

S13 / L4

DNS TRAFFIC, MYSQL ATTACKS,
SERVER LOGS

CONTENUTI

00

Traccia

01

Working with Text Files in the CLI

02

Getting Familiar with the Linux Shell

03

Linux Servers

04

Navigating the Linux Filesystem and Permission
Settings

00 TRACCIA

1) Working with Text Files in the CLI

In this lab, you will get familiar with Linux command-line text editors and configuration files.

2) Getting Familiar with the Linux Shell

In this lab, you will use the Linux command line to manage files and folders and perform some basic administrative tasks.

3) Linux Servers

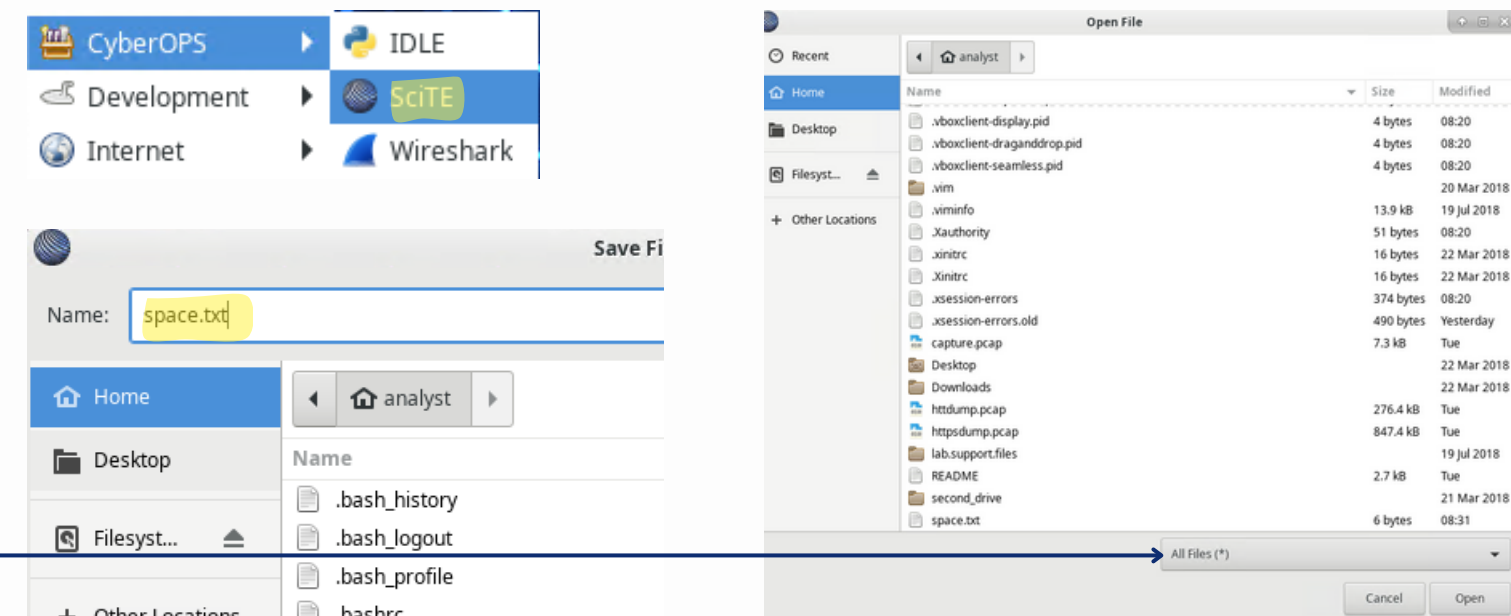
In this lab, you will use the Linux command line to identify servers that are running on a computer.

4) Navigating the Linux Filesystem and Permission Settings

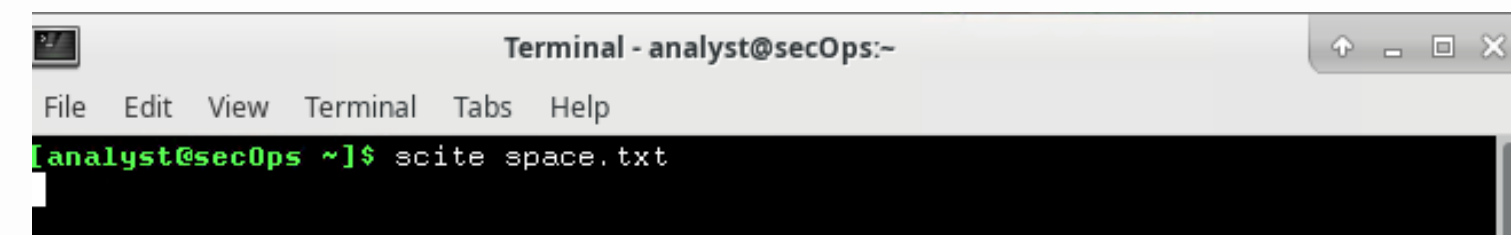
In this lab, you will familiarize yourself with Linux filesystems.

01 - WORKING WITH TEXT FILES IN THE CLI

Apro **SciTE** dal menu Applications > CyberOPS > SciTE, **scrivo qualcosa e salvo il file come space.txt** nella mia home directory. Mi accorgo che SciTE non mi mostra immediatamente il file salvato. Questo succede perché non riconosce automaticamente i file con estensione .txt. Seleziono "**Tutti i file**" dal menu a discesa per visualizzare correttamente il file salvato.



Provo ad aprire SciTE anche dal **terminale** usando il comando **scite space.txt**. Notando che il terminale rimane "bloccato" finché SciTE è aperto, capisco che il terminale sta eseguendo il programma in primo piano e non può ricevere altri comandi finché non chiudo l'applicazione.



01 - WORKING WITH TEXT FILES IN THE CLI

I file di configurazione sono vitali in Linux e possono essere usati per configurare applicazioni e servizi. Ci sono **due** principali categorie di file di configurazione: quelli a **livello utente**, che sono nascosti nella home directory, e quelli a **livello di sistema**, che si trovano in /etc.

Uso il comando **ls -la** per vedere i **file nascosti** nella mia home directory. Questo mi permette di trovare file come **.bashrc**, che contiene configurazioni personalizzate per il mio terminale. Uso cat per visualizzare il contenuto di .bashrc, scoprendo che questo file gestisce la **configurazione** del prompt e altri **alias** utili.

```
[analyst@sec0ps ~]$ ls -la
total 1264
drwx----- 15 analyst analyst 4096 Sep  5 08:31 .
drwxr-xr-x  3 root    root    4096 Mar 20  2018 ..
-rw-----  1 analyst analyst 1422 Sep  3 10:15 .bash_history
-rw-r--r--  1 analyst analyst  21 Feb  7  2018 .bash_logout
-rw-r--r--  1 analyst analyst  57 Feb  7  2018 .bash_profile
-rw-r--r--  1 analyst analyst  97 Mar 20  2018 .bashrc
```

```
PS1='\[\e[1;32m\][\u@\h \W]\$'\[\e[0m\] '
alias ls="ls --color"
alias vi="vim"
```

```
GNU nano 2.9.5 /etc/bash.bashrc
#
# /etc/bash.bashrc
#
# If not running interactively, don't do anything
[[ $- != *i* ]] && return

[[ $DISPLAY ]] && shopt -s checkwinsize

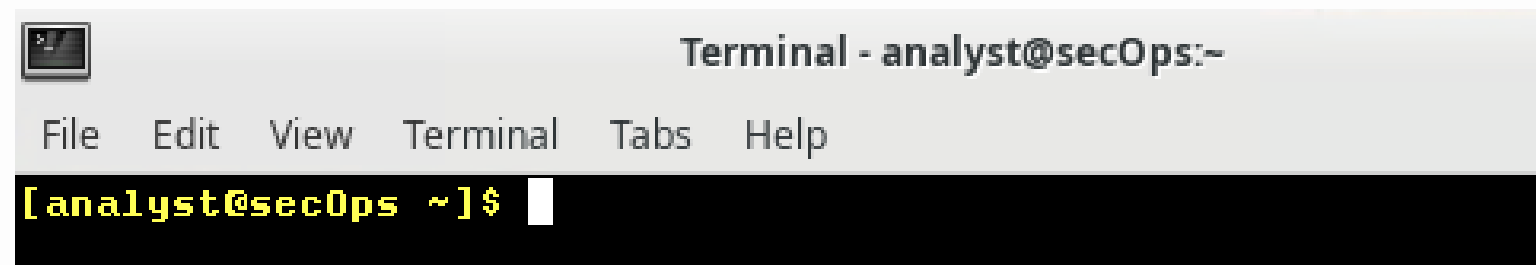
PS1='\u@\h \W]\$ '

case ${TERM} in
  xterm*|rxvt*|Eterm|aterm|kterm|gnome*)
    PROMPT_COMMAND=${PROMPT_COMMAND:+$PROMPT_COMMAND;} 'printf "\033[0;${@}s: ${@}s$'
    ;;
  screen*)
    PROMPT_COMMAND=${PROMPT_COMMAND:+$PROMPT_COMMAND;} 'printf "\033_ ${@}s: ${@}s\0$'
    ;;
  *)
    ;;
esac
```

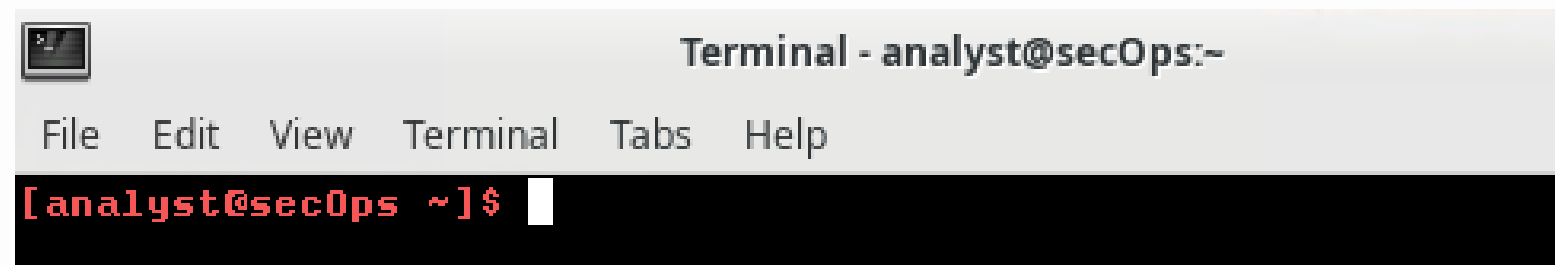
Passo poi a esplorare i file di configurazione di sistema in /etc, che richiedono permessi di root per essere modificati. Provo a visualizzare e modificare **/etc/bash.bashrc**, ma ovviamente bisogna essere **root** per farlo.

01 - WORKING WITH TEXT FILES IN THE CLI

Per cambiare il colore del prompt, apro **.bashrc** e cambio il codice colore da 32 (verde) a **31** (rosso). Dopo aver salvato le modifiche e riavviato il terminale, vedo il prompt colorato in rosso. Faccio lo stesso processo con nano, cambiando il colore del prompt a **giallo** e ricarico la configurazione con il comando `bash`.



```
Terminal - analyst@secOps:~  
File Edit View Terminal Tabs Help  
[analyst@secOps ~]$
```

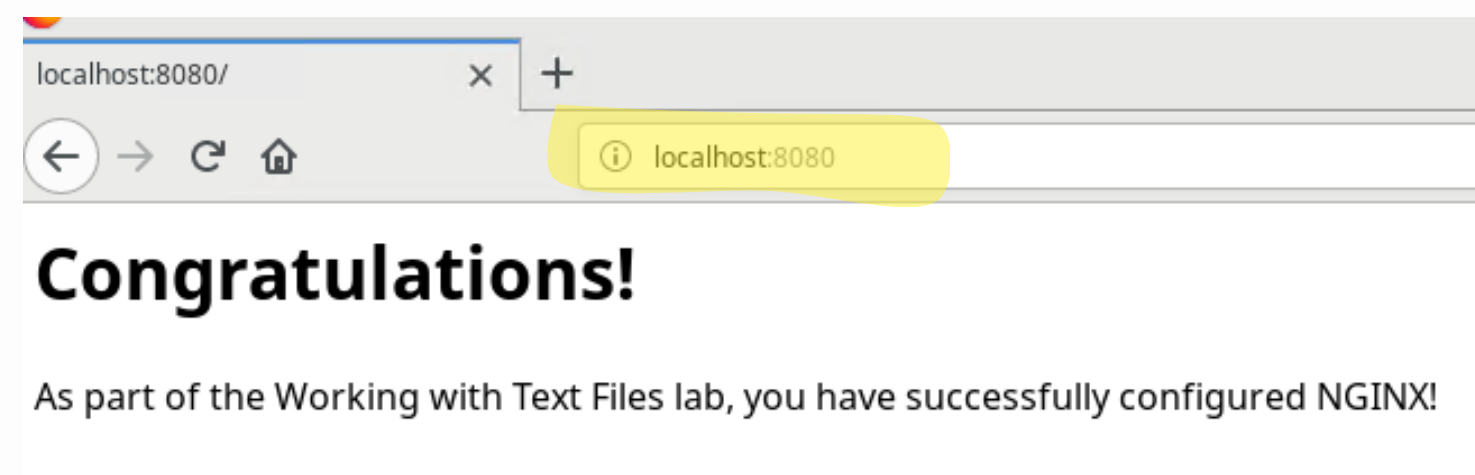


```
Terminal - analyst@secOps:~  
File Edit View Terminal Tabs Help  
[analyst@secOps ~]$
```

Infine, modifico il file di **configurazione di nginx**, cambiando la porta su cui il server web ascolta e la directory da cui serve le pagine web. Avvio nginx e verifico che tutto funzioni correttamente accedendo al server tramite il browser.

```
38     server {  
39         listen      8080;  
40         server_name localhost;  
41  
42         #charset koi8-r;
```

```
45     location / {  
46         root       /usr/share/nginx/html/text_ed_lab/  
47         index      index.html index.htm;  
48     }  
49 }
```



02 - GETTING FAMILIAR WITH THE LINUX SHELL

Il comando **man**, è la bibbia dei sistemi Linux/Unix-Like. Ecco alcuni esempi:

```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
MAN(1) Manual pager utils MAN(1)

NAME
  man - an interface to the on-line reference manuals

SYNOPSIS
  man [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L
  locale] [-m system[...]] [-M path] [-S list] [-e extension] [-i|-I]
  [--regex|--wildcard] [--names-only] [-a] [-u] [--no-subpages] [-P
  pager] [-r prompt] [-?] [-E encoding] [--no-hyphenation] [--no-justifi-
  cation] [-p string] [-t] [-T[device]] [-H[browser]] [-X[dpi]] [-Z]
  [[section] page[.section] ...] ...
  man -k [apropos options] regexp ...
  man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
  man -f [whatis options] page ...
  man -l [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L
  locale] [-P pager] [-r prompt] [-?] [-E encoding] [-p string] [-t]
  [-T[device]] [-H[browser]] [-X[dpi]] [-Z] file ...
  man -w|-W [-C file] [-d] [-D] page ...
  man -c [-C file] [-d] [-D] page ...
  man [-?U]

DESCRIPTION
  man is the system's manual pager. Each page argument given to man is
  normally the name of a program, utility or function. The manual page
  associated with each of these arguments is then found and displayed. A
  section, if provided, will direct man to look only in that section of
  the manual. The default action is to search in all of the available
  sections following a pre-defined order ("1 n 1 8 3 0 2 5 4 9 6 7" by
  default, unless overridden by the SECTION directive in
  /etc/man_db.conf), and to show only the first page found, even if page
  exists in several sections.

  The table below shows the section numbers of the manual followed by the
  types of pages they contain.

  1 Executable programs or shell commands
  2 System calls (functions provided by the kernel)
  3 Library calls (functions within program libraries)
  4 Special files (usually found in /dev)
  5 File formats and conventions eg /etc/passwd
  6 Games
  7 Miscellaneous (including macro packages and conventions), e.g.
  man(7), groff(7)
  8 System administration commands (usually only for root)
  9 Kernel routines [Non standard]

  A manual page consists of several sections.

Manual page man(1) line 1 (press h for help or q to quit)
```

```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
PWD(1) User Commands PWD(1)

NAME
  pwd - print name of current/working directory

SYNOPSIS
  pwd [OPTION]...

DESCRIPTION
  Print the full filename of the current working directory.

  -L, --logical
      use PWD from environment, even if it contains symlinks

  -P, --physical
      avoid all symlinks

  --help display this help and exit

  --version
      output version information and exit

  If no option is specified, -P is assumed.

  NOTE: your shell may have its own version of pwd, which usually super-
  sedes the version described here. Please refer to your shell's docu-
  mentation for details about the options it supports.

AUTHOR
  Written by Jim Meyering.

REPORTING BUGS
  GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
  Report pwd translation bugs to <https://translationproject.org/team/>

COPYRIGHT
  Copyright © 2017 Free Software Foundation, Inc. License GPLv3+: GNU
  GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
  This is free software: you are free to change and redistribute it.
  There is NO WARRANTY, to the extent permitted by law.

SEE ALSO
  getcwd(3)

  Full documentation at: <https://www.gnu.org/software/coreutils/pwd>
  or available locally via: info '(coreutils) pwd invocation'

GNU coreutils 8.29 December 2017 PWD(1)
Manual page pwd(1) line 1/48 (END) (press h for help or q to quit)
```

```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
CP(1) User Commands CP(1)

NAME
  cp - copy files and directories

SYNOPSIS
  cp [OPTION]... [-I] SOURCE DEST
  cp [OPTION]... SOURCE... DIRECTORY
  cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
  Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

  Mandatory arguments to long options are mandatory for short options
  too.

  -a, --archive
      same as -dR --preserve=all

  --attributes-only
      don't copy the file data, just the attributes

  --backup[=CONTROL]
      make a backup of each existing destination file

  -b like --backup but does not accept an argument

  --copy-contents
      copy contents of special files when recursive

  -d same as --no-dereference --preserve=links

  -f, --force
      if an existing destination file cannot be opened, remove it and
      try again (this option is ignored when the -n option is also
      used)

  -i, --interactive
      prompt before overwrite (overrides a previous -n option)

  -H follow command-line symbolic links in SOURCE

  -l, --link
      hard link files instead of copying

  -L, --dereference
      always follow symbolic links in SOURCE

  -n, --no-clobber
```

02 - GETTING FAMILIAR WITH THE LINUX SHELL

Ho usato **mkdir** per creare nuove cartelle e **cd** per spostarmi tra le directory. Con **ls -l** ho verificato il contenuto delle directory.

```
[analyst@secOps ~]$ mkdir cyops_folder1 cyops_folder2 cyops_folder3
[analyst@secOps ~]$ ls -l
total 1144
-rw-r--r-- 1 root    root      7326 Sep  3 08:41 capture.pcap
drwxr-xr-x 2 analyst analyst  4096 Sep  5 09:40 cyops_folder1
drwxr-xr-x 2 analyst analyst  4096 Sep  5 09:40 cyops_folder2
drwxr-xr-x 2 analyst analyst  4096 Sep  5 09:40 cyops_folder3
```

```
[analyst@secOps ~]$ echo il mio testo > miofile.txt
[analyst@secOps ~]$ cat miofile.txt
il mio testo
[analyst@secOps ~]$
```

Con **echo** ho stampato messaggi e ho reindirizzato il risultato in un file usando **>**. Con **>>** ho aggiunto nuove righe al file **senza sovrascrivere il contenuto precedente**.

Con **ls -la** ho visualizzato i file nascosti nella directory, come **.bashrc**, che contengono configurazioni utente.

```
[analyst@secOps ~]$ ls -la
total 1264
drwx----- 15 analyst analyst  4096 Sep  5 08:31 .
drwxr-xr-x  3 root    root      4096 Mar 20  2018 ..
-rw-----  1 analyst analyst  1422 Sep  3 10:15 .bash_history
-rw-r--r--  1 analyst analyst    21 Feb  7  2018 .bash_logout
-rw-r--r--  1 analyst analyst    57 Feb  7  2018 .bash_profile
-rw-r--r--  1 analyst analyst    97 Mar 20  2018 .bashrc
```


02 - GETTING FAMILIAR WITH THE LINUX SHELL

Ho usato cp per copiare un file da una cartella all'altra.

```
[analyst@secOps ~]$ cp miofile.txt cyops_folder2/  
[analyst@secOps ~]$
```

```
[analyst@secOps ~]$ rm miofile.txt  
[analyst@secOps ~]$
```

Ho usato **rm** per eliminare file e **rm -r** per eliminare intere cartelle e il loro contenuto.

Ho usato mv per spostare un file da una cartella alla directory home.

```
[analyst@secOps ~]$ mv cyops_folder2/miofile.txt .  
[analyst@secOps ~]$ ls  
capture.pcap  cyops_folder3  httdump.pcap  miofile.txt  space.txt  
cyops_folder1 Desktop        httpsdump.pcap README  
cyops_folder2 Downloads    lab.support.files second_drive  
[analyst@secOps ~]$
```

03 - LINUX SERVERS

In questo esercizio ho usato il comando **sudo ps -elf** che mi ha mostrato i **processi attivi sul sistema**. Ho dovuto usare sudo per vedere i processi che non appartengono all'utente corrente.

```
399 368 368 ? 00:00:00 Thunar
400 368 368 ? 00:00:00 xfce4-panel
409 368 368 ? 00:00:00 panel-6-systray
411 368 368 ? 00:00:00 panel-2-actions
405 368 368 ? 00:00:00 xfdesktop
446 368 368 ? 00:00:00 polkit-gnome-au
362 362 362 ? 00:00:00 systemd
363 362 362 ? 00:00:00 (sd-pam)
373 373 373 ? 00:00:00 dbus-daemon
391 373 373 ? 00:00:00 xfconfd
396 396 396 ? 00:00:00 gpg-agent
424 424 424 ? 00:00:00 at-spi-bus-laun
429 424 424 ? 00:00:00 dbus-daemon
433 424 424 ? 00:00:00 at-spi2-registr
645 373 373 ? 00:00:00 dconf-service
383 383 383 ? 00:00:00 polkitd
394 394 394 ? 00:00:00 ssh-agent
402 402 402 ? 00:00:00 xfsettingsd
421 421 421 ? 00:00:00 xfce4-power-man
434 434 434 ? 00:00:00 upowerd
458 456 456 ? 00:00:00 VBoxClient
459 456 456 ? 00:00:00 VBoxClient
470 469 469 ? 00:00:00 VBoxClient
471 469 469 ? 00:00:00 VBoxClient
481 477 477 ? 00:00:00 VBoxClient
482 477 477 ? 00:00:00 VBoxClient
486 485 485 ? 00:00:00 VBoxClient
487 485 485 ? 00:00:03 VBoxClient
840 840 840 ? 00:00:00 nginx
841 840 840 ? 00:00:00 nginx
1381 368 368 ? 00:00:00 xfce4-terminal
1385 1385 1385 pts/1 00:00:00 bash
1436 1436 1385 pts/1 00:00:00 sudo
1437 1436 1385 pts/1 00:00:00 ps
1432 1432 1432 ? 00:00:00 nginx
1433 1432 1432 ? 00:00:00 nginx
```

```
[analyst@sec0ps ~]$ 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 - 58535  Sys_ep 08:19 ?        00:00:00 /sbin
1 S root           2      0  0  80    0 -      0 -      08:19 ?        00:00:00 [kthr
1 I root           4      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [kwo
1 I root           6      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [mm_p
1 S root           7      2  0  80    0 -      0 -      08:19 ?        00:00:00 [ksof
1 I root           8      2  0  58   -   -      0 -      08:19 ?        00:00:00 [rcu_
1 I root           9      2  0  58   -   -      0 -      08:19 ?        00:00:00 [rcu_
1 I root          10      2  0  58   -   -      0 -      08:19 ?        00:00:00 [rcu_
1 S root          11      2  0  58   -   -      0 -      08:19 ?        00:00:00 [rcuc
1 S root          12      2  0  58   -   -      0 -      08:19 ?        00:00:00 [rcub
1 S root          13      2  0 -40   -   -      0 -      08:19 ?        00:00:00 [migr
5 S root          14      2  0 -40   -   -      0 -      08:19 ?        00:00:00 [watc
1 S root          15      2  0  80    0 -      0 -      08:19 ?        00:00:00 [cpuh
5 S root          16      2  0  80    0 -      0 -      08:19 ?        00:00:00 [kdev
1 I root          17      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [netn
1 S root          18      2  0  80    0 -      0 -      08:19 ?        00:00:00 [rcu_
1 S root          20      2  0  80    0 -  bios_c 08:19 ?        00:00:00 [khun
1 S root          21      2  0  80    0 -      0 -      08:19 ?        00:00:00 [oom_
1 I root          22      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [writ
1 S root          23      2  0  80    0 -      0 -      08:19 ?        00:00:00 [kcom
1 S root          24      2  0  85    5 -      0 -      08:19 ?        00:00:00 [ksmd
1 S root          25      2  0  99   19 -      0 -      08:19 ?        00:00:00 [khug
1 I root          26      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [cryp
1 I root          27      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [kint
1 I root          28      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [kblo
1 I root          29      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [edac
1 I root          30      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [devf
1 I root          31      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [watc
1 S root          33      2  0  80    0 -      0 -      08:19 ?        00:00:00 [kswa
1 I root          72      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [kthr
1 I root          73      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [acpi
1 I root          74      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [nvme
1 I root          75      2  0  60 -20   -      0 -      08:19 ?        00:00:00 [inve
```

Ho avviato il server **nginx** e ho usato il comando **ps** per visualizzare la gerarchia dei processi attivi.

03 - LINUX SERVERS

Ho poi usato **netstat** per analizzare le connessioni di rete, filtrando con opzioni combinate. Nel man questo è ciò che si impara riguardo i singoli parametri. t: Mostra le connessioni TCP, u: Mostra le connessioni UDP, n: Disabilita la risoluzione DNS e visualizza indirizzi e porte in formato numerico, a: Mostra sia i socket in ascolto che quelli non in ascolto, p: Mostra l'ID del processo e il nome del programma associato a ogni connessione.

```
[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 0.0.0.0:6633            0.0.0.0:*               LISTEN      284/python2.7
tcp        0      0 0.0.0.0:80             0.0.0.0:*               LISTEN      1432/nginx: master
tcp        0      0 0.0.0.0:8080            0.0.0.0:*               LISTEN      840/nginx: master p
tcp        0      0 0.0.0.0:21             0.0.0.0:*               LISTEN      299/vsftpd
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      301/sshd
tcp6       0      0 :::22                  :::*                   LISTEN      301/sshd
udp        0      0 192.168.1.74:68        0.0.0.0:*               *          207/systemd-network
```

Questo mi ha permesso di vedere i servizi attivi, come nginx, su specifiche porte. **Ho incrociato queste informazioni con il comando ps:**

```
[analyst@secOps ~]$ sudo ps -elf | grep 1432
1 S root      1432    1 0 80    0 -  7192 -      09:52 ?        00:00:00 nginx: master process /usr/sbin/nginx
5 S http      1433    0 80    0 -  8457 Sys_ep  09:52 ?        00:00:00 nginx: worker process
0 S analyst   1451   1385 0 80    0 -  2720 -      09:55 pts/1    00:00:00 grep 1432
[analyst@secOps ~]$ sudo ps -elf | grep 840
[sudo] password for analyst:
1 S root      840    1 0 80    0 -  7192 -      09:24 ?        00:00:00 nginx: master process nginx -c custom_s
er.conf
5 S http      841    0 80    0 -  8457 Sys_ep  09:24 ?        00:00:00 nginx: worker process
0 S analyst   1485   1385 0 80    0 -  2720 -     10:08 pts/1    00:00:00 grep 840
```

03 - LINUX SERVERS

Telnet è obsoleto e senza crittografia. In questo esempio l'ho usato per connettermi alla porta **80** e verificare la risposta del server web. Dopo aver inviato una richiesta casuale, il server nginx ha risposto con un errore HTTP, confermando che era attivo e in esecuzione.

```
[analyst@secOps ~]$ telnet 127.0.0.1 80
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
a
HTTP/1.1 400 Bad Request
Server: nginx/1.12.2
Date: Thu, 05 Sep 2024 14:10:14 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.
```

Ho ripetuto il test per la porta 22, associata al servizio **SSH**. Anche qui, il server SSH ha risposto con una stringa di identificazione, confermando la sua presenza.

```
[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_7.7
```

04 - NAVIGATING THE LINUX FILESYSTEM AND PERMISSION SETTINGS

Ho usato **lsblk** per visualizzare i dispositivi di blocco e ho notato che i dispositivi di archiviazione sono **sda** e **sdb**. Con il comando **mount**, ho verificato quali filesystem sono montati, trovando che /dev/sda1 è il filesystem principale, montato su /. Poi ho usato il comando **cd** per navigare nella directory radice e ho usato **ls** per elencare i file, che in realtà sono contenuti fisicamente in **/dev/sda1**. Per montare manualmente un filesystem, **ho creato una directory con mkdir e ho usato mount per montare /dev/sdb1 su ~/second_drive, successivamente ho verificato i contenuti con ls.**

```
[analyst@secOps ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   10G  0 disk
└─sda1       8:1    0   10G  0 part /
sdb          8:16   0    1G  0 disk
└─sdb1       8:17   0 1023M  0 part
sr0         11:0    1    51M  0 rom
```

```
/dev/sda1 on / type ext4 (rw,relatime,data=ordered)
securityfs on /sys/kernel/security type securityfs
```

```
[analyst@secOps ~]$ mkdir second_drive
mkdir: cannot create directory 'second_drive': File exists
[analyst@secOps ~]$ ls -l second_drive/
total 0
[analyst@secOps ~]$ sudo mount /dev/sdb1 ~/second_drive/
[sudo] password for analyst:
[analyst@secOps ~]$ ls -l second_drive/
total 20
drwx----- 2 root    root    16384 Mar 26  2018 lost+found
-rw-r--r--  1 analyst analyst  183 Mar 26  2018 myFile.txt
[analyst@secOps ~]$
```

04 - NAVIGATING THE LINUX FILESYSTEM AND PERMISSION SETTINGS

Ho usato `ls -l` nella directory `~/lab.support.files/scripts/` per visualizzare i **permessi** dei file. I file hanno permessi che specificano chi può leggere, scrivere o eseguire. Per esempio, il file **cyops.mn** ha permessi di lettura e scrittura per il proprietario e di sola lettura per gli altri. Ho poi usato `touch` per creare un file vuoto, ma ho ottenuto un errore di permessi in `/mnt`. Usando **chmod**, ho modificato i permessi di **myFile.txt** da `-rw-r--r--` a `-rw-rw-r-x`. Infine, ho cambiato il proprietario del file con `chown`, rendendo `analyst` il proprietario del file e confermando che potevo modificarlo con successo.

```
[analyst@secOps ~]$ ls -l ~/lab.support.files/scripts/
total 60
-rwxr-xr-x 1 analyst analyst  952 Mar 21  2018 configure_as_dhcp.sh
-rwxr-xr-x 1 analyst analyst 1153 Mar 21  2018 configure_as_static.sh
-rwxr-xr-x 1 analyst analyst 3459 Mar 21  2018 cyberops_extended_topo_no_fw.py
-rwxr-xr-x 1 analyst analyst 4062 Mar 21  2018 cyberops_extended_topo.py
-rwxr-xr-x 1 analyst analyst 3669 Mar 21  2018 cyberops_topo.py
-rw-r--r-- 1 analyst analyst 2871 Mar 21  2018 cyops.mn
-rwxr-xr-x 1 analyst analyst  458 Mar 21  2018 fw_rules
-rwxr-xr-x 1 analyst analyst   70 Mar 21  2018 mal_server_start.sh
drwxr-xr-x 2 analyst analyst 4096 Mar 21  2018 net_configuration_files
-rwxr-xr-x 1 analyst analyst   65 Mar 21  2018 reg_server_start.sh
-rwxr-xr-x 1 analyst analyst  189 Mar 21  2018 start_ELK.sh
-rwxr-xr-x 1 analyst analyst   85 Mar 21  2018 start_miniedit.sh
-rwxr-xr-x 1 analyst analyst   76 Mar 21  2018 start_pox.sh
-rwxr-xr-x 1 analyst analyst  106 Mar 21  2018 start_snort.sh
-rwxr-xr-x 1 analyst analyst   61 Mar 21  2018 start_tftpd.sh
[analyst@secOps ~]$
```

```
[analyst@secOps second_drive]$ ls -l
total 20
drwx----- 2 root      root      16384 Mar 26  2018 lost+found
-rw-r--r--  1 analyst  analyst   183 Mar 26  2018 myFile.txt
[analyst@secOps second_drive]$ sudo chmod 665 myFile.txt
[sudo] password for analyst:
[analyst@secOps second_drive]$ ls -l
total 20
drwx----- 2 root      root      16384 Mar 26  2018 lost+found
-rw-rw-r-x  1 analyst  analyst   183 Mar 26  2018 myFile.txt
[analyst@secOps second_drive]$
```

In Linux, i permessi dei file sono rappresentati in ottale (base 8) per semplificare la gestione dei permessi binari. Ogni cifra ottale rappresenta tre bit: lettura (r), scrittura (w) ed esecuzione (x). Ad esempio, "rw-" per il proprietario è 6 in ottale (110 in binario), mentre "r-x" per gli altri è 5 in ottale (101 in binario). La notazione ottale riassume i permessi in modo compatto.

04 - NAVIGATING THE LINUX FILESYSTEM AND PERMISSION SETTINGS

Ho creato un file simbolico con **ln -s** e un hard link con **ln**. Usando **ls -l**, ho verificato che il link simbolico **punta al nome del file originale**, mentre l'hard link **punta all'inode**, il che significa che cambia il contenuto di un file cambia anche l'altro. Quando ho rinominato i due file, il collegamento simbolico si è rotto, mentre l'hard link ha continuato a funzionare.

```
[analyst@secOps ~]$ echo "symbolic" > file1.txt
[analyst@secOps ~]$ cat file1.txt
symbolic
[analyst@secOps ~]$ echo "hard" > file2.txt
[analyst@secOps ~]$ cat file2.txt
hard
```

```
[analyst@secOps ~]$ ln -s file1.txt file1symbolic
[analyst@secOps ~]$ ln file2.txt file2hard
[analyst@secOps ~]$ ls -l
total 1160
-rw-r--r-- 1 root    root      7326 Sep  3 08:41 capture.pcap
drwxr-xr-x 2 analyst analyst  4096 Sep  5 09:40 cyops_folder1
drwxr-xr-x 2 analyst analyst  4096 Sep  5 09:47 cyops_folder2
drwxr-xr-x 2 analyst analyst  4096 Sep  5 09:40 cyops_folder3
drwxr-xr-x 5 analyst analyst  4096 Sep  5 09:39 Desktop
drwxr-xr-x 3 analyst analyst  4096 Mar 22  2018 Downloads
lrwxrwxrwx 1 analyst analyst    9 Sep  5 10:41 file1symbolic -> file1.txt
-rw-r--r-- 1 analyst analyst     9 Sep  5 10:37 file1.txt
-rw-r--r-- 2 analyst analyst     5 Sep  5 10:37 file2hard
-rw-r--r-- 2 analyst analyst     5 Sep  5 10:37 file2.txt
-rw-r--r-- 1 root    root    276408 Sep  3 10:17 httdump.pcap
-rw-r--r-- 1 root    root   847378 Sep  3 10:30 httpsdump.pcap
drwxr-xr-x 9 analyst analyst  4096 Jul 19  2018 lab.support.files
-rw-r--r-- 1 analyst analyst    13 Sep  5 09:46 miofile.txt
-rw-r--r-- 1 analyst analyst   2748 Sep  3 10:36 README
drwxr-xr-x 3 root    root      4096 Mar 26  2018 second_drive
-rw-r--r-- 1 analyst analyst     6 Sep  5 08:31 space.txt
```



GRAZIE

Flavio Scognamiglio