实验 - 在 CLI 中使用文本文件

1. 简介

在本实验中，同学们将熟悉 Linux 命令行文本编辑器和配置文件。

1. 所需资源

* CyberOps Workstation 虚拟机

1. 图形文本编辑器

在 Linux 中使用文本文件之前，同学们需要先熟悉文本编辑器。

文本编辑器是最早诞生的计算机应用程序之一。Linux 与许多其他操作系统一样，也有许多各具特色、功能各异的文本编辑器。有些文本编辑器包括图形界面，而有些文本编辑器只能通过命令行使用。每个文本编辑器都包含一个专为支持特定工作场景而设计的功能集。有些文本编辑器以程序员为中心，包含语法突出显示、括号匹配、查找和替换、多行 Regex 支持、拼写检查等功能以及其他侧重于编程的功能。

为节省空间和保持虚拟机的精简，**思科 CyberOps VM** 仅包含 **SciTE** 作为图形文本编辑器应用。**SciTE** 是一款小巧易用、运行敏捷的文本编辑器。它没有许多高级功能，却可以全面支持本课程中的任务。

**注**：文本编辑器的选择因人而异，并没有所谓的最佳文本编辑器。同学们感觉哪个文本编辑器用起来最自如、最顺手，哪个就是最佳文本编辑器。

* 1. 从 GUI 打开 SciTE
     1. 以 **analyst** 身份，使用密码 **cyberops** 登录 CyberOps VM。在本实验中，使用 **analyst** 作为示例用户帐户。
     2. 在顶部栏中，导航到**应用 > CyberOPS > SciTE**，启动 **SciTE** 文本编辑器。
     3. **SciTE** 虽然简单，但包括几个重要功能：标签式环境、语法突出显示等。花几分钟时间熟悉 SciTE。在主工作区域，键入或复制粘贴以下文本：

“Space, is big.Really big.You just won't believe how vastly, hugely, mindbogglingly big it is.I mean, you may think it's a long way down the road to the chemist, but that's just peanuts to space.”

― Douglas Adams, The Hitchhiker’s Guide to the Galaxy

* + 1. 点击**文件** > **保存**，保存文件。请注意，**SciTE** 会尝试将文件保存到当前用户的主目录，默认情况下是 analyst。将文件命名为 **space.txt**，并点击**保存**。
    2. 点击窗口右上角的 **X** 图标，关闭 **SciTE**，然后重新打开 **SciTE**。
    3. 点击**文件** > **打开…**，搜索刚刚保存的文件，即 **space.txt**。

能不能立即找到 **space.txt**？\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + 1. 即使 SciTE 查找的是正确目录 (/home/analyst)，space.txt 也不会显示。这是因为 SciTE 查找的是具有已知扩展名的文件，而 .txt 并不在其中。要显示所有文件，请点击**打开文件**窗口底部的下拉菜单，然后选择**所有文件 (\*)**。
    2. 选择 **space.txt** 打开它。

**注**：尽管 Linux 文件系统不依赖扩展名，但 **SciTE** 等有些应用可能会尝试使用扩展名来识别文件类型。

* + 1. 完成后，关闭 space.txt。
  1. 从终端打开 SciTE。
     1. 或者，同学们也可以从命令行打开 SciTE。点击底部停靠栏中的**终端**图标。**终端**仿真程序打开。
     2. 键入 **ls**，查看当前目录的内容。注意系统会列出 **space.txt**。这意味着，同学们不必提供路径信息，即可打开文件。
     3. 键入 **scite** **space.txt** 打开 **SciTE**。请注意，这不仅会在 GUI 中启动 **SciTE**，还将自动加载先前创建的 space.txt 文本文件。

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

* + 1. 请注意，尽管 **SciTE** 在前端打开，但用于启动它的终端窗口依然在后台打开。此外，请注意，用于启动 **SciTE** 的终端窗口不再显示提示符。

为什么不显示提示符？

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + 1. 像往常一样点击 X 图标关闭 **SciTE** 实例，或者将主显示内容切换回启动 **SciTE** 的终端窗口，并停止进程。也可以按 **CTRL+C** 停止进程。

**注**：当同学们想以 **root** 用户身份运行 **SciTE** 时，从命令行启动 **SciTE** 很有帮助。只需在 **scite** 前面添加 **sudo** 命令，即 **sudo scite**。

* + 1. 关闭 **SciTE**，移动到下一部分。

1. 命令行文本编辑器

尽管图形文本编辑器方便易用，但基于命令行的文本编辑器在 Linux 计算机中非常重要。基于命令行的文本编辑器的主要优势在于，它们允许从远程计算机上的远程 shell 编辑文本文件。

试想一下这样的场景：用户必须在 Linux 计算机上执行管理任务，但他不在这台计算机跟前。用户可以使用 **SSH** 启动与前面提到的计算机的远程 shell。在基于文本的远程 shell 下，图形界面可能不可用，因此也就无法使用图形文本编辑执行操作。在这种情况下，基于文本的文本编辑器至关重要。

**注：**当连接到没有 GUI 界面的远程无头服务器时，更是如此。

**思科 CyberOps VM** 包含一些基于命令行的文本编辑器。本课程重点介绍 **nano**。

**注**：另一个非常流行的文本编辑器叫作 **vi**。尽管 **vi** 学习起来很难，但 **vi** 是一个非常强大的基于命令行的文本编辑器。默认情况下，它包含在几乎所有的 Linux 分布中，其原始代码始创于 1976 年。**vi** 的更新版本名为 **vim**，意为“改进的 vi”。目前，大多数 **vi** 用户几乎都在使用更新版本（即 **vim**）。

由于缺少图形支持，**nano**（或 GNU **nano**）只能通过键盘控制。**CTRL+O** 保存当前文件；**CTRL+W** 打开搜索菜单。GNU **nano** 使用屏幕底部的两行快捷方式栏，其中列出了当前上下文的许多命令。打开 nano 后，按 **CTRL+G** 打开帮助屏幕和完整列表。

* + 1. 在终端窗口中，键入 **nano space.txt** 打开在第 1 部分创建的文本文件。

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

* + 1. **nano** 将启动并自动加载 **space.txt** 文本文件。尽管文本看起来可能已被截断或不完整，但事实并非如此。因为创建文本时没有使用回车符，而且自动换行在默认情况下不启用，所以 **nano** 显示很长的一行文本。

使用键盘上的 Home 和 End 键，可以分别快速导航到行首和行尾。

**Nano** 使用哪个字符表示行在屏幕边界以外继续显示？

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + 1. 如底部快捷方式行所示，可以使用 **CTRL+X** 退出 **nano**。**Nano** 会在退出前询问同学们是否要保存文件（“Y”表示“是”，或者“N”表示“否”）。如果选择“Y”，系统将提示同学们接受既定文件名、更改文件名，或者提供文件名（如果是新的未命名文档）。
    2. 要控制 **nano**，同学们可以使用 **CTRL**、**ALT**、**ESCAPE** 或 META 键。META 键是指键盘上带有 Windows 或 Mac 徽标的键，具体取决于同学们的键盘配置。
    3. 在 Nano 中导航十分简单。使用箭头键可切换选定的文件。同学们也可以使用 Page Up 和 Page Down 向前或向后翻页。花些时间了解 **nano** 及其帮助屏幕。要进入帮助屏幕，请按 **CTRL+G**。

1. 使用配置文件

**在 Linux 中，几乎一切都被视为文件。内存、磁盘、显示器输出、目录；从操作系统角度**，一切都是文件。因此，通过文件配置系统本身不足为奇。被称为配置文件，它们通常是文本文件，被各种应用和服务用来存储特定应用或服务的调整和设置。实际上，Linux 中的一切都依赖配置文件运行。有些服务的配置文件不止一个，而是有多个。

具有适当权限级别的用户使用文本编辑器更改此类配置文件的内容。做出更改后，保存文件，并且可供相关服务或应用使用。用户能够具体指定所有既定应用或服务的行为。启动时，服务和应用检查特定配置文件的内容，并相应地调整其行为。

* 1. 查找配置文件

程序创建者定义既定程序（服务或应用）的配置位置。因此，评估配置文件的位置时，应当查看文档。然而，在 Linux 中，用于配置用户应用的配置文件通常放在用户的主目录中，而用于控制系统范围的服务的配置文件放在 **/etc** 目录中。用户始终拥有写入自己的主目录的权限，并且能够配置他们使用的应用的行为。

* + 1. 使用 **ls** 命令列出 **analyst** 主目录中的所有文件：

[analyst@secOps ~]$ **ls –l**

total 20

drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop

drwx------ 3 analyst analyst 4096 Jul 14 11:28 Downloads

drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files

drwxr-xr-x 2 analyst analyst 4096 Mar 3 15:56 second\_drive

-rw-r--r-- 1 analyst analyst 254 Aug 16 13:32 space.txt

显示几个文件时，没有一个看上去像是配置文件。这是因为按惯例是通过在名字前面加上“.”（点号）字符来隐藏主目录托管的配置文件

* + 1. 再次使用 **ls** 命令，但这一次添加 **–a** 选项，以便将隐藏文件也包含在输出中：

[analyst@secOps ~]$ **ls –la**

total 268

drwxr-xr-x 19 analyst analyst 4096 Aug 2 15:43 .

drwxr-xr-x 3 root root 4096 Sep 26 2014 ..

-rw------- 1 analyst analyst 250 May 4 11:42 .atftp\_history

-rw------- 1 analyst analyst 13191 Aug 1 09:48 .bash\_history

-rw-r--r-- 1 analyst analyst 97 Mar 21 15:31 .bashrc

drwxr-xr-x 4 analyst analyst 4096 Jul 6 10:26 broken\_down

drwxr-xr-x 10 analyst analyst 4096 Nov 7 2016 .cache

drwxr-xr-x 12 analyst analyst 4096 Jun 5 11:45 .config

-rw-r--r-- 1 analyst analyst 16384 Apr 12 10:06 .cyberops\_topo.py.swp

drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop

-rw-r--r-- 1 analyst analyst 43 Sep 27 2014 .dmrc

drwx------ 3 analyst analyst 4096 Jul 14 11:28 Downloads

-rw-r--r-- 1 analyst analyst 72 Sep 26 2014 .fehbg

drwxr-xr-x 5 analyst analyst 4096 Sep 26 2014 .fluxbox

drwx------ 3 analyst analyst 4096 Sep 7 2016 .gnupg

-rw------- 1 analyst analyst 28920 Aug 2 15:01 .ICEauthority

drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 .idlerc

drwxr-xr-x 3 analyst analyst 4096 Sep 27 2014 .java

drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files

-rw------- 1 analyst analyst 290 Jul 6 15:15 .lesshst

drwxr-xr-x 3 analyst analyst 4096 Sep 26 2014 .local

<Some output omitted>

* + 1. 使用 **cat** 命令显示 **.bashrc** 文件的内容。使用此文件配置用户特定的终端行为和自定义。

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

export EDITOR=vim

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

alias ls="ls --color"

alias vi="vim"

此时，不要太担心 **.bashrc** 的语法。需要注意的是，**.bashrc** 包含终端配置。例如，行 **PS1='\[\e[1;32m\][\u@\h \W]\$\[\e[0m\] '** 定义终端显示的提示符的结构：**[username@hostname current\_dir]** 后跟一个美元符号，全部显示为绿色。其他几个配置包括命令的快捷方式，例如 **ls** 和 **vi**。在这种情况下，每当用户键入 **ls** 时，shell 会自动将其转换为 **ls –color**，显示彩色的 **ls** 输出（目录显示为蓝色，普通文件显示为灰色，可执行文件显示为绿色，等等）

具体语法不在此课程的范围之内。按照惯例，**用户配置在用户的主目录中存储为隐藏文件**，了解这一点非常重要。

* + 1. 尽管与用户应用相关的配置文件按惯例放在用户的主目录下，但与系统范围的服务相关的配置文件按惯例放在 **/etc** 目录中。例如，Web 服务、打印服务、FTP 服务、邮件服务等影响整个系统的服务，它们的配置文件都存储在 **/etc** 下。请注意，**普通用户对 /etc 没有写入权限。这一点非常重要，因为它限制了仅 root 用户能够更改系统范围的服务配置。**

使用 **ls** 命令列出 **/etc** 目录中的内容：

[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

* + 1. 使用 **cat** 命令显示 **bash\_bashrc** 文件中的内容：

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

#

# /etc/bash.bashrc

#

# If not running interactively, don't do anything

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

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/\~}"'

;;

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 ~]$

**bash\_bashrc** 的语法不在本课程范围之内。此文件为所有用户定义默认的 shell 行为。如果用户想自定义他/她自己的 shell 行为，可以通过编辑用户主目录中的 **.bashrc** 文件来覆盖默认行为。因为这是一个系统范围的配置，所以配置文件放在 **/etc** 下面，使其仅可由 **root** 用户编辑。因此，用户必须以 root 用户身份登录，才能修改 **.bashrc**。

为什么用户应用配置文件保存在用户的主目录下，而不是与所有其他系统范围的配置文件一样存储在 **/etc** 下？

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

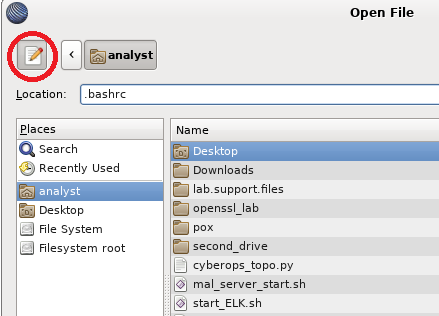
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. 编辑和保存配置文件

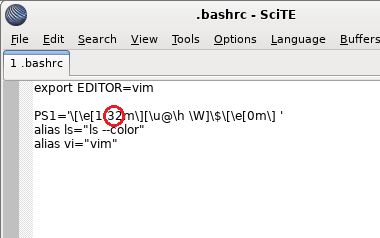
如前所述，可以使用文本编辑器编辑配置文件。

让我们来编辑 **.bashrc**，将向 **analyst** 用户显示的 shell 提示符的颜色从绿色改为红色。

* + 1. 首先，从**思科 CyberOPS VM** 屏幕上半部分的工具栏依次选择**应用** > **CyberOPS** > **SciTE** ，打开 **SciTE**。
    2. 选择**文件** > **打开**，启动 **SciTE** 的“打开文件”窗口。
    3. 因为 **.bashrc** 是隐藏文件且没有扩展名，所以 SciTE 不会在文件列表中显示它。如果“位置”功能在对话框中不可见，请点击**键入文件名**按钮，如下所示，然后键入 **.bashrc**。点击**打开**。



* + 1. 找到 32 并替换为 31。32 是绿色的颜色代码，31 是红色的颜色代码。



* + 1. 通过选择**文件** > **保存**来保存文件，然后点击 **X** 图标关闭 **SciTE**。
    2. 点击**思科 CyberOPS VM** 屏幕底部中间停靠栏上的终端应用图标。应当显示红色而非绿色提示符。

已打开的终端窗口的颜色是否也从绿色变成了红色？请说明原因。

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + 1. 可以使用 **nano** 等文本编辑器，从命令行执行相同的更改。从新的终端窗口，键入 **nano .bashrc**，启动 **nano** 并自动找到其中的 **.bashrc** 文件：

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

GNU nano 2.8.1 File: .bashrc

export EDITOR=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

* + 1. 将 31 更改为 33。33 是黄色的颜色代码。
    2. 按 **CTRL+X** 保存，然后按 **Y** 确认。Nano 还将为同学们提供更改文件名的机会。只需按 **ENTER**，即可使用同一名称，**.bashrc**。
    3. Nano 将结束，同学们将返回 shell 提示符。再次点击**思科 CyberOps VM** 屏幕底部中间停靠栏上的**终端**应用图标。现在，提示符应当显示为黄色而非红色。
  1. 编辑服务的配置文件

系统范围配置文件与用户应用文件没有太多不同。**nginx** 是一个轻型的 Web 服务器，可以安装在**思科 CyberOPS VM** 中。通过更改 **/etc/nginx** 下的配置文件，可以自定义 **nginx**。

* + 1. 首先，打开 **nano** 中的 **nginx** 配置文件。这里使用的配置文件名是 **custom\_server.conf**。注意下面，此命令的前面是 **sudo** 命令。键入 **nano** 后，再键入一个空格和 **-l** 开关命令标识符，打开行编码。

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

[sudo] password for analyst:

使用箭头按钮导航文件。

GNU nano 2.8.1 File: /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 server {

36 listen 81;

37 server\_name localhost;

38

39 #charset koi8-r;

40

41 #access\_log logs/host.access.log main;

42

43 location / {

44 root /usr/share/nginx/html;

45 index index.html index.htm;

46 }

47

48 #error\_page 404 /404.html;

49

^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 Li

**注**：按照惯例，使用 **.conf** 扩展名来标识配置文件。

* + 1. 尽管配置文件有许多参数，但我们仅配置两个：侦听传入连接的端口 nginx 以及它将从其中提供网页的目录，包括 index HTML 主页文件。
    2. 请注意，在窗口底部，nano 命令上面，突出显示并列出行编号。在第 36 行，将端口号从 **81** 更改为 **8080**。这将告知 nginx 侦听端口 **TCP 8080** 上的 HTTP 请求。
    3. 接下来，移动到第 44 行，将路径从 **/usr/share/nginx/html/** 更改为 **/home/analyst/lab.support.files/**

**注**：请务必小心，不要删除行尾的分号，否则 **nginx** 将在启动时报错。

* + 1. 按 **CTRL+X** 保存文件。依次按 **Y** 和 **ENTER** 进行确认，并使用 **custom\_server.conf** 作为文件名。
    2. 键入下面的命令，使用修改的配置文件执行 nginx：

[analyst@secOps ~]$ **sudo nginx -c custom\_server.conf -g “pid /var/run/nginx\_v.pid;”**

**注**：告知 nginx 在存储标识此 nginx 实例的进程 ID 时使用什么文件，需要 **-g “pid /var/run/nginx\_v.pid;”**。

* + 1. 点击停靠栏上的 Web 浏览器图标，启动 Firefox。
    2. 在地址栏上，键入 **127.0.0.1:8080**，连接到端口 8080 上的本地计算机托管的 Web 服务器。将显示与此实验相关的页面。
    3. 成功打开 **nginx** 主页后，在终端窗口中查看连接消息。错误消息指的是什么？

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + 1. 要关闭 **nginx** Web 服务器，请按 **ENTER**，获得命令提示符，然后在终端窗口中键入以下命令：

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

* + 1. 同学们可以首先清除 Web 浏览器中的近期历史记录，关闭再重新打开 Web 浏览器，然后转至 127.0.0.1:8080 上的 nginx 主页，测试 **nginx** 服务器是否真的关闭。是否显示网页？\_\_\_\_\_

**练习**：同学们是否会使用 SciTE 编辑 **/etc/nginx/custom\_configuration.conf** 文件？在下面描述过程。

切记，因为文件存储在 /etc 下面，所以同学们需要 root 权限来编辑它。

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 思考

根据服务，可能有更多选项可用于配置。

配置文件位置、语法和可用参数因服务而异。请始终参考文档了解信息。

权限是出现故障的常见原因。尝试编辑配置文件之前，请确保同学们有适当的权限。

通常情况下，必须重新启动服务，更改才能生效。