

Отчет по лабораторной работе 5

Мандатное разграничение прав в Linux

Шалыгин Георгий Эдуардович

Содержание

1	Цель работы	5
2	Теоретическое введение	6
2.0.1	Изменение владельца	7
2.1	Использование chmod	7
3	Выполнение лабораторной работы	9
4	Выводы	16
	Список литературы	17

Список иллюстраций

3.1	Файл httpd.conf	9
3.2	Задание имени сервера	9
3.3	Проверка работы	10
3.4	Запуск веб-сервера	10
3.5	Контекст сервера	11
3.6	Состояние переключателей	11
3.7	Статистика по политике	12
3.8	Проверка контекста	12
3.9	Доступ к серверу	13
3.10	Изменение контекста	13
3.11	Доступа нет	13
3.12	Лог файл	14
3.13	Добавление порта	14
3.14	Возвращение контекста	14
3.15	Сервер доступен	15
3.16	Окончание работы	15

Список таблиц

1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux¹. Проверить работу SELinx на практике совместно с веб-сервером Apache.

2 Теоретическое введение

В Linux, как и в любой многопользовательской системе, абсолютно естественным образом возникает задача разграничения доступа субъектов — пользователей к объектам — файлам дерева каталогов.

Один из подходов к разграничению доступа — так называемый дискреционный (от англ. *discretion* — чье-либо усмотрение) — предполагает назначение владельцев объектов, которые по собственному усмотрению определяют права доступа субъектов (других пользователей) к объектам (файлам), которыми владеют.

Дискреционные механизмы разграничения доступа используются для разграничения прав доступа процессов как обычных пользователей, так и для ограничения прав системных программ в (например, служб операционной системы), которые работают от лица псевдопользовательских учетных записей.

В Linux у каждого файла и каждого каталога есть два владельца: пользователь и группа.

Эти владельцы устанавливаются при создании файла или каталога. Пользователь, который создаёт файл становится владельцем этого файла, а первичная группа, в которую входит этот же пользователь, так же становится владельцем этого файла. Чтобы определить, есть ли у вас как у пользователя права доступа к файлу или каталогу, оболочка проверяет владение ими.

Это происходит в следующем порядке:

1. Оболочка проверяет, являетесь ли вы владельцем файла, к которому вы хотите получить доступ. Если вы являетесь этим владельцем, вы получаете разрешения и оболочка прекращает проверку.

2. Если вы не являетесь владельцем файла, оболочка проверит, являетесь ли вы участником группы, у которой есть разрешения на этот файл. Если вы являетесь участником этой группы, вы получаете доступ к файлу с разрешениями, которые для группы установлены, и оболочка прекратит проверку.
3. Если вы не являетесь ни пользователем, ни владельцем группы, вы получаете права других пользователей (Other).

Чтобы увидеть текущие назначения владельца, вы можете использовать команду **ls -l**. Эта команда показывает пользователя и группу-владельца.

Подробнее в [1].

2.0.1 Изменение владельца

Чтобы применить соответствующие разрешения, первое, что нужно учитывать, это владение. Для этого есть команда **chown**. Синтаксис этой команды несложен для понимания:

```
chown кто что
```

Например, следующая команда меняет владельца каталога `/home/account` на пользователя `linda`:

```
chown linda /home/account
```

2.1 Использование **chmod**

Для управления правами используется команда **chmod**. При использовании **chmod** вы можете устанавливать разрешения для пользователя (user), группы (group) и других (other). Вы можете использовать эту команду в двух режимах: относительный режим и абсолютный режим. В абсолютном режиме три цифры используются для установки основных разрешений.

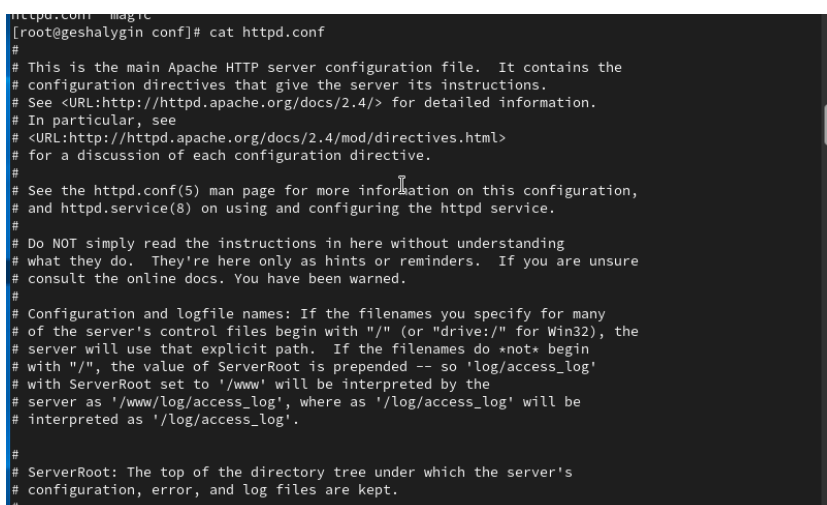
При настройке разрешений рассчитайте необходимое вам значение. Если вы хотите установить чтение, запись и выполнение для пользователя, чтение и выполнение для группы, а также чтение и выполнение для других в файле /somefile, то вы используете следующую команду **chmod**:

```
chmod 755 /somefile
```

Подробнее в [2].

3 Выполнение лабораторной работы

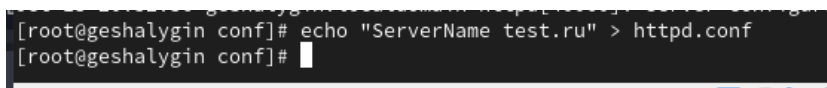
1. Файл /etc/httpd/httpd.conf (fig. 3.1).



```
[root@geshalygin conf]# cat httpd.conf
#
# This is the main Apache HTTP server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
#
# See the httpd.conf(5) man page for more information on this configuration,
# and httpd.service(8) on using and configuring the httpd service.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path. If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so 'log/access_log'
# with ServerRoot set to '/www' will be interpreted by the
# server as '/www/log/access_log', where as '/log/access_log' will be
# interpreted as '/log/access_log'.
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
```

Рис. 3.1: Файл httpd.conf

2. Также необходимо проследить, чтобы пакетный фильтр был отключён или в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола tcp. Отключить фильтр можно командами (fig. 3.2)



```
[root@geshalygin conf]# echo "ServerName test.ru" > httpd.conf
[root@geshalygin conf]#
```

Рис. 3.2: Задание имени сервера

3. Убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus.(fig. 3.3).

```
[root@geshalygin conf]# getenforce
Enforcing
[root@geshalygin conf]# sestatus
SELinux status:                enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:              targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
[root@geshalygin conf]#
```

Рис. 3.3: Проверка работы

4. Запустим веб-сервер: `service httpd start` (fig. 3.4).

```
[root@geshalygin conf]# service httpd status
Redirecting to /bin/systemctl status httpd.service
o httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
[root@geshalygin conf]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@geshalygin conf]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Fri 2023-10-13 20:32:36 MSK; 2s ago
     Docs: man:httpd.service(8)
  Main PID: 40060 (httpd)
    Status: "Started, listening on: port 80"
   Tasks: 213 (limit: 12210)
  Memory: 35.3M
    CPU: 108ms
  CGroup: /system.slice/httpd.service
          └─40060 /usr/sbin/httpd -DFOREGROUND
             └─40061 /usr/sbin/httpd -DFOREGROUND
                └─40065 /usr/sbin/httpd -DFOREGROUND
                   └─40066 /usr/sbin/httpd -DFOREGROUND
                      └─40068 /usr/sbin/httpd -DFOREGROUND

Oct 13 20:32:36 geshalygin.localdomain systemd[1]: Starting The Apache HTTP Server...
Oct 13 20:32:36 geshalygin.localdomain systemd[1]: Started The Apache HTTP Server.
Oct 13 20:32:36 geshalygin.localdomain httpd[40060]: Server configured, listening on: port 80
[root@geshalygin conf]#
```

Рис. 3.4: Запуск веб-сервера

5. Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности. Например, можно использовать команду `ps auxZ | grep httpd` (fig. 3.5).

```
[root@geshalygin conf]# echo "ServerName test.ru" > httpd.conf
[root@geshalygin conf]# ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 40060 0.0 0.5 20328 11672 ? Ss 20:32 0:00
/usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 40061 0.0 0.3 21664 7552 ? S 20:32 0:00
/usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 40065 0.0 0.8 1079476 17248 ? Sl 20:32 0:00
/usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 40066 0.0 0.8 1210612 17252 ? Sl 20:32 0:00
/usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 40068 0.0 0.6 1079476 13160 ? Sl 20:32 0:00
/usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 40385 0.0 0.1 221664 2244 pts/0 S+ 20:
40 0:00 grep --color=auto httpd
[root@geshalygin conf]#
```

Рис. 3.5: Контекст сервера

6. Посмотрите текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -b | grep httpd`. Многие отключены (fig. 3.6).

```
[root@geshalygin conf]# sestatus -b | grep httpd
httpd_anon_write off
httpd_builtin_scripting on
httpd_can_check_spam off
httpd_can_connect_ftp off
httpd_can_connect_ldap off
httpd_can_connect_mythtv off
httpd_can_connect_zabbix off
httpd_can_manage_courier_spool off
httpd_can_network_connect off
httpd_can_network_connect_cobbler off
httpd_can_network_connect_db off
httpd_can_network_memcache off
httpd_can_network_relay off
httpd_can_sendmail off
httpd_dbus_avahi off
httpd_dbus_sssd off
httpd_dontaudit_search_dirs off
httpd_enable_cgi on
httpd_enable_ftp_server off
httpd_enable_homedirs off
httpd_execmem off
httpd_graceful_shutdown off
httpd_manage_ipa off
httpd_mod_auth_ntlm_winbind off
httpd_mod_auth_pam off
httpd_read_user_content off
httpd_run_ipa off
httpd_run_upgrade off
```

Рис. 3.6: Состояние переключателей

7. Посмотрим статистику по политике с помощью команды `seinfo` (fig. 3.7).

```
[root@geshalygin conf]# seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:          33 (MLS enabled)
Target Policy:           selinux
Handle unknown classes:  allow

Classes:          135      Permissions:          457
Sensitivities:    1        Categories:          1024
Types:            5100     Attributes:           258
Users:            8        Roles:                14
Booleans:         353     Cond. Expr.:         384
Allow:            65000    Neverallow:           0
Auditallow:       170     Dontaudit:            8572
Type_trans:       265341   Type_change:          87
Type_member:       35      Range_trans:          6164
Role allow:       38       Role_trans:           420
Constraints:      70       Validatetrans:         0
MLS Constrains:  72       MLS Val. Tran:         0
Permissives:      2        Polcap:                6
Defaults:         7        Typebounds:            0
Allowxperm:        0       Neverallowxperm:       0
Auditallowxperm:   0       Dontauditxperm:        0
Ibendportcon:      0       Ibpkeycon:             0
Initial SIDs:      27      Fs_use:                35
Genfscon:          109     Portcon:               660
Netifcon:          0       Nodecon:               0

[root@geshalygin conf]#
```

Рис. 3.7: Статистика по политике

8. Определим тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды `ls -lZ /var/www`. (fig. 3.8). Проверим что guest не имеет доступ к файлу.

```
[root@geshalygin conf]# ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 May 16 23:21 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 May 16 23:21 html
[root@geshalygin conf]# ls -lZ /var/www/html
total 0
[root@geshalygin conf]#
```

Рис. 3.8: Проверка контекста

9. Создадим от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html

Обративсь к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Убедивсь, что файл был успешно отображён. (fig. 3.9).

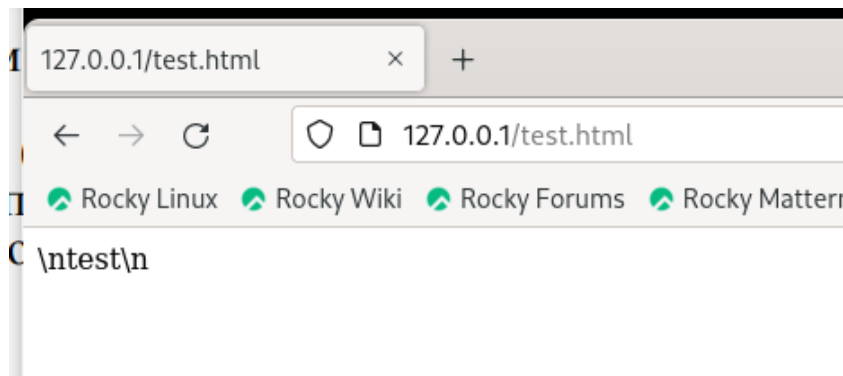


Рис. 3.9: Доступ к серверу

10. Изменив контекст файла `/var/www/html/test.html` с `httpd_sys_content_t` на `samba_share_t`: `chcon -t samba_share_t /var/www/html/test.html` `ls -Z /var/www/html/test.html` (fig. 3.10).

```
[root@geshalygin conf]# chcon -t samba_share_t /var/www/html/test.html
[root@geshalygin conf]# ls -Z /var/www/html/test.html
ls: cannot access '/var/www/html/test.html': No such file or directory
[root@geshalygin conf]# ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[root@geshalygin conf]#
```

Рис. 3.10: Изменение контекста

11. После этого файл недоступен (fig. 3.11).

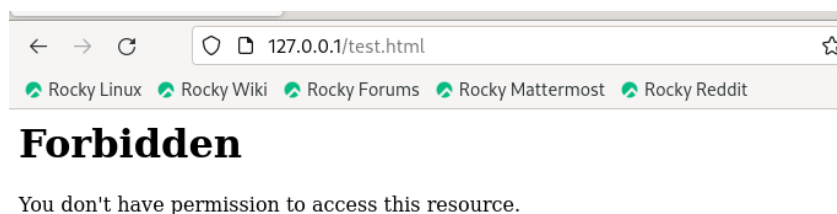


Рис. 3.11: Доступа нет

12. Просмотрим системный лог-файл: `tail /var/log/messages` (fig. 3.12).

```

l (1.41 confidence) suggests *****#012#012If you believe that httpd should
be allowed getattr access on the test.html file by default.#012Then you should report this as a bu
g.#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
Oct 13 20:51:34 geshalygin setroubleshoot[41243]: SELinux is preventing /usr/sbin/httpd from getat
tr access on the file /var/www/html/test.html.#012#012***** Plugin restorecon (92.2 confidence) s
uggests *****#012#012If you want to fix the label. #012/var/www/html/test.htm
l default label should be httpd_sys_content_t.#012Then you can run restorecon. The access attempt
may have been stopped due to insufficient permissions to access a parent directory in which case t
ry to change the following command accordingly.#012Do#012# /sbin/restorecon -v /var/www/html/test.
html#012#012***** Plugin public_content (7.83 confidence) suggests *****#012#012
If you want to treat test.html as public content#012Then you need to change the label on test.html
to public_content_t or public_content_rw_t.#012Do#012# semanage fcontext -a -t public_content_t '
/var/www/html/test.html'#012# restorecon -v '/var/www/html/test.html'#012#012***** Plugin catchal
l (1.41 confidence) suggests *****#012#012If you believe that httpd should
be allowed getattr access on the test.html file by default.#012Then you should report this as a bu
g.#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
Oct 13 20:51:44 geshalygin systemd[1]: dbus-1.1-org.fedoraproject.SetroubleshootPrivileged@0.serv
ice: Deactivated successfully.
Oct 13 20:51:44 geshalygin systemd[1]: dbus-1.1-org.fedoraproject.SetroubleshootPrivileged@0.serv
ice: Consumed 1.610s CPU time.
Oct 13 20:51:44 geshalygin systemd[1]: setroubleshootd.service: Deactivated successfully.
Oct 13 20:51:44 geshalygin systemd[1]: setroubleshootd.service: Consumed 1.172s CPU time.

```

Рис. 3.12: Лог файл

13. Выполним команду `semanage port -a -t http_port_t -p tcp 81`. После этого про-
верим список портов командой `semanage port -l | grep http_port_t`. Убедимся,
что порт 81 появился в списке. Теперь доступ к серверу есть, мы добавили
порт 81. (fig. 3.13).

```

[root@geshalygin conf]# semanage port -a -t http_port_t -p tcp 81
ValueError: Port tcp/81 already defined
[root@geshalygin conf]# semanage port -l | grep http_port_t
http_port_t          tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t  tcp      5988
[root@geshalygin conf]#

```

Рис. 3.13: Добавление порта

14. Вернем контекст `httpd_sys_content_t` к файлу `/var/www/html/ test.html`:
`chcon -t httpd_sys_content_t /var/www/html/test.html` После этого попро-
буем получить доступ к файлу через веб-сервер, введя в браузере адрес
`http://127.0.0.1:81/test.html`. (fig. 3.14).

```

[root@geshalygin conf]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@geshalygin conf]#

```

Рис. 3.14: Возвращение контекста

15. Сервер снова доступен (fig. 3.15).



Рис. 3.15: Сервер доступен

16. Исправим конфигурацию, удалим привязку к 81 порту и файл test (fig. 3.16).

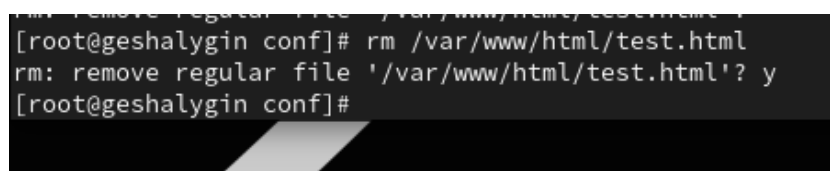


Рис. 3.16: Окончание работы

4 Выводы

В результате выполнения работы мы развили навыки администрирования ОС Linux. Получили первое практическое знакомство с технологией SELinux1 . Проверили работу SELinx на практике совместно с веб-сервером Apache.

Список литературы

1. Кетов Д.В. Внутреннее устройство Linux. BHV, 2017. 124 с.
2. Л. М. Ухлинов. Управление доступом в ОС GNU /Linux . ОКБ САПР», Москва, Россия, 2010.