

# **Лабораторная работа № 6**

**Мандатное разграничение прав в Linux**

Беличева Дарья Михайловна

# Содержание

1	Цель работы	4
2	Теоретическое введение	5
3	Выполнение лабораторной работы	6
4	Выводы	15
	Список литературы	16

# Список иллюстраций

3.1	Задание параметра ServerName . . . . .	6
3.2	Отключение пакетного фильтра . . . . .	6
3.3	Проверка режима работы SELinux . . . . .	7
3.4	Проверка статуса веб-сервера . . . . .	7
3.5	Контекст безопасности Apache . . . . .	7
3.6	Текущее состояние переключателей SELinux для Apache . . . . .	8
3.7	Статистика по политике . . . . .	9
3.8	Множество пользователей, ролей, типов . . . . .	10
3.9	Просмотр типов директорий в /var/www . . . . .	11
3.10	Содержимое файла /var/www/html/test.html . . . . .	11
3.11	Проверка контекста файла . . . . .	11
3.12	Запуск файла через веб-браузер . . . . .	12
3.13	Изменение контекста файла /var/www/html/test.html . . . . .	12
3.14	Отказ в доступе к html-странице через браузер . . . . .	12
3.15	Просмотр log-файлов веб-сервера Apache . . . . .	13
3.16	Замена прослушиваемого порта . . . . .	13
3.17	Открытие html-страницы через браузер при прослушивании 81 порта . . . . .	13
3.18	Проверка запуска сервера через порт 81 . . . . .	14
3.19	Список портов в semanage . . . . .	14
3.20	Возвращение прежнего контекста файла . . . . .	14
3.21	Открытие html-страницы через браузер . . . . .	14

# 1 Цель работы

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

## 2 Теоретическое введение

SELinux (англ. Security-Enhanced Linux — Linux с улучшенной безопасностью) — реализация системы мандатного управления доступа, которая может работать параллельно с классической избирательной системой контроля доступа[1].

Домен — список действий, которые может выполнять процесс. Обычно в качестве домена определяется минимально-возможный набор действий, при помощи которых процесс способен функционировать. Таким образом, если процесс дискредитирован, злоумышленнику не удастся нанести большого вреда.

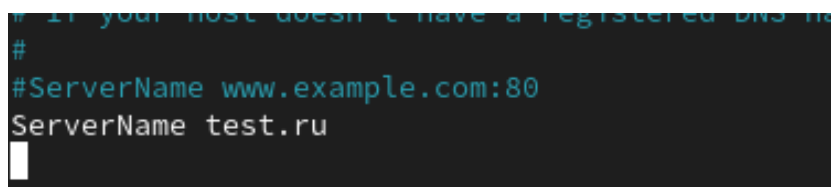
Роль — список доменов, которые могут быть применены. Если какого-то домена нет в списке доменов какой-то роли, то действия из этого домена не могут быть применены.

Тип — набор действий, которые допустимы по отношению к объекту. Тип отличается от домена тем, что он может применяться к пайпам, каталогам и файлам, в то время как домен применяется к процессам.

Контекст безопасности — все атрибуты SELinux — роли, типы и домены.

### 3 Выполнение лабораторной работы

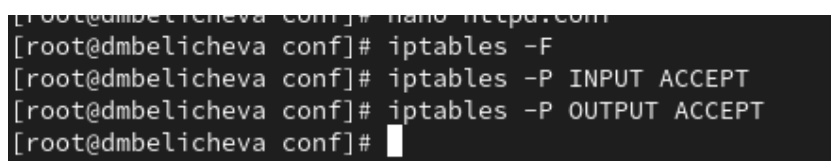
Подготовим лабораторный стенд: установим веб-сервер Apache, а в конфигурационном файле `/etc/httpd/httpd.conf` зададим параметр `ServerName` (рис. 3.1).



```
# If your host doesn't have a registered DNS name,
#
#ServerName www.example.com:80
ServerName test.ru
```

Рис. 3.1: Задание параметра `ServerName`

Далее отключим пакетный фильтр (рис. 3.2).



```
[root@dmbelicheva conf]# nano httpd.conf
[root@dmbelicheva conf]# iptables -F
[root@dmbelicheva conf]# iptables -P INPUT ACCEPT
[root@dmbelicheva conf]# iptables -P OUTPUT ACCEPT
[root@dmbelicheva conf]#
```

Рис. 3.2: Отключение пакетного фильтра

Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`.

```
[dmbelicheva@dmbelicheva conf]$ getenforce
Enforcing
[dmbelicheva@dmbelicheva 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
[dmbelicheva@dmbelicheva conf]$
```

Рис. 3.3: Проверка режима работы SELinux

Обратимся с помощью браузера к веб-серверу, запущенному на нашем компьютере, и убедимся, что последний работает(рис. 3.4).

```
[dmbelicheva@dmbelicheva conf]$ sudo systemctl enable httpd
[dmbelicheva@dmbelicheva conf]$ sudo systemctl start httpd
[dmbelicheva@dmbelicheva conf]$ sudo service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-10-12 00:12:50 EEST; 3min 56s ago
     Docs: man:httpd.service(8)
   Main PID: 41547 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
    Tasks: 177 (limit: 24675)
   Memory: 22.1M
      CPU: 300ms
   CGroup: /system.slice/httpd.service
           └─41547 /usr/sbin/httpd -DFOREGROUND
             └─41548 /usr/sbin/httpd -DFOREGROUND
               └─41549 /usr/sbin/httpd -DFOREGROUND
                 └─41550 /usr/sbin/httpd -DFOREGROUND
                   └─41551 /usr/sbin/httpd -DFOREGROUND

Oct 12 00:12:50 dmbelicheva.localdomain systemd[1]: Starting The Apache HTTP Server...
Oct 12 00:12:50 dmbelicheva.localdomain systemd[1]: Started The Apache HTTP Server.
Oct 12 00:12:50 dmbelicheva.localdomain httpd[41547]: Server configured, listening on: port 80
[dmbelicheva@dmbelicheva conf]$
```

Рис. 3.4: Проверка статуса веб-сервера

Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности (рис. 3.5). Информация о контексте находится в первом столбце.

```
[dmbelicheva@dmbelicheva conf]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root      41547  0.0  0.2 20152 11416 ?        Ss   00:12   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  41548  0.0  0.1 22032  7116 ?        S    00:12   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  41549  0.0  0.3 1571340 13252 ?       Sl   00:12   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  41550  0.0  0.2 1440204 10892 ?       Sl   00:12   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  41551  0.0  0.2 1440204 11020 ?       Sl   00:12   0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 dmbelicheva 41869  0.0  0.0 221664 2384 pts/0  S+   00:18   0:00 grep --color=auto httpd
[dmbelicheva@dmbelicheva conf]$
```

Рис. 3.5: Контекст безопасности Apache

Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды (рис. 3.6).

```
without options, show SELinux status.
[dmbelicheva@dmbelicheva 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
```

Рис. 3.6: Текущее состояние переключателей SELinux для Apache

Посмотрим статистику по политике с помощью команды `seinfo`, также определите множество пользователей, ролей, типов (рис. 3.7;3.8).



```

[dmbelicheva@dmbelicheva 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:                   5145     Attributes:           259
Users:                   8         Roles:                15
Booleans:                356     Cond. Expr.:         388
Allow:                   65500    Neverallow:           0
Auditallow:              176     Dontaudit:            8682
Type_trans:              271770  Type_change:          94
Type_member:              37     Range_trans:          5931
Role allow:              40      Role_trans:           417
Constraints:             70     Validatetrans:        0
MLS Constrains:          72     MLS Val. Tran:        0
Permissives:             4       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:              665
Netifcon:                0       Nodecon:              0
[dmbelicheva@dmbelicheva conf]$ █

```

Рис. 3.7: Статистика по политике

```

[dmbelicheva@dmbelicheva conf]$ seinfo -u
Users: 8
  guest_u
  root
  staff_u
  sysadm_u
  system_u
  unconfined_u
  user_u
  xguest_u
[dmbelicheva@dmbelicheva conf]$ seinfo -r
Roles: 15
  auditadm_r
  container_user_r
  dbadm_r
  guest_r
  logadm_r
  nx_server_r
  object_r
  secadm_r
  staff_r
  sysadm_r
  system_r
  unconfined_r
  user_r
  webadm_r
  xguest_r
[dmbelicheva@dmbelicheva conf]$ seinfo -t
Types: 5145
  NetworkManager_dispatcher_chronyc_script_t
  NetworkManager_dispatcher_chronyc_t
  NetworkManager_dispatcher_cloud_script_t
  NetworkManager_dispatcher_cloud_t
  NetworkManager_dispatcher_console_script_t
  NetworkManager_dispatcher_console_t
  NetworkManager_dispatcher_console_var_run_t
  NetworkManager_dispatcher_custom_t

```

Рис. 3.8: Множество пользователей, ролей, типов

Определите тип файлов и поддиректорий, находящихся в директории /var/www (рис. 3.9). Увидим, что есть директория, содержащая cgi-скрипты, и директория /var/www/html, содержащая все скрипты httpd(в данный момент пустая).

```
[dmbelicheva@dmbelicheva conf]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Aug 8 19:30 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 Aug 8 19:30 html
[dmbelicheva@dmbelicheva conf]$ ls -lZ /var/www/html
total 0
[dmbelicheva@dmbelicheva conf]$ cd /var/www/html
```

Рис. 3.9: Просмотр типов директорий в /var/www

Создадим от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания (рис. 3.10).

```
GNU nano 5.6.1
<html>
<body>test</body>
</html>
```

Рис. 3.10: Содержимое файла /var/www/html/test.html

Проверим контекст созданного вами файла, который был задан по умолчанию этому файлу (рис. 3.11). Увидим, что файлам по умолчанию сопоставляется свободный пользователь SELinux `unconfined_u`, указана роль `object_r` используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах и тип `httpd_sys_content_t`, который позволяет процессу `httpd` получить доступ к файлу.

```
[dmbelicheva@dmbelicheva html]$ ls -lZ /var/www/html/test.html
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 Oct 12 00:31 /var/www/html/test.html
[dmbelicheva@dmbelicheva html]$ secon --file /var/www/html/test.html
user: unconfined_u
role: object_r
type: httpd_sys_content_t
sensitivity: s0
clearance: s0
mls-range: s0
[dmbelicheva@dmbelicheva html]$
```

Рис. 3.11: Проверка контекста файла

Обратимся к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Убедимся, что файл успешно отображается (рис. 3.12).

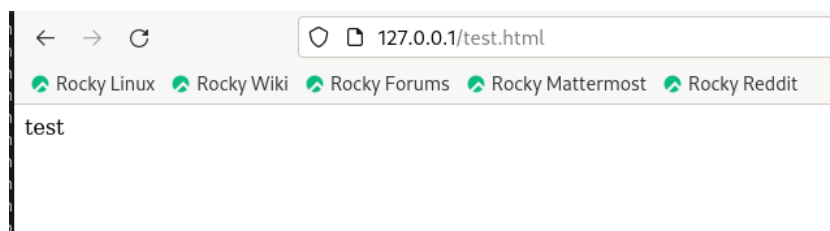


Рис. 3.12: Запуск файла через веб-браузер

Изучив справку `man httpd_selinux`, выясним, какие контексты файлов определены для `httpd`. Сопоставив их с типом файла `test.html` увидим, что его контекст `httpd_sys_content_t` для содержимого, которое должно быть доступно для всех скриптов `httpd` и для самого демона.

Изменим контекст файла `/var/www/html/test.html` с `httpd_sys_content_t` на тот, к которому процесс `httpd` не должен иметь доступа – `samba_share_t` (рис. 3.13).

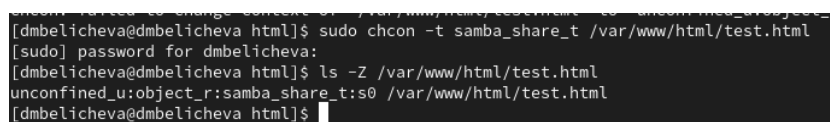


Рис. 3.13: Изменение контекста файла `/var/www/html/test.html`

Теперь снова попробуем получить доступ к файлу через браузер и получим отказ (рис. 3.14).

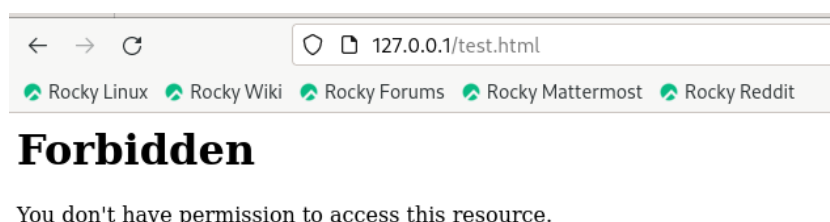


Рис. 3.14: Отказ в доступе к html-странице через браузер

Посмотрим `log`-файлы веб-сервера `Apache` и системный `log`-файл и увидим, что отказ происходит, так как доступ запрещен `SELinux` именно к веб-серверу (на просто просмотр текстовых файлов это не влияет) (рис. 3.15).

```

[root@dbelicheva ~]# tail /var/log/messages
Oct 12 01:03:37 dbelicheva setroubleshoot[42793]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www/html/test.html.
If you want to fix the label, #012/var/www/html/test.html default label should be httpd_sys_content_t.#012Then you can run restorecon. The access
in which case try to change the following command accordingly.#012Do#012# /sbin/restorecon -v /var/www/html/test.html#012#012***** Plugin public
tml 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 -
#012#012***** Plugin catchall (1.41 confidence) suggests *****#012#012If you believe that httpd should be allowed getattr
generate a local policy module to allow this access.#012Do#012allow this access for now by executing:#012# ausearch -c 'httpd' --raw | audit2all
Oct 12 01:03:46 dbelicheva systemd[1596]: Started Application launched by gnome-shell.
Oct 12 01:03:46 dbelicheva systemd[1596]: Started VTE child process 42874 launched by gnome-terminal-server process 2589.
Oct 12 01:03:47 dbelicheva systemd[1]: dbus-1.1-1-org.fedoraproject.SetroubleshootPrivileged@1.service: Deactivated successfully.
Oct 12 01:03:47 dbelicheva systemd[1]: dbus-1.1-1-org.fedoraproject.SetroubleshootPrivileged@1.service: Consumed 1.243s CPU time.
Oct 12 01:03:47 dbelicheva systemd[1]: setroubleshootd.service: Deactivated successfully.
Oct 12 01:03:51 dbelicheva systemd[1]: Starting Fingerprint Authentication Daemon...

```

Рис. 3.15: Просмотр log-файлов веб-сервера Apache

Запустим веб-сервер Apache на прослушивание TCP-порта 81. Для этого в файле /etc/httpd/httpd.conf найдем строчку Listen 80 и заменим её на Listen 81 (рис. 3.16).

```

#
#Listen 12.34.56.78:80
Listen 81

```

Рис. 3.16: Замена прослушиваемого порта

Выполним перезапуск веб-сервера Apache. Произошёл сбой, потому что порт 80 для локальной сети, а 81 нет (рис. 3.17).

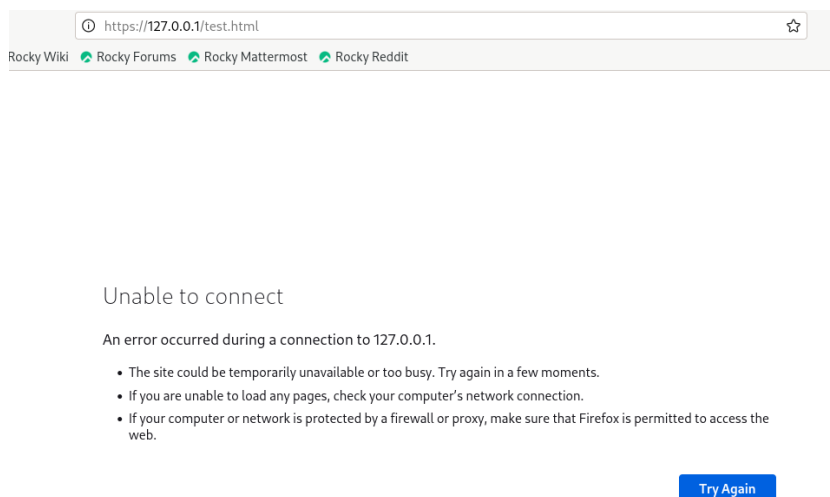


Рис. 3.17: Открытие html-страницы через браузер при прослушивании 81 порта

В логах можно увидеть запуск сервера через порт 81 (рис. 3.18).

```
Oct 12 01:24:11 dmbelicheva.localdomain systemd[1]: Starting The Apache HTTP Server...
Oct 12 01:24:11 dmbelicheva.localdomain systemd[1]: Started The Apache HTTP Server.
Oct 12 01:24:11 dmbelicheva.localdomain httpd[43820]: Server configured, listening on: port 81
[dmbelicheva@dmbelicheva conf]$
```

Рис. 3.18: Проверка запуска сервера через порт 81

Проверим какие порты у нас добавлены в semanage (рис. 3.19).

```
[root@dmbelicheva ~]# 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@dmbelicheva ~]#
```

Рис. 3.19: Список портов в semanage

Вернем контекст `httpd_sys_content_t` к файлу `/var/www/html/ test.html` (рис. 3.20).. После этого попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html`. Доступ успешно получен. Увидим содержимое файла – слово «test» (рис. 3.21).

```
[root@dmbelicheva ~]# systemctl restart httpd
[root@dmbelicheva ~]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@dmbelicheva ~]# systemctl restart httpd
[root@dmbelicheva ~]#
```

Рис. 3.20: Возвращение прежнего контекста файла

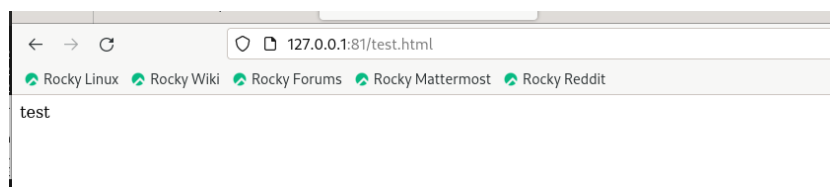


Рис. 3.21: Открытие html-страницы через браузер

Исправим обратно конфигурационный файл `apache`, вернув `Listen 80`. Удалим привязку `http_port_t` к 81 порту. Удалим файл `/var/www/html/test.html`:

## 4 Выводы

В результате выполнения данной лабораторной работы мною были развиты навыки администрирования ОС Linux, получено первое практическое знакомство с технологией SELinux, а также проверена работа SELinux на практике совместно с веб-сервером Apache.

# Список литературы

1. SELinux [Электронный ресурс]. 2024. URL: <https://ru.wikipedia.org/wiki/SELinux>.