ESERCIZIO S11-L2

Wireshark e TCPdump Obiettivo del Laboratorio

L'attività pratica si concentra sull'utilizzo di Wireshark per l'osservazione del processo di TCP threeway handshake attraverso tre fasi principali:

Configurazione degli host per l'acquisizione del traffico di rete

Esame dei pacchetti attraverso l'interfaccia grafica di Wireshark

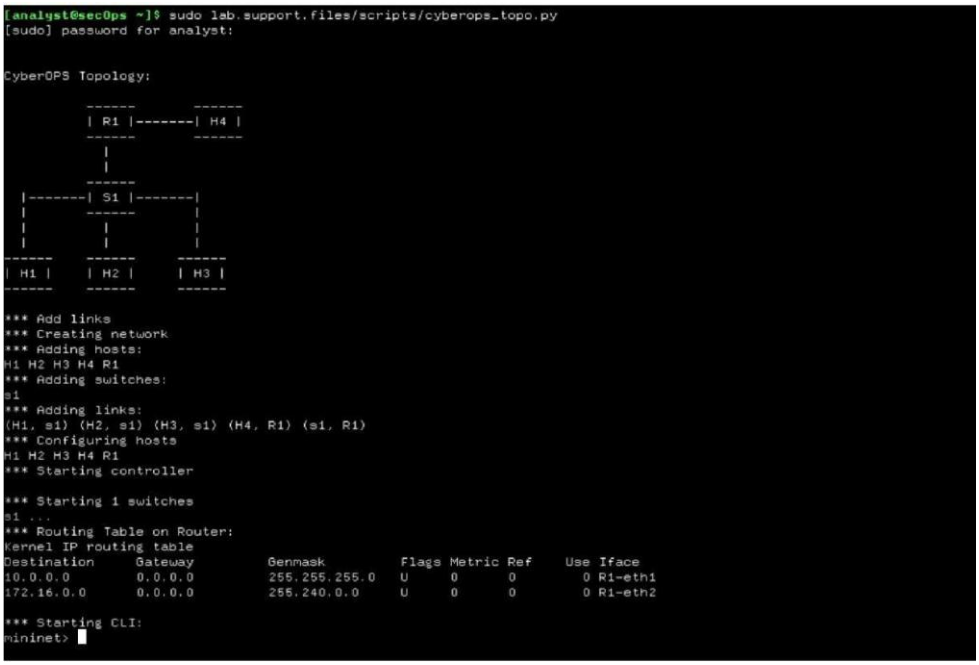
Analisi dei dati catturati mediante il tool da riga di comando tcpdump

Realizzazione del Laboratorio

Configurazione dell'Ambiente di Lavoro

Inizialmente, ho impostato l'ambiente di laboratorio utilizzando la macchina virtuale CyberOps.

Per avviare l'infrastruttura di rete, ho lanciato il comando: ***sudo lab.support.files/scripts/cyberops_topo.py*** che attiva Mininet.
Questo genera un ambiente virtuale completo con switch e host simulati.



Procedo quindi con l'attivazione degli host H1 e H4 utilizzando i comandi *xterm H1* e *xterm H4* .
Successivamente, sul nodo H4 avvio il servizio web server eseguendo ***/home/analyst/lab.support.files/scripts/reg_server_start.sh.***

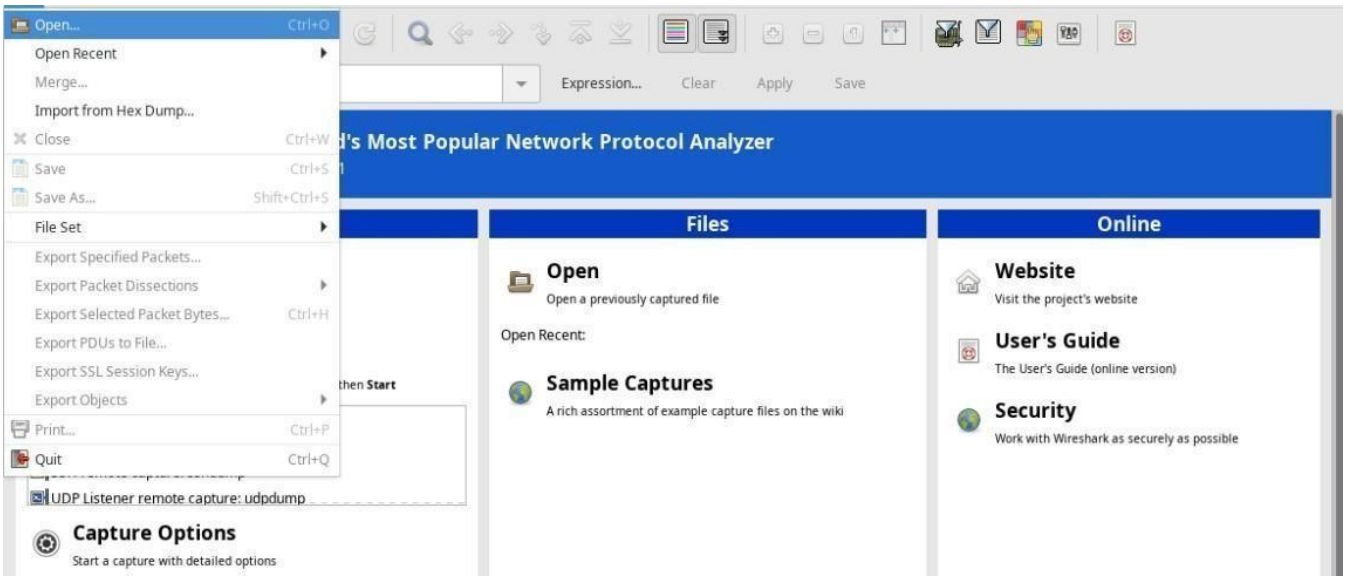
Dato che per motivi di sicurezza non è possibile eseguire firefox con privilegi di root, sull'host H1 cambio l'utente corrente ad analyst tramite *su analyst* e lancio il browser con il comando ***firefox & .*** Durante l'attesa dell'apertura della finestra del browser, attivo una sessione di cattura tcpdump sull'host H1, memorizzando i risultati nel file **capture.pcap** attraverso il comando ***sudo tcpdump -i H1eth0 -v -c 50 w /home/analyst/capture.pcap .***

Generazione del Traffico di Rete

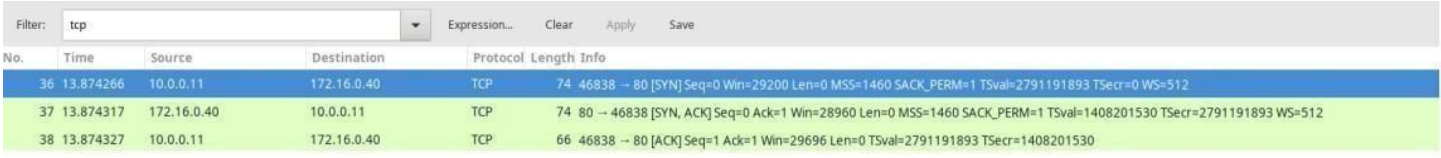
Una volta completata la configurazione, navigo con firefox verso l'indirizzo **172.16.0.40** *per generare il traffico HTTP necessario all'analisi.*

Esame del Traffico con Wireshark

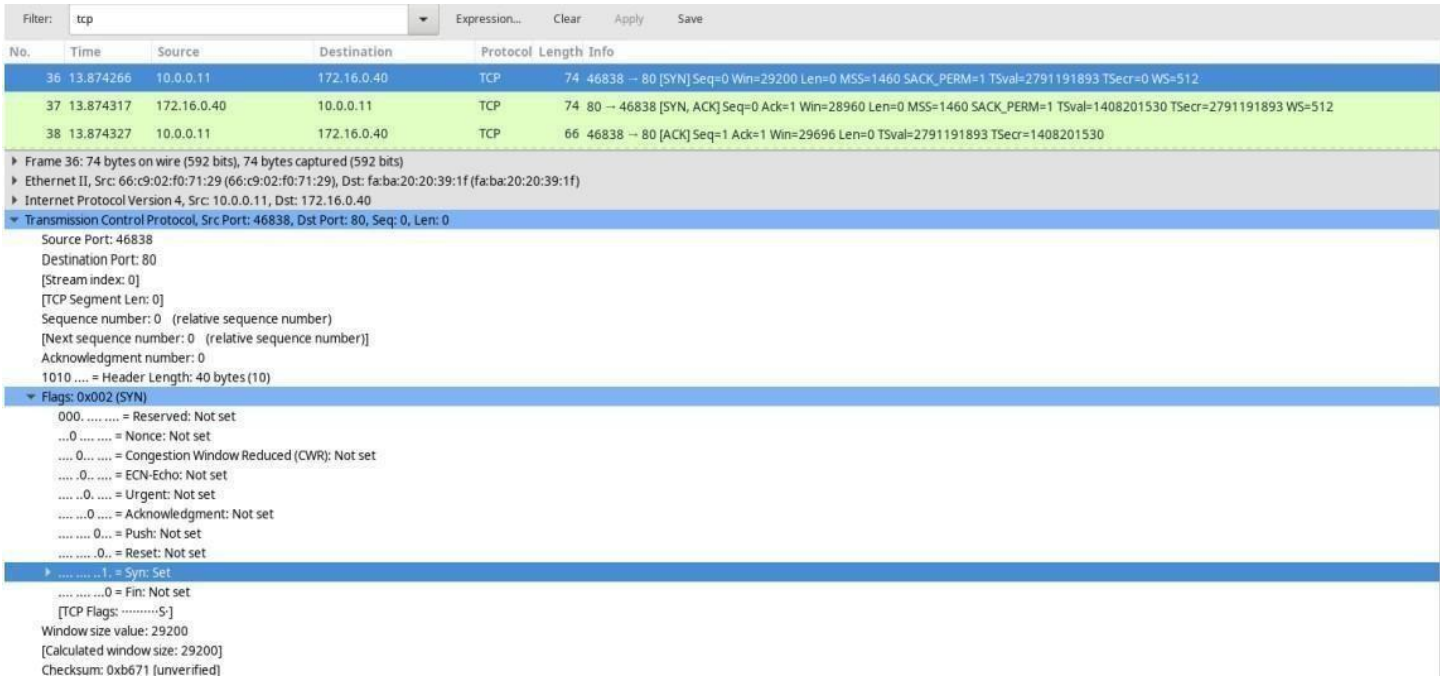
Per procedere con l'analisi dettagliata, lancio Wireshark e carico la cattura precedentemente effettuata attraverso tcpdump. Accedo al menu *File >>> Open e seleziono il file da esaminare.*



Applico quindi il **filtro tcp** per concentrarmi esclusivamente sui pacchetti del protocollo TCP. I frame **36, 37 e 38** risultano essere quelli di particolare interesse, *poiché rappresentano la sequenza completa del three-way handshake TCP.*



Il frame **36** *dà inizio alla procedura di handshake tra il **PC H1** e il **server H4**.* Attraverso il pannello Packet List nella parte inferiore dell'interfaccia, posso esaminare tutte le informazioni relative al pacchetto.



Nella sezione inferiore della finestra vengono visualizzate tutte le informazioni significative, inclusi gli indirizzi IP di origine e destinazione, i numeri di porta e gli indirizzi MAC.

Espandendo le informazioni specifiche del protocollo TCP, posso verificare che il flag SYN è impostato al valore 1.

Analizzando nel dettaglio il primo pacchetto:

- ***Numero di porta TCP di origine***: 46838 (porta dinamica/effimera)

- ***Numero di porta TCP di destinazione***: 80 (well-known port HTTP)

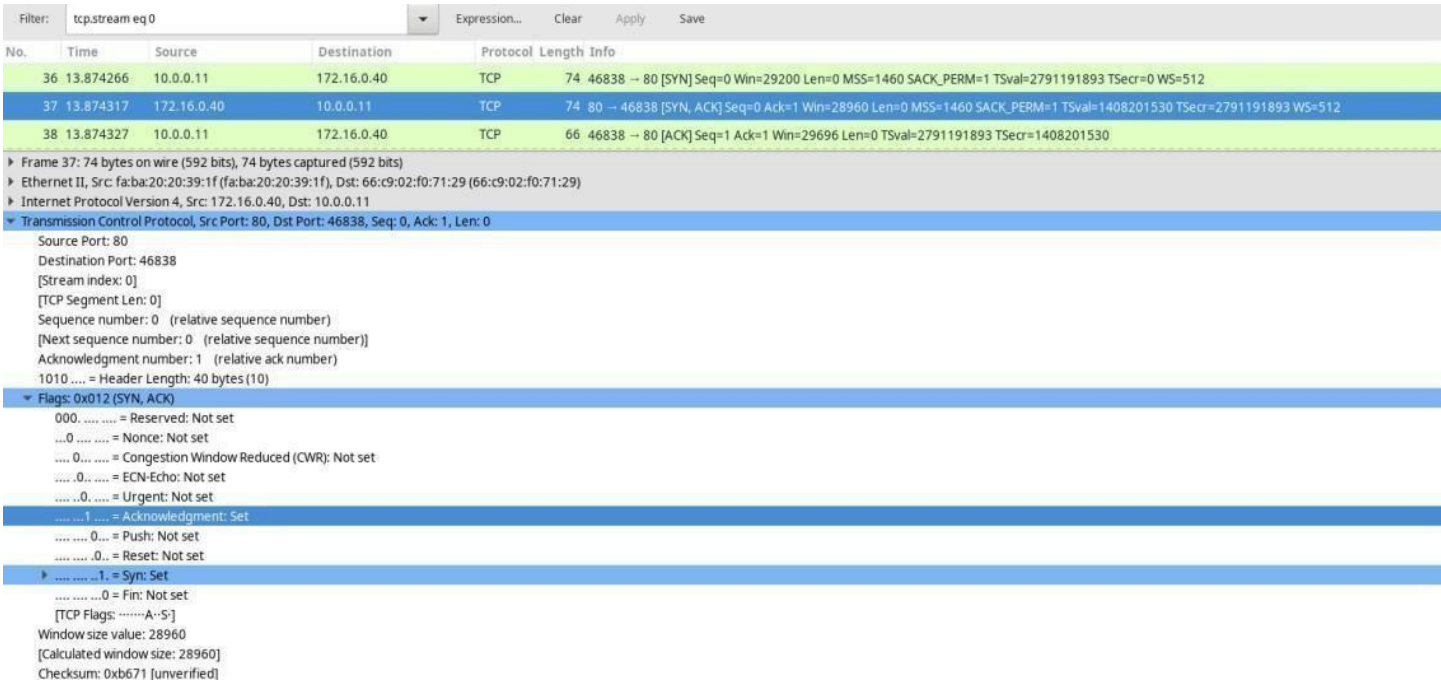
- ***Flag impostato***: SYN = 1

- ***Numero di sequenza relativo***: 0

Il pacchetto 37 rappresenta la risposta del server verso il client H1. In questo caso, nella sezione TCP, risultano impostati a 1 entrambi i flag SYN e ACK.

Esaminando i dettagli del secondo pacchetto:

- ***Porte di origine e destinazione***: 80 → 46838 (invertite rispetto al primo)
- ***Flag impostati***: SYN = 1 e ACK = 1
- ***Sequence number relativo***: 0
- ***Acknowledgment number***: 1 (sequence number ricevuto + 1)



Il terzo pacchetto, che completa la sequenza di handshake, presenta come previsto il flag ACK attivato.

Dettagli del terzo pacchetto:

- ***Flag impostato***: ACK = 1

- ***Sequence number***: 1

- ***Acknowledgment number***: 1

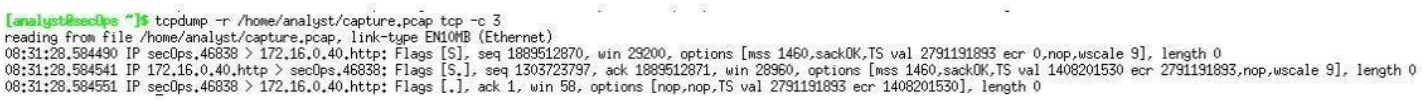
La connessione TCP è ora stabilita e pronta per il trasferimento dati.

È particolarmente significativo osservare i valori di ***sequence number*** e ***acknowledgment number***: nel primo pacchetto il sequence number risulta *pari a 0*, il server risponde con sequence number *uguale a 0* e acknowledgment number *pari a 1* (*sequence number ricevuto + 1*). Infine, il client conclude l'handshake con sequence number 1 e acknowledge number 1 .

Analisi tramite tcpdump

Per completare l'esame, posso anche analizzare il traffico catturato direttamente tramite **tcpdump.**

Eseguo il comando **tcpdump -r /home/analyst/capture.pcap tcp -c 3** per visualizzare la cattura precedente in formato testuale.



L'opzione -r permette di leggere pacchetti di file salvati in precedenza.

Conclusioni

L'attività ha permesso di osservare concretamente il meccanismo del TCP three-way handshake attraverso due diverse metodologie di analisi. L'utilizzo combinato di Wireshark e tcpdump fornisce una comprensione completa sia dal punto di vista grafico che testuale del processo di stabilimento delle connessioni TCP, evidenziando l'importanza della corretta sequenza SYN → SYN-ACK → ACK per l'instaurazione di connessioni affidabili.

Domande di Riflessione

Tre filtri utili per un amministratore di rete in Wireshark:

1. **ip.addr == x.x.x.x** - Filtra tutto il traffico da/verso un indirizzo IP specifico, utile per monitorare l'attività di un host particolare o per il troubleshooting di problemi di connettività.
2. **tcp.flags.reset == 1** - Identifica le connessioni TCP che sono state resettate improvvisamente, indicativo di problemi di rete, timeout o potenziali attacchi.
3. **http.response.code >= 400** - Filtra le risposte HTTP con codici di errore (4xx, 5xx), permettendo di identificare rapidamente problemi applicativi o di configurazione del server web.

Altri utilizzi di Wireshark in una rete di produzione:

- ***Rilevamento di anomalie di sicurezza***: Identificazione di pattern di traffico sospetti, tentativi di intrusione, scansioni di porte o attacchi DDoS attraverso l'analisi dei pacchetti.
- ***Ottimizzazione delle performance***: Analisi dei tempi di risposta, identificazione di colli di bottiglia nella rete, monitoraggio della latenza e del throughput per ottimizzare le prestazioni.
- ***Troubleshooting applicativo***: Diagnosi di problemi di comunicazione tra applicazioni, analisi di protocolli specifici, identificazione di errori nella logica di rete delle applicazioni.
- ***Compliance e auditing***: Monitoraggio del traffico per verificare il rispetto delle policy aziendali, documentazione delle comunicazioni di rete per audit di sicurezza.
- ***Capacity planning***: Raccolta di statistiche sul traffico di rete per pianificare l'espansione dell'infrastruttura e dimensionare correttamente le risorse di rete.