text/html)
38	7.515691	172.16.0.40	10.0.0.11	HTTP	374	HTTP/1.1 404 Not Found (text/html)
20	7.436352	10.0.0.11	172.16.0.40	TCP	74	40594 → 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM TSval=1564391224 TSecr=0 WS=512
22	7.436394	10.0.0.11	172.16.0.40	TCP	66	40594 → 80 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1564391224 TSecr=1522920915
23	7.436436	10.0.0.11	172.16.0.40	HTTP	397	GET / HTTP/1.1
26	7.436628	10.0.0.11	172.16.0.40	TCP	66	40594 → 80 [ACK] Seq=332 Ack=239 Win=42496 Len=0 TSval=1564391224 TSecr=1522920915
28	7.436644	10.0.0.11	172.16.0.40	TCP	66	40594 → 80 [ACK] Seq=332 Ack=854 Win=41984 Len=0 TSval=1564391224 TSecr=1522920915
37	7.515522	10.0.0.11	172.16.0.40	HTTP	411	GET /favicon.ico HTTP/1.1

Frame 25: 304 bytes on wire (2432 bits), 304 bytes captured (2432 bits) on interface 0
Ethernet II, Src: c6:ae:cf:4f:cc:20 (c6:ae:cf:4f:cc:20), Dst: 96:89:28:e2:92:c7 (96:89:28:e2:92:c7)
Internet Protocol Version 4, Src: 172.16.0.40, Dst: 10.0.0.11
Transmission Control Protocol, Src Port: 80, Dst Port: 40594, Seq: 1, Ack: 332, Len: 238
Source Port: 80
Destination Port: 40594
[Stream index: 0]
[Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 238]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 1300498740
[Next Sequence Number: 239 (relative sequence number)]
Acknowledgment Number: 332 (relative ack number)
Acknowledgment number (raw): 1268392175
1000 ... = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window: 85
[Calculated window size: 43520]
[Window size scaling factor: 512]
Checksum: 0xb757 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0

0000 96 89 28 e2 92 c7 c6 ae cf 4f cc 20 08 00 45 00 --{-----O---E-
0010 01 22 5c ae 40 00 3f 06 27 e5 ac 10 00 28 0a 00 --\ @? - ----(
0020 00 00 00 50 9e 92 4d 04 09 34 4b 9a 20 ef 80 18 --[P-M-4K-
0030 00 55 b7 57 00 00 01 01 08 0a 5a c5 ed d3 5d 3e --[U-W-...-Z-...]
0040 b7 38 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f --8HTTP/1.1 200 O
0050 4b 0d 0a 53 65 72 76 65 72 3a 20 6e 67 69 6e 78 K Serve r: ngxin
0060 2f 31 2e 32 38 2e 30 0d 0a 44 61 74 65 3a 20 54 /1.28.0 Date: T
0070 75 65 2c 20 32 33 20 53 65 70 20 32 30 32 35 20 ue, 23 S ep 2025
0080 31 32 3a 31 34 3a 30 33 20 47 4d 54 0d 0a 43 6f 12:14:03 GMT Co
0090 6e 74 65 6e 74 2d 54 79 70 65 3a 20 74 65 78 74 ntent-Ty pe: text
00a0 2f 68 74 6d 6c 0d 0a 43 6f 6e 74 65 6e 74 2d 4c /html Content-L
00b0 65 6e 67 74 68 3a 20 36 31 35 0d 0a 4c 61 73 74 engh: 6 15 Last
00c0 2d 4d 6f 64 69 66 69 65 64 3a 20 57 65 64 2c 20 -Modifie d: Wed,
00d0 33 30 20 41 70 72 20 32 30 32 35 20 32 33 3a 34 30 Apr 2 025 23:4
00e0 37 3a 33 36 20 47 4d 54 0d 0a 43 6f 6e 6e 65 63 7:36 GMT Connec
00f0 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65 tion: ke ep-alive
0100 0d 0a 45 54 61 67 3a 20 22 36 38 31 32 62 36 39 ETag: "6812b69
0110 38 2d 32 36 37 22 0d 0a 41 63 63 65 70 74 2d 52 8-267" Accept-R
0120 61 6e 67 65 73 3a 20 62 79 74 65 73 0d 0a 0d 0a anges: b ytes ---

Quale flag è impostato?

Il flag **ACK** (Acknowledgment) è impostato.

L'impostazione di questo flag conferma che il client ha ricevuto il pacchetto precedente del server (il SYN-ACK) e che la connessione è stata stabilita con successo, pronta per lo scambio di dati.

```
tcpdump [ -AbDefhHIJKlLnNOpqStuUvxX# ] [ -B buffer size ]  
[ -c count ] [ --count ] [ -C file size ]  
[ -E spi@ipaddr algo:secret,... ]  
[ -F file ] [ -G rotate seconds ] [ -i interface ]  
[ --immediate-mode ] [ -j tstamp type ] [ -m module ]  
[ -M secret ] [ --number ] [ --print ] [ -Q in|out|inout ]  
[ -r file ] [ -s snaplen ] [ -T type ] [ --version ]  
[ -V file ] [ -w file ] [ -W filecount ] [ -y datalinktype ]  
[ -z postrotate-command ] [ -Z user ]  
[ --time-stamp-precision=tstamp precision ]  
[ --micro ] [ --nano ]  
[ expression ]
```

Cosa fa l'opzione -r?

-r è utilizzata per leggere i pacchetti da un file.

Ci sono centinaia di filtri disponibili in Wireshark. Una rete di grandi dimensioni potrebbe avere numerosi filtri e molti tipi diversi di traffico. Elenca tre filtri che potrebbero essere utili a un amministratore di rete.

(ricerca per la risposta svolta tramite ai)

tcp.flags.syn == 1 and tcp.flags.ack == 0: Questo filtro permette di visualizzare i pacchetti TCP con il solo flag SYN impostato. È uno strumento per identificare scansioni di porte o tentativi di connessione non riusciti, che potrebbero indicare attività di ricognizione o attacchi.

http.request.method == "POST": Se un amministratore ha bisogno di analizzare il traffico legato all'invio di dati come moduli web questo filtro mostra solo le richieste HTTP di tipo POST, rendendo più semplice il debugging delle applicazioni web.

icmp.type == 8: Questo filtro isola specificamente i messaggi di richiesta Echo (echo request) detti anche ping . Può essere utilizzato per monitorare la latenza e la raggiungibilità degli host, o per analizzare il traffico di diagnostica

In quali altri modi di Wireshark potrebbe essere utilizzato in una rete di produzione?

Sicurezza e analisi forense: Gli amministratori possono catturare il traffico per rilevare attività malevole, come tentativi di esfiltrazione di dati, comunicazioni con server di comando e controllo (C2) o l'uso di protocolli non autorizzati.

Debug di applicazioni: Gli sviluppatori e gli ingegneri di sistema lo utilizzano per esaminare i dettagli a livello di pacchetto delle comunicazioni tra applicazioni. Permette di verificare che le chiamate API, i protocolli personalizzati e le risposte del server funzionino correttamente, facilitando la diagnosi di errori in ambienti complessi.

Analisi delle prestazioni: Wireshark può essere utilizzato per individuare i colli di bottiglia della rete. Analizzando i pacchetti, si possono identificare problemi come elevata latenza, pacchetti persi (mostrati come retransmissions) e problemi di congestione, permettendo di ottimizzare il flusso dei dati.