

Analisi dei Pacchetti TCP con Wireshark

Obiettivo del Laboratorio Il laboratorio mira a catturare e analizzare pacchetti di rete generati durante una sessione HTTP tra un browser web su un host e un server web remoto. L'attività prevede l'uso di strumenti come Mininet per simulare la rete, tcpdump per catturare i pacchetti e Wireshark per analizzarli.

Cattura dei Pacchetti

Preparazione dell'Ambiente

- È stata avviata la macchina virtuale (VM) CyberOps, effettuando l'accesso con le credenziali:
 - Nome utente: **analyst**
 - Password: **cyberops**
- Successivamente, è stato avviato Mininet tramite il comando:
- `sudo lab.support.files/scripts/cyberops_topo.py`

Configurazione degli Host

- Nel terminale Mininet, sono stati avviati gli host H1 e H4 utilizzando:
- `xterm H1`
- `xterm H4`
- Sul nodo H4, è stato avviato il server web tramite il comando:
- `/home/analyst/lab.support.files/scripts/reg_server_start.sh`

Accesso e Avvio del Browser Web

- Sul nodo H1, è stato effettuato lo switch all'account utente analyst con:
- `su analyst`
- Successivamente, è stato avviato Firefox tramite il comando:
- `firefox &`

Cattura dei Pacchetti

- Sul nodo H1, è stato avviato il comando tcpdump per catturare 50 pacchetti, salvando l'output in un file denominato capture.pcap:
- `sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analista/capture.pcap`
- Mentre tcpdump era attivo, è stato effettuato l'accesso all'indirizzo IP del server web (172.16.0.40) utilizzando il browser Firefox.

Analisi dei Pacchetti con Wireshark

Avvio di Wireshark

- Sul nodo H1, è stato avviato Wireshark tramite il comando:
- `wireshark &`
- All'avvio, è stato aperto il file di cattura (capture.pcap) tramite il percorso: **File > Apri**.

Applicazione del Filtro TCP

- In Wireshark, è stato applicato un filtro per visualizzare esclusivamente il traffico TCP. Questo filtro ha permesso di identificare i pacchetti relativi all'handshake a tre vie e alla sessione HTTP stabilita tra H1 e il server web H4.
- Sono stati osservati i seguenti elementi chiave nei primi 3 frame:
 - **SYN:** Richiesta di sincronizzazione inviata dall'host H1 al server H4.
 - **SYN-ACK:** Risposta del server H4 per confermare la richiesta.
 - **ACK:** Conferma finale dell'host H1 per completare l'handshake.

Dettagli dell'Analisi dei Pacchetti

- Fai clic sulla freccia a sinistra di Flags. Un valore di 1 significa che il flag è impostato. Individua il flag impostato in questo pacchetto.
 - **Numero della porta sorgente TCP:** 58716 (esempio). Classificazione: dinamico o privato.
 - **Numero della porta di destinazione TCP:** 80. Classificazione: noto, registrato (protocollo HTTP o web).
 - **Flag impostato:** Bandiera SYN.
 - **Numero di sequenza relativo:** 0.
- Seleziona il pacchetto successivo nell'handshake a tre vie (esempio: frame 2). Questo è il server web che risponde alla richiesta iniziale di avviare una sessione.
 - **Porta di origine:** 80.
 - **Porta di destinazione:** 58716.
 - **Flag impostati:** Flag di riconoscimento (ACK) e Syn (SYN).
 - **Numeri di sequenza e di conferma relativi:** Numero di sequenza relativo: 0. Numero di conferma relativo: 1.
- Infine, seleziona il terzo pacchetto nell'handshake a tre vie.
 - **Flag impostato:** Flag di conferma (ACK).
 - **Numeri di sequenza e di conferma relativi:** Entrambi impostati su 1 come punto di partenza.
- La connessione TCP è stata stabilita e la comunicazione tra il computer sorgente e il server Web può iniziare.

Visualizza i Pacchetti Utilizzando tcpdump

È anche possibile visualizzare il file pcap e filtrare le informazioni desiderate.

Visualizzare il Manuale di tcpdump

- Apri una nuova finestra del terminale e digita:
- `man tcpdump`
- Nota: potrebbe essere necessario premere INVIO per visualizzare il prompt.
- Utilizzando le pagine del manuale disponibili con il sistema operativo Linux, è possibile leggere o cercare al loro interno le opzioni per selezionare le informazioni desiderate dal file pcap. Per cercare nelle pagine man, puoi usare / (ricerca in avanti) o ? (ricerca all'indietro) per trovare termini specifici, e n per andare avanti alla corrispondenza successiva e q per uscire.
 - **A cosa serve l'opzione -r?** L'opzione -r consente di leggere il pacchetto dal file salvato utilizzando l'opzione -w con tcpdump o altri strumenti che scrivono file pcap o pcap-ng, come Wireshark.

Visualizzazione dei Pacchetti Acquisiti

- Nel terminale, apri il file di acquisizione per visualizzare i primi 3 pacchetti TCP acquisiti:
- `tcpdump -r /home/analista/capture.pcap -c 3`
- Per visualizzare l'handshake a 3 vie, potrebbe essere necessario aumentare il numero di righe dopo l'opzione -c.

Pulizia e Chiusura di Mininet


- Vai al terminale utilizzato per avviare Mininet. Termina Mininet immettendo:
- `quit`
- nella finestra principale del terminale CyberOps VM.
- Dopo aver chiuso Mininet, esegui:
- `sudo mn -c`
- per ripulire i processi avviati da Mininet. Inserisci la password cyberops quando richiesto.

Conclusioni Il laboratorio ha permesso di:

- Comprendere il processo di handshake a tre vie utilizzato da TCP per stabilire una connessione affidabile.
- Utilizzare tcpdump per catturare pacchetti di rete e salvare i dati in formato pcap.
- Analizzare il traffico di rete catturato utilizzando Wireshark, applicando filtri per isolare pacchetti specifici.
- Visualizzare e analizzare i file pcap direttamente dal terminale con tcpdump, utilizzando le opzioni disponibili nel manuale del comando.

Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
*** 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> 

"Node: H1"
[root@secOps analyst]# su analyst
[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
50 packets received by filter
0 packets dropped by kernel
[analyst@secOps ~]\$ wireshark &
[1] 1534
[analyst@secOps ~]\$ bash: wireshark: command not found

[1]+ Exit 127 wireshark
[analyst@secOps ~]\$ firefox &
[1] 1592
[analyst@secOps ~]\$ sudo tcpdump -i H1-eth0 -v -c 50 -w /home/analyst/capture.pcap
pcap
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
[analyst@secOps ~]\$
sh
bash: /home/analyst/lab.support.file/scripts/reg_server_start.sh: No such file or directory
[root@secOps analyst]# /home/analyst/lab.support.files/scripts/reg_server_start.sh
[root@secOps analyst]# 2024/12/11 07:04:21 [error] 1003#1003: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 10.0.0.11, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "172.16.0.40"
2024/12/11 08:03:07 [error] 1003#1003: *3 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 10.0.0.11, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "172.16.0.40"
0

Welcome to nginx! - Mozilla Firefox
Welcome to nginx! x +
172.16.0.40

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

capture.pcap [Wireshark 2.5.1]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: tcp Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
37	10.656837	10.0.0.11	172.16.0.40	TCP	74	43118 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1558876665 TSecr=0 WS=512
38	10.656858	172.16.0.40	10.0.0.11	TCP	74	80 → 43118 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2640495309 TSecr=1558876665
39	10.656863	10.0.0.11	172.16.0.40	TCP	66	43118 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=1558876665 TSecr=2640495309

```

analyst@secOps ~]$ sudo rm -c
[sudo] password for analyst:
** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller udpbwtest mnexec ixs 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller udpbwtest mnexec ixs 2> /dev/null
kill -9 -f "sudo mnexec"
** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
** Removing old X11 tunnels
** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([-.[:alnum:]]+-eth[[:digit:]]+)'
ip link show
** Killing stale mininet node processes
kill -9 -f mininet:
** Shutting down stale tunnels
kill -9 -f Tunnel=Ethernet
kill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
** Cleanup complete.
analyst@secOps ~]$

```