

LABORATORIO 20 FEBBRAIO 2025 S11-L4

Laboratorio:

Utilizzo dei file di testo nella CLI

In questo laboratorio acquisiremo familiarità con gli editor di testo della riga di comando e i file di configurazione di Linux.

<https://itexamanswers.net/4-2-6-lab-working-with-text-files-in-the-cli-answers.html>

☰ Contenuto

- ▼ 4.2.6 Lab – Lavorare con i file di testo nella CLI (versione per istruttore)
 - Obiettivi
 - Risorse necessarie
 - > Disposizioni
 - Riflessione

4.2.6 Lab – Lavorare con i file di testo nella CLI (ve

Obiettivi

In questo laboratorio, acquisirai familiarità con gli editor di testo della riga di comando e i file di configurazione di Linux.

- **Parte 1: Editor di testo grafico**
- **Parte 2: Editor di testo della riga di comando**
- **Parte 3: Lavorare con i file di configurazione**

Risorse necessarie

- Macchina virtuale CyberOps Workstation

Procediamo con il laboratorio avviando la VM CyberOps Workstation.

Passaggio 1: apri SciTE dalla GUI

un. Accedere alla VM CyberOps come **analista** utente utilizzando la password **cyberops**.

L'analista di account viene utilizzato come account utente di esempio in questo lab.

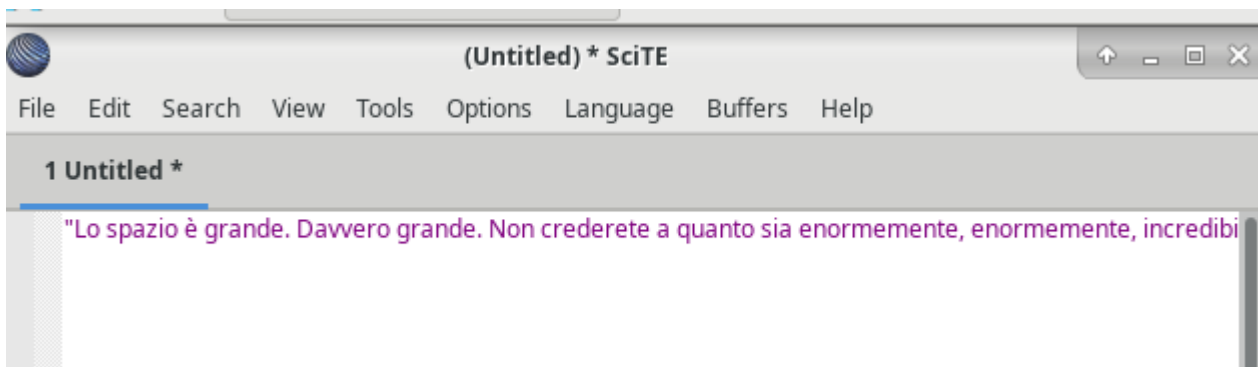
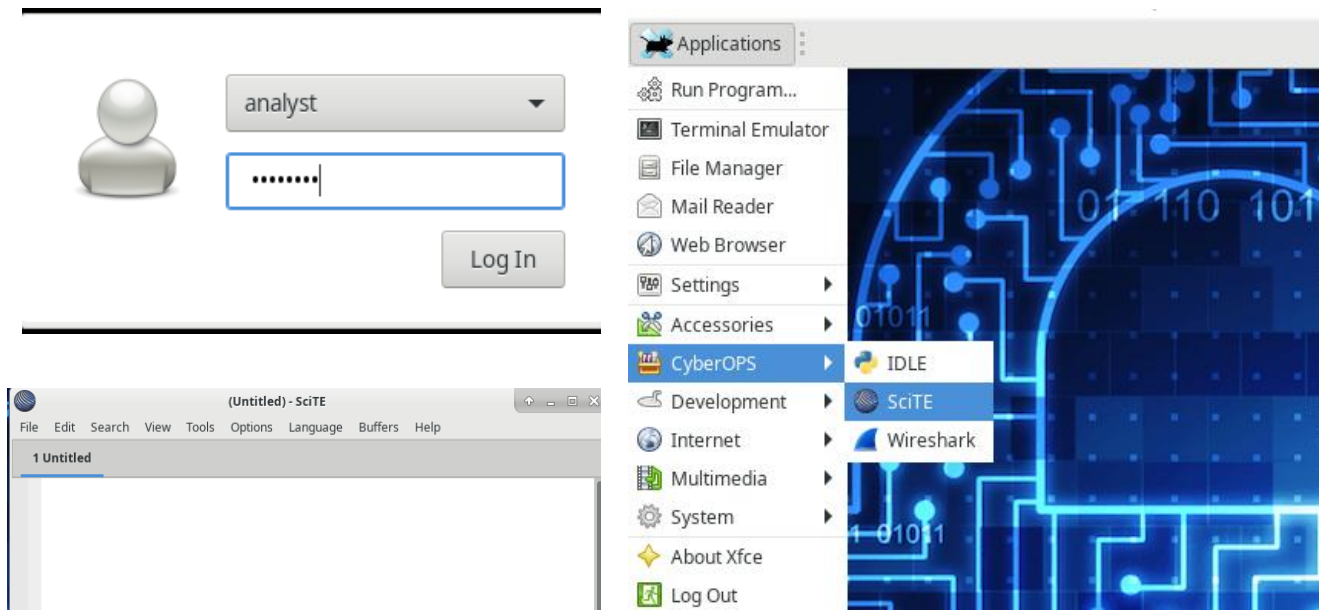
b. Nella barra in alto, vai su **Applicazioni > CyberOPS > SciTE** per avviare l'editor di testo **SciTE**.

c. **SciTE** è semplice ma include alcune importanti funzionalità: ambiente a schede, evidenziazione della sintassi e altro ancora. Trascorri qualche minuto con SciTE. Nell'area di lavoro principale, digita o copia e incolla il testo seguente:

"Lo spazio è grande. Davvero grande. Non crederete a quanto sia enormemente, enormemente, incredibilmente grande. Voglio dire, potresti pensare che la strada per il chimico sia ancora lunga, ma sono solo noccioline per lo spazio".

- Douglas Adams, Guida galattica per gli autostoppisti

Effettuiamo l'accesso come utente **analyst** e apriamo l'editor di testo **SciTE**. Copiamo e incolliamo il testo fornitoci.



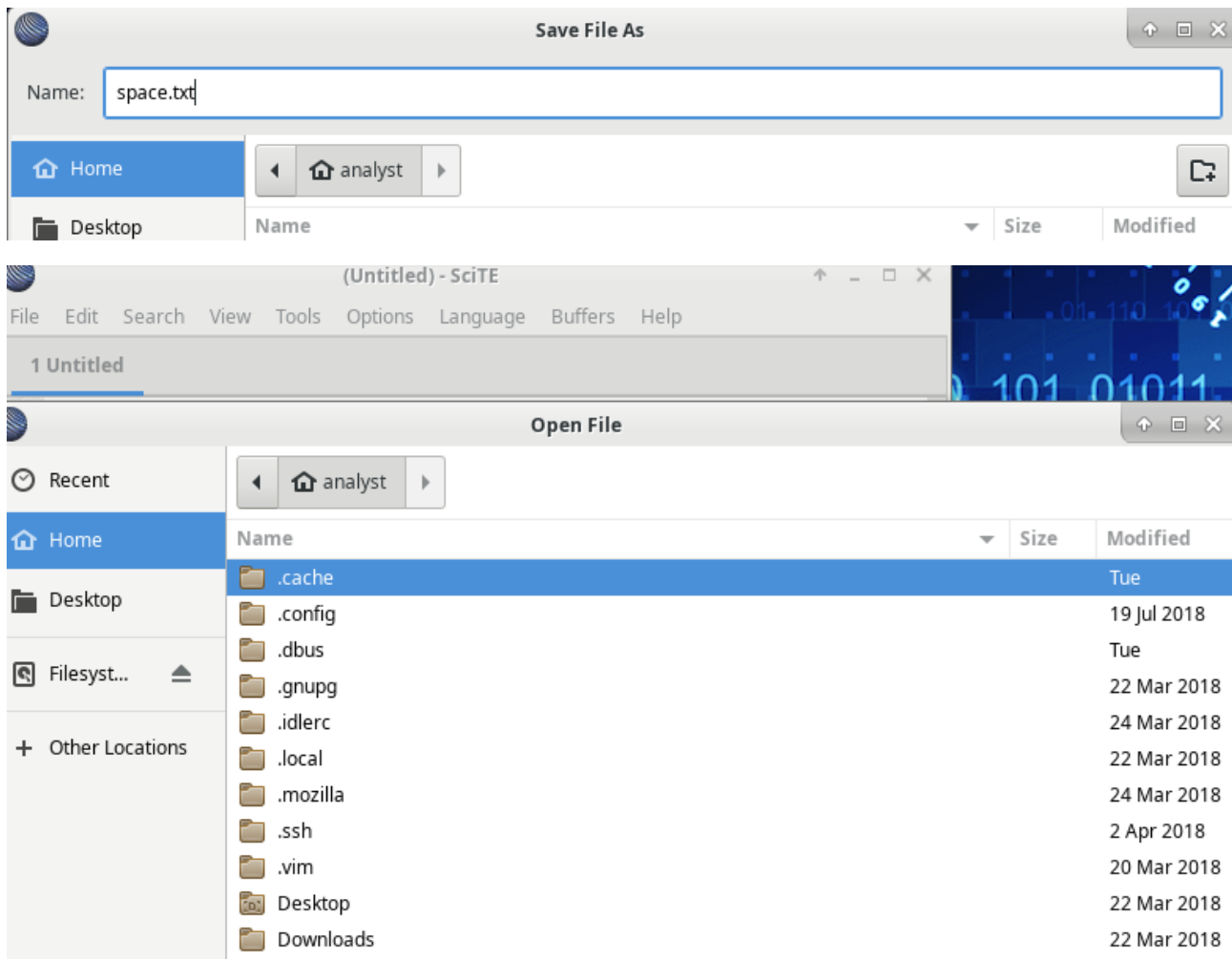
d. Fare clic su **File > Salva** per salvare il file. Si noti che **SciTE** tenta di salvare il file nella directory home dell'utente corrente, che è **analyst**, per impostazione predefinita. Assegna al file il **nome txt** e fai clic su **Salva**.

e. Chiudi **SciTE** facendo clic sull'icona **X** nella parte in alto a destra della finestra, quindi riapri **SciTE**.

f. Fare clic su **File > Apri...** e cercare il file appena salvato, **space.txt**.

Riusciresti a trovare subito **space.txt**?

Salviamo il file e chiudiamo SciTe. Quando lo riapriamo cerchiamo il file per riaprirlo e ci accorgiamo che non si trova facilmente.

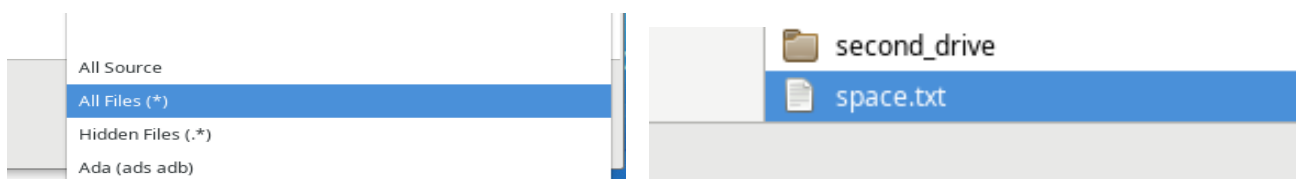


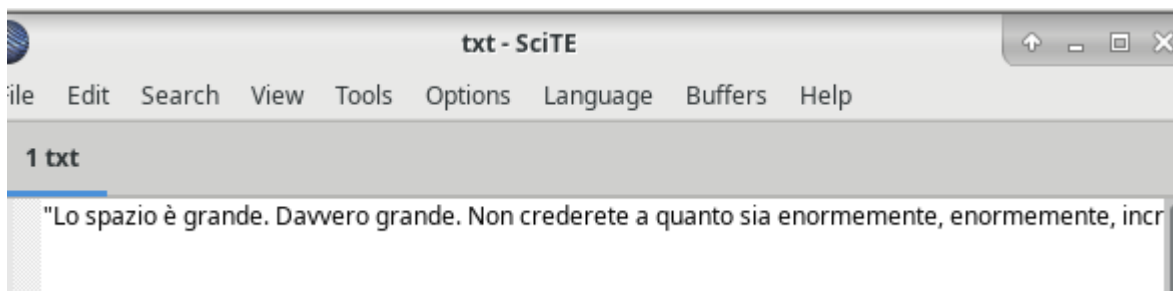
g. Anche se SciTE sta guardando la directory corretta (/home/analyst), space.txt non viene visualizzata. Questo perché SciTE è alla ricerca di estensioni note e .txt non è una di queste. Per visualizzare tutti i file, fare clic sul menu a discesa nella parte inferiore della finestra **Apri file** e selezionare **Tutti i file (*)**.

h. Seleziona **txt** per aprirlo.

Nota: Sebbene i file system Linux non si basino sulle estensioni, alcune applicazioni come **SciTE** potrebbero tentare di utilizzarle per identificare i tipi di file.

i. Chiudere space.txt al termine.





Passaggio 2: apri SciTE dal terminale.

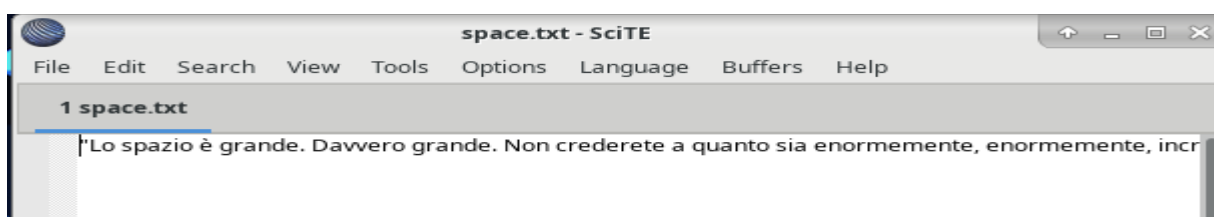
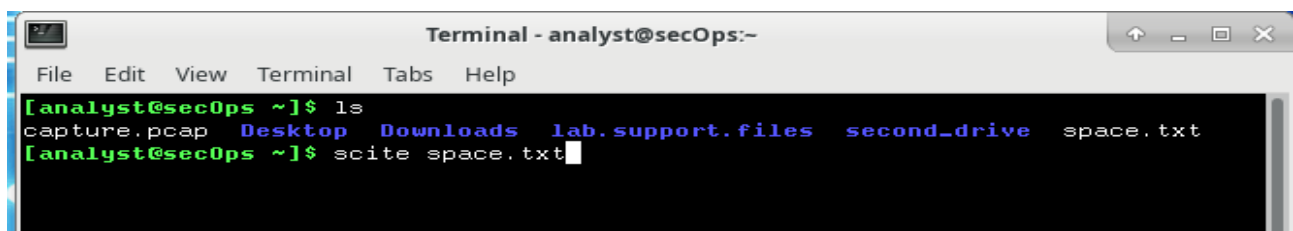
un. In alternativa, puoi anche aprire SciTE dalla riga di comando. Fai clic sull'icona del **terminale** situata nel Dock nella parte inferiore della scrivania. Si apre l'emulatore **di terminale**.

b. Digitare **ls** per visualizzare il contenuto della directory corrente. Si noti che **txt** è elencato. Ciò significa che non è necessario fornire informazioni sul percorso per aprire il file.

c. Digita **scite txt** per aprire **SciTE**. Si noti che questo non solo avvierà **SciTE** nella GUI, ma caricherà automaticamente anche il file di testo **space.txt** che è stato creato in precedenza.

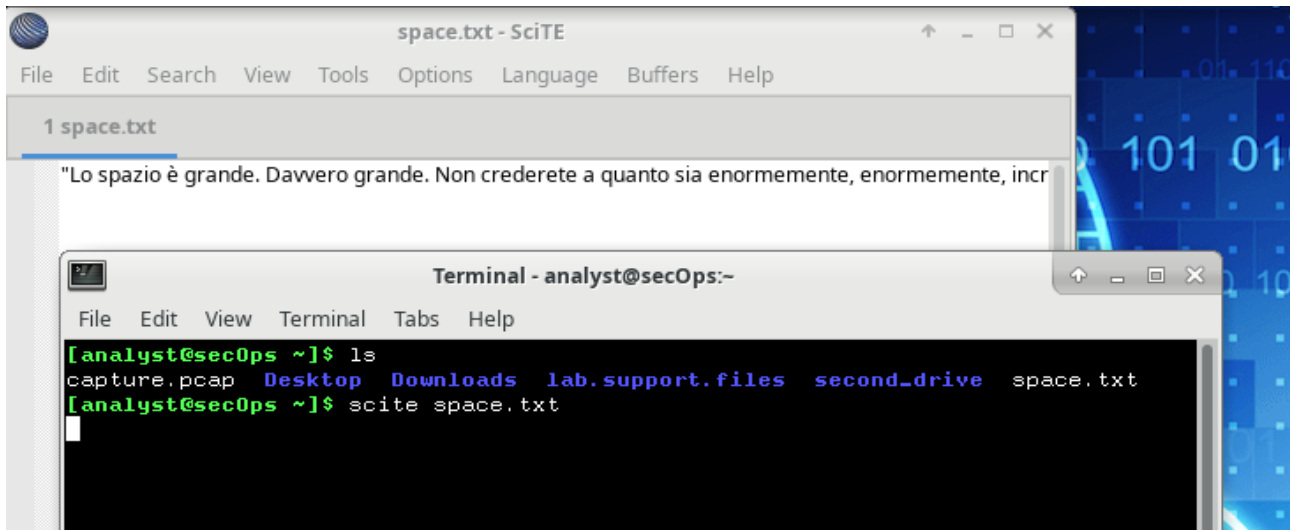
```
[analyst@secOps ~]$ scite space.txt
```

Procediamo con l'apertura tramite terminale. Apriamo quest'ultimo e con il comando **ls** visualizziamo la lista dei contenuti presenti in directory. A questo punto digitiamo **scite space.txt** per aprire il file direttamente con l'editor.



d. Si noti che mentre **SciTE** è aperto in primo piano, la finestra del terminale utilizzata per avviarlo è ancora aperta in background. Inoltre, si noti che la finestra del terminale utilizzata per avviare **SciTE** non visualizza più il prompt.

Perché il prompt non viene visualizzato nel terminale?



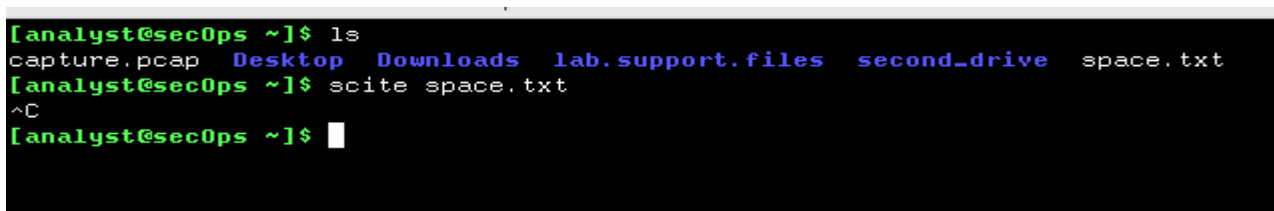
Effettivamente possiamo notare che non viene più visualizzato il prompt. Questo è dovuto dal fatto che il terminale sta eseguendo SciTe e non è in grado di ricevere altri comandi.

e. Chiudi questa istanza di **SciTE** facendo clic sull'icona X come prima o riportando lo stato attivo sulla finestra del terminale che ha avviato **SciTE** e interrompendo il processo. È possibile interrompere il processo premendo **CTRL+C**.

Nota: l'avvio di **SciTE** dalla riga di comando è utile quando si desidera eseguire **SciTE** come **root**. Basta far precedere **scite** con il comando **sudo**, **sudo scite**.

f. Chiudi **SciTE** e passa alla sezione successiva.

Con il comando **ctrl+c** il prompt è tornato visibile ed utilizzabile e la sessione SciTe si è chiusa autonomamente.



PARTE 2:

un. Nella finestra del terminale, digita **nano space.txt** per aprire il file di testo creato nella Parte 1.

```
[analyst@secOps ~]$ nano space.txt
```

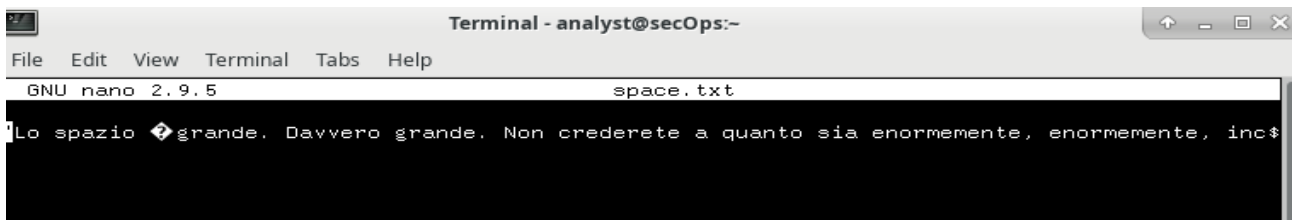
b. **nano** si avvierà e caricherà automaticamente il file di testo **txt**. Anche se il testo può sembrare troncato o incompleto, non lo è. Poiché il testo è stato creato senza caratteri di ritorno e il ritorno a capo non è abilitato, per impostazione predefinita **nano** visualizza una lunga riga di testo.

Utilizzare i tasti Home e Fine della tastiera per spostarsi rapidamente rispettivamente all'inizio e alla fine di una riga.

Quale carattere usa nano per rappresentare che una linea continua oltre i confini dello schermo?

Apriamo il file con **nano**. Questa versione utilizza il carattere \$ per farci capire che il testo continua oltre lo schermo.

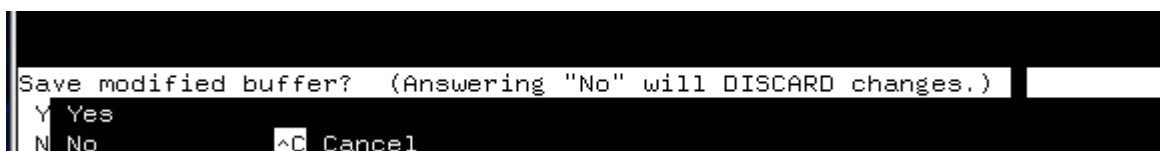
```
^C  
[analyst@secOps ~]$ nano space.txt
```



c. Come mostrato nelle righe di scelta rapida in basso, **CTRL+X** può essere utilizzato per uscire da **nano**. **nano** ti chiederà se vuoi salvare il file prima di uscire ('Y' per Sì o N per 'No'). Se si sceglie 'Y', verrà richiesto di premere Invio per accettare il nome del file specificato, o modificare il nome del file, o fornire un nome file se si tratta di un nuovo documento senza nome.

d. Per controllare **nano**, puoi usare **CTRL**, **ALT**, **ESCAPE** o i tasti META. Il tasto META è il tasto sulla tastiera con il logo Windows o Mac, a seconda della configurazione della tastiera.

Salviamo il file uscendo con il comando **ctrl+x** e digitiamo **Y** per salvare le modifiche e confermare.



PARTE 3:

Passaggio 1: individuazione dei file di configurazione

L'autore del programma definisce la posizione di configurazione per un determinato programma (servizio o applicazione). Per questo motivo, è necessario consultare la documentazione quando si valuta la posizione del file di configurazione. Convenzionalmente, tuttavia, in Linux, i file di configurazione utilizzati per configurare le applicazioni utente vengono spesso inseriti nella directory home dell'utente, mentre i file di configurazione utilizzati per controllare i servizi a livello di sistema vengono inseriti nella directory **/etc**. Gli utenti hanno sempre il permesso di scrivere nelle proprie directory home e sono in grado di configurare il comportamento delle applicazioni che utilizzano.

un. Utilizzare il comando per elencare tutti i file nella home directory **dell'analista**: `ls`

```
[analyst@secOps ~]$ ls -l
total 20
drwxr-xr-x 2 analyst analyst 4096 Mar 22 2018 Desktop
drwxr-xr-x 3 analyst analyst 4096 Apr 2 14:44 Downloads
drwxr-xr-x 9 analyst analyst 4096 Jul 19 2018 lab.support.files
drwxr-xr-x 2 analyst analyst 4096 Mar 21 2018 second_drive
-rw-r--r-- 1 analyst analyst 255 Apr 17 16:42 space.txt
```

Utilizziamo il comando **ls -l** per elencare i file nella directory **analyst**.

```
[analyst@secOps ~]$ ls -l
total 28
-rw-r--r-- 1 root root 7044 Feb 18 09:13 capture.pcap
drwxr-xr-x 2 analyst analyst 4096 Mar 22 2018 Desktop
drwxr-xr-x 3 analyst analyst 4096 Mar 22 2018 Downloads
drwxr-xr-x 9 analyst analyst 4096 Jul 19 2018 lab.support.files
drwxr-xr-x 2 analyst analyst 4096 Mar 21 2018 second_drive
-rw-r--r-- 1 analyst analyst 233 Feb 20 09:38 space.txt
[analyst@secOps ~]$
```

b. Usa di nuovo il comando, ma questa volta aggiungi l'opzione **-la** per includere anche i file nascosti nell'output: `ls`

```
[analyst@secOps ~]$ ls -la
total 144
drwx----- 14 analyst analyst 4096 Apr 17 16:34 .
drwxr-xr-x 3 root root 4096 Mar 20 2018 ..
-rw----- 1 analyst analyst 424 Apr 17 12:52 .bash_history
-rw-r--r-- 1 analyst analyst 21 Feb 7 2018 .bash_logout
```


Ripetiamo l'operazione ma stavolta tramite il comando `ls -la` per visualizzare anche i file nascosti.

```
[analyst@secOps ~]$ ls -la
total 160
drwx----- 15 analyst analyst 4096 Feb 20 09:12 .
drwxr-xr-x  3 root    root    4096 Mar 20 2018 ..
-rw-----  1 analyst analyst  235 Feb 18 11:05 .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
-rw-r--r--  1 analyst analyst  141 Feb  7 2018 .bashrc_stock
drwxr-xr-x  8 analyst analyst 4096 Feb 18 09:07 .cache
-rw-r--r--  1 root    root    7044 Feb 18 09:13 capture.pcap
drwxr-xr-x 10 analyst analyst 4096 Jul 19 2018 .config
drwx-----  3 analyst analyst 4096 Feb 18 09:07 .dbus
```

c. Utilizzare il comando per visualizzare il contenuto del file `.bashrc`. Questo file viene utilizzato per configurare il comportamento e la personalizzazione del terminale specifici dell'utente. `cat`

```
[analyst@secOps ~]$ cat .bashrc
export EDITOR=vim

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

```
[analyst@secOps ~]$ cat .bashrc
export EDITOR=vim

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

d. Mentre i file di configurazione relativi alle applicazioni utente sono convenzionalmente collocati nella home directory dell'utente, i file di configurazione relativi ai servizi a livello di sistema vengono collocati nella directory `/etc`, per convenzione. I servizi web, i servizi di stampa, i servizi ftp e i servizi di posta elettronica sono esempi di servizi che interessano l'intero sistema e i cui file di configurazione sono memorizzati in `/etc`. Si noti che gli utenti regolari non hanno accesso in scrittura a `/etc`. Questo è importante in quanto limita la possibilità di modificare la configurazione del servizio a livello di sistema solo all'utente `root`.

Utilizzare il comando per elencare il contenuto della directory `/etc`: `ls`

```
[analyst@secOps ~]$ ls /etc
adjtime             host.conf           mke2fs.conf         rc_maps.cfg
apache-ant          hostname            mkinitcpio.conf     request-key.c
apparmor.d          hosts               mkinitcpio.d        request-key.d
arch-release        ifplugd             modprobe.d           resolv.conf
```

```
[analyst@secOps ~]$ ls /etc
adjtime          gtk-3.0          mailcap
apparmor.d       healthd.conf    mail.rc
arch-release     host.conf        makepkg.conf
avahi            hostname         man_db.conf
bash.bash_logout hosts            mdadm.conf
bash.bashrc      ifplugd         mime.types
```

e. Utilizzare il comando per visualizzare il contenuto del file **bash.bashrc**: `cat`

```
[analyst@secOps ~]$ cat /etc/bash.bashrc
#
# /etc/bash.bashrc
#

# If not running interactively, don't do anything
[[ $- != *i* ]] && return
```

```
[analyst@secOps ~]$ cat /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:+$PROMPT_COMMAND; } 'printf "\033]0;%s@%s:%s\007" "${USER}"
    "${HOSTNAME%%.*}" "${PWD/#$HOME/\~}"'
  *)
    # Default PS1 for Bash
    PS1='\u@\h \w)\$ '
```

Perché i file di configurazione dell'applicazione utente vengono salvati nella directory home dell'utente e non in **/etc** con tutti gli altri file di configurazione a livello di sistema?

Gli utenti regolari non hanno il permesso di scrivere su **/etc**. Poiché Linux è un sistema operativo multiutente, l'inserimento dei file di configurazione dell'applicazione utente in **/etc** impedirebbe agli utenti di personalizzare le proprie applicazioni.

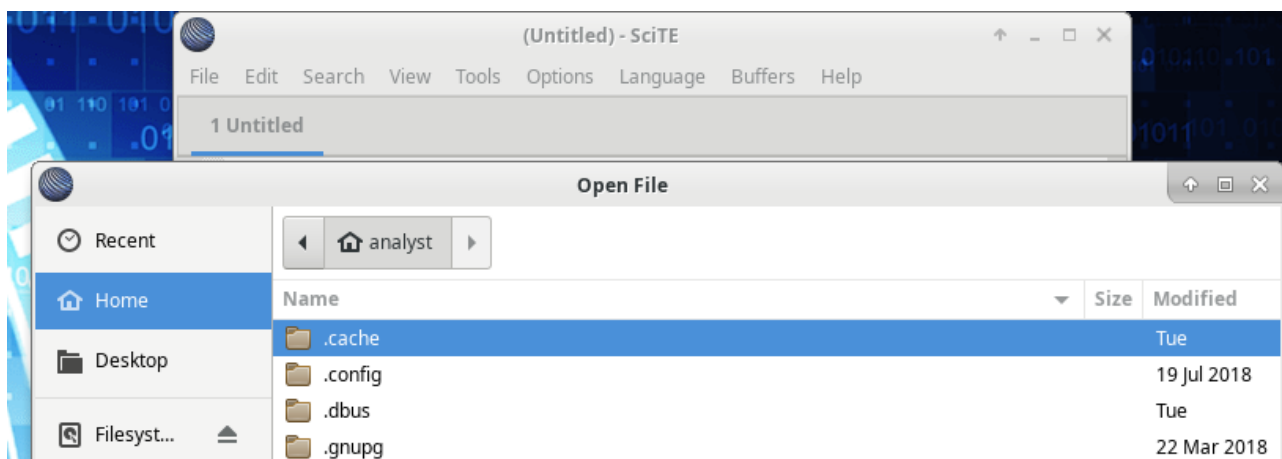
Passaggio 2: modifica e salvataggio dei file di configurazione

Come accennato in precedenza, i file di configurazione possono essere modificati con gli editor di testo.

Modifichiamo **.bashrc** per cambiare il colore del prompt della shell da verde a rosso per l'utente **analista**.

un. Innanzitutto, aprire **SciTE** selezionando **Applicazioni > CyberOPS > SciTE** dalla barra degli strumenti situata nella parte superiore della **macchina virtuale Cisco CyberOPS**

b. Seleziona **File > Apri** per avviare la finestra Apri file di **SciTE**.



c. Poiché **.bashrc** è un file nascosto senza estensione, SciTE non lo visualizza nell'elenco dei file. Se la funzione Posizione (Location) non è visibile nella finestra di dialogo, modificate il tipo di file visualizzato selezionando **Tutti i file (*)** dalla casella di riepilogo del tipo, come illustrato di seguito. Vengono visualizzati tutti i file nella home directory dell'analista.

d. Seleziona **.bashrc** e fai clic su **Apri**.

e. Individua 32 e sostituiscilo con 31. 32 è il codice colore del verde, mentre 31 rappresenta il rosso.



f. Salva il file selezionando **File > Salva** e chiudi **SciTE** facendo clic sulla **X**

Filesystem...	.bash_logout	41 bytes	7 Feb 2018
	.bash_profile	57 bytes	7 Feb 2018
	.bashrc	97 bytes	20 Mar 2018
	.bashrc_stock	141 bytes	7 Feb 2018
+ Other Locations	.cache		Tue
	.config		19 Jul 2018
	.dbus		Tue
	.dmrc	23 bytes	23 Mar 2018
	.gnupg		22 Mar 2018
	.ICEauthority	2.8 kB	08:56
	.idlcrc		24 Mar 2018
	.lessht	94 bytes	Tue
	.local		22 Mar 2018
	.mozilla		24 Mar 2018
	.ssh		2 Apr 2018
	.vboxclient-clipboard.pid	4 bytes	08:56
	.vboxclient-display.pid	4 bytes	08:56
	.vboxclient-draganddrop.pid	4 bytes	08:56

All Files (*)

Cancel Open

1 .bashrc

```
export EDITOR=vim

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

1 .bashrc *

```
export EDITOR=vim

PS1='\[\e[1;31m\][u@\h \W]\$\[\e[0m\] '
alias ls="ls --color"
alias vi="vim"
```

g. Fare clic sull'icona dell'applicazione Terminale situata sul Dock, in basso al centro della **macchina virtuale Cisco CyberOPS**. Il prompt dovrebbe essere visualizzato in rosso anziché in verde.

Anche la finestra del terminale che era già aperta ha cambiato colore da verde a rosso? Spiegare.

```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
gtk-2.0 machine-id pam.d skel
[analyst@secOps ~]$ cat /etc/bash.bashrc
# /etc/bash.bashrc
#
# If not running interactively, don't do anything
[[ $- != *i* ]] && return

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

PS1='\[\e[1;32m\][u@\h \W]\$\[\e[0m\] '

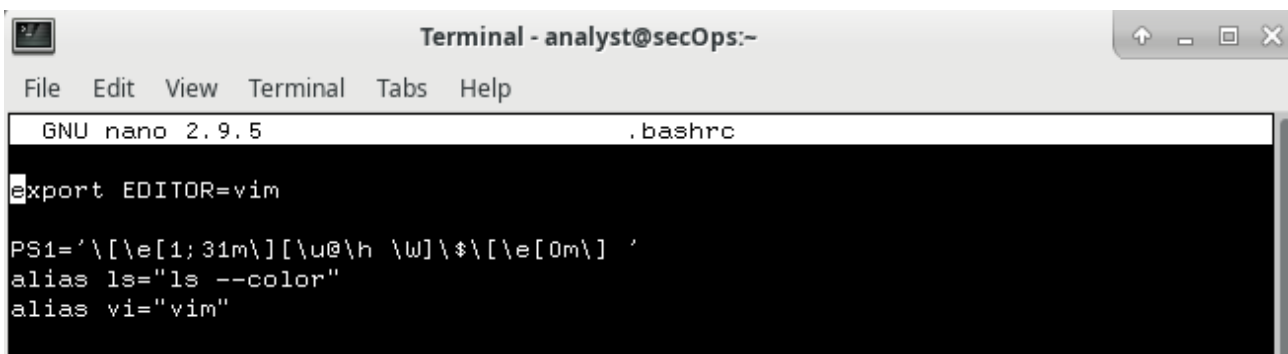
case ${TERM} in
  xterm*|rxvt*|Eterm|aterm|kterm|gnome*)
    PROMPT_COMMAND=${PROMPT_COMMAND:+$PROMPT_COMMAND;} 'printf "\033[0;32m${HOSTNAME%*.} ${PWD/#$HOME/\~}"'
    ;;
  screen*)
    PROMPT_COMMAND=${PROMPT_COMMAND:+$PROMPT_COMMAND;} 'printf "\033[0;31m${HOSTNAME%*.} ${PWD/#$HOME/\~}"'
    ;;
  *)
    ;;
esac

[ -r /usr/share/bash-completion/bash_completion ] && . /usr/share/bash-completion/bash_completion
[analyst@secOps ~]$
```

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

No. Il file `.bashrc` viene eseguito e applicato quando un terminale viene aperto per la prima volta, quindi tutti i terminali aperti in precedenza non saranno influenzati dalle modifiche al file `.bashrc`.

f. La stessa modifica avrebbe potuto essere effettuata dalla riga di comando con un editor di testo come **nano**. Da una nuova finestra del terminale, digita **nano .bashrc** per avviare **nano** e caricare automaticamente il file `.bashrc` al suo interno:

A screenshot of a terminal window titled "Terminal - analyst@secOps:-". The terminal is running the GNU nano 2.9.5 editor, editing the file .bashrc. The editor's menu bar shows "File Edit View Terminal Tabs Help". The content of the file is as follows:

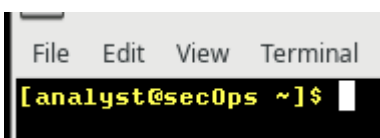
```
export EDITOR=vim

PS1='\[\e[1;31m\]\[\u@\h \w\]\$'\[\e[0m\] '
```

i. Modifica da 31 a 33. 33 è il codice colore del giallo.

j. Premere **CTRL+X** per salvare, quindi premere **Y** per confermare. L'editor di testo **nano** ti offrirà anche la possibilità di cambiare il nome del file. Basta premere **INVIO** per utilizzare lo stesso nome, `.bashrc`.

Okay. L'editor di testo **nano** terminerà e tornerai al prompt della shell. Questa volta ricarica il terminale bash inserendo il comando **bash** nel terminale. Il prompt dovrebbe ora apparire in giallo anziché in rosso.

A screenshot of a terminal window showing the shell prompt. The menu bar shows "File Edit View Terminal". The prompt is displayed in yellow text on a black background:

```
[analyst@secOps ~]$
```