实验 - 浏览 Linux 文件系统和权限设置

1. 目标

在此实验中，同学们将熟悉 Linux 文件系统的使用。

1. 所需资源

* CyberOps Workstation VM

1. 了解 Linux 中的文件系统

Linux 文件系统是其最常用的功能之一。虽然 Linux 支持许多不同类型的文件系统，但此实验的重点是 **ext** 系列，它是 Linux 中最常见的一个文件系统。

* 1. 访问命令行。

启动 CyberOps Workstation 虚拟机并打开一个终端窗口。

* 1. 显示当前已安装的文件系统。

必须先*挂载*文件系统，然后才能访问并使用这些文件系统。在计算机术语中，*挂载文件系统*意味着该文件系统可供操作系统访问。挂载文件系统是指将块设备（硬盘驱动器、SSD 驱动器、笔式驱动器等）的物理分区链接到某个目录的过程，通过该目录可以访问整个文件系统。由于上述目录成为新挂载的文件系统的 root，因此它也被称为*挂载点*。

* + 1. 使用 **lsblk** 命令显示所有块设备：

[analyst@secOps ~]$ **lsblk**

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 5.9G 0 disk

└─sda1 8:1 0 5.9G 0 part /

sdb 8:16 0 1G 0 disk

└─sdb1 8:17 0 1023M 0 part

sr0 11:0 1 1024M 0 rom

上面的输出显示 CyberOps Workstation 虚拟机安装了三台块设备，分别是：sr0、sda 和 sdb。这个树状输出还显示了 sda 和 sdb 下的分区。通常情况下，Linux 使用 /dev/sdX 表示硬盘驱动器，尾部数字表示该设备内的分区号。具有多个硬盘驱动器的计算机可能会显示更多 /dev/sdX 设备。例如，如果 Linux 在一台有四个硬盘驱动器的计算机上运行，则默认情况下它会将这些硬盘驱动器显示为 /dev/sda、/dev/sdb、/dev/sdc 和 /dev/sdd。输出表明 sda 和 sdb 均为硬盘驱动器，各包含一个分区。输出还显示 sda 是一个容量为 5.9GB 的磁盘，而 sdb 的容量为 1GB。

注：Linux 也经常将 USB 闪存驱动器显示为 /dev/sdX，具体取决于其固件类型。

* + 1. 使用 **mount** 命令可显示 CyberOps Workstation 虚拟机中目前挂载的文件系统的更多详细信息。

[analyst@secOps ~]$ **mount**

proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)

sys on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)

dev on /dev type devtmpfs (rw,nosuid,relatime,size=1030408k,nr\_inodes=218258,mode=755)

run on /run type tmpfs (rw,nosuid,nodev,relatime,mode=755)

/dev/sda1 on / type ext4 (rw,relatime,data=ordered)

securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)

tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)

devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)

tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)

<output omitted>

上面的许多文件系统超出了本课程的范围，并且与此实验无关。让我们把重点放在 *root 文件系统*上面，即存储在 **/dev/sda1** 中的文件系统。Root 文件系统是 Linux 操作系统本身的存储位置；默认情况下，所有程序、工具、配置文件都存储在 root 文件系统中。

* + 1. 再次运行 **mount** 命令，但这一次使用管道 **|** 将 mount 输出发送到 **grep**，以过滤输出且只显示 root 文件系统：

[analyst@secOps ~]$ **mount | grep sda1**

/dev/sda1 on / type ext4 (rw,relatime,data=ordered)

在上面过滤后的输出中，mount 向我们显示 root 文件系统位于 sda 块设备的第一个分区中 (/dev/sda1)。我们之所以知道这是 root 文件系统，是因为使用了挂载点：“/”（斜杠符号）。通过输出还可以得知分区中使用的格式化类型，在本例中为 ext4。括号内的信息与分区挂载选项有关。

* + 1. 在 **CyberOps Workstation 虚拟机**上发出以下两个命令：

[analyst@secOps ~]$ **cd /**

[analyst@secOps /]$ **ls -l**

输出的含义是什么？所列的文件存储在哪些物理位置？

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

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

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

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

为什么在上面的输出中没有显示 **/dev/sdb1**？

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

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

* 1. 手动挂载和取消挂载文件系统

**mount** 命令还可用于挂载和取消挂载文件系统。如步骤 1 所示，CyberOps Workstation 虚拟机安装了两个硬盘驱动器。内核将第一个硬盘驱动器识别为 /dev/sda，将第二个识别为 /dev/sdb。必须有挂载点，才能挂载块设备。

* + 1. 使用 **ls -l** 命令验证目录 **second\_drive** 位于分析师的主目录中。

[analyst@secOps /]$ **cd ~**

[analyst@secOps ~]$ **ls –l**

total 28

drwxr-xr-x 3 analyst analyst 4096 Aug 16 15:15 cyops\_folder2

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 142 Aug 16 15:11 some\_text\_file.txt

-rw-r--r-- 1 analyst analyst 254 Aug 16 13:38 space.txt

**注**：如果目录 **second\_drive** 不存在，请使用 **mkdir second\_drive** 命令创建目录。

[analyst@secOps ~]$ **mkdir second\_drive**

**注**：根据虚拟机的状态，同学们的列表很可能会包含不同的文件和目录。

* + 1. 再次使用 **ls -l** 列出新建的 second\_drive 目录的内容。

[analyst@secOps ~]$ **ls -l second\_drive/**

total 0

注意该目录为空。

* + 1. 使用 **mount** 命令在新建的 **second\_drive** 目录中挂载 **/dev/sdb1**。挂载的语法是：**mount [options] <device to be mounted> <mounting point>**。

[analyst@secOps ~]$ **sudo mount /dev/sdb1 ~/second\_drive/**

[sudo] password for analyst:

无输出则意味着挂载过程已成功。

* + 1. **/Dev/sdb1** 现已挂载到 **/home/analyst/second\_drive**，现在再次使用 **ls -l** 列出该目录的内容。

[analyst@secOps ~]$ **ls -l second\_drive/**

total 20

drwx------ 2 root root 16384 Mar 3 10:59 lost+found

-rw-r--r-- 1 root root 183 Mar 3 15:42 myFile.txt

为什么该目录不再为空？所列的文件存储在哪些物理位置？

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

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

* + 1. 再次发出 **mount** 命令，且不附带任何选项显示 **/dev/sdb1** 分区的详细信息。和之前一样，使用 **grep** 命令只显示 **/dev/sdX** 文件系统：

[analyst@secOps ~]$ **mount | grep sd**

/dev/sda1 on / type ext4 (rw,relatime,data=ordered)

/dev/sdb1 on /home/analyst/second\_drive type ext2 (rw,relatime,block\_validity,barrier,user\_xattr,acl)

* + 1. 取消挂载文件系统同样很简单。确保将目录更改为挂载点之外的位置，并使用  **umount** 命令，如下所示：

[analyst@secOps ~]$ **sudo umount /dev/sdb1**

[sudo] password for analyst:

[analyst@secOps ~]$

[analyst@secOps ~]$ **ls -l second\_drive/**

total 0

1. 文件权限

Linux 文件系统具有内置功能，可以控制用户查看、更改、浏览和执行文件系统内容的权限。本质上，文件系统中的每个文件都有自己的一组权限，并且总是带有一组关于用户和组可以对文件执行哪些操作的定义。

* 1. 显示并更改文件权限。
     1. 导航到 /home/analyst/lab.support.files/scripts/。

[analyst@secOps ~]$ **cd lab.support.files/scripts/**

* + 1. 使用 **ls -l** 命令显示文件权限。

[analyst@secOps scripts]$ **ls -l**

total 60

-rwxr-xr-x 1 analyst analyst 190 Jun 13 09:45 configure\_as\_dhcp.sh

-rwxr-xr-x 1 analyst analyst 192 Jun 13 09:45 configure\_as\_static.sh

-rwxr-xr-x 1 analyst analyst 3459 Jul 18 10:09 cyberops\_extended\_topo\_no\_fw.py

-rwxr-xr-x 1 analyst analyst 4062 Jul 18 10:09 cyberops\_extended\_topo.py

-rwxr-xr-x 1 analyst analyst 3669 Jul 18 10:10 cyberops\_topo.py

-rw-r--r-- 1 analyst analyst 2871 Apr 28 11:27 cyops.mn

-rwxr-xr-x 1 analyst analyst 458 May 1 13:50 fw\_rules

-rwxr-xr-x 1 analyst analyst 70 Apr 28 11:27 mal\_server\_start.sh

drwxr-xr-x 2 analyst analyst 4096 Jun 13 09:55 net\_configuration\_files

-rwxr-xr-x 1 analyst analyst 65 Apr 28 11:27 reg\_server\_start.sh

-rwxr-xr-x 1 analyst analyst 189 Dec 15 2016 start\_ELK.sh

-rwxr-xr-x 1 analyst analyst 85 Dec 22 2016 start\_miniedit.sh

-rwxr-xr-x 1 analyst analyst 76 Jun 22 11:38 start\_pox.sh

-rwxr-xr-x 1 analyst analyst 106 Jun 27 09:47 start\_snort.sh

-rwxr-xr-x 1 analyst analyst 61 May 4 11:45 start\_tftpd.sh

以 **cyops. mn** 文件为例。该文件的所有者是谁？小组呢？

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

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

**Cyops.mn** 的权限为 **–rw-r--r--**。这意味着什么？

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

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

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

* + 1. **touch** 命令非常简单有用。使用它可以快速创建一个空文本文件。使用下面的命令在 **/mnt** 目录中创建一个空文件：

[analyst@secOps scripts]$ **touch /mnt/myNewFile.txt**

touch: cannot touch '/mnt/myNewFile.txt': Permission denied

文件是否已创建？列出 **/mnt** 目录的权限、所有权和内容，并解释所发生的情况。在下方的横线上记录答案。

[analyst@secOps ~]$ **ls -l /mnt**

total 4

drwxr-xr-x 2 root root 4096 Mar 3 11:13 second\_drive

添加 -d 选项后，它列出了父目录的权限。

[analyst@secOps ~]$ **ls -ld /mnt**

drwxr-xr-x 3 root root 4096 Mar 3 15:43 /mnt

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

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

执行哪些操作可以使上面所示的 **touch** 命令执行成功？

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

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

* + 1. **chmod** 命令用于更改文件或目录的权限。和之前一样，在此实验早些时候创建的 **/home/analyst/second\_drive** 目录上挂载 **/dev/sdb1** 分区：

[analyst@secOps scripts]$ **sudo mount /dev/sdb1 ~/second\_drive/**

* + 1. 改为 **second\_drive** 目录并列出其内容：

[analyst@secOps scripts]$ **cd ~/second\_drive**

[analyst@secOps second\_drive]$ **ls -l**

total 20

drwx------ 2 root root 16384 Mar 3 10:59 lost+found

-rw-r--r-- 1 root root 183 Mar 3 15:42 myFile.txt

**myFile.txt** 文件有哪些权限？

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

* + 1. 使用 **chmod** 命令更改 **myFile.txt** 的权限。

[analyst@secOps second\_drive]$ **sudo** **chmod 665 myFile.txt**

[analyst@secOps second\_drive]$ **ls -l**

total 20

drwx------ 2 root root 16384 Mar 3 10:59 lost+found

-rw-rw-r-x 1 root root 183 Mar 3 15:42 myFile.txt

权限是否已更改？**myFile.txt** 有哪些权限？

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

**Chmod** 命令的权限以八进制格式显示。因此，665 分解如下：

八进制中的 6 表示二进制中的 110。假定文件权限的每个位置都可以是 1 或 0，则 110 表示 rw-（读取=1，写入=1，执行=0）。

因此，**chmod 665 myFile.txt** 命令将权限改为：

**所有者**：rw-（八进制中的 6 或 110）

**组**：rw-（八进制中的 6 或 110）

**其他**：rw-（八进制中的 5 或 101）

使用什么命令可以将 myFile.txt 的权限更改为 rwxrwxrwx，从而向系统中的任何用户授予该文件的完全访问权限？

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

* + 1. **chown** 命令用于更改文件或目录的所有权。发出下面的命令，使分析师用户成为 **myFile.txt** 的所有者：

[analyst@secOps second\_drive]$ **sudo chown analyst myFile.txt**

[sudo] password for analyst:

[analyst@secOps second\_drive]$ **ls -l**

total 20

drwx------ 2 root root 16384 Mar 3 10:59 lost+found

-rw-rw-r-x 1 analyst root 183 Mar 3 15:42 myFile.txt

[analyst@secOps second\_drive]$

**注**：要将所有者和组同时改为分析师，请使用 **sudo chown analyst:analyst myFile.txt** 格式。

* + 1. 现在，分析师是文件所有者，请尝试将“test”一词附加到 **myFile.txt** 的末尾。

[analyst@secOps second\_drive]$ **echo test >> myFile.txt**

[analyst@secOps second\_drive]$ **cat myFile.txt**

操作是否成功？请说明原因。

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

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

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

* 1. 目录和权限

与常规文件相似，目录也带有权限。但目录在权限中有一个额外的位。

* + 1. 重新改回 /home/analyst/lab.support.files 目录，并发出 **ls -l** 命令列出所有文件以及详细信息：

[analyst@secOps second\_drive]$ **cd ~/lab.support.files/**

[analyst@secOps lab.support.files]$ **ls -l**

total 580

-rw-r--r-- 1 analyst analyst 649 Jun 28 18:34 apache\_in\_epoch.log

-rw-r--r-- 1 analyst analyst 126 Jun 28 11:13 applicationX\_in\_epoch.log

drwxr-xr-x 4 analyst analyst 4096 Aug 7 15:29 attack\_scripts

-rw-r--r-- 1 analyst analyst 102 Jul 20 09:37 confidential.txt

-rw-r--r-- 1 analyst analyst 2871 Dec 15 2016 cyops.mn

-rw-r--r-- 1 analyst analyst 75 May 24 11:07 elk\_services

-rw-r--r-- 1 analyst analyst 373 Feb 16 16:04 h2\_dropbear.banner

-rw-r--r-- 1 analyst analyst 147 Mar 21 15:30 index.html

-rw-r--r-- 1 analyst analyst 255 May 2 13:11 letter\_to\_grandma.txt

-rw-r--r-- 1 analyst analyst 24464 Feb 7 2017 logstash-tutorial.log

drwxr-xr-x 2 analyst analyst 4096 May 25 13:01 malware

-rwxr-xr-x 1 analyst analyst 172 Jul 25 16:27 mininet\_services

drwxr-xr-x 2 analyst analyst 4096 Feb 14 2017 openssl\_lab

drwxr-xr-x 2 analyst analyst 4096 Aug 7 15:25 pcaps

drwxr-xr-x 7 analyst analyst 4096 Sep 20 2016 pox

-rw-r--r-- 1 analyst analyst 473363 Feb 16 15:32 sample.img

-rw-r--r-- 1 analyst analyst 65 Feb 16 15:45 sample.img\_SHA256.sig

drwxr-xr-x 3 analyst analyst 4096 Jul 18 10:10 scripts

-rw-r--r-- 1 analyst analyst 25553 Feb 13 2017 SQL\_Lab.pcap

将 **malware** 目录的权限与 **mininet\_services** 文件进行比较。它们的权限有什么区别？

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

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

字母“d”表示该特定条目是目录而不是文件。文件权限和目录权限之间的另一个区别是执行位。如果文件的执行位已打开，则表示它可以由系统执行。目录不同于设置了执行位的文件（设置了执行位的文件是可执行脚本或程序）。设置了执行位的目录指定用户是否可以进入该目录。

**chmod** 和 **chown** 命令对目录的处理方式与对文件的处理方式相同。

1. 符号链接和其他特殊文件类型

同学们现在已经了解到 Linux 中一些不同的文件类型。**ls –l** 命令中每个文件列表的第一个字符显示文件类型。Linux 中的三种不同的文件类型（包括子类型和字符）为：

* **常规文件 (-)**，包括：
  1. 可读文件 - 文本文件
  2. 二进制文件 - 程序
  3. 映像文件
  4. 压缩文件
* **目录文件 (d)**
  1. 文件夹
* **特殊文件**，包括：
  1. **块文件 (b)**  - 用于访问物理硬件的文件，就像访问硬盘驱动器的挂载点一样。
  2. **字符设备文件 (c)** - 提供输入和输出串行流的文件。tty 终端就属于此类文件。
  3. **管道文件 (p)** - 用于传递信息的文件，其中前面的字节始终在前面。这也称为 FIFO（先进先出）。
  4. **符号链接文件 (l)**  - 用于链接到其他文件或目录的文件。有两种类型：符号链接和硬链接。
  5. **套接字文件**  - 这些文件用于在应用间传递信息，以便通过网络进行通信。
  6. 检查文件类型。
     1. 使用 **ls -l** 命令显示文件。注意，每一行的第一个字符都是表示文件的“–”或表示目录的“d”

[analyst@secOps ~]$ **ls -l**

total 28

drwxr-xr-x 3 analyst analyst 4096 Aug 16 15:15 cyops\_folder2

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 3 analyst analyst 4096 Mar 3 18:23 second\_drive

-rw-r--r-- 1 analyst analyst 142 Aug 16 15:11 some\_text\_file.txt

-rw-r--r-- 1 analyst analyst 254 Aug 16 13:38 space.txt

* + 1. 生成 **/dev** 目录列表。滚动到输出的中间，注意块文件以“b”开头，字符设备文件如何以“c”开头，符号链接文件以“l”开头：

[analyst@secOps ~]$ **ls -l /dev/**

<output omitted>

crw-rw-rw- 1 root tty 5, 2 May 29 18:32 ptmx

drwxr-xr-x 2 root root 0 May 23 06:40 pts

crw-rw-rw- 1 root root 1, 8 May 23 06:41 random

crw-rw-r-- 1 root root 10, 56 May 23 06:41 rfkill

lrwxrwxrwx 1 root root 4 May 23 06:41 rtc -> rtc0

crw-rw---- 1 root audio 253, 0 May 23 06:41 rtc0

brw-rw---- 1 root disk 8, 0 May 23 06:41 sda

brw-rw---- 1 root disk 8, 1 May 23 06:41 sda1

brw-rw---- 1 root disk 8, 16 May 23 06:41 sdb

brw-rw---- 1 root disk 8, 17 May 23 06:41 sdb1

drwxrwxrwt 2 root root 40 May 28 13:47 shm

crw------- 1 root root 10, 231 May 23 06:41 snapshot

drwxr-xr-x 2 root root 80 May 23 06:41 snd

brw-rw----+ 1 root optical 11, 0 May 23 06:41 sr0

lrwxrwxrwx 1 root root 15 May 23 06:40 stderr -> /proc/self/fd/2

lrwxrwxrwx 1 root root 15 May 23 06:40 stdin -> /proc/self/fd/0

lrwxrwxrwx 1 root root 15 May 23 06:40 stdout -> /proc/self/fd/1

crw-rw-rw- 1 root tty 5, 0 May 29 17:36 tty

crw--w---- 1 root tty 4, 0 May 23 06:41 tty0

<output omitted>

* + 1. Linux 中的符号链接就像 Windows 中的快捷方式一样。Linux 中有两种链接：符号链接和硬链接。符号链接和硬链接的区别在于，符号链接文件指向另一个文件的名称，而硬链接文件指向另一个文件的内容。使用 echo 创建两个文件：

[analyst@secOps ~]$ **echo "symbolic" > file1.txt**

[analyst@secOps ~]$ **cat file1.txt**

symbolic

[analyst@secOps ~]$ **echo "hard" > file2.txt**

[analyst@secOps ~]$ **cat file2.txt**

hard

* + 1. 使用 **ln –s** 创建指向 file1.txt 的符号链接，使用 **ln** 创建指向 file2.txt 的硬链接：

[analyst@secOps ~]$ **ln –s file1.txt file1symbolic**

[analyst@secOps ~]$ **ln file2.txt file2hard**

* + 1. 使用 ls –l 命令并检查目录列表：

[analyst@secOps ~]$ **ls -l**

total 40

drwxr-xr-x 3 analyst analyst 4096 Aug 16 15:15 cyops\_folder2

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

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

lrwxrwxrwx 1 analyst analyst 9 Aug 17 16:43 file1symbolic -> file1.txt

-rw-r--r-- 1 analyst analyst 9 Aug 17 16:41 file1.txt

-rw-r--r-- 2 analyst analyst 5 Aug 17 16:42 file2hard

-rw-r--r-- 2 analyst analyst 5 Aug 17 16:42 file2.txt

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

drwxr-xr-x 3 analyst analyst 4096 Mar 3 18:23 second\_drive

-rw-r--r-- 1 analyst analyst 142 Aug 16 15:11 some\_text\_file.txt

-rw-r--r-- 1 analyst analyst 254 Aug 16 13:38 space.txt

请注意，文件 **file1symbolic** 如何在行的开头使用 **l** 并使用指向 **file1.txt** 的指针而成为符号链接。**file2hard** 似乎是常规文件，因为实际上它是恰好指向硬盘驱动器上与 **file2.txt** 相同的索引节点的常规文件。换言之，**file2hard** 指向与 **file2.txt** 相同的属性和磁盘块位置。

* + 1. 更改原始文件的名称：file1.txt 和 file2.txt，并注意它会给链接文件带来怎样的影响。

[analyst@secOps ~]$ **mv file1.txt file1new.txt**

[analyst@secOps ~]$ **mv file2.txt file2new.txt**

[analyst@secOps ~]$ **cat file1symbolic**

cat: file1symbolic: no such file or directory

[analyst@secOps ~]$ **cat file2hard**

hard

请注意，**file1symbolic** 现在是一个断开的符号链接，因为它指向的 **file1.txt** 文件的名称已更改，但硬链接文件 **file2hard** 仍然正常工作，因为它指向 **file2.txt** 的索引节点而不是其现在的名称 **file2new.txt**。

如果打开了文本编辑器并更改了 **file2new.txt** 中的文本，同学们认为 **file2hard** 会发生什么情况？

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

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

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

1. 思考

文件权限和所有权是 Linux 中最重要的两个属性。它们也是造成问题的常见原因。权限或所有权设置错误的文件将无法供需要访问它的程序使用。在这种情况下，该程序通常会中断并且会遇到错误。