Laboratório - Trabalhando com arquivos de texto na CLI

# Objetivos

Neste laboratório, você se familiarizará com editores de texto de linha de comando Linux e arquivos de configuração.

Parte 1: Editores Gráficos de Texto

Parte 2: Editores de texto de linha de comando

Parte 3: Trabalhando com Arquivos de Configuração

# Recursos necessários

=    Máquina virtual Security Workstation.

# Instruções

## Parte 1: Editores de Texto Gráficos

Antes de poder trabalhar com arquivos de texto no Linux, você deve estar familiarizado com editores de texto.

Os editores de texto são uma das categorias mais antigas de aplicativos criados para computadores. Linux, como muitos outros sistemas operacionais, tem vários editores de texto diferentes, com vários recursos e funções. Alguns editores de texto incluem interfaces gráficas, enquanto outros só podem ser usados através da linha de comando. Cada editor de texto inclui um conjunto de recursos projetado para suportar um cenário de trabalho específico. Alguns editores de texto se concentram no programador e incluem recursos como realce de sintaxe, correspondência de colchete, encontrar e substituir, suporte a Regex multi-linha, verificação ortográfica e outros recursos focados na programação.

Para economizar espaço e manter a máquina virtual enxuta, a **VM Cisco CyberOps Workstation** inclui apenas **ScITE** como um aplicativo de editor de texto gráfico. O **SCite** é um editor de texto simples, pequeno e rápido. Ele não tem muitos recursos avançados, mas suporta totalmente o trabalho realizado neste curso.

**Nota**: A escolha do editor de texto é pessoal. Não existe tal coisa como um melhor editor de texto. O melhor editor de texto é aquele com o qual você se sente mais confortável e funciona melhor para você.

### Etapa 1: Abrir SCite a partir da GUI

a.     Faça logon na Security Workstation VM como **analyst** , usando a senha **cyberops**. A conta **analyst** é usada como exemplo de conta de usuário neste laboratório.

b.     Na barra superior, navegue até **Aplicativos > CyberOps > ScITE** para iniciar o editor de texto **ScITE** .

c.     **O SCite** é simples, mas inclui alguns recursos importantes: ambiente com abas, realce de sintaxe e muito mais. Passe alguns minutos com SCite. Na área de trabalho principal, digite ou copie e cole o texto abaixo:

“O espaço é grande. Realmente grande. Você só não vai acreditar o quão grande, imensamente, incrivelmente grande é. Quero dizer, você pode pensar que é um longo caminho até o químico, mas isso é apenas amendoins para o espaço.”

― Douglas Adams, O Guia do Caroneiro para a Galáxia

d.     Clique em **Arquivo** > **Salvar** para salvar o arquivo. Observe que o **ScITE** tenta salvar o arquivo no diretório home do usuário atual, que é o analyst, por padrão. Nomeie o arquivo **space.txt** e clique em **Salvar**.

e.     Feche o **SciTE** clicando no ícone **X** no lado superior direito da janela e, em seguida, reabra o **SciTE**.

f.       Clique em **Arquivo** > **Abrir...** e procure o arquivo recém-salvo, **space.txt**.

#### Pergunta:

Você poderia encontrar imediatamente space.txt?

***Área de Resposta***

g.     Mesmo que SciTE esteja olhando para o diretório correto (/home/analyst), space.txt não é exibido. Isso ocorre porque o SciTE está procurando extensões conhecidas e .txt não é uma delas. Para exibir todos os arquivos, clique no menu suspenso na parte inferior da janela **Abrir Arquivo** e selecione **Todos os arquivos (\*)**.

h.     Selecione **space.txt** para abri-lo.

**Observação**: Embora os sistemas de arquivos Linux não dependam de extensões, alguns aplicativos, como o SciTE, podem tentar usá-los para identificar tipos de arquivos.

i.       Feche space.txt quando terminar.

### Etapa 2: Abra o SciTE a partir do Terminal.

a.     Alternativamente, você também pode abrir o SciTE a partir da linha de comando. Clique no ícone do **terminal** localizado na Dock, na parte inferior da área de trabalho. O emulador de **terminal** é aberto.

b.     Digite **ls** para ver o conteúdo do diretório atual. Observe que **space.txt** está listado. Isso significa que você não precisa fornecer informações de caminho para abrir o arquivo.

c.     Digite **scite** **space.txt** para abrir o **SciTE**. Observe que isso não só iniciará o **SciTE** na GUI, mas também carregará automaticamente o arquivo de texto space.txt que foi criado anteriormente.

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

d.     Observe que, enquanto o **SciTE** está aberto em primeiro plano, a janela do terminal usada para iniciá-lo ainda está aberta em segundo plano. Além disso, observe que a janela do terminal usada para iniciar o **SciTE** não exibe mais o prompt.

#### Pergunta:

Por que o prompt não é mostrado no terminal?

***Área de Resposta***

e.     Feche essa instância do **SciTE** clicando no ícone X como antes ou alternando o foco de volta para a janela de terminal que iniciou o **SciTE** e interrompendo o processo. Você pode parar o processo pressionando **CTRL+C**.

**Observação**: Iniciar o **SciTE** a partir da linha de comando é útil quando você deseja executar o **SciTE** como **root**. Basta preceder **scite** com o **comando sudo** , **sudo scite**.

f.       Feche o **SciTE** e passe para a próxima seção.

## Parte 2: Editores de texto de linha de comando

Embora os editores de texto gráficos sejam convenientes e fáceis de usar, os editores de texto baseados em linha de comando são muito importantes em computadores Linux. O principal benefício dos editores de texto baseados em linha de comando é que eles permitem a edição de arquivos de texto a partir de um shell remoto em um computador remoto.

Considere o seguinte cenário. Um usuário deve executar tarefas administrativas em um computador Linux, mas não está sentado na frente desse computador. Usando **SSH**, o usuário inicia um shell remoto para o computador acima mencionado. Sob o shell remoto baseado em texto, a interface gráfica pode não estar disponível, o que torna impossível confiar em editores de texto gráficos. Neste tipo de situação, editores de texto baseados em texto são cruciais.

**Nota**: Isto é principalmente verdadeiro quando se conecta a servidores remotos sem cabeça que não possuem uma interface GUI.

A **VM do Cisco CyberOps Workstation** inclui alguns editores de texto baseados em linha de comando. Este curso centra-se em **nano**.

**Nota**: Outro editor de texto extremamente popular é chamado **vi**.Embora a curva de aprendizagem para **vi** seja considerada íngreme, **vi** é um editor de texto baseado em linha de comando muito poderoso. Ele está incluído por padrão em quase todas as distribuições Linux e seu código original foi criado pela primeira vez em 1976. Uma versão atualizada do **vi** é chamado vim que significa vi-improved. Hoje a maioria dos usuários do **vi** estão realmente usando a versão atualizada, vim.

Devido à falta de suporte gráfico, nano(ou GNU nano) pode ser controlado apenas através do teclado. **CTRL+O** salva o arquivo atual; **CTRL+W** abre o menu de pesquisa. GNU nano usa uma barra de atalho de duas linhas na parte inferior da tela, onde vários comandos para o contexto atual são listados. Depois que nano estiver aberto, pressione **CTRL+G** para a tela de ajuda e uma lista completa.

a.     Na janela do terminal, digite **nano space.txt** para abrir o arquivo de texto criado na Parte 1.

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

b.     nano iniciará e carregará automaticamente o arquivo de texto space.txt. Embora o texto possa parecer truncado ou incompleto, não é. Como o texto foi criado sem caracteres de retorno e a quebra de linha não está habilitada, por padrão, nano está exibindo uma longa linha de texto.

Use as teclas de teclado Início e End para navegar rapidamente até o início e o fim de uma linha, respectivamente.

Que personagem nano usa para representar que uma linha continua além dos limites da tela?

***Área de Resposta***

c.     Como mostrado nas linhas de atalho inferiores, **CTRL+X** pode ser usado para sair do **nano**. **nano** perguntará se você deseja salvar o arquivo antes de sair ('Y' para Sim, ou N para 'Não'). Se 'Y' for escolhido, você será solicitado a pressionar Enter para aceitar o nome do arquivo fornecido, ou alterar o nome do arquivo, ou fornecer um nome de arquivo se for um novo documento sem nome.

d.     Para controlar **nano**, você pode usar **CTRL**, **ALT**, **ESCAPE**ou as teclas META. A tecla META é a tecla do teclado com um logotipo do Windows ou Mac, dependendo da configuração do teclado.

Navegação em nano é muito amigável. Use as setas para mover os arquivos. Page Up e Page Down também podem ser usados para avançar ou retroceder páginas inteiras. Passe algum tempo com nano e sua tela de ajuda. Para entrar na tela de ajuda, pressione **CTRL+G**. Pressione **q** para sair da tela de ajuda e retornar à edição de documentos em nano.

## Parte 3: Trabalhando com Arquivos de Configuração

No Linux, tudo é tratado como um arquivo, incluindo a memória, os discos, a saída do monitor, os arquivos e os diretórios. Do ponto de vista do sistema operacional, tudo é um arquivo. Não deve ser surpresa que o próprio sistema esteja configurado através de arquivos. Conhecidos como arquivos de configuração, eles geralmente são arquivos de texto e são usados por vários aplicativos e serviços para armazenar ajustes e configurações para esse aplicativo ou serviço específico. Praticamente tudo no Linux depende de arquivos de configuração para funcionar. Alguns serviços têm não um, mas vários arquivos de configuração.

Os usuários com níveis de permissão apropriados usam editores de texto para alterar o conteúdo de tais arquivos de configuração. Depois que as alterações são feitas, o arquivo é salvo e pode ser usado pelo serviço ou aplicativo relacionado. Os usuários podem especificar exatamente como querem que qualquer aplicativo ou serviço se comporte. Quando iniciados, os serviços e aplicativos verificam o conteúdo de arquivos de configuração específicos e ajustam seu comportamento de acordo.

### Etapa 1: Localizando Arquivos de Configuração

O autor do programa define a localização da configuração de um determinado programa (serviço ou aplicação). Por isso, a documentação deve ser consultada ao avaliar a localização do arquivo de configuração. Convencionalmente, no entanto, no Linux, os arquivos de configuração que são usados para configurar aplicativos do usuário são frequentemente colocados no diretório home do usuário, enquanto os arquivos de configuração usados para controlar serviços de todo o sistema são colocados no diretório **/etc** . Os usuários sempre têm permissão para gravar em seus próprios diretórios home e são capazes de configurar o comportamento dos aplicativos que usam.

a.     Use o comando **ls** para listar todos os arquivos no diretório home do **analyst**:

[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

Enquanto alguns arquivos são exibidos, nenhum deles parece ser arquivos de configuração. Isso ocorre porque é convenção para ocultar arquivos de configuração hospedados pelo diretório doméstico precedendo seus nomes com um “.” caractere (ponto).

b.     Use o comando **ls** novamente, mas desta vez adicione a opção **–a** para incluir também arquivos ocultos na saída:

[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

-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 Mar 25 12:18 .cache

drwxr-xr-x 10 analyst analyst  4096 Jul 19  2018 .config

drwxr-xr-x  2 analyst analyst  4096 Mar 22  2018 Desktop

-rw-r--r--  1 analyst analyst    23 Mar 23  2018 .dmrc

drwxr-xr-x  3 analyst analyst  4096 Apr  2 14:44 Downloads

drwx------  3 analyst analyst  4096 Mar 22  2018 .gnupg

-rw-------  1 analyst analyst  2520 Mar 24 12:32 .ICEauthority

drwxr-xr-x  2 analyst analyst  4096 Mar 24  2018 .idlerc

drwxr-xr-x  9 analyst analyst  4096 Jul 19  2018 lab.support.files

drwxr-xr-x  3 analyst analyst  4096 Mar 22  2018 .local

drwx------  5 analyst analyst  4096 Mar 24  2018 .mozilla

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

<Alguma saída omitida>

c.     Use o comando **cat** para exibir o conteúdo do arquivo **.bashrc**. Este arquivo é usado para configurar o comportamento de terminal específico do usuário e a personalização.

[analyst@secOps ~]$ **cat .bashrc**

editor de exportação=vim

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

alias ls="ls —color”

alias vi="vim”

Não se preocupe muito com a sintaxe de.bashrc neste ponto. O importante a notar é que .bashrc contém configuração para o terminal. Por exemplo, a linha PS1='\ [\ e [1; 32m\] [\ u@\ h\ W]\ $\ [\ e [0m\] 'define a estrutura de prompt do prompt exibido pelo terminal: [username @hostname current\_dir] seguido de um cifrão, tudo em verde. Algumas outras configurações incluem atalhos para comandos como ls e vi. Nesse caso, toda vez que o usuário digita ls, o shell converte automaticamente para ls —color para exibir uma saída codificada por cores para ls (diretórios em azul, arquivos regulares em cinza, arquivos executáveis em verde, etc.)

A sintaxe específica está fora do escopo deste curso. O importante é entender que as configurações do usuário são convencionalmente armazenadas como arquivos ocultos no diretório pessoal do usuário.

d.     Enquanto os arquivos de configuração relacionados aos aplicativos do usuário são convencionalmente colocados sob o diretório home do usuário, os arquivos de configuração relacionados aos serviços de todo o sistema são colocados no diretório /etc, por convenção. Serviços Web, serviços de impressão, serviços de ftp e serviços de e-mail são exemplos de serviços que afetam todo o sistema e dos quais arquivos de configuração são armazenados em /etc. Observe que os usuários regulares não têm acesso de gravação para /etc. Isso é importante, pois restringe a capacidade de alterar a configuração do serviço em todo o sistema somente para o usuário root.

Use o comando ls para listar o conteúdo do diretório /etc.

[analyst@secOps ~]$ **ls /etc**

adjtime           host.conf          mke2fs.conf         rc\_maps.cfg

apache-ant        hostname           mkinitcpio.conf     request-key.conf

apparmor.d        hosts              mkinitcpio.d        request-key.d

arch-release      ifplugd            modprobe.d          resolv.conf

avahi             initcpio           modules-load.d      resolvconf.conf

bash.bash\_logout  inputrc            motd                rpc

bash.bashrc       iproute2           mtab                rsyslog.conf

binfmt.d          iptables           nanorc              securetty

ca-certificates   issue              netconfig           security

crypttab          java-7-openjdk     netctl              services

dbus-1            java-8-openjdk     netsniff-ng         shadow

default           kernel             nginx               shadow-

depmod.d          krb5.conf          nscd.conf           shells

dhcpcd.conf       ld.so.cache        nsswitch.conf       skel

dhcpcd.duid       ld.so.conf         ntp.conf            ssh

dkms              ld.so.conf.d       openldap            ssl

drirc             libnl              openvswitch         sudoers

elasticsearch     libpaper.d         os-release          sudoers.d

environment       lightdm            pacman.conf         sudoers.pacnew

ethertypes        locale.conf        pacman.conf.pacnew  sysctl.d

filebeat          locale.gen         pacman.d            systemd

fonts             locale.gen.pacnew  pam.d               tmpfiles.d

fstab             localtime          pango               trusted-key.key

gai.conf          login.defs         papersize           udev

gemrc             logrotate.conf     passwd              UPower

group             logrotate.d        passwd-             vdpau\_wrapper.cfg

group-            logstash           pcmcia              vimrc

group.pacnew      lvm                pkcs11              webapps

grub.d            machine-id         polkit-1            wgetrc

gshadow           mail.rc            profile             X11

gshadow-          makepkg.conf       profile.d           xdg

gshadow.pacnew    man\_db.conf        protocols           xinetd.d

gtk-2.0           mdadm.conf         pulse               yaourtrc

gtk-3.0           mime.types         rc\_keymaps

e.     Use o comando **cat** para exibir o conteúdo do arquivo **bash.bashrc**.

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

#

# /etc/bash.bashrc

#

# Se não correr interativamente, não faça nada

[$-!= \*i\*]] && return

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

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

case $ {TERM} em

  xterm\*|rxvt\*|Eterm|aterm|kterm|gnome\*)

    PROMPT\_COMMAND=${PROMPT\_COMMAND:+$PROMPT\_COMMAND; }'printf "\033]0;%s@%s:%s\007" "${USER}" "${HOSTNAME%%.\*}" "${PWD/#$HOME/\~}"'

    ;;

  screen)

    PROMPT\_COMMAND=${PROMPT\_COMMAND:+$PROMPT\_COMMAND; }'printf "\033\_%s@%s:%s\033\\" "${USER}" "${HOSTNAME%%.\*}" "${PWD/#$HOME/\~}"'

    ;;

esac

[ -r /usr/share/bash-completion/bash\_completion   ] && . /usr/share/bash-completion/bash\_completion

[analyst@secOps ~]$

A sintaxe do **bash.bashrc** está fora do escopo deste curso. Este arquivo define o comportamento padrão do shell para todos os usuários. Se um usuário quiser personalizar seu próprio comportamento de shell, o comportamento padrão pode ser substituído editando o arquivo .bashrc localizado no diretório home do usuário. Como esta é uma configuração em todo o sistema, o arquivo de configuração é colocado sob /etc, tornando-o editável apenas pelo usuário root. Portanto, o usuário terá que fazer login como root para modificar **bash.bashrc**.

#### Pergunta:

Por que os arquivos de configuração do aplicativo do usuário são salvos no diretório home do usuário e não sob **/etc** com todos os outros arquivos de configuração de todo o sistema?

***Área de Resposta***

### Etapa 2: Editando e salvando arquivos de configuração

Como mencionado anteriormente, os arquivos de configuração podem ser editados com editores de texto.

Vamos editar **.bashrc** para alterar a cor do prompt do shell de verde para vermelho para o usuário do **analyst** .

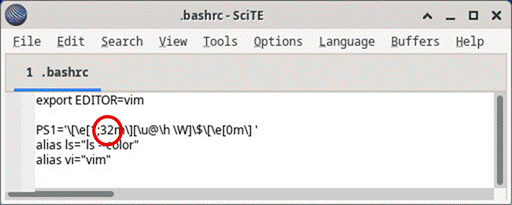
a.     Primeiro, abra **SciTE** selecionando **Aplicativos**> **CyberOps** > **SciTE** na barra de ferramentas localizada na parte superior da tela **Cisco CyberOps VM**.

b.     Selecione **Arquivo**> **Abrir** para iniciar a janela Abrir arquivo do **SciTE**.

c.     Como **.bashrc** é um arquivo oculto sem extensão, **SCite** não o exibe na lista de arquivos. Se o recurso Localização não estiver visível na caixa de diálogo, Altere o tipo de arquivo mostrado selecionando **Todos os arquivos (\*)** na caixa suspensa tipo, como mostrado abaixo. Todos os arquivos no diretório home do analista são mostrados.

d.     Selecione **.bashrc**e clique em **Abrir**.

e.     Localize 32 e substitua por 31. 32 é o código de cor para verde, enquanto 31 representa vermelho.



f.       Salve o arquivo selecionando **Arquivo**> **Salvar** e feche **SciTE** clicando no ícone **X**.

g.     Clique no ícone do aplicativo Terminal localizado na Dock, na parte inferior central da tela **Cisco CyberOPS VM**. O prompt deve aparecer em vermelho em vez de verde.

#### Pergunta:

Será que a janela do terminal que já estava aberta também mudou de cor de verde para vermelho? Explique.

***Área de Resposta***

h.     A mesma alteração poderia ter sido feita a partir da linha de comando com um editor de texto como **nano**. Em uma nova janela de terminal, digite **nano .bashrc** para iniciar o **nano** e carregue automaticamente o arquivo **.bashrc** nele:

[analyst@secOps ~]$ **nano .bashrc**

  GNU nano 4.9.2                    File: .bashrc

editor de exportação=vim

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

alias ls="ls —color”

alias vi="vim”

                                [ Read 5 lines ]

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos

^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^\_ Go To Line

i.       Mude de 31 para 33. 33 é o código de cor para amarelo.

j.       Pressione **CTRL+X** para salvar e, em seguida, pressione **Y** para confirmar. O editor de texto **nano** também lhe dará a chance de mudar o nome do arquivo. Basta pressionar ENTER para usar o mesmo nome, **.bashrc**.

k.     O editor de texto **nano** terminará e você estará de volta no prompt do shell. Desta vez, recarregue o terminal bash inserindo o comando **bash** no terminal. O prompt deve agora aparecer em amarelo em vez de vermelho.

### Etapa 3: Editando Arquivos de Configuração para Serviços

Arquivos de configuração em todo o sistema não são muito diferentes dos arquivos de aplicativo do usuário. O **nginx** é um servidor web leve instalado na **Security Workstation VM**. O **nginx** pode ser personalizado alterando seu arquivo de configuração, que está localizado em **/etc/nginx**.

a.     Primeiro, abra o arquivo de configuração do nginx no **nano**.. O nome do arquivo de configuração usado aqui é **custom\_server.conf**. Observe abaixo que o comando é precedido pelo comando **sudo**. Depois de digitar **nano**, inclua um espaço e a opção **-l** para ativar a numeração de linha.

[analyst@secOps ~]$ **sudo nano -l /etc/nginx/custom\_server.conf**

[sudo] password for analyst:

Use as teclas de seta para navegar pelo arquivo.

  GNU nano 4.9.2                /etc/nginx/custom\_server.conf

  1

  2 #user html;

  3 worker\_processes  1;

  4

  5 #error\_log  logs/error.log;

  6 #error\_log  logs/error.log  notice;

  7 #error\_log  logs/error.log  info;

  8

  9 #pid        logs/nginx.pid;

 10

 11

 12 events {

 13     worker\_connections  1024;

 14 }

 15

 16

 17 http {

 18     include       mime.types;

 19     default\_type  application/octet-stream;

 20

 21     #log\_format  main  '$remote\_addr - $remote\_user [$time\_local] "$request" '

 22     #                  '$status $body\_bytes\_sent "$http\_referer" '

 23     #                  '"$http\_user\_agent" "$http\_x\_forwarded\_for"';

 24

 25     #access\_log  logs/access.log  main;

 26

 27     sendfile        on;

 28     #tcp\_nopush     on;

 29

 30     #keepalive\_timeout  0;

 31     keepalive\_timeout  65;

 32

 33     #gzip  on;

 34

 35     types\_hash\_max\_size 4096;

 36     server\_names\_hash\_bucket\_size 128;

 37

 38     server {

 39         listen       81;

 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;

 48             index  index.html index.htm;

 49         }

<Alguma saída omitida>

**Observação**: Convencionalmente, as extensões .conf são usadas para identificar arquivos de configuração.

b.     Enquanto o arquivo de configuração tem muitos parâmetros, vamos configurar apenas dois: a porta nginx escuta conexões de entrada, e o diretório a partir do qual servirá páginas web, incluindo o arquivo de página inicial HTML índice.

c.     Observe que na parte inferior da janela, acima dos comandos nano, o número da linha é destacado e listado. Na linha 39, altere o número da porta de **81** para **8080**. Isto irá dizer ao nginx para ouvir pedidos HTTP na porta **TCP 8080**.

d.     Em seguida, mova para a linha 47 e altere o caminho de **/usr/share/nginx/html/** para **/usr/share/nginx/html/text\_ed\_lab/**

**Nota**: Tenha cuidado para não remover o ponto-e-vírgula no final da linha ou o **nginx** lançará um erro na inicialização.

e.     Pressione **CTRL+X** para salvar o arquivo. Pressione **Y** e ENTER para confirmar e usar **custom\_server.conf** como o nome do arquivo.

f.       Digite o comando abaixo para executar o nginx usando o arquivo de configuração modificado:

[analyst@secOps ~]$ **sudo nginx -c custom\_server.conf**

g.     Clique no ícone do navegador da Web no Dock para iniciar o Firefox.

h.     Na barra de endereços, digite **127.0.0.1:8080** para se conectar a um servidor Web hospedado na máquina local na porta 8080. Uma página relacionada a este laboratório deve aparecer.

i.       Depois de abrir com sucesso a página inicial do **nginx**, veja a mensagem de conexão na janela do terminal.

#### Pergunta:

A que se refere a mensagem de erro?

***Área de Resposta***

j.       Para desligar o servidor web **nginx**, pressione **ENTER** para obter um prompt de comando e digite o seguinte comando na janela do terminal:

[analyst@secOps ~]$ **sudo pkill nginx**

k.      Você pode testar se o servidor **nginx** é realmente desligado, primeiro limpando o histórico recente no navegador da Web, depois feche e reabra o navegador da Web e, em seguida, vá para a página inicial do nginx em 127.0.0. 1:8080.

#### Pergunta:

A página da Web aparece?

***Área de Resposta***

**Pergunta Desafio**: Você pode editar o arquivo **/etc/nginx/custom\_configuration.conf** com o SciTE? Descreva o processo abaixo.

Lembre-se, como o arquivo é armazenado em /etc, você precisará de permissões de root para editá-lo.

***Área de Resposta***

# Reflexão

Dependendo do serviço, mais opções podem estar disponíveis para configuração.

A localização do arquivo de configuração, a sintaxe e os parâmetros disponíveis variam de serviço para serviço. Consulte sempre a documentação para obter informações.

As permissões são uma causa muito comum de problemas. Certifique-se de que você tem as permissões corretas antes de tentar editar arquivos de configuração.

Na maioria das vezes, os serviços devem ser reiniciados antes que as alterações entrem em vigor.

*Fim do documento*