```
[root@servera ~]# grep '#UserDir' /etc/httpd/conf.d/userdir.conf
#UserDir disabled
[root@servera ~]# grep '^ *UserDir' /etc/httpd/conf.d/userdir.conf
UserDir public_html
```

5. 启动并启用 Apache Web 服务，以使更改生效。

```
[root@servera ~]# systemctl enable --now httpd
```

6. 在另一个终端窗口中，以 student 身份登录。通过 SSH 连接到 servera。创建一些会从用户主目录发布的 Web 内容。

6.1. 在另一个终端窗口中，以 student 身份登录。使用 **ssh** 命令，以 student 用户身份登录 servera。

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

6.2. 使用 **mkdir** 命令，创建一个名为 **~/public_html** 的目录。

```
[student@servera ~]$ mkdir ~/public_html
```

6.3. 创建包含以下内容的文件 **index.html**：

```
[student@servera ~]$ echo 'This is student content on SERVERA.' > \
~/public_html/index.html
```

6.4. 使用 **chmod** 命令更改 student 主目录上的权限，以便 Apache 能够访问 **public_html** 子目录。

```
[student@servera ~]$ chmod 711 ~
```

7. 在 workstation 上打开 Web 浏览器，并尝试查看以下 URL：http://servera/~student/index.html。您会收到一条错误消息，提示您不具有访问该文件的权限。

8. 在具有 **root** 访问权限的终端窗口中，使用 **getsebool** 命令查看是否存在任何限制访问主目录的布尔值。

```
[root@servera ~]# getsebool -a | grep home
...output omitted...
httpd_enable_homedirs --> off
...output omitted...
```

9. 在具有 **root** 访问权限的终端窗口中，使用 **setsebool** 命令持久启用主目录访问权限。

```
[root@servera ~]# setsebool -P httpd_enable_homedirs on
```

海量视频题库 myitpub.com QQ:556462
www.52myit.com

▶ **10.** 再次尝试查看 `http://servera/~student/index.html`。您应看到消息：**`This is student content on SERVERA`**。

▶ **11.** 从 `servera` 退出。

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

完成

在 `workstation` 上，运行 **`lab selinux-booleans finish`** 脚本来完成本练习。

```
[student@workstation ~]$ lab selinux-booleans finish
```

本引导式练习到此结束。

海量视频题库 myitpub.com QQ:556546 2
www.52myit.com

调查和解决 SELINUX 问题

培训目标

学完本节后，您应能够：

- 使用 SELinux 日志分析工具。
- 使用 **sealert** 命令，在 SELinux 故障排除期间显示有用的信息。

对 SELINUX 问题进行故障排除

在 SELinux 阻止访问服务器上的文件时（您知道本应该可以访问的），了解必须采取哪些操作至关重要。请将以下步骤作为解决这些问题的指南：

1. 在考虑做任何调整之前，应了解到 SELinux 禁止意图访问的这一做法也许非常正确。当 Web 服务器尝试访问 **/home** 中的文件时，如果用户并未发布 Web 内容，则可能表明服务遭入侵。如果已授予访问权限，则需要采取其他步骤来解决此问题。

2. 最常见的 SELinux 问题是使用不正确的文件上下文。此问题会在以下情况中发生：即，使用一个文件上下文在某个位置创建了文件，而该文件又被移至预期会使用其他上下文的地方。在大多数情况下，运行 **restorecon** 将会更正此问题。以这种方式更正问题对系统剩余部分的安全性具有非常小的影响。

3. 对于严苛限制性访问的另一个补救措施可以是调整布尔值。例如，**ftpd_anon_write** 布尔值控制匿名 FTP 用户能否上传文件。您必须打开此布尔值以允许匿名 FTP 用户将文件上传到服务器。调整布尔值时需格外谨慎，因为布尔值会对系统的安全性造成广泛影响。

4. SELinux 策略可能存在阻止合法访问的漏洞。由于 SELinux 技术已经成熟，这种情况极少发生。一旦明确了某个策略漏洞，请联系红帽支持并汇报此漏洞，以便问题得到解决。

监控 SELINUX 冲突

安装 setroubleshoot-server 软件包，以便将 SELinux 消息发送至 **/var/log/messages**。**setroubleshoot-server** 侦听 **/var/log/audit/audit.log** 中的审核消息，并发送简短摘要到 **/var/log/messages**。该摘要包括 SELinux 冲突的唯一标识符 (UUID)，可用于收集更多信息。**sealert -l UUID** 命令用于生成特定事件的报告。使用 **sealert -a /var/log/audit/audit.log** 可以在该文件中生成所有事件的报告。

考虑在标准 Apache Web 服务器上执行以下示例命令序列的情况：

```
[root@host ~]# touch /root/file3
[root@host ~]# mv /root/file3 /var/www/html
[root@host ~]# systemctl start httpd
[root@host ~]# curl http://localhost/file3
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
```

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

```
<p>You don't have permission to access /file3
on this server.</p>
</body></html>
```

您期望 Web 服务器提供 **file3** 的内容，但它却返回 **permission denied** 错误。查看 **/var/log/audit/audit.log** 和 **/var/log/messages** 可以发现关于此错误的一些额外信息。

```
[root@host ~]# tail /var/log/audit/audit.log
...output omitted...
type=AVC msg=audit(1392944135.482:429): avc:  denied  { getattr } for
  pid=1609 comm="httpd" path="/var/www/html/file3" dev="vda1" ino=8980981
  scontext=system_u:system_r:httpd_t:s0
  tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file
...output omitted...
[root@host ~]# tail /var/log/messages
...output omitted...
Feb 20 19:55:42 host setroubleshoot: SELinux is preventing /usr/sbin/httpd
  from getattr access on the file . For complete SELinux messages. run
  sealert -l 613ca624-248d-48a2-a7d9-d28f5bbe2763
```

这两个日志文件都指出 SELinux 拒绝是问题所在。作为 **/var/log/messages** 输出的一部分，**sealert** 命令可以提供一些额外信息，包括可能的修复办法。

```
[root@host ~]# sealert -l 613ca624-248d-48a2-a7d9-d28f5bbe2763
SELinux is preventing /usr/sbin/httpd from getattr access on the file .

*****  Plugin catchall (100. confidence) suggests   **************************

If you believe that httpd should be allowed getattr access on the
  file by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# grep httpd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp


Additional Information:
Source Context                system_u:system_r:httpd_t:s0
Target Context                unconfined_u:object_r:admin_home_t:s0
Target Objects                 [ file ]
Source                        httpd
Source Path                   /usr/sbin/httpd
Port                          <Unknown>
Host                          servera
Source RPM Packages           httpd-2.4.6-14.el7.x86_64
Target RPM Packages
Policy RPM                    selinux-policy-3.12.1-124.el7.noarch
Selinux Enabled               True
Policy Type                   targeted
Enforcing Mode                Enforcing
Host Name                     servera
```

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

```
Platform                    Linux servera 3.10.0-84.el7.x86_64 #1
                            SMP Tue Feb 4 16:28:19 EST 2014 x86_64 x86_64
Alert Count                 2
First Seen                  2014-02-20 19:55:35 EST
Last Seen                   2014-02-20 19:55:35 EST
Local ID                    613ca624-248d-48a2-a7d9-d28f5bbe2763

Raw Audit Messages
type=AVC msg=audit(1392944135.482:429): avc:  denied  { getattr } for
  pid=1609 comm="httpd" path="/var/www/html/file3" dev="vda1" ino=8980981
  scontext=system_u:system_r:httpd_t:s0
  tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file

type=SYSCALL msg=audit(1392944135.482:429): arch=x86_64 syscall=lstat
  success=no exit=EACCES a0=7f9fed0edea8 a1=7fff7bffc770 a2=7fff7bffc770
  a3=0 items=0 ppid=1608 pid=1609 auid=4294967295 uid=48 gid=48 euid=48
  suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295
  comm=httpd exe=/usr/sbin/httpd subj=system_u:system_r:httpd_t:s0 key=(null)

Hash: httpd,httpd_t,admin_home_t,file,getattr
```

注意

Raw Audit Messages 部分显示，目标文件 **/var/www/html/file3** 正是问题所在。同样，目标上下文 **tcontext** 似乎并不属于 Web 服务器。使用 **restorecon /var/www/html/file3** 命令修复此文件上下文。如果还需要调整其他文件，**restorecon** 可递归地重置上下文：**restorecon -R /var/www/**。

sealert 命令的 **Raw Audit Messages** 部分包含来自 **/var/log/audit.log** 的信息。要搜索 **/var/log/audit.log** 文件，请使用 **ausearch** 命令。**-m** 将按消息类型执行搜索。**-ts** 选项则根据时间执行搜索。

```
[root@host ~]# ausearch -m AVC -ts recent
----
time->Tue Apr  9 13:13:07 2019
type=PROCTITLE msg=audit(1554808387.778:4002):
 proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1554808387.778:4002): arch=c000003e syscall=49
 success=no exit=-13 a0=3 a1=55620b8c9280 a2=10 a3=7ffed967661c items=0
 ppid=1 pid=9340 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd"
 subj=system_u:system_r:httpd_t:s0 key=(null)
type=AVC msg=audit(1554808387.778:4002): avc:  denied  { name_bind }
 for  pid=9340 comm="httpd" src=82 scontext=system_u:system_r:httpd_t:s0
 tcontext=system_u:object_r:reserved_port_t:s0 tclass=tcp_socket permissive=0
```

WEB 控制台

如果安装了 Web 控制台，它也可用于对 SELinux 问题进行故障排除。登录 Web 控制台并从左侧的菜单中选择 SELinux。SELinux 策略窗口会通知您当前的执行策略。所有的问题都会在 SELinux Access Control Errors 部分进行详细说明。

海量视频题库 itpub.com QQ:556462
www.52mzit.com

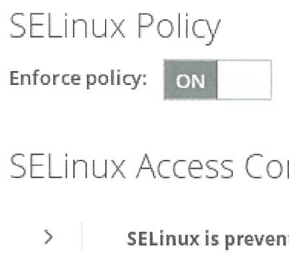

图 5.3: Web 控制台中的 SELinux 策略

单击 **>** 字符显示错误详细信息。单击 solution details 显示所有详细信息及可能的解决方案。

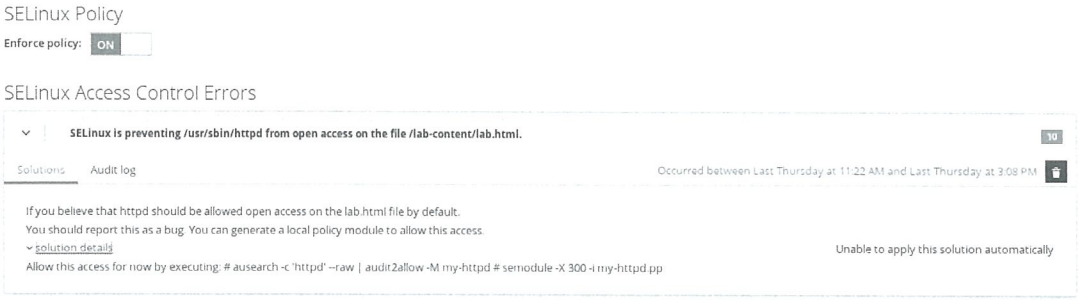

图 5.4: Web 控制台中的 SELinux 策略解决方案

一旦问题得到解决，SELinux Access Control Errors 部分便不再显示错误。如果显示消息 **No `SELinux alerts`**，则说明所有问题均已解决。

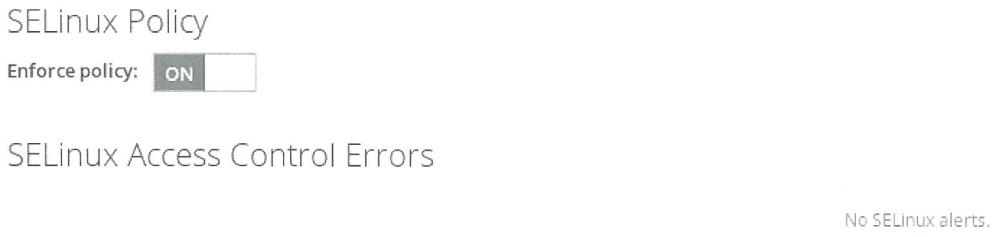

图 5.5: Web 控制台中的“No SELinux Alerts”消息

参考文献

`sealert` (8) man page

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

指导练习

调查和解决 SELINUX 问题

在本实验中，您将学习如何对 SELinux 安全拒绝问题进行故障排除。

成果

您将获得 SELinux 故障排除工具的一些使用经验。

在你开始之前

在 workstation 上，以 student 用户身份并使用密码 student 进行登录。

在 workstation 上，运行 **lab selinux-issues start** 命令。此命令将运行一个起始脚本，它将确定 servera 计算机是否可从网络访问。它将安装 httpd 服务，配置 servera 上的防火墙以允许进行 HTTP 连接，以及删除 **/custom** 目录的 SELinux 上下文。

```
[student@workstation ~]$ lab selinux-issues start
```

1. 使用 **ssh** 命令，以 student 用户身份登录 servera。系统已配置为使用 SSH 密钥来进行身份验证，因此不需要提供密码。

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

2. 使用 **sudo -i** 命令，切换为 root 用户。student 用户的密码为 student。

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

3. 在 workstation 上打开 Web 浏览器，并尝试查看 http://servera/index.html。您将会得到一个错误消息，提示您不具有访问该文件的权限。

4. 使用 **less** 命令，查看 **/var/log/messages** 的内容。使用 **/** 键并搜索 **sealert**。复制建议的 **sealert** 命令，以便它可以在下一步中使用。使用 **q** 键退出 **less** 命令。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

```
[root@servera ~]# less /var/log/messages
...output omitted...
Mar 28 06:07:03 servera setroubleshoot[15326]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /custom/index.html. For complete SELinux
 messages run: sealert -l b1c9cc8f-a953-4625-b79b-82c4f4f1fee3
Mar 28 06:07:03 servera platform-python[15326]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /custom/index.html.#012#012*****  Plugin
 catchall (100. confidence) suggests   **************************#012#012If
 you believe that httpd should be allowed getattr access on the index.html file
 by default.#012Then you should report this as a bug.#012You can generate a
 local policy module to allow this access.#012Do#012allow this access for now by
 executing:#012# ausearch -c 'httpd' --raw | audit2allow -M my-httpd#012# semodule
 -X 300 -i my-httpd.pp#012
Mar 28 06:07:04 servera setroubleshoot[15326]: failed to retrieve rpm info for /
custom/index.html
...output omitted...
```

5. 运行所建议的 **sealert** 命令。请注意源上下文、目标对象、策略及执行模式。

```
[root@servera ~]# sealert -l b1c9cc8f-a953-4625-b79b-82c4f4f1fee3
SELinux is preventing /usr/sbin/httpd from getattr access on the file /custom/
index.html.

*****  Plugin catchall (100. confidence) suggests   **************************

If you believe that httpd should be allowed getattr access on the index.html file
 by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# ausearch -c 'httpd' --raw | audit2allow -M my-httpd
# semodule -X 300 -i my-httpd.pp


Additional Information:
Source Context                system_u:system_r:httpd_t:s0
Target Context                unconfined_u:object_r:default_t:s0
Target Objects                /custom/index.html [ file ]
Source                        httpd
Source Path                   /usr/sbin/httpd
Port                          <Unknown>
Host                          servera.lab.example.com
Source RPM Packages
Target RPM Packages
Policy RPM                    selinux-policy-3.14.1-59.el8.noarch
Selinux Enabled               True
Policy Type                   targeted
Enforcing Mode                Enforcing
Host Name                     servera.lab.example.com
Platform                      Linux servera.lab.example.com 4.18.0-67.el8.x86_64
                              #1 SMP Sat Feb 9 12:44:00 UTC 2019 x86_64 x86_64
```

海量视频题库 mvitpub.com QQ:5565462
www.52mvit.com

```
                                Alert Count                    18
                                First Seen                     2019-03-25 19:25:28
 CET
                                Last Seen                      2019-03-28 11:07:00
 CET
                                Local ID                       b1c9cc8f-a953-4625-
b79b-82c4f4f1fee3

                                Raw Audit Messages
                                type=AVC msg=audit(1553767620.970:16958):
 avc:  denied  { getattr } for  pid=15067 comm="httpd" path="/custom/
index.html" dev="vda1" ino=4208311 scontext=system_u:system_r:httpd_t:s0
 tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0


                                Hash: httpd,httpd_t,default_t,file,getattr
```

6. **sealert** 命令的 **Raw Audit Messages** 部分包含来自 **/var/log/audit/audit.log** 的信息。使用 **ausearch** 命令，搜索 **/var/log/audit/audit.log** 文件。**-m** 选项将按消息类型执行搜索。**-ts** 选项则根据时间执行搜索。此条目标识了导致警报的相关进程和文件。进程是 **httpd** Apache Web 服务器，文件是 **/custom/index.html**，上下文是 **system_r:httpd_t**。

```
[root@servera ~]# ausearch -m AVC -ts recent
----
time->Thu Mar 28 13:39:30 2019
type=PROCTITLE msg=audit(1553776770.651:17000):
 proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1553776770.651:17000): arch=c000003e syscall=257
 success=no exit=-13 a0=ffffff9c a1=7f8db803f598 a2=80000 a3=0 items=0 ppid=15063
 pid=15065 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48
 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd"
 subj=system_u:system_r:httpd_t:s0 key=(null)
type=AVC msg=audit(1553776770.651:17000): avc:  denied
 { open } for  pid=15065 comm="httpd" path="/custom/index.html"
 dev="vda1" ino=4208311 scontext=system_u:system_r:httpd_t:s0
 tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0
```

7. 要解决此问题，请使用 **semanage** 和 **restorecon** 命令。要管理的上下文是 **httpd_sys_content_t**。

```
[root@servera ~]# semanage fcontext -a -t httpd_sys_content_t '/custom(/.*)?'
[root@servera ~]# restorecon -Rv /custom
Relabeled /custom from unconfined_u:object_r:default_t:s0 to
 unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled /custom/index.html from unconfined_u:object_r:default_t:s0 to
 unconfined_u:object_r:httpd_sys_content_t:s0
```

8. 再次尝试查看 http://servera/index.html。您应当看到显示的消息 **This is SERVERA**。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

▶ **9.** 从 servera 退出。

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

完成

在 workstation 上，运行 **lab selinux-issues finish** 脚本来完成本练习。

```
[student@workstation ~]$ lab selinux-issues finish
```

本引导式练习到此结束。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

开放研究实验

管理 SELINUX 安全性

任务执行清单

在本实验中，您将解决 SELinux 访问被拒的问题。在 SELinux 处于强制模式时，系统管理员使用新 Web 服务器向客户端交付内容时遇到问题。

成果

您应能够：

- 识别系统日志文件中的问题。
- 调整 SELinux 配置。

在你开始之前

以 `student` 用户身份并使用 `student` 作为密码登录 `workstation`。

在 `workstation` 上，运行 **`lab selinux-review start`** 命令。此命令将运行一个起始脚本，它将确定 `serverb` 计算机是否可从网络访问。此外，它还将安装 httpd Apache 服务器，为 Apache 创建新的文档根目录，以及更新配置文件。

```
[student@workstation ~]$ lab selinux-review start
```

1. 以 root 用户身份登录 `serverb`。
2. 在 `workstation` 上启动 Web 浏览器并浏览到 `http://serverb/lab.html`。您将看到错误消息：**`You do not have permission to access/lab.html on this server.`**
3. 研究并确定正在阻止 Apache 提供 Web 内容服务的 SELinux 问题。
4. 显示新 HTTP 文档根目录和原始 HTTP 文档根目录的 SELinux 上下文。解决正在阻止 Apache 提供 Web 内容服务的 SELinux 问题。
5. 验证 SELinux 问题已解决，并且 Apache 能够提供 Web 内容服务。
6. 从 `serverb` 退出。

评估

在 `workstation` 上，运行 **`lab selinux-review grade`** 脚本来确认是否成功完成本练习。

```
[student@workstation ~]$ lab selinux-review grade
```

完成

在 `workstation` 上，运行 **`lab selinux-review finish`** 脚本来完成实验。

```
[student@workstation ~]$ lab selinux-review finish
```

海量视频题库 myitpub.com QQ:556546 2
www.52myit.com

本实验到此结束。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

解决方案

管理 SELINUX 安全性

任务执行清单

在本实验中，您将解决 SELinux 访问被拒的问题。在 SELinux 处于强制模式时，系统管理员使用新 Web 服务器向客户端交付内容时遇到问题。

成果

您应能够：

- 识别系统日志文件中的问题。
- 调整 SELinux 配置。

在你开始之前

以 `student` 用户身份并使用 `student` 作为密码登录 `workstation`。

在 `workstation` 上，运行 **`lab selinux-review start`** 命令。此命令将运行一个起始脚本，它将确定 `serverb` 计算机是否可从网络访问。此外，它还将安装 httpd Apache 服务器，为 Apache 创建新的文档根目录，以及更新配置文件。

```
[student@workstation ~]$ lab selinux-review start
```

1. 以 root 用户身份登录 `serverb`。

   1.1. 使用 **`ssh`** 命令，以 `student` 用户身份登录 `serverb`。系统已配置为使用 SSH 密钥来进行身份验证，因此不需要提供密码。

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

   1.2. 使用 **`sudo -i`** 命令，切换为 `root` 用户。`student` 用户的密码为 `student`。

```
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

2. 在 `workstation` 上启动 Web 浏览器并浏览到 `http://serverb/lab.html`。您将看到错误消息：**`You do not have permission to access/lab.html on this server.`**
3. 研究并确定正在阻止 Apache 提供 Web 内容服务的 SELinux 问题。

   3.1. 使用 **`less`** 命令，查看 **`/var/log/messages`** 的内容。使用 **`/`** 键并搜索 **`sealert`**。使用 **`q`** 键退出 **`less`** 命令。

海量视频题库 myitpub.com QQ:556462 www.52myit.com

```
[root@serverb ~]# less /var/log/messages
Mar 28 10:19:51 serverb setroubleshoot[27387]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /lab-content/lab.html. For complete SELinux
 messages run: sealert -l 8824e73d-3ab0-4caf-8258-86e8792fee2d
Mar 28 10:19:51 serverb platform-python[27387]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /lab-content/lab.html.#012#012*****  Plugin
 catchall (100. confidence) suggests   **************************#012#012If
 you believe that httpd should be allowed getattr access on the lab.html file
 by default.#012Then you should report this as a bug.#012You can generate a
 local policy module to allow this access.#012Do#012allow this access for now by
 executing:#012# ausearch -c 'httpd' --raw | audit2allow -M my-httpd#012# semodule
 -X 300 -i my-httpd.pp#012
```

3.2. 运行所建议的 **sealert** 命令。请注意源上下文、目标对象、策略及执行模式。

```
[root@serverb ~]# sealert -l 8824e73d-3ab0-4caf-8258-86e8792fee2d
SELinux is preventing /usr/sbin/httpd from getattr access on the file /lab-
content/lab.html.

*****  Plugin catchall (100. confidence) suggests   **************************

If you believe that httpd should be allowed getattr access on the lab.html file by
 default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# ausearch -c 'httpd' --raw | audit2allow -M my-httpd
# semodule -X 300 -i my-httpd.pp


Additional Information:
Source Context                system_u:system_r:httpd_t:s0
Target Context                unconfined_u:object_r:default_t:s0
Target Objects                /lab-content/lab.html [ file ]
Source                        httpd
Source Path                   /usr/sbin/httpd
Port                          <Unknown>
Host                          serverb.lab.example.com
Source RPM Packages
Target RPM Packages
Policy RPM                    selinux-policy-3.14.1-59.el8.noarch
Selinux Enabled               True
Policy Type                   targeted
Enforcing Mode                Enforcing
Host Name                     serverb.lab.example.com
Platform                      Linux serverb.lab.example.com 4.18.0-67.el8.x86_64
                              #1 SMP Sat Feb 9 12:44:00 UTC 2019 x86_64 x86_64
                              Alert Count                   2
                              First Seen                    2019-03-28 15:19:46
 CET
                              Last Seen                     2019-03-28 15:19:46
 CET
```

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

```
                                Local ID
 8824e73d-3ab0-4caf-8258-86e8792fee2d

                                Raw Audit Messages
                                type=AVC msg=audit(1553782786.213:864):
 avc:  denied  { getattr } for  pid=15606 comm="httpd" path="/lab-content/
lab.html" dev="vda1" ino=8763212 scontext=system_u:system_r:httpd_t:s0
 tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0


Hash: httpd,httpd_t,default_t,file,getattr
```

3.3. **sealert** 命令的 **Raw Audit Messages** 部分包含来自 **/var/log/audit/audit.log** 的信息。使用 **ausearch** 命令，搜索 **/var/log/audit/audit.log** 文件。**-m** 选项将按消息类型执行搜索。**ts** 选项则根据时间执行搜索。此条目标识了导致警报的相关进程和文件。进程是 **httpd** Apache Web 服务器，文件是 **/lab-content/lab.html**，上下文是 **system_r:httpd_t**。

```
[root@serverb ~]# ausearch -m AVC -ts recent
time->Thu Mar 28 15:19:46 2019
type=PROCTITLE msg=audit(1553782786.213:864):
 proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1553782786.213:864): arch=c000003e syscall=6 success=no
 exit=-13 a0=7fb900004930 a1=7fb92dfca8e0 a2=7fb92dfca8e0 a3=1 items=0 ppid=15491
 pid=15606 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48
 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd"
 subj=system_u:system_r:httpd_t:s0 key=(null)
type=AVC msg=audit(1553782786.213:864): avc:  denied  { getattr } for
 pid=15606 comm="httpd" path="/lab-content/lab.html" dev="vda1" ino=8763212
 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0
 tclass=file permissive=0
```

4. 显示新 HTTP 文档根目录和原始 HTTP 文档根目录的 SELinux 上下文。解决正在阻止 Apache 提供 Web 内容服务的 SELinux 问题。

4.1. 使用 **ls -dZ** 比较 **/lab-content** 和 **/var/www/html** 的文档根目录。

```
[root@serverb ~]# ls -dZ /lab-content /var/www/html
unconfined_u:object_r:default_t:s0 /lab-content/
 system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
```

4.2. 创建一条文件上下文规则，以便将 **/lab-content** 及其下面的所有文件的默认类型设置为 **httpd_sys_content_**。

```
[root@serverb ~]# semanage fcontext -a -t httpd_sys_content_t '/lab-content(/.*)?'
```

4.3. 使用 **restorecon** 命令，为 **/lab-content** 中的文件设置 SELinux 上下文。

```
[root@serverb ~]# restorecon -R /lab-content/
```

5. 验证 SELinux 问题已解决，并且 Apache 能够提供 Web 内容服务。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

使用您的 Web 浏览器刷新 `http://serverb/lab.html` 链接。现在，您应看到一些 Web 内容。

```
This is the html file for the SELinux final lab on SERVERB.
```

6. 从 `serverb` 退出。

```
[root@serverb ~]# exit
logout
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

评估

在 `workstation` 上，运行 **`lab selinux-review grade`** 脚本来确认是否成功完成本练习。

```
[student@workstation ~]$ lab selinux-review grade
```

完成

在 `workstation` 上，运行 **`lab selinux-review finish`** 脚本来完成实验。

```
[student@workstation ~]$ lab selinux-review finish
```

本实验到此结束。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

总结

在本章中，您学到了：

- 使用 **getenforce** 和 **setenforce** 命令来管理系统的 SELinux 模式。
- 使用 **semanage** 命令来管理 SELinux 策略规则。使用 **restorecon** 命令来应用策略所定义的上下文。
- 布尔值是可更改 SELinux 策略行为的参数。您可以启用或禁用布尔值，也可以使用它们来调整策略。
- **sealert** 可显示有助于 SELinux 故障排除的有用信息。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

章 6

管理基本存储

目标	使用命令行创建和管理存储设备、分区、文件系统和交换空间。
培训目标	· 创建存储分区，将其格式化为相应的文件系统，然后进行挂载以供使用。 · 创建和管理交换空间以补充物理内存。
章节	· 添加分区、文件系统和持久挂载（及引导式练习） · 管理交换空间（及引导式练习）
实验	管理基本存储

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

添加分区、文件系统和持久挂载

培训目标

学完本节后，您应能够创建存储分区，将其格式化为相应的文件系统，然后进行挂载以供使用。

对磁盘进行分区

磁盘分区允许系统管理员将硬盘驱动器划分为多个逻辑存储单元，这些单元称为分区。通过将磁盘划分为多个分区，系统管理员可以使用不同的分区执行不同功能。

例如，在以下情况下，磁盘分区是必要或有益的：

- 限制应用或用户的可用空间。
- 将操作系统和程序文件与用户文件分隔开。
- 创建用于内存交换的单独区域。
- 限制磁盘空间使用，以提高诊断工具和备份映像的性能。

MBR 分区方案

自 1982 年以来，主启动记录 (MBR) 分区方案指定了在运行 BIOS 固件的系统上如何对磁盘进行分区。此方案支持最多四个主分区。在 Linux 系统上，管理员可以使用扩展分区和逻辑分区来创建最多 15 个分区。由于分区大小数据以 32 位值存储，使用 MBR 方案分区时，最大磁盘和分区大小为 2 TiB。

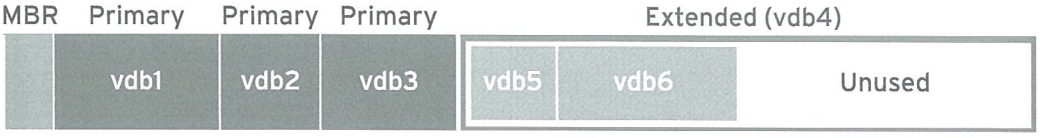

图 6.1: /dev/vdb 存储设备的 MBR 分区

由于物理磁盘变得越来越大，而基于 SAN 的卷甚至更大，因此针对 MBR 分区方案的 2 TiB 磁盘和分区大小限制已不再是理论限制，而是系统管理员在生产环境中越来越频繁遇到的实际问题。因此，新的 GUID 分区表 (GPT) 正在取代传统的 MBR 方案用于磁盘分区。

GPT 分区方案

对于运行统一可扩展固件接口 (UEFI) 固件的系统，GPT 是在物理硬盘上布置分区表的标准。GPT 是 UEFI 标准的一部分，可以解决原有基于 MBR 的方案所带来的许多限制。

GPT 最多可提供 128 个分区。MBR 使用 32 位来存储逻辑块地址和大小信息；与此不同，GPT 为逻辑块地址分配 64 位。这就使得 GPT 可支持最多 8 泽字节 (ZiB)，即 80 亿太字节的分区和磁盘。

除解决 MBR 分区方案的限制以外，GPT 还可提供一些其他功能特性和优势。GPT 使用全局唯一标识符 (GUID) 来识别每个磁盘和分区。与 MBR 存在单一故障点不同，GPT 提供分区表信息的冗余。主 GPT 位于磁盘头部，而备份副本（次要 GPT）位于磁盘尾部。GPT 使用校验和来检测 GPT 头和分区表中的错误与损坏。

海量视频题库 itpub.com QQ:556546 2
www.52nit.com

图 6.2: /dev/vdb 存储设备的 GPT 分区

使用 PARTED 管理分区

借助分区编辑器程序，管理员可对磁盘的分区进行更改，例如创建分区、删除分区和更改分区类型。要执行这些操作，管理员可以对 MBR 和 GPT 分区方案使用 Parted 分区编辑器。

parted 命令将整个磁盘的设备名称作为第一个参数，并且还有一个或多个子命令。以下示例使用 **print** 子命令显示 **/dev/vda** 磁盘上的分区表。

```
[root@host ~]# parted /dev/vda print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 53.7GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      1049kB  10.7GB  10.7GB  primary  xfs          boot
 2      10.7GB  53.7GB  42.9GB  primary  xfs
```

如果不提供子命令，**parted** 就会打开用于发出命令的交互式会话。

```
[root@host ~]# parted /dev/vda
GNU Parted 3.2
Using /dev/vda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 53.7GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      1049kB  10.7GB  10.7GB  primary  xfs          boot
 2      10.7GB  53.7GB  42.9GB  primary  xfs

(parted) quit
[root@host ~]#
```

默认情况下，**parted** 显示以 10 的幂次方表示的所有空间大小（KB、MB、GB）。您可以使用 **unit** 子命令来更改默认设置，该子命令接受以下参数：

- **s** 表示扇区
- **B** 表示字节

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

- **MIB**、**GiB** 或 **TiB**（2 的幂次方）
- **MB**、**GB** 或 **TB**（10 的幂次方）

```
[root@host ~]# parted /dev/vda unit s print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start       End         Size        Type      File system  Flags
 1      2048s       20971486s   20969439s   primary   xfs          boot
 2      20971520s   104857535s  83886016s   primary   xfs
```

如上例所示，您还可以在同一行上指定多个子命令（此处的 **unit** 和 **print**）。

向新磁盘写入分区表

要对新驱动器进行分区，首先必须为其写入磁盘标签。磁盘标签指示了所用的分区方案。

> **注意**
> 请记住，**parted** 会令更改立即生效。如果误用了 **parted**，注定会导致数据丢失。

以 root 用户身份，使用以下命令将 MBR 磁盘标签写入磁盘。

```
[root@host ~]# parted /dev/vdb mklabel msdos
```

要写入 GPT 磁盘标签，请使用以下命令。

```
[root@host ~]# parted /dev/vdb mklabel gpt
```

> **警告**
> **mklabel** 子命令可擦除现有的分区表。仅当想重复使用磁盘而不考虑现有数据时，才应使用 **mklabel**。如果新标签更改了分区边界，则现有文件系统中的所有数据都将无法访问。

创建 MBR 分区

创建 MBR 磁盘分区涉及到几个步骤：

1. 指定要在其上创建分区的磁盘设备。

   以 root 用户身份执行 **parted** 命令，并指定该磁盘设备名称作为参数。这样将会以交互模式启动 **parted** 命令并显示命令提示符。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

2. 使用 **mkpart** 子命令创建新的主分区或扩展分区。

```
(parted) mkpart
Partition type?  primary/extended? primary
```

> **注意**
> 对于在 MBR 分区的磁盘上需要超过四个分区的情况，可以创建三个主分区和一个扩展分区。此扩展分区用作一个容器，可在其中创建多个逻辑分区。

3. 指示要在分区上创建的文件系统类型，如 **xfs** 或 **ext4**。这并不会在分区上创建文件系统；它仅仅指示分区类型。

```
File system type?  [ext2]? xfs
```

要获取受支持的文件系统类型的列表，请使用以下命令：

```
[root@host ~]# parted /dev/vdb help mkpart
  mkpart PART-TYPE [FS-TYPE] START END     make a partition

    PART-TYPE is one of: primary, logical, extended
    FS-TYPE is one of: btrfs, nilfs2, ext4, ext3, ext2, fat32, fat16, hfsx,
    hfs+, hfs, jfs, swsusp, linux-swap(v1), linux-swap(v0), ntfs, reiserfs,
    hp-ufs, sun-ufs, xfs, apfs2, apfs1, asfs, amufs5, amufs4, amufs3,
    amufs2, amufs1, amufs0, amufs, affs7, affs6, affs5, affs4, affs3, affs2,
    affs1, affs0, linux-swap, linux-swap(new), linux-swap(old)
    START and END are disk locations, such as 4GB or 10%.  Negative values
    count from the end of the disk.  For example, -1s specifies exactly the
    last sector.

    'mkpart' makes a partition without creating a new file system on the
    partition.  FS-TYPE may be specified to set an appropriate partition
    ID.
```

4. 指定磁盘上新分区开始的扇区。

```
Start? 2048s
```

请注意，**s** 后缀提供了扇区的值。您也可以使用 **MIB**、**GiB**、**TiB**、**MB**、**GB** 或 **TB** 后缀。如果未提供后缀，则默认为 **MB**。**parted** 可以舍入您所提供的值，以满足磁盘的限制条件。

当 **parted** 启动时，它将检索设备的磁盘拓扑。例如，磁盘的物理块大小通常是一个由 **parted** 收集的参数。利用这些信息，**parted** 可确保您所提供的起始位置能让分区与磁

海量视频题库 myitpub.com QQ:556462
www.520vit.com

盘结构正确对齐。正确的分区对齐对于获得最佳性能至关重要。如果起始位置导致分区错位，**parted** 就会显示警告。对于大多数磁盘而言，假设起始扇区为 2048 的倍数较为安全。

5. 指定应结束新分区的磁盘扇区。

```
End? 1000MB
```

通过 **parted** 并不能直接提供分区的大小，但可以使用以下公式进行快速计算：

```
Size = End - Start
```

一旦提供了结束位置，**parted** 即可利用新分区的详细信息来更新磁盘上的分区表。

6. 退出 **parted**。

```
(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

7. 运行 **udevadm settle** 命令。此命令会等待系统检测新分区并在 **/dev** 目录下创建关联的设备文件。只有在完成上述操作后，它才会返回。

```
[root@host ~]# udevadm settle
[root@host ~]#
```

作为交互模式的替代方法，您也可以按如下方式创建分区：

```
[root@host ~]# parted /dev/vdb mkpart primary xfs 2048s 1000MB
```

创建 GPT 分区

GTP 方案同样使用 **parted** 命令来创建新分区：

1. 指定要在其上创建分区的磁盘设备。

作为 root 用户，以磁盘设备作为唯一参数执行 **parted** 命令，从而通过命令提示符在交互模式下启动 **parted**。

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

2. 使用 **mkpart** 子命令开始创建新分区。

对于 GPT 方案而言，每个分区都会获得一个名称。

```
(parted) mkpart
Partition name?  []? usersdata
```

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

3. 指示要在分区上创建的文件系统类型，如 **xfs** 或 **ext4**。这并不会在分区上创建文件系统；它仅仅指示分区类型。

```
File system type?  [ext2]? xfs
```

4. 指定磁盘上新分区开始的扇区。

```
Start? 2048s
```

5. 指定应结束新分区的磁盘扇区。

```
End? 1000MB
```

一旦提供了结束位置，**parted** 即可利用新分区的详细信息来更新磁盘上的分区表。

6. 退出 **parted**。

```
(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

7. 运行 **udevadm settle** 命令。此命令会等待系统检测新分区并在 **/dev** 目录下创建关联的设备文件。只有在完成上述操作后，它才会返回。

```
[root@host ~]# udevadm settle
[root@host ~]#
```

作为交互模式的替代方法，您也可以按如下方式创建分区：

```
[root@host ~]# parted /dev/vdb mkpart usersdata xfs 2048s 1000MB
```

删除分区

以下步骤适用于 MBR 和 GPT 两种分区方案。

1. 指定包含要删除的分区的磁盘。

作为 root 用户，以磁盘设备作为唯一参数执行 **parted** 命令，从而通过命令提示符在交互模式下启动 **parted**。

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

2. 确定要删除的分区的分区编号。

海量视频题库 myitpub.com QQ:556462
www.52myit.com