

Three-way-handshake del protocollo TCP tramite Wireshark

Il *three-way handshake* (stretta di mano a tre vie) è il processo utilizzato dal protocollo TCP (Transmission Control Protocol) per stabilire una connessione affidabile tra un client e un server. Questo meccanismo garantisce che entrambe le parti siano pronte per comunicare e che il collegamento sia stabile prima di inviare dati.

Per verificare se la connessione è stata stabilita tra host e server, si andrà a simulare un ambiente realistico di comunicazione.

Come primo passaggio avvieremo la *mininet* sulla macchina virtuale *CyberOps*.

```
CyberOPS Topology:
      -----
      | R1 |-----| H4 |
      -----
      |
      |
      -----
      |-----| S1 |-----|
      |         |         |
      |         |         |
      |         |         |
      -----
      | H1 |   | H2 |   | H3 |
      -----

*** Add links
*** Creating network
*** Adding hosts:
H1 H2 H3 H4 R1
*** Adding switches:
s1
*** Adding links:
(H1, s1) (H2, s1) (H3, s1) (H4, R1) (s1, R1)
*** Configuring hosts
H1 H2 H3 H4 R1
*** Starting controller

*** Starting 1 switches
s1 ...
*** Routing Table on Router:
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U        0      0        0 R1-eth1
172.16.0.0      0.0.0.0         255.240.0.0     U        0      0        0 R1-eth2

*** Starting CLI:
mininet> xterm H1
mininet> xterm H4
mininet>
```

Successivamente si avviano il server (H4) e l'host (H1). Su quest'ultimo si aprirà il browser e si navigherà all'indirizzo *172.16.0.40*. Si avvierà *tcpdump* per la cattura del traffico della connessione sulla porta 80 con il comando *[analyst@secOps ~]\$ sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analyst/capture.pcap*, che verrà salvato su un file denominato *capture.pcap*.

In seguito avviare Wireshark e aprire il file salvato, descritto poco sopra. Una volta avviato, si inserirà nel filtro "TCP" in modo da avere come risultato solo quello che ci interessa, cioè la connessione con la porta 80.

SYN

The screenshot shows the Wireshark 2.5.1 interface with a packet capture of a SYN request. The filter is set to 'tcp'. The packet list shows a SYN packet from 10.0.0.11 to 172.16.0.40 on port 80. The packet details pane shows the TCP segment length, sequence number, and flags. The 'Flags' section is expanded, showing 'Syn: Set' highlighted with a red box. The 'Expert Info' section shows the connection establish request (SYN) for server port 80.

No.	Time	Source	Destination	Protocol	Length	Info
10	3.383134	10.0.0.11	172.16.0.40	TCP	74	42622 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1
11	3.383170	172.16.0.40	10.0.0.11	TCP	74	80 → 42622 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1
12	3.383177	10.0.0.11	172.16.0.40	TCP	66	42622 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=252713 TSecr=0
13	3.383275	10.0.0.11	172.16.0.40	HTTP	377	GET / HTTP/1.1
14	3.383286	172.16.0.40	10.0.0.11	TCP	66	80 → 42622 [ACK] Seq=1 Ack=312 Win=30708 Len=0 TSval=7275 TSecr=0
15	3.383287	172.16.0.40	10.0.0.11	TCP	304	80 → 42622 [RST] Seq=1 Len=0 Window=0

[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
[Next sequence number: 0 (relative sequence number)]
Acknowledgment number: 0
1010 = Header Length: 40 bytes (10)

▼ 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
▼ [Expert Info (Chat/Sequence): Connection establish request (SYN): server port 80]
[Connection establish request (SYN): server port 80]

Nella prima riga selezionata, si vede il primo messaggio SYN inviato. Questo lo si può vedere nel dettaglio in basso, alla voce Flags. Come si può notare tutte le voci sono con valore 0, tranne SYN che ha 1.

SYN/ACK

Con lo stesso procedimento di prima per il SYN, si può verificare il secondo messaggio SYN/ACK.

capture.pcap [Wireshark 2.5.1]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Show the capture options... Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
10	3.383134	10.0.0.11	172.16.0.40	TCP	74	42622 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
11	3.383170	172.16.0.40	10.0.0.11	TCP	74	80 → 42622 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0
12	3.383177	10.0.0.11	172.16.0.40	TCP	66	42622 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSv
13	3.383275	10.0.0.11	172.16.0.40	HTTP	377	GET / HTTP/1.1
14	3.383286	172.16.0.40	10.0.0.11	TCP	66	80 → 42622 [ACK] Seq=1 Ack=312 Win=30208 Len=0 T
15	3.38327	172.16.0.40	10.0.0.11	TCP	304	80 → 42622 [RST] Seq=1 Len=0

Destination Port: 42622

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 0 (relative sequence number)

[Next sequence number: 0 (relative sequence number)]

Acknowledgment number: 1 (relative ack number)

1010 = Header Length: 40 bytes (10)

▼ Flags: 0x012 (SYN, ACK)

- 000. = Reserved: Not set
- ...0 = Nonce: Not set
- 0... = Congestion Window Reduced (CWR): Not set
-0.. = ECN-Echo: Not set
-0... = Urgent: Not set
- ...1 = Acknowledgment: Set
-0... = Push: Not set
-0 = Reset: Not set
- ▼1. = Syn: Set

ACK

Ovviamente questo vale anche per l' ultimo messaggio ACK che chiude il Three-way Handshake, come si può verificare nell' immagine sottostante.

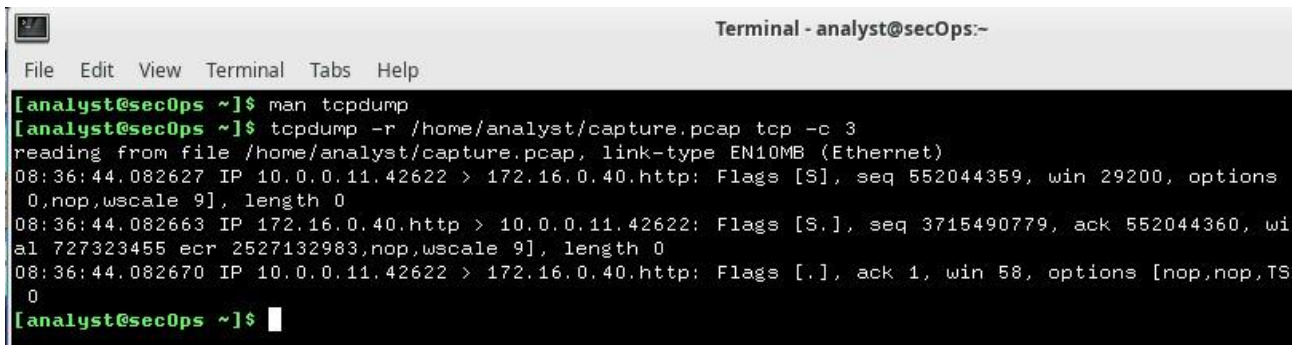
The image shows a Wireshark 2.5.1 capture of a TCP three-way handshake. The packet list shows packets 10 through 15. Packet 12 is selected, showing the details of the TCP segment. The 'Flags' field is expanded, and the 'Acknowledgment' flag is highlighted with a red box, indicating it is set.

No.	Time	Source	Destination	Protocol	Length	Info
10	3.383134	10.0.0.11	172.16.0.40	TCP	74	42622 → 80 [SYN] Seq
11	3.383170	172.16.0.40	10.0.0.11	TCP	74	80 → 42622 [SYN, AC
12	3.383177	10.0.0.11	172.16.0.40	TCP	66	42622 → 80 [ACK] Seq
13	3.383275	10.0.0.11	172.16.0.40	HTTP	377	GET / HTTP/1.1
14	3.383286	172.16.0.40	10.0.0.11	TCP	66	80 → 42622 [ACK] Seq
15	3.383827	172.16.0.40	10.0.0.11	TCP	304	80 → 42622 [ACK] Seq

[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
1000 = Header Length: 32 bytes (8)
▼ Flags: 0x010 (ACK)
000. = Reserved: Not set
...0 = Nonce: Not set
.... 0... = Congestion Window Reduced (CWR): Not set
.... .0.. = ECN-Echo: Not set
.... .0. = Urgent: Not set
.... ..1 = Acknowledgment: Set
.... 0... = Push: Not set
....0.. = Reset: Not set
....0. = Syn: Not set
....0 = Fin: Not set

Vedere il pacchetto con Tcpdump

Come si può vedere nell' immagine sottostante, si può aprire il file con il traffico salvato e visualizzarlo sul terminale tramite Tcpdump. Usando il comando `tcpdump -r /home/analyst/capture.pcap tcp -c 3` vengono mostrare solo le prime tre righe che interessano a noi, cioè quelle del Three-way handshake.

A terminal window titled "Terminal - analyst@secOps:~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The prompt is [analyst@secOps ~]\$ and the user has entered 'man tcpdump'. The prompt is again [analyst@secOps ~]\$ and the user has entered 'tcpdump -r /home/analyst/capture.pcap tcp -c 3'. The terminal shows the output of the command: 'reading from file /home/analyst/capture.pcap, link-type EN10MB (Ethernet)' followed by three lines of network traffic data representing a three-way handshake. The first line is from 10.0.0.11 to 172.16.0.40, the second is from 172.16.0.40 to 10.0.0.11, and the third is from 10.0.0.11 to 172.16.0.40. The prompt is [analyst@secOps ~]\$ with a cursor.

```

[analyst@secOps ~]$ man tcpdump
[analyst@secOps ~]$ tcpdump -r /home/analyst/capture.pcap tcp -c 3
reading from file /home/analyst/capture.pcap, link-type EN10MB (Ethernet)
08:36:44.082627 IP 10.0.0.11.42622 > 172.16.0.40.http: Flags [S], seq 552044359, win 29200, options
 0,nop,wscale 9], length 0
08:36:44.082663 IP 172.16.0.40.http > 10.0.0.11.42622: Flags [S.], seq 3715490779, ack 552044360, wi
al 727323455 ecr 2527132983,nop,wscale 9], length 0
08:36:44.082670 IP 10.0.0.11.42622 > 172.16.0.40.http: Flags [.], ack 1, win 58, options [nop,nop,TS
 0
[analyst@secOps ~]$
```