Build Week 3

Esercizio 2

LANDA

TRACKER SPA

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

PARTE 1: SERVER

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

E' stato necessario eseguire ps come root perché tramite il superutente (**root**) si acquisiscono i privilegi più alti, ciò consente di vedere tutti i processi in esecuzione, inclusi quelli degli altri utenti e i processi di sistema.

Come viene rappresentata la gerarchia dei processi da ps?

Usando il comando **ps -ejH**, otteniamo una visualizzazione dettagliata dei processi in esecuzione, includendo informazioni sulla gerarchia dei processi e altre informazioni relative al controllo delle sessioni e dei gruppi di processi.

```
  472   472   472 ?        00:00:00   systemd
  473   472   472 ?        00:00:00     (sd-pam)
  483   483   483 ?        00:00:00     dbus-daemon
  501   483   483 ?        00:00:00     xfconfd
  506   506   506 ?        00:00:01     gpg-agent
  545   545   545 ?        00:00:00     at-spi-bus-laun
  550   545   545 ?        00:00:01       dbus-daemon
  553   545   545 ?        00:00:20     at-spi2-registr
  493   493   493 ?        00:00:00   polkitd
  504   504   504 ?        00:00:00   ssh-agent
  512   512   512 ?        00:00:10   xfsettingsd
  529   529   529 ?        00:00:00   xfce4-power-man
  559   559   559 ?        00:00:01   upowerd
 1055   478   478 ?        00:00:12   xfce4-terminal
 1059  1059  1059 pts/0    00:00:00     bash
 1100  1100  1059 pts/0    00:00:00       sudo
 1101  1100  1059 pts/0    00:00:00         ps
 1095  1095  1095 ?        00:00:00   nginx
 1096  1095  1095 ?        00:00:00     nginx
```

Qual è il significato delle opzioni -t, -u, -n, -a e -p in netstat? (usa man netstat per rispondere) L'ordine delle opzioni è importante per netstat?

L'ordine delle opzioni non influisce sul comportamento del comando **netstat**.

- **-t**: Connessioni TCP. Se non specificato, netstat mostra tutte le connessioni
- **-u**: Connessioni UDP. Se non specificato, netstat mostra tutte le connessioni
- **-n**: Indirizzi e porte numerici (no risoluzione dei nomi).
- **-a**: Connessioni e porte di ascolto.
- **-p**: Mostra il PID e il nome del processo.

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?

Il protocollo di livello 4 mostrato è il protocollo **TCP**, in stato **LISTEN** e il suo PID è **395** di ngnix: master.

```
[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 0.0.0.0:6633            0.0.0.0:*               LISTEN      257/python2.7
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      395/nginx: master
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN      279/vsftpd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      277/sshd: /usr/bin
tcp6       0      0 :::22                   :::*                    LISTEN      277/sshd: /usr/bin
udp        0      0 192.168.1.15:68         0.0.0.0:*                           237/systemd-network
```

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

Il servizio in esecuzione sulla porta TCP **80** è molto probabilmente **HTTP**, poiché è una porta standard e nota riservata a questo tipo di traffico.

Il processo PID 395 è nginx. Come si potrebbe concludere questo dall'output sopra?

Possiamo arrivare alla conclusione che il PID di nginx è **395** leggendo la l'ultima colonna della prima riga (*PID/Program name*)

Cos'è nginx? Qual è la sua funzione?

Nginx è un software open-source ad alte prestazioni che funge primariamente da:

- **Server Web**: Serve contenuti web in modo efficiente e veloce.

- **Reverse Proxy e Load Balancer**: essenziale per la moderna architettura web. Riceve le richieste dei client e le inoltra ai server applicativi interni (reverse proxy), distribuendo il carico di traffico tra essi per garantire scalabilità e alta disponibilità (load balancing).

Il suo punto di forza è l'architettura asincrona che gli permette di gestire un numero elevatissimo di connessioni simultanee con un consumo di risorse molto basso.

La seconda riga mostra che il processo 396 è di proprietà di un utente chiamato http e ha il processo numero 395 come processo genitore. Cosa significa? È un comportamento comune?

E' un comportamento di sicurezza standard e molto comune per i server web. Il processo **master nginx 395** ha creato il **worker 396** con un utente non privilegiato per aderire al Principio del Minimo Privilegio.

Questa è un'architettura tipica di nginx, il master process (root) legge i file di configurazione, apre le porte e fa da supervisore, mentre il worker process (utente limitato) gestisce le connessioni con i client.

In questo modo, se il processo del worker viene compromesso, l'attaccante avrà solo i privilegi minimi impedendo, così, l'accesso o il danneggiamento dei file di sistema. Questa è una misura di sicurezza fondamentale.

```
[analyst@secOps ~]$ sudo ps -elf | grep 395
           [sudo] password for analyst:
           1 S root       395     1  0  80   0 -  1829     19:33 ?        00:00:00 nginx: master process /usr/bin/nginx
           5 S http       396   395  0  80   0 -  1866     19:33 ?        00:00:00 nginx: worker process
```

Perché l'ultima riga mostra `grep 395`?

Grep è un'utilità a riga di comando che cerca linee di testo che corrispondono al pattern richiesto. In questo caso noi abbiamo richiesto **395**, quindi l'output di **ps -elf** contiene la lista di processi in esecuzione che corrispondono alla nostra richiesta (395) comprendendo anche il grep stesso perché attivato.

```
[analyst@secOps ~]$ sudo ps -elf | grep 395
           [sudo] password for analyst:
           1 S root       395     1  0  80   0 -  1829     19:33 ?        00:00:00 nginx: master process /usr/bin/nginx
           5 S http       396   395  0  80   0 -  1866     19:33 ?        00:00:00 nginx: worker process
           0 S analyst   3789  1872  0  80   0 -  1190     19:53 pts/0    00:00:00 grep 395
```

PARTE 2: USARE TELNET PER TESTARE I SERVIZI TCP

Perché l'errore è stato inviato come pagina web?

L'errore è stato inviato come pagina web perché connessi tramite porta **80** che utilizza il protocollo **HTTP**.

```
[analyst@secOps ~]$ telnet 127.0.0.1 80
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
dsdvdfdf
HTTP/1.1 400 Bad Request
Server: nginx/1.12.2
Date: Mon, 29 Sep 2025 14:28:49 GMT
Content-Type: text/html
Content-Length: 173
Connection: close

<html>
<head><title>400 Bad Request</title></head>
<body bgcolor="white">
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx/1.12.2</center>
</body>
</html>
Connection closed by foreign host.
```

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

Usando Telnet per connetterci alla porta **68** riceviamo come output "*Connection refused*" poiché Telnet non è il tool giusto per interagire con la porta 68 (UDP), e, soprattutto, non c'è alcun servizio TCP in ascolto su quella porta per accettare la connessione.

```
[analyst@secOps ~]$ telnet 127.0.0.1 68
Trying 127.0.0.1...
telnet: Unable to connect to remote host: Connection refused
[analyst@secOps ~]$ nestat -tunap
bash: nestat: command not found
[analyst@secOps ~]$ netstat -tunap
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:6633            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:22            127.0.0.1:33176         TIME_WAIT   -
tcp6       0      0 :::22                   :::*                    LISTEN      -
```

Quali sono i vantaggi dell'uso di netstat?

I vantaggi di usare **netstat** (Network Statistics) sono:

- **Sicurezza**: Identifica le porte aperte (LISTEN) e, se usato con le opzioni appropriate (-p o -o), rivela il processo (PID) che le sta utilizzando, aiutando a rilevare malware o servizi non autorizzati.

- **Diagnostica**: Mostra lo stato di tutte le connessioni attive, permettendo di verificare rapidamente se un servizio sta comunicando correttamente o è bloccato.

- **Informazioni di Rete**: Visualizza la tabella di routing (-r) e le statistiche di traffico per protocollo (TCP, UDP), fondamentali per comprendere il flusso di rete.

Quali sono i vantaggi dell’uso di Telnet? E’ sicuro?

Telnet, pur essendo un protocollo obsoleto per l'amministrazione remota, ha alcuni vantaggi che lo rendono ancora utile in contesti come il testing. I suoi vantaggi sono semplicità, compatibilità universale, utile a livello didattico.

Essendo datato, come anticipato prima, non è sicuro e il suo utilizzo per l'accesso remoto è fortemente sconsigliato e superato da decenni. (Trasmissione in chiaro, vulnerabilità, è stato sostituito da **SSH**).