Progetto Cyber Security & Ethical Hacking

Team: Secure Sentinels

Modulo: Cybersecurity & Ethical Hacking (Epicode)

Data: 23 Febbraio 2026

Esercizio 2: Server Linux

Obiettivi

In questo laboratorio, userai la riga di comando Linux per identificare i server in esecuzione su un dato computer.
● **Parte 1**: Server
● **Parte 2**: Usare Telnet per Testare i Servizi TCP

1. Questo laboratorio si concentra su server e client basati su rete IP e vede in azione la Macchina Virtuale CyberOps Workstation tramite Terminale.

Parte 1: Server

Molti programmi, chiamati anche processi, sono eseguiti in background

Istruzioni

a.Usa il comando **ps** per visualizzare tutti i programmi in esecuzione in background

sudo ps -elf

```
[analyst@secOps ~]$ ps
    PID TTY          TIME CMD
    703 pts/0    00:00:00 bash
    718 pts/0    00:00:00 ps
[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 -  5502 do_epo 07:33 ?        00:00:00 /sbin/init
1 S root           2       0  0  80   0 -     0 kthrea 07:33 ?        00:00:00 [kthreadd]
1 S root           3       2  0  80   0 -     0 kthrea 07:33 ?        00:00:00 [pool_workqueue_release]
1 I root           4       2  0  60 -20 -     0 rescue 07:33 ?        00:00:00 [kworker/R-rcu_gp]
1 I root           5       2  0  60 -20 -     0 rescue 07:33 ?        00:00:00 [kworker/R-sync_wq]
1 I root           6       2  0  60 -20 -     0 rescue 07:33 ?        00:00:00 [kworker/R-kvfree_rcu_reclaim]
1 I root           7       2  0  60 -20 -     0 rescue 07:33 ?        00:00:00 [kworker/R-slub_flushwq]
1 I root           8       2  0  60 -20 -     0 rescue 07:33 ?        00:00:00 [kworker/R-netns]
1 I root           9       2  0  80   0 -     0 worker 07:33 ?        00:00:00 [kworker/0:0-events]
1 I root          10       2  0  80   0 -     0 worker 07:33 ?        00:00:00 [kworker/0:1-events]
1 I root          11       2  0  60 -20 -     0 worker 07:33 ?        00:00:00 [kworker/0:0H-kblockd]
1 I root          12       2  0  80   0 -     0 worker 07:33 ?        00:00:00 [kworker/u8:0-events_unbound]
1 I root          13       2  0  80   0 -     0 worker 07:33 ?        00:00:00 [kworker/u8:1-ipv6_addrconf]
1 I root          14       2  0  60 -20 -     0 rescue 07:33 ?        00:00:00 [kworker/R-mm_percpu_wq]
1 S root          15       2  0  80   0 -     0 smpboo 07:33 ?        00:00:00 [ksoftirqd/0]
1 I root          16       2  0  58   - -     0 rcu_gp 07:33 ?        00:00:00 [rcu_preempt]
1 S root          17       2  0  58   - -     0 rcu_bo 07:33 ?        00:00:00 [rcub/0]
1 S root          18       2  0  80   0 -     0 kthrea 07:33 ?        00:00:00 [rcu_exp_par_gp_kthread_worker/0]
1 S root          19       2  0  80   0 -     0 kthrea 07:33 ?        00:00:00 [rcu_exp_gp_kthread_worker]
1 S root          20       2  0 -40   - -     0 smpboo 07:33 ?        00:00:00 [migration/0]
1 S root          21       2  0   9   - -     0 smpboo 07:33 ?        00:00:00 [idle_inject/0]
1 S root          22       2  0  80   0 -     0 smpboo 07:33 ?        00:00:00 [cpuhp/0]
1 S root          23       2  0  80   0 -     0 smpboo 07:33 ?        00:00:00 [cpuhp/1]
1 S root          24       2  0   9   - -     0 smpboo 07:33 ?        00:00:00 [idle_inject/1]
1 S root          25       2  0 -40   - -     0 smpboo 07:33 ?        00:00:00 [migration/1]
1 S root          26       2  0  80   0 -     0 smpboo 07:33 ?        00:00:00 [ksoftirqd/1]
1 I root          27       2  0  80   0 -     0 worker 07:33 ?        00:00:00 [kworker/1:0-ata_sff]
1 I root          28       2  0  60 -20 -     0 worker 07:33 ?        00:00:00 [kworker/1:0H-kblockd]
5 I root          29       2  0  80   0 -     0 worker 07:33 ?        00:00:00 [kworker/u9:0-events_unbound]
1 I root          30       2  0  80   0 -     0 worker 07:33 ?        00:00:00 [kworker/u10:0-events_power_efficient]
5 S root          31       2  0  80   0 -     0 devtmp 07:33 ?        00:00:00 [kdevtmpfs]
```

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

Il comando **ps** di base mostra solo i processi dell'utente corrente ma molti processi in background sono di proprietà di **root(Amministratore)** o di altri utenti di sistema
ps: mostra i processi attivi
-e: tutti i processi del sistema
-l:long format**,** informazioni dettagliate.
-f: full format**,** mostra informazioni complete e l'albero dei processi

b.In Linux, i programmi possono anche chiamare altri programmi. Il comando **ps** può anche essere usato per visualizzare tale gerarchia di processi.

Usa le opzioni **-ejH** per visualizzare l'albero dei processi attualmente in esecuzione dopo aver avviato il **webserver nginx** con privilegi elevati.

sudo /usr/sbin/nginx
sudo ps -ejH

-e Mostra tutti i processi in esecuzione.
-j Aggiunge colonne utili per il controllo (PGID, SID, TTY).
-H Visualizza i processi in struttura ad albero (gerarchia).

```
[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
      9       0       0 ?        00:00:00   kworker/0:0-events
     10       0       0 ?        00:00:00   kworker/0:1-events
     11       0       0 ?        00:00:00   kworker/0:0H-kblockd
     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 ?        00:00:00   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 ?        00:00:00   migration/1
     26       0       0 ?        00:00:00   ksoftirqd/1
     28       0       0 ?        00:00:00   kworker/1:0H-kblockd
     30       0       0 ?        00:00:00   kworker/u10:0-events_power_efficient
     31       0       0 ?        00:00:00   kdevtmpfs
     32       0       0 ?        00:00:00   kworker/R-inet_frag_wq
     33       0       0 ?        00:00:00   rcu_tasks_kthread
```

```
  613     607     607 ?        00:00:01     VBoxClient
  615     615     615 ?        00:00:00   polkitd
  658     658     658 ?        00:00:00   upowerd
  678     677     677 ?        00:00:00   VBoxClient
  679     677     677 ?        00:00:00     VBoxClient
  697     468     468 ?        00:00:01   xfce4-terminal
  703     703     703 pts/0    00:00:00     bash
  789     789     703 pts/0    00:00:00       sudo
  791     791     791 pts/2    00:00:00         sudo
  792     792     791 pts/2    00:00:00           ps
  783     783     783 ?        00:00:00   nginx
  784     783     783 ?        00:00:00     nginx
```

Come viene rappresentata la gerarchia dei processi da ps?

L'albero dei processi di **ps -ejH** mostra la relazione genitore -> figlio tramite indentazione, ogni processo figlio è indentato sotto il processo genitore.

c. I server sono essenzialmente programmi, il compito svolto da un server è chiamato servizio. In questo modo, un web server fornisce servizi web.
Il comando **netstat** è un ottimo strumento per aiutare a identificare i server di rete in esecuzione su un computer.

Nella finestra del terminale, digita **netstat**.

```
[analyst@secOps ~]$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  3      [ ]         STREAM     CONNECTED     5360
unix  2      [ ]         DGRAM      CONNECTED     2215
unix  3      [ ]         STREAM     CONNECTED     3571     /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     5461     /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     5366
unix  3      [ ]         DGRAM      CONNECTED     3655
unix  2      [ ]         DGRAM                    2783
unix  3      [ ]         STREAM     CONNECTED     2590
unix  3      [ ]         STREAM     CONNECTED     5470     /tmp/.ICE-unix/468
unix  3      [ ]         STREAM     CONNECTED     2797     /run/dbus/system_bus_socket
unix  2      [ ]         DGRAM      CONNECTED     3616
unix  3      [ ]         STREAM     CONNECTED     4748     /run/user/1000/bus
unix  3      [ ]         DGRAM      CONNECTED     2217
unix  3      [ ]         DGRAM      CONNECTED     805
unix  3      [ ]         STREAM     CONNECTED     4899
unix  3      [ ]         STREAM     CONNECTED     2727
unix  3      [ ]         DGRAM      CONNECTED     800      /run/systemd/notify
unix  3      [ ]         STREAM     CONNECTED     4887
unix  3      [ ]         STREAM     CONNECTED     2599
unix  2      [ ]         DGRAM      CONNECTED     5376
unix  3      [ ]         DGRAM      CONNECTED     3654
unix  3      [ ]         DGRAM      CONNECTED     802
unix  3      [ ]         STREAM     CONNECTED     2829
unix  3      [ ]         STREAM     CONNECTED     2646
unix  3      [ ]         STREAM     CONNECTED     5484     /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     5377
unix  3      [ ]         DGRAM      CONNECTED     3657
unix  3      [ ]         STREAM     CONNECTED     4931
```

d.Usa **netstat** con le opzioni **-tunap** per regolare l'output di netstat.

```
[analyst@secOps ~]$  sudo netstat -tunap
[sudo] password for analyst:
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN      445/vsftpd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      379/sshd: /usr/bin/
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      783/nginx: master p
tcp        0      0 0.0.0.0:6633            0.0.0.0:*               LISTEN      373/python3.9
tcp6       0      0 :::22                   :::*                    LISTEN      379/sshd: /usr/bin/
udp        0      0 10.0.2.15:68            0.0.0.0:*                           291/systemd-network
```

Qual è il significato delle opzioni -t, -u, -n, -a -p in netstat? (usa man netstat per rispondere)

```
netstat {--statistics|-s} [--tcp|-t] [--udp|-u] [--udplite|-U] [--sctp|-S] [--raw|-w]
```

```
--numeric, -n
    Show numerical addresses instead of trying to determine symbolic host, port or user names.
```

```
-a, --all
    Show both listening and non-listening sockets.  With the --interfaces option, show interfaces that are not up
```

```
-p, --program
   Show the PID and name of the program to which each socket belongs.  A hyphen is shown if the socket belongs to the kernel (e.g. a kernel service, or the process has exited but the socket hasn't  finished
   closing yet).
```

-t: tcp
-u: udp
-n: mostra indirizzi numerici
-a: mostra gli entrambi socket, listening e non-listening
-p: mostra il PID e il nome del programma alla quale appartengono ogni socket.

L'ordine delle opzioni è importante per netstat?
No, l'ordine delle opzioni in netstat non è importante perché vengono interpretate come flag indipendenti**.**

Basandosi sull'output di netstat mostrato al punto (d), qual è il protocollo di Livello 4, lo stato della connessione e il PID del processo in esecuzione sulla porta 80?

In base all'output di **netstat** il protocollo di livello 4 è tcp, lo stato di connessione è sul Listen (il servizio è in ascolto per connessioni in ingresso), il PID del processo di esecuzione è il **783 /nginx: master p**

Sebbene i numeri di porta siano solo una convenzione, puoi indovinare che tipo di servizio è in esecuzione sulla porta 80 TCP?

La porta 80/TCP è la convenzione standard per HTTP.
Dato che il processo è **nginx**, il servizio in esecuzione è un web server HTTP, nello specifico **NGINX**.

e. A volte è utile incrociare le informazioni fornite da netstat con ps. Basandosi sull'output del punto (d), si sa che un processo con PID *** è associato alla porta TCP 80. Usa **ps** e **grep** per elencare tutte le righe dell'output di **ps** che contengono PID ***.

sudo ps -elf | grep 783

```
[analyst@secOps ~]$ man netstat
[analyst@secOps ~]$ sudo ps -elf | grep 783
[sudo] password for analyst:
1 S root         783       1  0  80   0 -  3723 sigsus 07:50 ?        00:00:00 nginx: master process /usr/sbin/nginx
5 S http         784     783  0  80   0 -  3827 do_epo 07:50 ?        00:00:00 nginx: worker process
0 S analyst     1027     703  0  80   0 -  1615 anon_p 09:10 pts/0    00:00:00 grep 783
```

● Il processo è PID 783 nginx. Come si potrebbe concludere questo dall'output sopra?
Si potrebbe concludere questo dall'output **nginx: master process /usr/sbin/nginx**

● Cos'è nginx? Qual è la sua funzione? (Usa google per saperne di più su nginx)

Estratto della ricerca attraverso Google

"NGINX (pronunciato "engine-x") è un web server open-source ad altissime prestazioni, utilizzato anche come reverse proxy, bilanciatore di carico e cache HTTP. Progettato per gestire migliaia di connessioni simultanee con un basso consumo di risorse, è fondamentale per accelerare i siti web, gestire il traffico intenso e aumentare la sicurezza.

Le funzioni principali di NGINX includono:

- ***Web Server:** Ospita contenuti statici (come immagini, CSS, HTML) in modo estremamente veloce.*
- ***Reverse Proxy:** Agisce da intermediario, inoltrando le richieste dei client ai server di backend (es. applicazioni Node.js o Python), migliorando la sicurezza e la gestione del traffico.*
- ***Bilanciamento del Carico (Load Balancer):** Distribuisce il traffico in entrata su più server per evitare sovraccarichi e garantire l'alta disponibilità.*
- ***Caching HTTP:** Memorizza le risposte dei server per velocizzare le richieste successive.*
- ***Sicurezza:** Funge da firewall applicativo (WAF), supporta SSL/TLS per la crittografia e protegge da attacchi DDoS.*

È noto per la sua architettura asincrona ad eventi, che lo rende superiore a server tradizionali come Apache in scenari ad alto traffico"

●La seconda riga mostra che il processo 784 è di proprietà di un utente chiamato http e ha il processo numero 783 come processo genitore. Cosa significa? È un comportamento comune?

Significa che Il **master process** gira come root per potere aprire porte privilegiate (es. 80, 443) e leggere certificati e file protetti.

I worker process invece girano come utente non privilegiato (http, www-data, nginx, ecc.) per ridurre l'impatto di un eventuale exploit. Questo è un pattern di sicurezza standard nei demoni di rete (principio del least privilege).

● **Perché l'ultima riga mostra `grep 783`?**
grep 783 è esso stesso un processo in esecuzione in questo momento.Poiché la stringa 783 è presente nel comando **grep 783**, **ps** lo mostra e **grep** trova sé stesso.

Parte 2: Usare Telnet per Testare i Servizi TCP

Istruzioni

a. Nella Parte 1, si è scoperto che nginx era in esecuzione e assegnato alla **porta 80 TCP**. Sebbene una rapida ricerca su internet abbia rivelato che nginx è un server web leggero, come potrebbe un analista esserne sicuro? E se un attaccante avesse cambiato il nome di un programma malware in nginx, solo per farlo sembrare il popolare webserver? Usa **telnet** per connetterti all'host locale sulla porta **80 TCP**

telnet 127.0.0.1 80

```
[analyst@secOps ~]$ telnet 127.0.0.1 80
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
```

b. Premi alcune lettere sulla tastiera. Qualsiasi tasto funzionerà. Dopo aver premuto alcuni tasti, premi INVIO

*** asdasdaddasd**

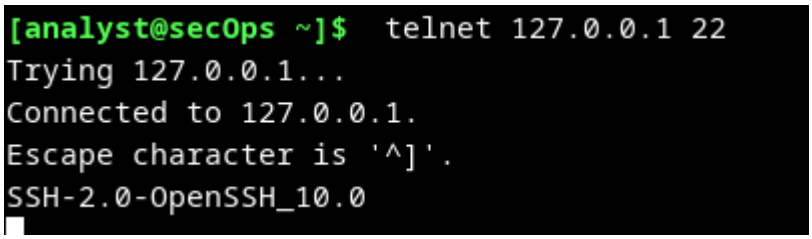
```
asdasdaddasd
HTTP/1.1 400 Bad Request
Server: nginx/1.28.0
Date: Mon, 23 Feb 2026 14:32:48 GMT
Content-Type: text/html
Content-Length: 157
Connection: close

<html>
<head><title>400 Bad Request</title></head>
<body>
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx/1.28.0</center>
</body>
</html>
Connection closed by foreign host.
```

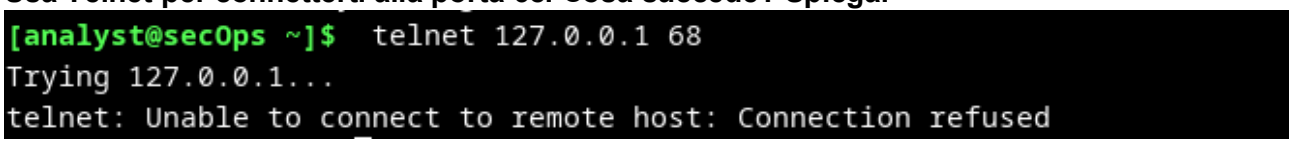
Perché l'errore è stato inviato come pagina web?
Perché si sta parlando con un **web server HTTP** (nello specifico **NGINX**) e l'errore provocato è un errore di protocollo HTTP, quindi la risposta deve rispettare il protocollo HTTP.
***asdasaddasd**: è una stringa non valida come richiesta HTTP. Il server si aspetterebbe qualcosa come GET / HTTP/1.1 la quale sarebbe una richiesta HTTP valida.

c. Guardando l'output di **netstat** presentato prima, è possibile vedere un processo associato alla porta 22. Usa Telnet per connetterti ad esso.

telnet 127.0.0.1 22



Usa Telnet per connetterti alla porta 68. Cosa succede? Spiega.



Rifiuta la connessione, la porta 68 non espone un servizio interattivo TCP. La porta 68/UDP è usata da DHCP client che non usa TCP ma usa UDP. Non c'è un servizio in ascolto sulla porta 68, inoltre Telnet funziona solo con servizi TCP testuali.

Domande di Riflessione

1. Quali sono i vantaggi dell'uso di netstat?
netstat è uno strumento di diagnostica di rete locale. I principali vantaggi sono:
Visibilità dello stato di rete (es. porte in ascolto, connessione attive, mostra quali servizi o processi usano le porte)
Correlazione porta -> processo (es. mappa la porta, PID, programma e individuare servizi inattesi in ascolto)
Supporto al troubleshooting (es. debug di servizi che non rispondono, verifica se un servizio è in ascolto e individua i conflitti di porta)

2. Quali sono i vantaggi dell'uso di Telnet? È sicuro?
Telnet permette di testare se una porta tcp è aperta, vedere i banner di servizio e inviare richieste testuali grezze. Molto utile nei laboratori e per una verifica rapida di servizii
Telnet non è sicuro. È vulnerabile ad attacchi Man in The Middle, session hijacking e sniffing. Ad oggi è sostituito da SecureShell (SSH) per l'accesso remoto sicuro