

Data: 10 Giugno 2025

Oggetto: Osservazione e Analisi dell'Handshake TCP a Tre Vie e Applicazioni di Wireshark in Ambiente di Produzione

Autore: Stefano Gugliotta

1. Introduzione

Il presente report documenta l'esecuzione di un laboratorio pratico finalizzato all'osservazione e all'analisi del processo di handshake a tre vie del protocollo Transmission Control Protocol (TCP). L'attività è stata condotta utilizzando una macchina virtuale CyberOps con Mininet per simulare una topologia di rete, e gli strumenti tcpdump e Wireshark per la cattura e l'analisi del traffico. Vengono inoltre esaminate le potenziali applicazioni di Wireshark in un contesto di rete di produzione.

2. Procedura di Laboratorio

2.1 Parte 1: Preparazione degli Host per la Cattura del Traffico

2.1.1 Avvio della VM CyberOps: La macchina virtuale CyberOps è stata avviata con successo. Si è riscontrato un errore iniziale di avvio che è stato risolto modificando le impostazioni della scheda di rete passando ad una scheda di rete "Intel Wifi 6E" che ha permesso l'accessione della macchina. L'accesso è stato effettuato con le credenziali fornite: nome utente analyst e password cyberops.

2.1.2 Avvio di Mininet e Configurazione della Topologia: Mininet è stato avviato eseguendo lo script `sudo`

`lab.support.files/scripts/cyberops_topo.py`. Questa operazione ha creato la topologia di rete predefinita CyberOPS. La topologia include gli host H1, H2, H3, H4 e il router R1, interconnessi tramite uno switch virtuale S1. È stata verificata la tabella di routing su R1, che ha mostrato correttamente le rotte per le reti 10.0.0.0/8 (via R1-eth1) e

```
CyberOPS Topology:
graph TD
    R1[R1] --- S1[S1]
    S1 --- H1[H1]
    S1 --- H2[H2]
    S1 --- H3[H3]
    S1 --- H4[H4]
    R1 --- H4

*** 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
0.0.0.0        0.0.0.0         0.0.0.0         U        0      0        0 R1-eth1
172.16.0.0     0.0.0.0         0.0.0.0         U        0      0        0 R1-eth2

*** Starting CLI:
mininet> xterm H1
mininet> xterm H4
mininet> quit
*** Stopping 0 controllers

*** Stopping 2 terms
*** Stopping 5 links
...
*** Stopping 1 switches
s1
*** Stopping 5 hosts
H1 H2 H3 H4 R1
*** Done
```

```

Node: H1
[root@secOps analyst]# su analyst
[analyst@secOps ~]$ firefox &
[1] 1045
Sempre su H1 avvio il tcpdump
[analyst@secOps ~]$ sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analyst/capture.pcap
[sudo] password for analyst:
tcpdump: listening on H1-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
50 packets captured
52 packets received by filter
0 packets dropped by kernel
```

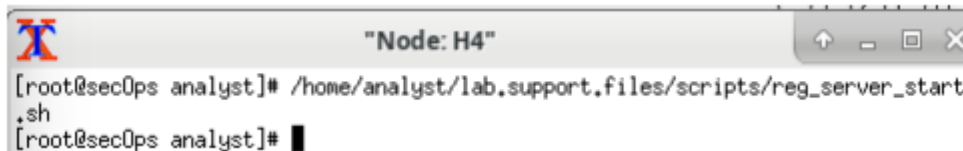
172.16.0.0/16 (via R1-eth2).

Successivamente, sono stati aperti terminali xterm dedicati per gli host H1 e H4, come richiesto dalla traccia.

2.1.3 Avvio del Server Web su H4 e del Browser su H1: Un server web è stato avviato sull'host H4 utilizzando

lo script `/home/analyst/lab.support.files/scripts/reg_server_start.sh`. Questo ha configurato

un server Nginx, la cui interfaccia di benvenuto è stata successivamente



```
[root@sec0ps analyst]# /home/analyst/lab.support.files/scripts/reg_server_start
.sh
[root@sec0ps analyst]#
```

osservata. Sull'host H1, per ragioni di sicurezza e per consentire l'esecuzione del browser Firefox, l'utente è stato commutato da root ad analyst tramite il comando su analyst. Successivamente, Firefox è stato avviato in background (firefox &).

2.1.4 Cattura del Traffico con tcpdump: Contemporaneamente all'avvio del browser, è stata iniziata una sessione di cattura del traffico sull'interfaccia H1-eth0 dell'host H1 utilizzando il comando `sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analyst/capture.pcap`. Questa configurazione ha permesso di acquisire un massimo di 50 pacchetti in modalità verbosa, salvando l'output nel file capture.pcap. Subito dopo l'avvio di tcpdump, è stata effettuata una navigazione rapida a 172.16.0.40 nel browser Firefox su H1, indirizzo corrispondente al server web su H4.

2.2 Parte 2: Analisi dei Pacchetti con Wireshark

2.2.1 Caricamento del File di Cattura e Applicazione del Filtro: Wireshark è stato avviato su H1 (`wireshark-gtk &`). È stato confermato l'avviso relativo all'esecuzione come superutente. Il file di cattura capture.pcap è stato aperto tramite File > Open. Per concentrare l'analisi sui pacchetti rilevanti per l'handshake TCP, è stato applicato il filtro di visualizzazione tcp nella barra dei filtri di Wireshark.

2.2.2 Analisi Dettagliata dell'Handshake a 3 Vie TCP:

- **Frame 1 (Richiesta SYN) - Identificato come Frame 7 nella cattura (vedi "dettagli.png"):**
 - **Numero di porta TCP di origine:** 47832.
 - **Classificazione della porta di origine:** Si tratta di una porta effimera (o dinamica), utilizzata dal sistema operativo client (H1) per stabilire una connessione. La sua selezione è temporanea e non è associata a servizi specifici.
 - **Numero di porta TCP di destinazione:** 80.
- Transmission Control Protocol, Src Port: 47832, Dst Port: 80,

Source Port: 47832

Destination Port: 80

[Stream index: 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)

Window size value: 29200

[Calculated window size: 29200]

Checksum: 0xb671 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

- **Classificazione della porta di destinazione:** Questa è una porta ben nota (well-known port), standardizzata per il protocollo HTTP. Indica che il client H1 sta cercando di connettersi a un servizio web sul server H4.
- **Flag impostato:** Il flag SYN (Synchronize) è impostato (valore 1). Questo indica una richiesta di apertura di una nuova connessione TCP.
- **Valore del numero di sequenza relativo:** Nello screenshot "dettagli.png", il "Relative sequence number" è visualizzato come 0. Tuttavia, disabilitando l'opzione "Relative sequence numbers" (Analyze -> Expert Information -> TCP Analysis -> Relative sequence numbers) si osserva il numero di sequenza assoluto 2432755549, coerentemente con l'output di tcpdump. Questo valore rappresenta il numero di sequenza iniziale (ISN) scelto dal client.
- **Frame 2 (Risposta SYN-ACK) - Identificato come Frame 8 nella cattura :**
 - **Valori delle porte di origine e destinazione:**

| | | | | | |
|----|-----------|-------------|-------------|-----|---------------|
| 48 | 34.807432 | 10.0.0.11 | 172.16.0.40 | TCP | 74 47832 → 80 |
| 49 | 34.807469 | 172.16.0.40 | 10.0.0.11 | TCP | 74 80 → 47832 |
| 50 | 34.807476 | 10.0.0.11 | 172.16.0.40 | TCP | 66 47832 → 80 |

Porta di origine: 80 (il server risponde dal suo servizio web); Porta di destinazione: 47832 (la porta effimera del client).
 - **Flag impostati:** I flag SYN e ACK sono entrambi impostati. Il flag SYN indica che il server sta stabilendo la propria sequenza numerica, mentre il flag ACK conferma la ricezione del SYN del client.
 - **Valori dei numeri relativi di sequenza e acknowledgment:** Il "Relative Sequence number" è 0 (per il server) e il "Relative Acknowledgment number" è 1. Il numero di acknowledgment (assoluto 2432755550) è il numero di sequenza del client incrementato di 1, confermando la ricezione del suo SYN. Il numero di sequenza (assoluto 1766419191) è l'ISN del server.
- **Frame 3 (ACK Finale) - Identificato come Frame 9 nella cattura:**
 - **Flag impostato:** Il flag ACK è impostato. Questo pacchetto conclude l'handshake, confermando al server la ricezione del suo SYN-ACK.
 - **Numeri relativi di sequenza e acknowledgment:** Entrambi i numeri relativi di sequenza e acknowledgment sono impostati a 1, indicando che la connessione TCP è ora pienamente stabilita e la trasmissione dei dati applicativi può iniziare.

2.3 Parte 3: Visualizzazione dei Pacchetti con tcpdump

2.3.1 Consultazione delle Pagine man di tcpdump: È stato aperto un nuovo terminale e consultata la pagina del manuale per tcpdump (man tcpdump). La ricerca dell'opzione -r ha confermato che questa opzione è utilizzata per leggere pacchetti da un file (file) precedentemente catturato.

2.3.2 Visualizzazione dei Pacchetti Catturati con tcpdump: Il comando `tcpdump -r /home/analyst/capture.pcap tcp -c 3` è stato eseguito per visualizzare i primi tre pacchetti TCP dal file `capture.pcap`. L'output ha mostrato i dettagli dei pacchetti, inclusi gli indirizzi IP di origine e destinazione, le porte, i flag TCP ([S] per SYN, [S.] per SYN-ACK, [.] per ACK), e i numeri di sequenza (seq) e acknowledgement (ack). L'output ha mostrato la coerenza tra i flag e i numeri di sequenza assoluti con quanto osservato in Wireshark dopo aver disabilitato i numeri di sequenza relativi.

2.3.3 Terminazione e Pulizia di Mininet: Al termine dell'analisi, Mininet è stato spento correttamente digitando `quit` nella CLI di Mininet. Successivamente, per assicurare la pulizia di tutti i processi e le configurazioni residue, è stato eseguito il comando `sudo mn -c`, inserendo la password `cyberops` quando richiesto.

3. Domande di Riflessione

3.1. Tre filtri utili in Wireshark per un amministratore di rete: Wireshark offre una vasta gamma di filtri di visualizzazione che sono inestimabili per gli amministratori di rete. Ecco tre esempi significativi:

1. `ip.addr == [indirizzo IP specifico]`:

- **Funzione:** Questo filtro visualizza tutti i pacchetti il cui indirizzo IP sorgente o destinazione corrisponde all'indirizzo IP specificato.
- **Utilità:** Permette di isolare il traffico relativo a un singolo host, facilitando la diagnosi di problemi di connettività specifici, l'analisi del comportamento di un dispositivo o il monitoraggio di attività sospette da/verso quella macchina.

2. `tcp.port == 80` (o qualsiasi altra porta specifica, es. `tcp.port == 443` per HTTPS, `udp.port == 53` per DNS):

- **Funzione:** Filtra il traffico TCP (o UDP) che coinvolge la porta specificata, sia come porta sorgente che come porta di destinazione.
- **Utilità:** Indispensabile per analizzare il traffico di un servizio specifico (es. HTTP, HTTPS, DNS, SSH, FTP). Aiuta a diagnosticare problemi di applicazione, verificare la disponibilità del servizio o monitorare l'utilizzo di una particolare risorsa di rete.

3. `http.request`:

- **Funzione:** Questo filtro seleziona specificamente tutti i pacchetti che contengono una richiesta HTTP (es. GET, POST, PUT, DELETE).
- **Utilità:** Estremamente utile per gli sviluppatori web e gli amministratori di server per analizzare il comportamento delle applicazioni web, ispezionare le richieste HTTP inviate dai client, verificare gli URL, gli header HTTP, i metodi di richiesta e i parametri inviati, facilitando il debug e l'ottimizzazione.

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

Oltre all'osservazione dell'handshake TCP e al debug di base, Wireshark è uno strumento estremamente versatile e potente in un ambiente di rete di produzione, con applicazioni che vanno ben oltre la semplice cattura di pacchetti. Alcuni utilizzi cruciali includono:

1. Risoluzione Avanzata dei Problemi di Rete:

- **Diagnosi di problemi di performance:** Identificare la causa di rallentamenti della rete, come latenza elevata, ritrasmissioni eccessive, frammentazione di pacchetti o problemi di congestione. Wireshark può mostrare i tempi di risposta tra client e server, l'efficienza della finestra TCP e la gestione del controllo di flusso.
- **Individuazione di problemi di connettività:** Debuggare connessioni interrotte, timeout inaspettati o problemi di risoluzione DNS, analizzando i messaggi di errore a livello di protocollo.
- **Troubleshooting di applicazioni distribuite:** Analizzare le interazioni tra diversi componenti di un'applicazione (database, server applicativi, bilanciatori di carico) per identificare dove si verifica un ritardo o un errore.

2. Analisi e Forensica della Sicurezza:

- **Rilevamento di attività sospette:** Identificare scansioni di porte (port scans), tentativi di brute-force, attacchi Denial of Service (DoS) o la presenza di malware che genera traffico anomalo.
- **Indagine su incidenti di sicurezza:** Analizzare il traffico catturato durante o dopo un incidente per comprendere la catena di attacco, identificare le vulnerabilità sfruttate e raccogliere prove forensi.
- **Verifica della configurazione di sicurezza:** Assicurarsi che le politiche di sicurezza (es. firewall, IPS) stiano bloccando il traffico indesiderato o che le comunicazioni sensibili siano correttamente crittografate (anche se Wireshark non può decifrare il traffico HTTPS senza le chiavi private del server).

3. Verifica della Conformità e Audit:

- **Monitoraggio della conformità normativa:** Assicurarsi che le politiche di sicurezza e di privacy dei dati siano rispettate, verificando che determinati tipi di traffico (es. dati sensibili) non vengano trasmessi in chiaro.
- **Validazione della configurazione di rete:** Convalidare che i dispositivi di rete (router, switch, firewall) stiano inoltrando e gestendo il traffico come previsto dalle policy aziendali.

4. Ottimizzazione delle Prestazioni della Rete:

- **Analisi del throughput e della banda:** Valutare l'utilizzo effettivo della banda e identificare le applicazioni o i dispositivi che consumano più risorse.
- **Ottimizzazione del protocollo:** Analizzare le impostazioni TCP (finestre di ricezione, algoritmi di controllo della congestione) per ottimizzare la trasmissione dei dati e ridurre la latenza.

5. Formazione e Sviluppo:

- **Comprensione approfondita dei protocolli:** Fornire uno strumento visivo per studenti e professionisti per capire come i protocolli funzionano a basso livello.
- **Debug di nuove applicazioni:** Gli sviluppatori possono utilizzare Wireshark per verificare il traffico generato dalle loro applicazioni, assicurandosi che comunichino correttamente e in modo efficiente.

In sintesi, Wireshark non è solo uno strumento diagnostico, ma una suite completa per la visibilità della rete che consente agli amministratori di mantenere reti robuste, sicure ed efficienti.