

# Laboratori giorno 2 – Cisco CyberOps

## Relazione: Analisi del Traffico di Rete con Wireshark e tcpdump

### Introduzione

L'obiettivo di questo esercizio è comprendere il funzionamento del protocollo TCP (Transmission Control Protocol) e analizzare il traffico di rete utilizzando strumenti come tcpdump e Wireshark. Attraverso l'uso di una macchina virtuale (CyberOps Workstation) e la simulazione di una rete con Mininet, abbiamo catturato e analizzato i pacchetti scambiati tra un client (H1) e un server web (H4). Questo esercizio ci ha permesso di osservare il processo di handshake a tre vie (three-way handshake) e di esaminare i dettagli dei pacchetti TCP.

### Obiettivi

1. Preparare gli host per catturare il traffico :
  - Configurare una rete simulata con Mininet.
  - Avviare un server web su H4 e un browser su H1.
  - Catturare il traffico di rete utilizzando tcpdump.
2. Analizzare i pacchetti con Wireshark :
  - Esaminare il three-way handshake TCP.
  - Identificare le porte sorgente e destinazione, i flag TCP e i numeri di sequenza/acknowledgment.
3. Visualizzare i pacchetti con tcpdump :
  - Leggere il file .pcap generato da tcpdump.
  - Filtrare e visualizzare i pacchetti TCP.

### Ambiente di Lavoro

- Macchina Virtuale : CyberOps Workstation.
- Strumenti Utilizzati :
  - Mininet : Per simulare una rete con due nodi (H1 e H4).
  - tcpdump : Per catturare il traffico di rete.
  - Wireshark : Per analizzare i pacchetti catturati.
- Configurazione della Rete :
  - La VM è stata configurata in modalità NAT per consentire la connettività internet.
  - Mininet ha creato una rete interna con due host virtuali:
    - H1 : Simula un client con indirizzo IP 10.0.0.11.

- H4 : Simula un server web con indirizzo IP 172.16.0.40.

## **Procedura**

### **Parte 1: Preparazione degli Host**

1. Avvio della VM :
  - La VM CyberOps Workstation è stata avviata e accesso effettuato con l'utente analyst.
2. Simulazione della Rete con Mininet :
  - È stato eseguito lo script `cyberops_topo.py` per creare una rete simulata con due nodi (H1 e H4).
3. Avvio del Server Web :
  - Su H4, è stato avviato uno script (`reg_server_start.sh`) per simulare un server web in ascolto sulla porta 80.
4. Navigazione sul Browser :
  - Su H1, è stato avviato Firefox e si è navigato verso l'indirizzo IP del server web (`http://172.16.0.40`).
5. Cattura del Traffico :
  - Utilizzando `tcpdump`, sono stati catturati 50 pacchetti di rete e salvati in un file chiamato `capture1.pcap`.

### **Parte 2: Analisi dei Pacchetti con Wireshark**

1. Apertura del File .pcap :
  - Il file `capture.pcap` è stato aperto in Wireshark per analizzare il traffico catturato.
2. Filtro TCP :
  - È stato applicato un filtro TCP (`tcp`) per visualizzare solo i pacchetti relativi al protocollo TCP.



19	4.941664	172.16.0.40	10.0.0.11	TCP	74	80 → 55388 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3096250453 TSecr=100340
▶ Frame 19: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) ▶ Ethernet II, Src: 9a:d7:04:ad:5d:8b (9a:d7:04:ad:5d:8b), Dst: 12:e5:cb:ce:43:e8 (12:e5:cb:ce:43:e8) ▶ Internet Protocol Version 4, Src: 172.16.0.40, Dst: 10.0.0.11 ▼ Transmission Control Protocol, Src Port: 80, Dst Port: 55388, Seq: 0, Ack: 1, Len: 0						
Source Port: 80 Destination Port: 55388 [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 ....00 = Fin: Not set [TCP Flags: .....A..S.]						
0000 12 e5 cb ce 43 e8 9a d7 04 ad 5d 8b 08 00 45 00 ...C... ..E. 0010 00 3c 00 00 40 00 3f 06 85 79 ac 10 00 28 0a 00 ...<.@.?.y...[.. 0020 00 0b 00 50 d8 5c 4f 36 ed 75 47 21 00 d3 a0 12 ...P\O6.uGI.... 0030 71 20 b6 71 00 00 02 04 05 b4 04 02 08 0a b8 8d q.q.... .....						

18	4.941633	10.0.0.11	172.16.0.40	TCP	74	55388 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1003401865 TSecr=0 WS=512
19	4.941664	172.16.0.40	10.0.0.11	TCP	74	80 → 55388 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3096250453 TSecr=100340
20	4.941672	10.0.0.11	172.16.0.40	TCP	66	55388 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=1003401865 TSecr=3096250453
▶ Frame 20: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) ▶ Ethernet II, Src: 12:e5:cb:ce:43:e8 (12:e5:cb:ce:43:e8), Dst: 9a:d7:04:ad:5d:8b (9a:d7:04:ad:5d:8b) ▶ Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40 ▼ Transmission Control Protocol, Src Port: 55388, Dst Port: 80, Seq: 1, Ack: 1, Len: 0						
Source Port: 55388 Destination Port: 80 [Stream index: 0] [TCP Segment Len: 0] Sequence number: 1 (relative sequence number) [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 ....00 = Fin: Not set						
0000 9a d7 04 ad 5d 8b 12 e5 cb ce 43 e8 08 00 45 00 ....C... ..E.. 0010 00 34 39 0e 40 00 04 06 4b 73 0a 00 00 0b ac 10 ...49.@.@.KS..... 0020 00 28 d8 5c 00 50 47 21 00 d3 4f 36 ed 76 80 10 ...(\.PGI..O6.v.. 0030 00 3a b6 69 00 00 01 01 08 0a 3b ce b2 89 b8 8d ...!.....;.....						

### Parte 3: Visualizzazione dei Pacchetti con tcpdump

1. Lettura del File .pcap :
  - Il comando `tcpdump -r /home/analyst/capture.pcap tcp -c 3` è stato utilizzato per visualizzare i primi 3 pacchetti TCP.
2. Verifica del Three-Way Handshake :
  - I pacchetti visualizzati corrispondono a quelli analizzati in Wireshark:
    - Primo pacchetto: Flag SYN.
    - Secondo pacchetto: Flag SYN e ACK.
    - Terzo pacchetto: Flag ACK.

```
[analyst@secOps ~]$ man tcpdump
[analyst@secOps ~]$ tcpdump -r /home/analyst/capture1.pcap tcp -c 3
reading from file /home/analyst/capture1.pcap, link-type EN10MB (Ethernet)
08:47:41.567483 IP 10.0.0.11.55388 > 172.16.0.40.http: Flags [S], seq 1193345234, win 29200, options [mss 1460,sackOK,TS val 1003401865 ecr 0,nop,wscale 9], length 0
08:47:41.567514 IP 172.16.0.40.http > 10.0.0.11.55388: Flags [S.], seq 1320999797, ack 1193345235, win 28960, options [mss 1460,sackOK,TS val 3096250453 ecr 1003401865,nop,wscale 9], length 0
08:47:41.567522 IP 10.0.0.11.55388 > 172.16.0.40.http: Flags [.], ack 1, win 58, options [nop,nop,TS val 1003401865 ecr 3096250453], length 0
[analyst@secOps ~]$
```

## Risultati

- Three-Way Handshake :
  - Il processo di handshake a tre vie è stato osservato chiaramente nei pacchetti catturati.
  - I numeri di sequenza e acknowledgment hanno confermato l'inizio della comunicazione tra il client e il server.
- Porte e Flag TCP :
  - La porta sorgente del client era dinamica, mentre la porta destinazione era 80 (HTTP).
  - I flag TCP (SYN, ACK) hanno dimostrato il corretto funzionamento del protocollo.

## Conclusioni

Questo esercizio ha fornito una comprensione pratica del funzionamento del protocollo TCP e delle tecniche di cattura e analisi del traffico di rete. Gli strumenti utilizzati, tcpdump e Wireshark , sono essenziali per monitorare e diagnosticare problemi di rete, nonché per analizzare attacchi o anomalie. La simulazione della rete con Mininet ha reso possibile eseguire l'esercizio in un ambiente controllato e sicuro.