

Build Week 3 – ESERCIZIO 2

Progetto S12 - L1

Titolo: Server Linux

Executive Summary

Nel presente laboratorio è stato analizzato il funzionamento dei servizi di rete su un server Linux all'interno della VM CyberOps Workstation.

Sono stati utilizzati comandi di sistema per osservare i processi in esecuzione, identificare le porte in ascolto e correlare i servizi attivi ai rispettivi PID.

Attraverso l'impiego di `ps`, `netstat` e `telnet`, è stato verificato il comportamento di un web server (nginx) e del servizio SSH, comprendendo la relazione tra processi, porte TCP e protocolli di Livello 4. L'attività ha consentito di consolidare competenze fondamentali di troubleshooting e analisi dei servizi di rete.

Introduzione

L'obiettivo dell'esercizio è stato comprendere come identificare e analizzare i servizi attivi su un server Linux.

In particolare, è stato richiesto di:

- Osservare i processi in esecuzione
- Identificare i servizi di rete in ascolto
- Comprendere la relazione tra porte TCP, PID e processi
- Testare manualmente i servizi tramite Telnet

L'attività riproduce una tipica fase di analisi tecnica e verifica dei servizi in ambito cybersecurity.

Parte 1 — Server

Passo 1: Accesso alla riga di comando

È stata avviata la VM CyberOps Workstation.

È stato effettuato l'accesso con:

- Utente: **analyst**
- Password: **cyberops**

Successivamente è stato aperto il Terminale dal Dock.

Passo 2: Visualizzazione dei servizi in esecuzione

Nel terminale è stato eseguito:

sudo ps -elf

Quando richiesto, è stata inserita la password.

DOMANDA

Perché è stato necessario eseguire ps come root (premettendo il comando con sudo)?

RISPOSTA

È stato necessario utilizzare privilegi elevati perché molti processi di sistema appartengono all'utente root. Senza sudo la visualizzazione sarebbe stata limitata ai soli processi dell'utente corrente.

```
[analyst@secOps ~]$ sudo ps -elf
[sudo] password for analyst:
F S UID          PID     PPID    C  PRI   NI     ADDR  SZ  WCHAN    STIME TTY          TIME CMD
4 S root           1         0    0   80    0    -   5449 do_epo  08:14 ?        00:00:00 /sbin/init
1 S root           2         0    0   80    0    -         0 kthrea  08:14 ?        00:00:00 [kthreadd]
1 S root           3         2    0   80    0    -         0 kthrea  08:14 ?        00:00:00 [pool_workqueue_release]
1 I root           4         2    0   60  -20   -         0 rescue  08:14 ?        00:00:00 [kworker/R-rcu_gp]
1 I root           5         2    0   60  -20   -         0 rescue  08:14 ?        00:00:00 [kworker/R-sync_wq]
1 I root           6         2    0   60  -20   -         0 rescue  08:14 ?        00:00:00 [kworker/R-kvfree_rcu_reclaim]
1 I root           7         2    0   60  -20   -         0 rescue  08:14 ?        00:00:00 [kworker/R-slub_flushwq]
1 I root           8         2    0   60  -20   -         0 rescue  08:14 ?        00:00:00 [kworker/R-netns]
1 I root          12         2    0   80    0    -         0 worker  08:14 ?        00:00:00 [kworker/u8:0-events_unbound]
1 I root          13         2    0   80    0    -         0 worker  08:14 ?        00:00:00 [kworker/u8:1-ipv6_addrconf]
1 I root          14         2    0   60  -20   -         0 rescue  08:14 ?        00:00:00 [kworker/R-mm_percpu_wq]
1 S root          15         2    0   80    0    -         0 smpboo  08:14 ?        00:00:00 [ksoftirqd/0]
1 I root          16         2    0   58    -    -         0 rcu_gp   08:14 ?        00:00:01 [rcu_preempt]
1 S root          17         2    0   58    -    -         0 rcu_bo   08:14 ?        00:00:00 [rcub/0]
1 S root          18         2    0   80    0    -         0 kthrea   08:14 ?        00:00:00 [rcu_exp_par_gp_kthread_worker/0]
1 S root          19         2    0   80    0    -         0 kthrea   08:14 ?        00:00:00 [rcu_exp_gp_kthread_worker]
1 S root          20         2    0  -40    -    -         0 smpboo  08:14 ?        00:00:00 [migration/0]
1 S root          21         2    0    9    -    -         0 smpboo  08:14 ?        00:00:00 [idle_inject/0]
1 S root          22         2    0   80    0    -         0 smpboo  08:14 ?        00:00:00 [cpuhp/0]
1 S root          23         2    0   80    0    -         0 smpboo  08:14 ?        00:00:00 [cpuhp/1]
1 S root          24         2    0    9    -    -         0 smpboo  08:14 ?        00:00:00 [idle_inject/1]
```

- Output di `sudo ps -elf`

Successivamente è stato avviato nginx:

```
sudo /usr/sbin/nginx
```

È stata quindi visualizzata la gerarchia dei processi:

```
sudo ps -ejH
```

DOMANDA

Come viene rappresentata la gerarchia dei processi da ps?

RISPOSTA

La gerarchia viene rappresentata come una struttura ad albero: i processi figli risultano indentati sotto il processo padre, rendendo visibile la relazione parent/child.

```
[analyst@secOps ~]$ sudo /usr/sbin/nginx
[analyst@secOps ~]$ sudo ps -ejH
```

PID	PGID	SID	TTY	TIME	CMD
2	0	0	?	00:00:00	kthreadd
3	0	0	?	00:00:00	pool_workqueue_release
4	0	0	?	00:00:00	kworker/R-rcu_gp
5	0	0	?	00:00:00	kworker/R-sync_wq
6	0	0	?	00:00:00	kworker/R-kvfree_rcu_reclaim
7	0	0	?	00:00:00	kworker/R-slub_flushwq
8	0	0	?	00:00:00	kworker/R-netns
12	0	0	?	00:00:00	kworker/u8:0-events_unbound
13	0	0	?	00:00:00	kworker/u8:1-ipv6_addrconf
14	0	0	?	00:00:00	kworker/R-mm_percpu_wq
15	0	0	?	00:00:00	ksoftirqd/0
16	0	0	?	00:00:01	rcu_preempt
17	0	0	?	00:00:00	rcub/0
18	0	0	?	00:00:00	rcu_exp_par_gp_kthread_worker/0
19	0	0	?	00:00:00	rcu_exp_gp_kthread_worker
20	0	0	?	00:00:00	migration/0
21	0	0	?	00:00:00	idle_inject/0
22	0	0	?	00:00:00	cpuhp/0
23	0	0	?	00:00:00	cpuhp/1
24	0	0	?	00:00:00	idle_inject/1
25	0	0	?	00:00:01	migration/1
26	0	0	?	00:00:00	ksoftirqd/1
28	0	0	?	00:00:00	kworker/1:0H-events_highpri
31	0	0	?	00:00:00	kdevtmpfs

- Output con struttura ad albero (master → worker)

Identificazione dei server di rete con netstat

Sono stati eseguiti i seguenti comandi:

netstat

e successivamente:

sudo netstat -tunap

DOMANDA

Qual è il significato delle opzioni -t, -u, -n, -a e -p in netstat?

RISPOSTA

- **-t** → mostra le connessioni TCP
- **-u** → mostra le connessioni UDP
- **-n** → visualizza indirizzi e porte in formato numerico

- **-a** → mostra tutte le connessioni e le porte in ascolto
 - **-p** → mostra il PID e il nome del processo associato
-

DOMANDA

L'ordine delle opzioni è importante per netstat?

RISPOSTA

No, l'ordine delle opzioni non è determinante. È rilevante la presenza delle opzioni, non la loro sequenza.

```
[analyst@secOps ~]$ sudo netstat -tunap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      390/sshd: /usr/bin/
tcp        0      0 0.0.0.0:21             0.0.0.0:*               LISTEN      507/vsftpd
tcp        0      0 0.0.0.0:80             0.0.0.0:*               LISTEN      2246/nginx: master
tcp        0      0 0.0.0.0:6633           0.0.0.0:*               LISTEN      375/python3.9
tcp6       0      0 :::22                  :::*                     LISTEN      390/sshd: /usr/bin/
udp        0      0 10.0.2.15:68          0.0.0.0:*               292/systemd-network
```

- Output completo di `sudo netstat -tunap`
-

DOMANDA

Basandosi sull'output di netstat, qual è il protocollo di Livello 4, lo stato della connessione e il PID del processo sulla porta 80?

RISPOSTA

- Protocollo L4: TCP
 - Stato: LISTEN
 - PID: quello associato a nginx (può variare in base alla VM)
-

DOMANDA

Che tipo di servizio è in esecuzione sulla porta 80 TCP?

RISPOSTA

La porta 80/TCP è comunemente associata al protocollo HTTP, quindi è stato identificato un servizio web.

Incrocio tra netstat e ps

È stato eseguito il comando:

```
sudo ps -elf | grep 2246
```

DOMANDE

Come si conclude che il processo è nginx?

È stato possibile verificarlo osservando la colonna CMD, nella quale compare il riferimento a nginx, con PID coincidente a quello individuato tramite netstat.

Cos'è nginx?

È un web server e reverse proxy che gestisce richieste HTTP e HTTPS.

Perché sono presenti processi master e worker?

Il processo master avvia e gestisce uno o più processi worker che elaborano le richieste. Si tratta di un comportamento standard nei web server moderni.

Perché compare grep 395?

Perché anche il comando grep è un processo attivo e, contenendo il numero cercato nella riga di comando, viene incluso nei risultati.

```
[analyst@secOps ~]$ sudo ps -elf | grep 2246
[sudo] password for analyst:
1 S root      2246      1  0  80   0 -  3723 sigsus 14:27 ?        00:00:00 nginx: master proces
s /usr/sbin/nginx
5 S http      2247    2246  0  80   0 -  3827 do_epo 14:27 ?        00:00:00 nginx: worker proces
s
0 S analyst   2302    2207  0  80   0 -  1615 anon_p 14:38 pts/0    00:00:00 grep 2246
```

- Output di `ps -elf | grep 2246`
-

Parte 2 — Utilizzo di Telnet per testare i servizi TCP

Test della porta 80 (nginx)

È stata effettuata la connessione:

telnet 127.0.0.1 80

Sono stati inviati caratteri casuali seguiti da INVIO.

DOMANDA

Perché l'errore è stato inviato come pagina web?

RISPOSTA

Perché nginx comunica tramite protocollo HTTP. Anche un errore viene restituito sotto forma di risposta HTTP (header + contenuto HTML), come se fosse una normale pagina web.

```
[analyst@secOps ~]$ telnet 127.0.0.1 80
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
GET / HTTP/1.1
Host: localhost

HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Mon, 23 Feb 2026 19:45:43 GMT
Content-Type: text/html
Content-Length: 615
Last-Modified: Wed, 30 Apr 2025 23:47:36 GMT
Connection: keep-alive
ETag: "6812b698-267"
Accept-Ranges: bytes
```

- Connessione Telnet porta 80 con risposta HTTP
-

Test della porta 22 (SSH)

È stata effettuata una connessione TCP alla porta 22 utilizzando Telnet:

telnet 127.0.0.1 22

La connessione è stata stabilita correttamente, come indicato dal messaggio di conferma.

Subito dopo l'apertura della sessione è stato visualizzato il banner del servizio:

SSH-2.0-OpenSSH_10.0

```
[analyst@secOps ~]$ telnet 127.0.0.1 22
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
SSH-2.0-OpenSSH_10.0
```

La presenza del banner conferma che:

- La porta 22 è in ascolto
- Il protocollo attivo è SSH
- Il servizio in esecuzione è OpenSSH

La visualizzazione del banner rappresenta un esempio di **banner grabbing**, tecnica utilizzata per identificare i servizi attivi su una determinata porta TCP.

DOMANDA

Cosa succede se ci si connette alla porta 68?

```
telnet 127.0.0.1 68
```

RISPOSTA

La porta 68 è tipicamente utilizzata dal client DHCP su protocollo UDP. Poiché Telnet opera su TCP, la connessione non viene stabilita e si verifica un errore o un timeout.

```
[analyst@secOps ~]$ telnet 127.0.0.1 68
Trying 127.0.0.1...
telnet: Unable to connect to remote host: Connection refused
```

- Output Errore o timeout sulla porta 68

Domande di Riflessione

Quali sono i vantaggi dell'uso di netstat?

Consente di identificare porte aperte, connessioni attive, protocolli utilizzati e PID associati ai processi. È uno strumento utile per attività di troubleshooting e per l'individuazione di servizi sospetti.

Quali sono i vantaggi dell'uso di Telnet? È sicuro?

Telnet consente di testare rapidamente la risposta di un servizio TCP e di effettuare verifiche manuali (banner grabbing).

Non è sicuro per accessi remoti perché trasmette dati in chiaro, senza cifratura.

Conclusioni

Attraverso questo laboratorio **è stata approfondita l'analisi dei processi e dei servizi di rete su un server Linux.**

L'uso combinato di `ps`, `netstat` e `telnet` ha consentito di **verificare il funzionamento di un web server e del servizio SSH, correlando porte, PID e protocolli di Livello 4.**

Le competenze acquisite risultano fondamentali per attività di amministrazione di sistema, troubleshooting e analisi tecnica in ambito cybersecurity.