



REPORT TECNICO

Lavorare con File di Testo nella CLI e la Shell Linux

Redatto da: *Nicolò Calì Cybersecurity Student*

Data: *19/02/2026*

Oggetto: *Utilizzo di editor di testo e gestione dei file di configurazione in ambiente Linux*

1. Introduzione

Il presente report documenta l'attività pratica focalizzata sull'utilizzo degli editor di testo in ambiente Linux, impiegando sia strumenti dotati di interfaccia grafica, come **SciTE**, sia strumenti basati su riga di comando, come **nano**. L'obiettivo principale dell'esercitazione è acquisire familiarità con questi applicativi per la creazione, la gestione e la modifica dei file di sistema e di configurazione.

L'ambiente di test utilizzato per l'intera durata del laboratorio è la macchina virtuale **CyberOps Workstation** basata su Linux.

2. Utilizzo degli Editor di Testo Grafici

La prima fase dell'attività ha previsto l'esplorazione dell'editor di testo grafico **SciTE**. L'applicativo è stato avviato dall'interfaccia Desktop della **VM CyberOps**. All'interno dell'area di lavoro è stato inserito un testo di prova, successivamente salvato all'interno della directory home dell'utente (**/home/analyst**) con il nome "**space.txt**".

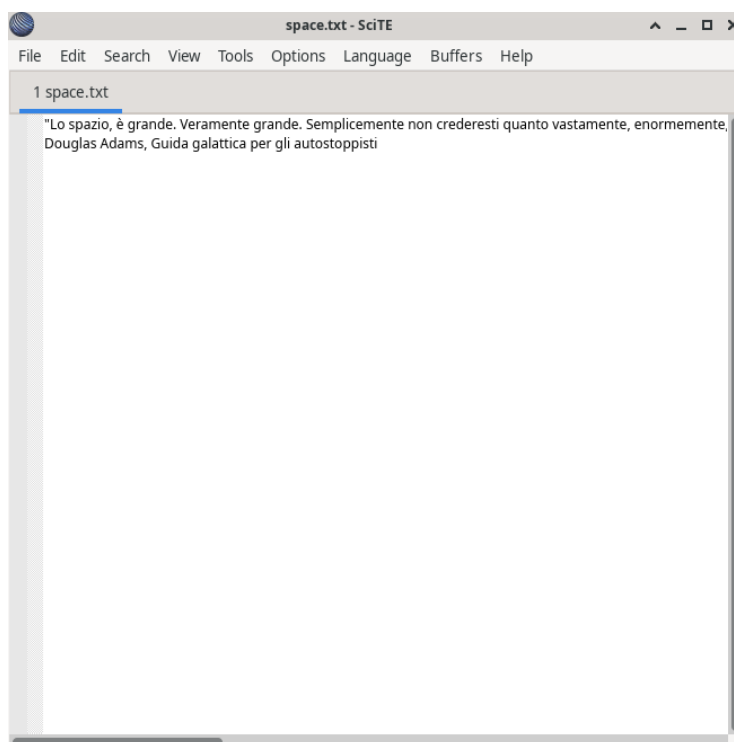


Fig. 1 Inserimento del testo e salvataggio del file space.txt tramite SciTE.

2.1 Avvio dell'editor da riga di comando

L'applicativo SciTE è stato successivamente richiamato tramite l'emulatore di terminale. Utilizzando il comando "**scite space.txt**", l'editor grafico è stato lanciato caricando automaticamente il file di testo. Si è potuto osservare il comportamento del processo lanciato in primo piano (**foreground**): la finestra del terminale utilizzata per l'avvio rimane in attesa e non mostra il prompt dei comandi fino alla terminazione dell'interfaccia grafica, eseguibile anche tramite l'interruzione del processo (**CTRL+C**).

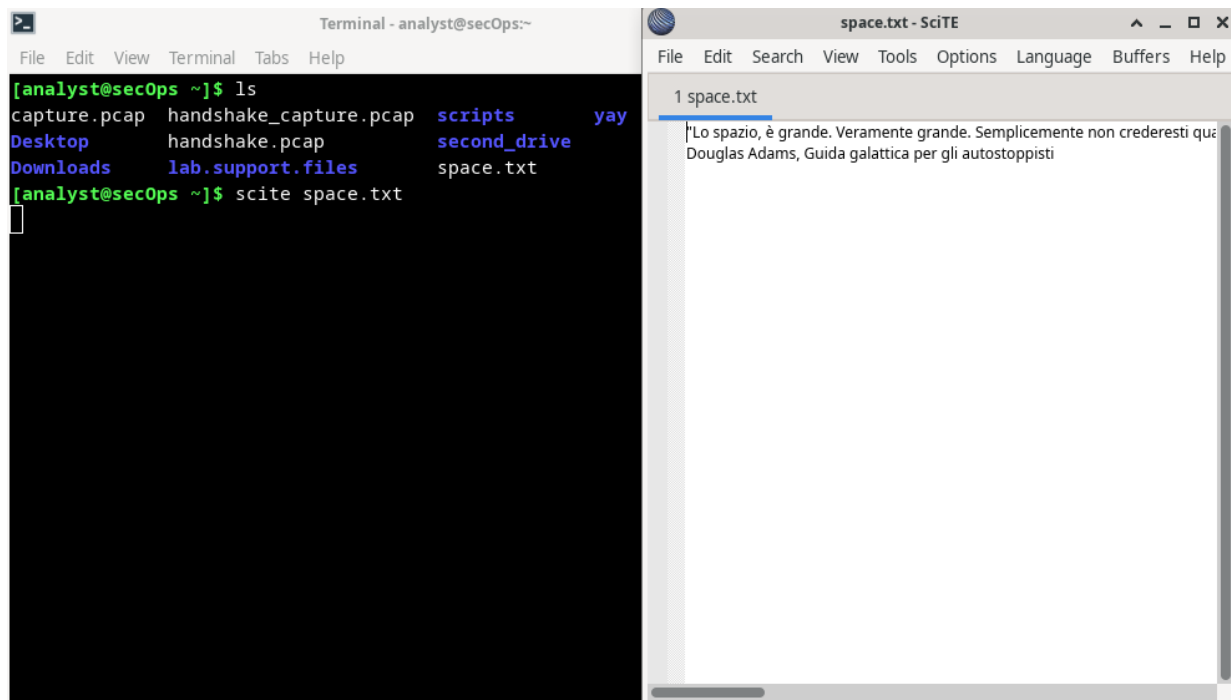


Fig. 2 Avvio dell'editor grafico SciTE tramite riga di comando.

3. Utilizzo degli Editor da Riga di Comando

La seconda parte dell'esercitazione si è concentrata sugli editor basati esclusivamente su interfaccia a riga di comando (**CLI**), strumenti essenziali per l'amministrazione di sistemi privi di interfaccia grafica.

Utilizzando l'emulatore di terminale, è stato lanciato l'applicativo **GNU nano** per aprire il file "space.txt" precedentemente creato tramite il comando: "**nano space.txt**".

È stato possibile osservare come l'intera interfaccia dell'editor venga renderizzata all'interno della shell stessa, richiedendo l'utilizzo esclusivo di scorciatoie da tastiera (es. **CTRL+X** per uscire, **CTRL+O** per salvare) per la navigazione e la gestione del documento.

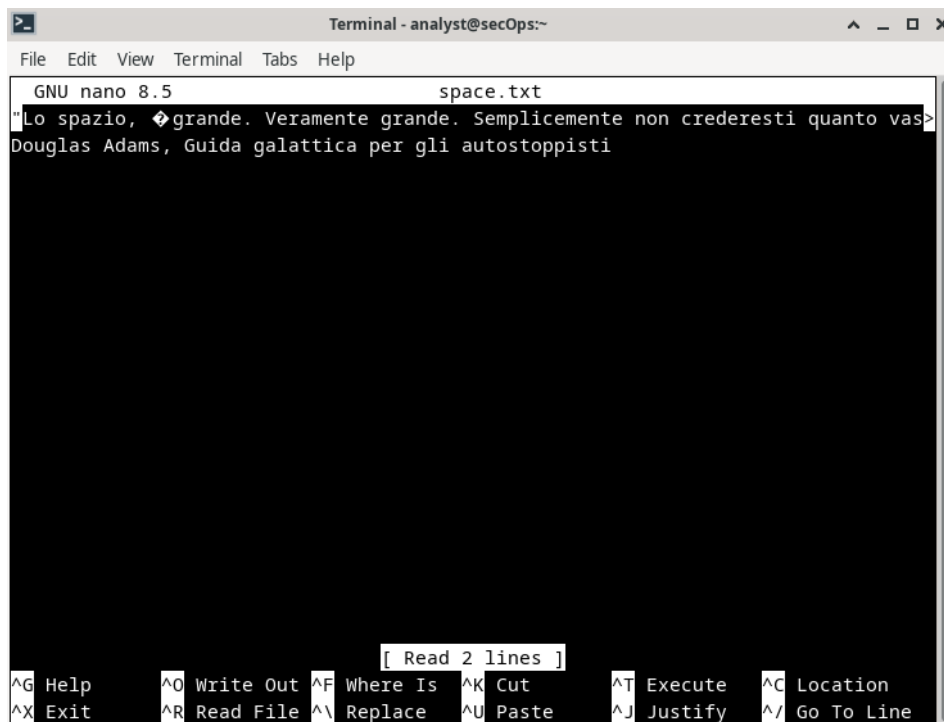


Fig. 3 Visualizzazione e modifica del file di testo tramite l'editor a riga di comando GNU nano.

4. Lavorare con i File di Configurazione

Il laboratorio è proseguito con l'analisi dei file di configurazione, essenziali per definire il comportamento di applicazioni e servizi in ambiente Linux. Tramite il comando "**ls -la**" è stato possibile visualizzare i file nascosti all'interno della directory **home** dell'utente, i quali, per convenzione, presentano un punto come prefisso nel nome (es. **.bashrc**). Utilizzando il comando "**cat**", è stato ispezionato il contenuto del file "**.bashrc**" (*configurazione utente locale*) e successivamente il file **"/etc/bash.bashrc"** (*configurazione a livello di sistema*), evidenziando la distinzione architettonica tra le impostazioni personalizzate e quelle globali.

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

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

Fig. 4 Lettura del file di configurazione nascosto **.bashrc** tramite il comando **cat**.

4.1 Modifica dei file di configurazione

Per comprendere l'impatto pratico delle configurazioni utente, è stata effettuata una modifica al file **".bashrc"** utilizzando l'editor **SciTE**. Nello specifico, si è intervenuti sulla variabile d'ambiente **"PS1"**, responsabile della formattazione del prompt dei comandi.

Il codice colore ANSI predefinito **"32" (verde)** è stato sostituito con **"31" (rosso)**. In seguito al salvataggio del file e all'avvio di una nuova sessione del terminale, è stato possibile osservare l'immediata applicazione della nuova direttiva grafica.

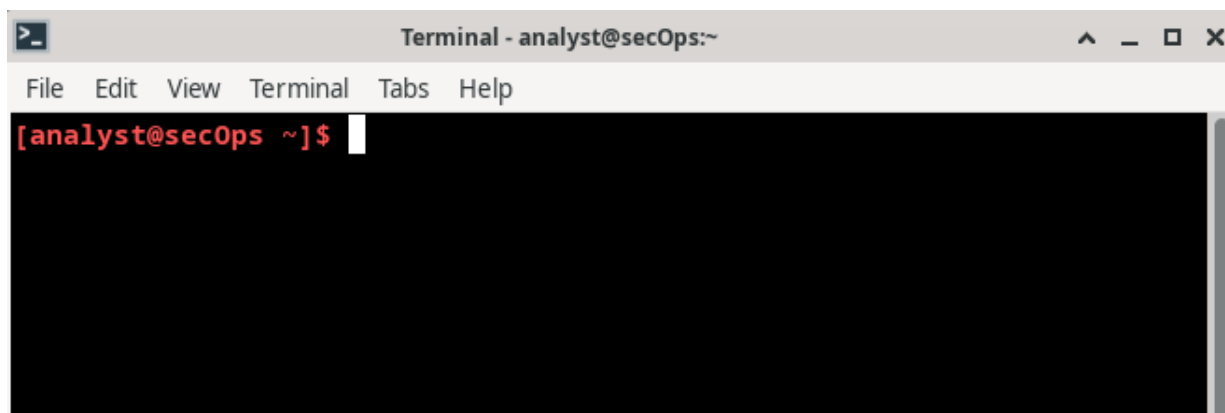


Fig. 5 Modifica del file .bashrc e conseguente variazione del colore del prompt in rosso.

4.2 Modifica dei file di configurazione tramite CLI

A scopo comparativo, la medesima operazione di personalizzazione del prompt è stata eseguita utilizzando l'editor a riga di comando **GNU nano**.

Il file ".bashrc" è stato aperto direttamente dal terminale, e il codice colore della variabile "PS1" è stato alterato dal valore **"31" (rosso)** al valore **"33" (giallo)**.

Una volta salvato il documento, le nuove direttive sono state applicate alla sessione corrente avviando una nuova istanza della shell tramite l'esecuzione del comando **"bash"**, confermando l'avvenuto cambio cromatico senza la necessità di riavviare l'emulatore di terminale.

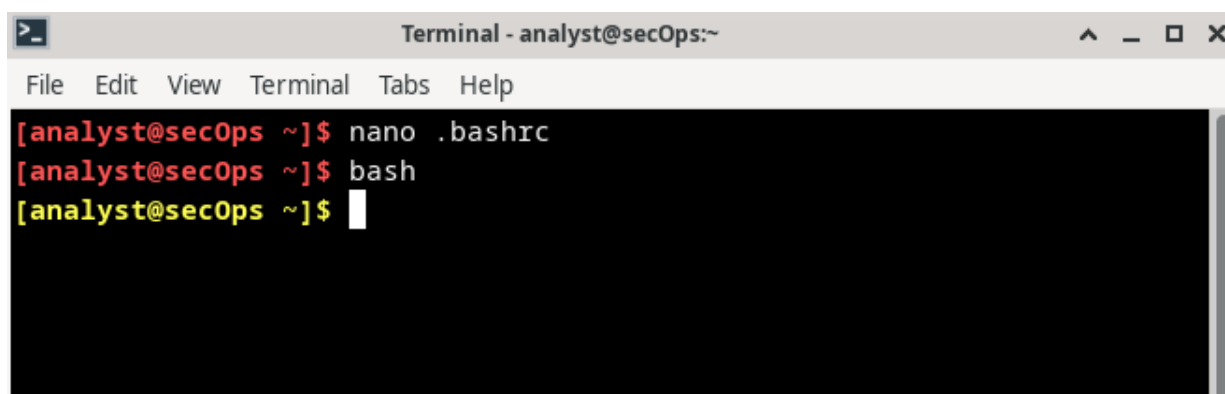


Fig. 6 - Modifica del file .bashrc tramite nano.

4.3 Modifica dei file di configurazione dei servizi di sistema

La fase conclusiva dell'analisi dei file di configurazione ha riguardato i servizi a livello di sistema, prendendo in esame il server web leggero "**nginx**".

Essendo i file globali ospitati nella directory "**/etc**", è stato necessario operare con privilegi di root tramite il comando "**sudo**".

Utilizzando "**sudo nano -l /etc/nginx/custom_server.conf**", il file è stato aperto con la numerazione delle righe attiva.

Sono state modificate la **porta** di ascolto (*da 81 a TCP 8080*) e la **web root directory** (*impostata su /usr/share/nginx/html/text_ed_lab/*).

```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
GNU nano 8.5 /etc/nginx/custom_server.conf Modified
32
33 #gzip on;
34
35 types_hash_max_size 4096;
36 server_names_hash_bucket_size 128;
37
38 server {
39 listen 8080;
40 server_name localhost;
41
42 #charset koi8-r;
43
44 #access_log logs/host.access.log main;
45
46 location / {
47 root /usr/share/nginx/html/text_ed_lab/;
48 index index.html index.htm;
49 }
50
51 #error_page 404 /404.html;
^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line
```

Fig 7 Modifica della configurazione del web server nginx.

Al termine del salvataggio, il servizio **nginx** è stato istanziato puntando al file customizzato, e il corretto funzionamento è stato validato raggiungendo la pagina web locale tramite browser all'indirizzo **127.0.0.1:8080**.

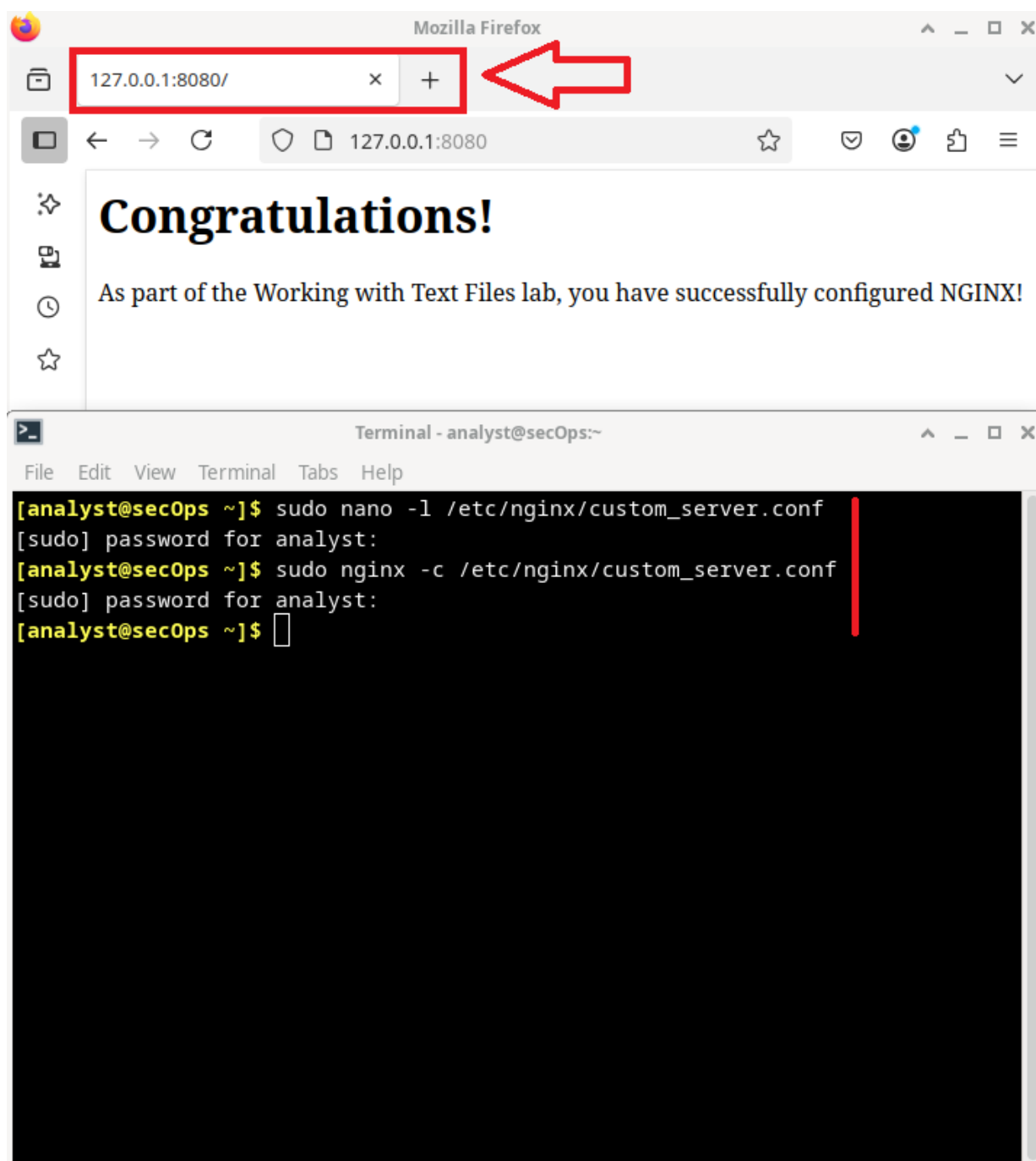


Fig. 8 Verifica dell'avvio tramite connessione in localhost (porta 8080).

4.4 Terminazione del servizio di rete

Al fine di verificare la corretta gestione del ciclo di vita del servizio, si è proceduto allo spegnimento del server web precedentemente istanziato.

Utilizzando l'utility "**pkill**" con privilegi elevati (comando "**sudo pkill nginx**"), il processo è stato terminato inviando l'apposito segnale di arresto all'eseguibile.

L'effettiva disattivazione del demone è stata confermata tentando un nuovo accesso tramite browser all'indirizzo locale (**127.0.0.1:8080**), operazione che si è conclusa con un fisiologico fallimento della connessione, attestando l'assenza del servizio in ascolto.

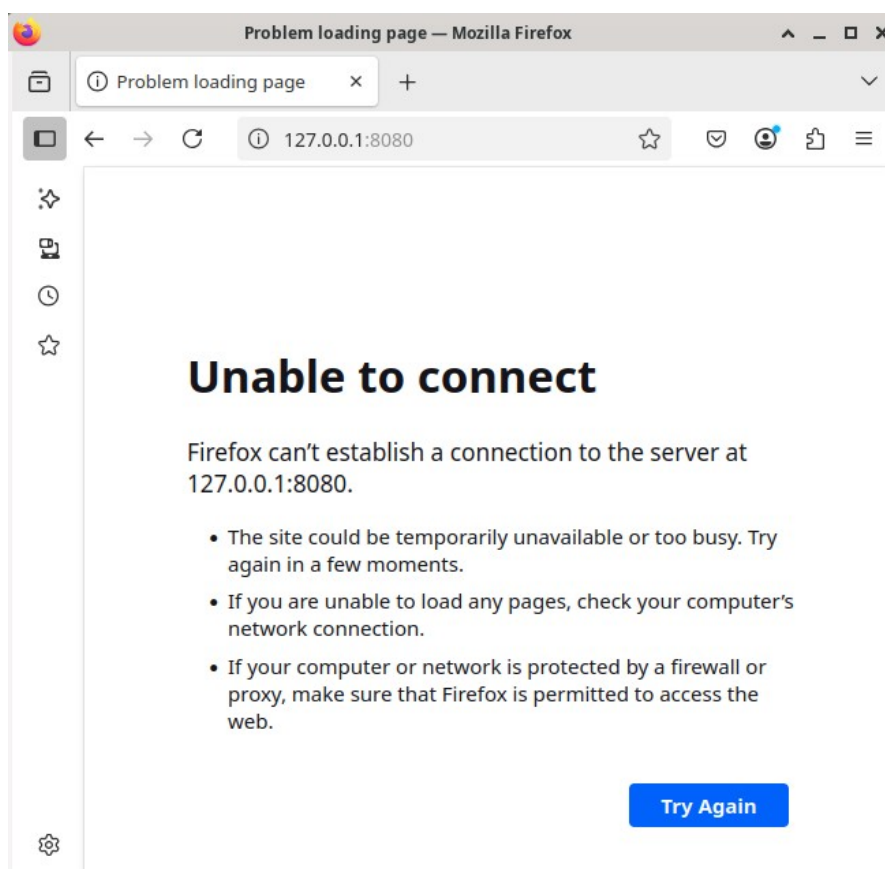


Fig. 8 Verifica dello spegnimento del demone nginx tramite fallimento della connessione.

5. Conclusioni

L'attività di laboratorio ha permesso di consolidare le competenze relative all'utilizzo degli editor di testo in ambiente Linux, sia in modalità grafica (**SciTE**) che a riga di comando (**GNU nano**).

Si è inoltre approfondita la gestione dei file di configurazione, comprendendo la netta separazione architetturale e di sicurezza tra le personalizzazioni del singolo utente (file nascosti nella directory **home**) e le configurazioni globali dei servizi (directory **/etc**).

5.1 Risposte ai Quesiti di Analisi

Quesito 1: Perché il prompt non viene mostrato nel terminale quando si avvia SciTE da riga di comando?

Risposta: Quando si lancia un applicativo grafico dal terminale senza specificare l'esecuzione in background, il processo viene eseguito in primo piano (**foreground**). Il terminale rimane in attesa della terminazione del processo figlio (**SciTE**) prima di restituire il controllo all'utente e mostrare nuovamente il prompt.

Quesito 2: Quale carattere usa nano per rappresentare che una linea continua oltre i bordi dello schermo?

Risposta: L'editor nano utilizza il carattere speciale ">" per indicare che la riga di testo continua oltre l'area visibile del terminale.

Quesito 3: Perché i file di configurazione delle applicazioni utente sono salvati nella directory home dell'utente e non sotto /etc con tutti gli altri file di configurazione a livello di sistema?

Risposta: Per un fondamentale principio di sicurezza e gestione dei privilegi. La directory /etc contiene configurazioni che influenzano l'intero sistema ed è modificabile solo dall'utente **root**. I file nella directory **home** influenzano esclusivamente il singolo utente, garantendo che quest'ultimo possa personalizzare il proprio ambiente senza rischiare di compromettere il sistema operativo.

Quesito 4: Dopo aver modificato il file .bashrc, anche la finestra del terminale che era già aperta ha cambiato colore da verde a rosso? Spiega.

Risposta: No, la vecchia finestra non ha cambiato colore. Il file di configurazione .bashrc viene letto ed eseguito dalla shell Bash esclusivamente durante la fase di avvio della sessione (**startup**). Le finestre già aperte mantengono in cache i parametri caricati al momento della loro inizializzazione.

Quesito 5: Puoi modificare il file `/etc/nginx/custom_server.conf` con SciTE? Descrivi il processo.

Risposta: Sì, è possibile. Poiché il file risiede in una directory protetta (`/etc`), l'editor grafico standard non disporrebbe dei permessi di scrittura. Tuttavia, avviando l'editor dal terminale e antepoendo il comando per l'elevazione dei privilegi, è possibile operare regolarmente.

Il comando da eseguire è: **"`sudo scite /etc/nginx/custom_server.conf`".**