

# Relazione Tecnica: Analisi dell'Handshake a 3 Vie TCP

**Studente:** Victor Rosati

**Laboratorio:** Cisco CyberOps - Analisi del Traffico Web

## 1. Obiettivo e Scenario

L'esercitazione ha avuto come scopo la cattura e l'analisi forense dell'**handshake a tre vie TCP**.

Utilizzando la VM CyberOps Workstation, abbiamo simulato una connessione tra un client (10.0.0.11) e un server web (172.16.0.40) per osservare come viene stabilita una sessione affidabile.

## 2. Fase Operativa di Cattura

Dopo aver avviato il server su **H4** e il browser su **H1**, abbiamo utilizzato lo strumento **tcpdump** per intercettare i dati.

- **Comando di cattura:** `sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analyst/capture.pcap`.
- **Risultato:** Sono stati catturati correttamente **50 pacchetti** durante il caricamento della pagina web di nginx.

## 3. Analisi Dettagliata con Wireshark

Attraverso Wireshark, abbiamo isolato la sequenza di instaurazione della connessione.

### A. La Sequenza dell'Handshake

L'immagine seguente mostra l'intera sequenza temporale dei pacchetti catturati, evidenziando il passaggio da SYN a SYN-ACK e infine ad ACK.

```

▼ Transmission Control Protocol, Src Port
  Source Port: 80
  Destination Port: 52410
  [Stream index: 0]
  [Stream Packet Number: 2]
  ▶ [Conversation completeness: Incomplete]
    [TCP Segment Len: 0]
    Sequence Number: 0      (relative sequence number)
    Sequence Number (raw): 3518945642
    [Next Sequence Number: 1      (relative sequence number)]
    Acknowledgment Number: 1      (relative sequence number)
    Acknowledgment number (raw): 32332530
    1010 .... = Header Length: 40 bytes (relative sequence number)
  ▶ Flags: 0x012 (SYN, ACK)
  Window: 43440
  [Calculated window size: 43440]
  Checksum: 0xb671 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  ▶ Options: (20 bytes), Maximum segment size: 1460
  ▶ [Timestamps]
  ▶ [SEQ/ACK analysis]

```

Transmission Control Protocol: Protocol

*Lista dei pacchetti filtrata per protocollo TCP.*

## B. Analisi del Secondo Pacchetto (SYN-ACK)

Questo pacchetto rappresenta la risposta del server che accetta la connessione.

- **Flag:** **SYN** e **ACK** sono entrambi impostati a 1.
- **Numeri Relativi:** Il numero di sequenza è **0**, mentre l'Acknowledgment è **1**.

## C. Analisi del Terzo Pacchetto (ACK Finale)

Il client invia la conferma finale per iniziare lo scambio di dati HTTP.

- **Flag:** È impostato esclusivamente il flag **ACK**.
- **Numeri Relativi:** Sia **Sequence** che **Acknowledgment** sono impostati a **1**.

```

▶ Frame 27: 66 bytes on wire (528 bits),
▶ Ethernet II, Src: 7e:2d:86:36:9b:ef (7e:2d:86:36:9b:ef)
▶ Internet Protocol Version 4, Src: 10.0
└ Transmission Control Protocol, Src Port: 52410
    Destination Port: 80
    [Stream index: 0]
    [Stream Packet Number: 3]
    [Conversation completeness: Incomplete]
    [TCP Segment Len: 0]
    Sequence Number: 1 (relative sequence number)
    Sequence Number (raw): 3233253041
    [Next Sequence Number: 1 (relative sequence number)]
    Acknowledgment Number: 1 (relative sequence number)
    Acknowledgment number (raw): 3518945
    1000 . . . = Header Length: 32 bytes
    └ Flags: 0x010 (ACK)
        Window: 83
        [Calculated window size: 42496]
        [Window size scaling factor: 512]
        Checksum: 0xb669 [unverified]
        [Checksum Status: Unverified]
        Urgent Pointer: 0
    └ Options: (12 bytes), No-Operation (NOP)
    └ [Timestamps]
    └ [SEQ/ACK analysis]

```

*Dettagli del pacchetto ACK (Frame 27).*

## 4. Analisi via Riga di Comando (tcpdump)

Abbiamo inoltre verificato i dati leggendo il file di cattura tramite terminale con l'opzione `-r` (read).

- **Flag Identificati:** `[S]` per SYN, `[S.]` per SYN-ACK e `[.]` per ACK.

*Output testuale di tcpdump dell'handshake.*

## 5. Riflessioni Finali

1. **Filti Utili:** Un amministratore di rete può trarre grande beneficio dall'uso di filtri come `tcp.port == 80` per il traffico web o `ip.addr == [IP]` per isolare specifici host durante un'analisi di sicurezza.
2. **Utilizzo Professionale:** Wireshark è fondamentale in contesti di produzione per il troubleshooting della latenza di rete e per rilevare possibili tentativi di attacco o esfiltrazione di dati non autorizzati.

### Analisi del primo pacchetto (SYN)

- Qual è il numero di porta TCP di origine? `52410`.
- Come classificheresti la porta di origine? Porta effimera o dinamica.
- Qual è il numero di porta TCP di destinazione? `80`.
- Come classificheresti la porta di destinazione? Porta Well-Known (HTTP).
- Quale flag è impostato? `SYN`.

- A quale valore è impostato il numero di sequenza relativo? 0.

### Analisi del secondo pacchetto (SYN, ACK)

- Quali sono i valori delle porte di origine e destinazione? Origine 80, Destinazione 52410.
- Quali flag sono impostati? SYN e ACK.
- A quali valori sono impostati i numeri relativi di sequenza e acknowledgment? Sequence 0, Acknowledgment 1.

### Analisi del terzo pacchetto (ACK)

- Quale flag è impostato? Solo ACK.
- A quali valori sono impostati i numeri relativi di sequenza e acknowledgment? Sequence 1, Acknowledgment 1.

### Analisi con tcpdump

- Cosa fa l'opzione -r? Permette a `tcpdump` di leggere i pacchetti da un file salvato (file .pcap) anziché catturarli dal vivo.

### Domande di Riflessione

- Elenca tre filtri che potrebbero essere utili a un amministratore di rete:
  1. `tcp.port == 80`: Filtra solo il traffico web HTTP.
  2. `ip.addr == 172.16.0.40`: Mostra il traffico da/verso uno specifico host.
  3. `icmp`: Filtra i pacchetti di controllo e di errore (come il ping).
- In quali altri modi Wireshark potrebbe essere utilizzato in una rete di produzione?
  1. Risoluzione di problemi di latenza o connettività.
  2. Analisi forense per individuare intrusioni o attacchi malware.
  3. Monitoraggio della conformità dei protocolli di sicurezza.