

Лабораторная работа №6

Мандатное разграничение прав в Linux

Латыпова Диана. НФИбд-02-21

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	10
4	Выполнение лабораторной работы	12
5	Выводы	24
	Список литературы	25

Список иллюстраций

4.1	Пункт 1	12
4.2	Пункт 2	13
4.3	Пункт 3	13
4.4	Пункт 4	14
4.5	Пункт 5	15
4.6	Пункты 6-8	15
4.7	Пункты 9-10	16
4.8	Содержимое файла test.html	16
4.9	Пункт 11	16
4.10	Пункт 12	17
4.11	Пункт 13	17
4.12	Пункт 14	17
4.13	Пункт 15	18
4.14	Замена Listen 80 на Listen 81	19
4.15	Пункты 16-17	19
4.16	Пункт 18	20
4.17	Пункты 19-20	21
4.18	Пункт 21	21
4.19	Проверка сайта	22
4.20	Пункт 22	22
4.21	Пункт 23	23
4.22	Пункт 24	23

Список таблиц

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 Задание

1. Войдите в систему с полученными учётными данными и убедитесь, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`.
2. Обратитесь с помощью браузера к веб-серверу, запущенному на вашем компьютере, и убедитесь, что последний работает: `service httpd status` или `/etc/rc.d/init.d/httpd status` Если не работает, запустите его так же, но с параметром `start`.
3. Найдите веб-сервер Apache в списке процессов, определите его контекст безопасности и занесите эту информацию в отчёт. Например, можно использовать команду `ps auxZ | grep httpd` или `ps -eZ | grep httpd`
4. Посмотрите текущее состояние переключателей SELinux для Apache с помощью команды

```
sestatus -bigrep httpd
```

Обратите внимание, что многие из них находятся в положении «off». 5. Посмотрите статистику по политике с помощью команды `seinfo`, также определите множество пользователей, ролей, типов. 6. Определите тип файлов и поддиректорий, находящихся в директории `/var/www`, с помощью команды

```
ls -lZ /var/www
```

7. Определите тип файлов, находящихся в директории `/var/www/html`:

```
ls -lZ /var/www/html
```

8. Определите круг пользователей, которым разрешено создание файлов в директории `/var/www/html`.
9. Создайте от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл `/var/www/html/test.html` следующего содержания:

```
<html>
<body>test</body>
</html>
```

10. Проверьте контекст созданного вами файла. Занесите в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории `/var/www/html`.
11. Обратитесь к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Убедитесь, что файл был успешно отображён.
12. Изучите справку `man httpd_selinux` и выясните, какие контексты файлов определены для `httpd`. Сопоставьте их с типом файла `test.html`. Проверить контекст файла можно командой `ls -Z`.

```
ls -Z /var/www/html/test.html
```

Рассмотрим полученный контекст детально. Обратите внимание, что так как по умолчанию пользователи CentOS являются свободными от типа (`unconfined` в переводе с англ. означает свободный), созданному нами файлу `test.html` был сопоставлен SELinux, пользователь `unconfined_u`. Это первая часть контекста. Далее политика ролевого разделения доступа RBAC используется процессами, но не файлами, поэтому роли не имеют никакого значения для файлов. Роль `object_r` используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах. (В директории `/ргос` файлы, относящиеся к процессам, могут иметь роль `system_r`. Если активна политика MLS, то могут использоваться и другие роли, например, `secadm_r`. Данный случай мы рассматривать не будем, как и предназначение `:s0`).

Тип `httpd_sys_content_t` позволяет процессу `httpd` получить доступ к файлу. Благодаря наличию последнего типа мы получили доступ к файлу при обращении к нему через браузер. 13. Измените контекст файла `/var/www/html/test.html` с `httpd_sys_content_t` на любой другой, к которому процесс `httpd` не должен иметь доступа, например, на `samba_share_t`:

```
chcon -t samba_share_t /var/www/html/test.html
ls -Z /var/www/html/test.html
```

После этого проверьте, что контекст поменялся. 14. Попробуйте ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Вы должны получить сообщение об ошибке:

Forbidden

You don't have permission to access /test.html on this server.

15. Проанализируйте ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю?

```
ls -l /var/www/html/test.html
```

Просмотрите log-файлы веб-сервера Apache. Также просмотрите системный лог-файл:

```
tail /var/log/messages
```

Если в системе окажутся запущенными процессы `setroubleshootd` и `audtd`, то вы также сможете увидеть ошибки, аналогичные указанным выше, в файле `/var/log/audit/audit.log`. Проверьте это утверждение самостоятельно. 16. Попробуйте запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в `/etc/services`). Для этого в файле `/etc/httpd/httpd.conf` найдите строчку `Listen 80` и замените её на `Listen 81`. 17. Выполните перезапуск веб-сервера Apache. Произошёл сбой? Поясните почему? 18. Проанализируйте лог-файлы:


```
tail -n1 /var/log/messages
```

Просмотрите файлы /var/log/http/error_log, /var/log/http/access_log и /var/log/audit/audit.log и выясните, в каких файлах появились записи. 19. Выполните команду

```
semanage port -a -t http_port_t -p tcp 81
```

После этого проверьте список портов командой

```
semanage port -l | grep http_port_t
```

Убедитесь, что порт 81 появился в списке. 20. Попробуйте запустить веб-сервер Apache ещё раз. Поняли ли вы, почему он сейчас запустился, а в предыдущем случае не смог? 21. Верните 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 попробуйте получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html. Вы должны увидеть содержимое файла — слово «test». 22. Исправьте обратно конфигурационный файл apache, вернув Listen 80. 23. Удалите привязку http_port_t к 81 порту:

```
semanage port -d -t http_port_t -p tcp 81
```

и проверьте, что порт 81 удалён. 24. Удалите файл /var/www/html/test.html:

```
rm /var/www/html/test.html
```

3 Теоретическое введение

SELinux (Security-Enhanced Linux) — это система управления доступом на уровне ядра, которая реализует обязательное управление доступом (MAC) в операционных системах Linux. Она позволяет администраторам задавать политику доступа, которая определяет, каким процессам разрешено взаимодействовать с объектами системы (файлы, сокеты и т.д.) [1].

SELinux использует три режима работы [2]:

- Enforcing — политика SELinux активно применяется, и все действия, не соответствующие политике, блокируются.
- Permissive — политика не блокирует действия, но все нарушения записываются в журнал.
- Disabled — SELinux отключен.

Политики SELinux подразделяются на несколько видов, среди которых наиболее распространённой является Targeted Policy, которая защищает только определённые процессы, такие как веб-серверы и службы безопасности, оставляя остальные процессы менее защищёнными.

Команды для управления и проверки статуса SELinux [2]:

- getenforce — показывает текущий режим работы SELinux (Enforcing, Permissive, Disabled)
- sestatus — выводит подробную информацию о текущем состоянии SELinux

Apache — это один из самых популярных веб-серверов, который используется для обслуживания веб-сайтов и приложений. При установке и настройке

веб-сервера на системе с SELinux важно учитывать, что политика SELinux контролирует доступ Apache к файлам и ресурсам системы [3].

Основные команды [4]:

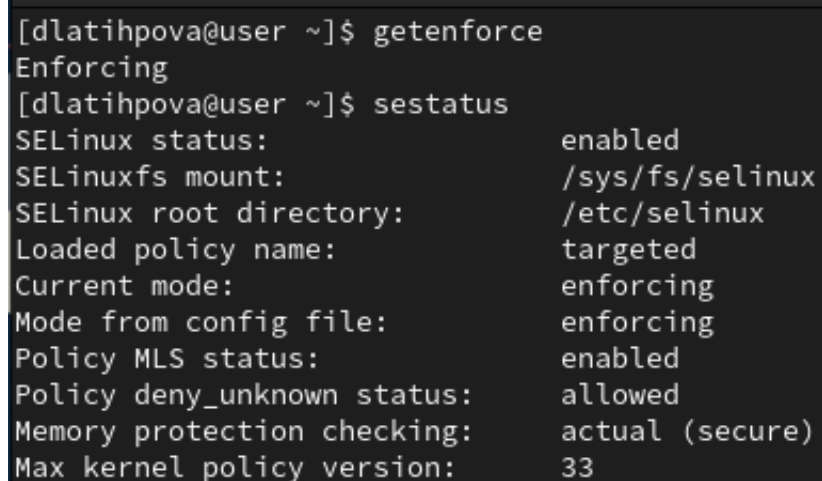
- `service httpd status` или `/etc/rc.d/init.d/httpd status` — проверяет, работает ли веб-сервер
- `service httpd start` или `/etc/rc.d/init.d/httpd start` — запускает веб-сервер, если он выключен
- `ps auxZ | grep httpd` — показывает контекст безопасности процессов веб-сервера Apache
- `getsebool -a | grep httpd` — выводит список всех SELinux переключателей для Apache и их текущее состояние
- `ls -Z /var/www/html/test.html` — показывает контекст безопасности файла
- `chcon -t samba_share_t /var/www/html/test.html` — изменит тип контекста на `samba_share_t`, который запрещает доступ Apache к этому файлу
- `semanage port -a -t http_port_t -p tcp 81` — добавляет порт 81 к списку допустимых для Apache
- `semanage port -l | grep http_port_t` — просмотр списка портов, разрешённых SELinux

При работе с веб-сервером и SELinux важно отслеживать логи, чтобы своевременно обнаруживать и устранять ошибки. Основные файлы логов для Apache:

- `/var/log/messages` — системный лог.
- `/var/log/httpd/error_log` — лог ошибок веб-сервера.
- `/var/log/audit/audit.log` — лог SELinux, в котором фиксируются события, связанные с нарушениями политики безопасности.

4 Выполнение лабораторной работы

Я вошла в систему с полученными учётными данными и убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus` (рис. 4.1):



```
[dlatihpova@user ~]$ getenforce
Enforcing
[dlatihpova@user ~]$ 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
```

Рис. 4.1: Пункт 1

Обратилась с помощью браузера к веб-серверу, запущенному на моем компьютере, и убедилась, что последний работает (рис. 4.2):

```
service httpd status
```

```
[dlatihpova@user ~]$ service httpd status~
The service command supports only basic LSB actions (start, stop, restart, try-restart,
reload, reload-or-restart, try-reload-or-restart, force-reload, status, condrestart). Fo
r other actions, please try to use systemctl.
[dlatihpova@user ~]$ systemctl start httpd
[dlatihpova@user ~]$ systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/sys
temd/system/httpd.service.
[dlatihpova@user ~]$ 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: disab
   Active: active (running) since Sat 2024-10-12 14:05:27 MSK; 37s ago
     Docs: man:httpd.service(8)
   Main PID: 112408 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes>
     Tasks: 177 (limit: 12208)
   Memory: 26.7M
     CPU: 66ms
   CGroup: /system.slice/httpd.service
           └─112408 /usr/sbin/httpd -DFOREGROUND
             └─112417 /usr/sbin/httpd -DFOREGROUND
               └─112422 /usr/sbin/httpd -DFOREGROUND
                 └─112423 /usr/sbin/httpd -DFOREGROUND
                   └─112424 /usr/sbin/httpd -DFOREGROUND

Oct 12 14:05:27 user.localdomain systemd[1]: Starting The Apache HTTP Server...
Oct 12 14:05:27 user.localdomain systemd[1]: Started The Apache HTTP Server.
Oct 12 14:05:27 user.localdomain httpd[112408]: Server configured, listening on>
lines 1-19/19 (END)...skipping...
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disab
```

Рис. 4.2: Пункт 2

Нашла веб-сервер Apache в списке процессов, определила его контекст без-опасности - httpd_t (рис. 4.3):

ps auxZ | grep httpd

```
[dlatihpova@user ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 112408 0.0 0.5 20364 11464 ? Ss 14:05 0:00 /usr/sbin/h
ttpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 112417 0.0 0.3 22096 7248 ? S 14:05 0:00 /usr/sbin/h
ttpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 112422 0.0 0.6 1112656 13476 ? Sl 14:05 0:00 /usr/sbin/h
ttpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 112423 0.0 0.5 981520 11096 ? Sl 14:05 0:00 /usr/sbin/h
ttpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 112424 0.0 0.5 981520 11096 ? Sl 14:05 0:00 /usr/sbin/h
ttpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 dlatihp+ 112704 0.0 0.4 236780 9008 pts/0 S+ 14:06 0:
00 /bin/systemctl status httpd.service
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 dlatihp+ 112767 0.0 0.1 221664 2176 pts/1 S+ 14:08 0:
00 grep --color=auto httpd
```

Рис. 4.3: Пункт 3

Посмотрела текущее состояние переключателей SELinux для Apache с помощью команды:

sestatus -bigrep httpd

Многие из них находятся в положении «off» (рис. 4.4):

```
[dlatihpova@user ~]$ sestatus -b httpd
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

Policy booleans:
abrt_anon_write                 off
abrt_handle_event               off
abrt_upload_watch_anon_write    on
antivirus_can_scan_system       off
antivirus_use_jit               off
auditadm_exec_content           on
authlogin_nsswitch_use_ldap     off
authlogin_radius                off
authlogin_yubikev               off
```

Рис. 4.4: Пункт 4

Посмотрела статистику по политике с помощью команды `seinfo`, также выделила множество пользователей, ролей, типов (рис. 4.5):

```
[dlatihpova@user ~]$ seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:          33 (MLS enabled)
Target Policy:           selinux
Handle unknown classes:  allow
Classes:                 135
Sensitivities:           1
Types:                   5145
Users:                   8
Booleans:                356
Allow:                   65504
Auditallow:              176
Type_trans:              271770
Type_member:             37
Role allow:              40
Constraints:             70
MLS Constrain:           72
Permissives:             4
Defaults:                7
Allowxperm:              0
Auditallowxperm:         0
Ibendportcon:            0
Initial SIDs:            27
Genfscon:                109
Netifcon:                0
Permissions:             457
Categories:              1024
Attributes:              259
Roles:                   15
Cond. Expr.:            388
Neverallow:              0
Dontaudit:               8682
Type_change:             94
Range_trans:             5931
Role_trans:              417
Validatetrans:           0
MLS Val. Tran:           0
Polcap:                  6
Typebounds:              0
Neverallowxperm:         0
Dontauditxperm:          0
Ibpkeycon:               0
Fs_use:                  35
Portcon:                 665
Nodecon:                 0
```

Рис. 4.5: Пункт 5

Определила тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды:

```
ls -lZ /var/www
```

Определила тип файлов, находящихся в директории /var/www/html:

```
ls -lZ /var/www/html
```

Определила круг пользователей, которым разрешено создание файлов в директории /var/www/html - 0 (рис. 4.6):

```
[dlatihpova@user ~]$ 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
[dlatihpova@user ~]$ ls -lZ /var/www/html
total 0
```

Рис. 4.6: Пункты 6-8

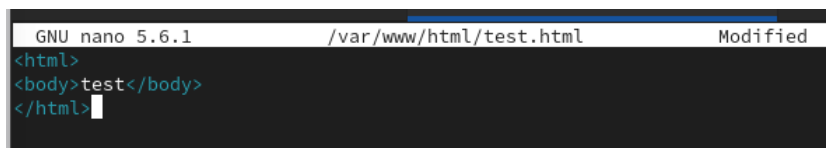
Создала от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания (рис. 4.8):

```
<html>
<body>test</body>
</html>
```

Проверила контекст созданного мной файла (рис. 4.7):

```
[dlatihpova@user ~]$ su
Password:
[root@user dlatihpova]# touch /var/www/html/test.html
[root@user dlatihpova]# nano /var/www/html/test.html
[root@user dlatihpova]# cat /var/www/html/test.html
<html>
<body>test</body>
</html>
[root@user dlatihpova]# exit
exit
[dlatihpova@user ~]$ ls -lZ /var/www/html
total 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 0
ct 12 14:17 test.html
```

Рис. 4.7: Пункты 9-10



```
GNU nano 5.6.1 /var/www/html/test.html Modified
<html>
<body>test</body>
</html>
```

Рис. 4.8: Содержимое файла test.html

Обратилась к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html>. Файл успешно отображён (рис. 4.9):

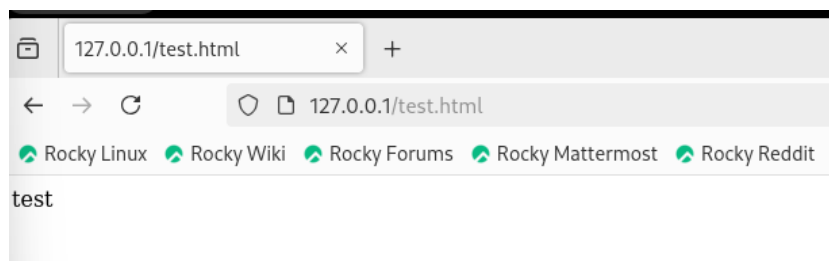


Рис. 4.9: Пункт 11

Изучила справку `man httpd_selinux` и выяснила, какие контексты файлов определены для `httpd`. Проверила контекст файла командой (рис. 4.10):

```
ls -Z /var/www/html/test.html
```

```
[dlatihpova@user ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
```

Рис. 4.10: Пункт 12

Изменила контекст файла `/var/www/html/test.html` с `httpd_sys_content_t` на `samba_share_t` (рис. 4.11):

```
chcon -t samba_share_t /var/www/html/test.html
```

```
ls -Z /var/www/html/test.html
```

```
[dlatihpova@user ~]$ su
Password:
[root@user dlatihpova]# chcon -t samba_share_t /var/www/html/test.html
[root@user dlatihpova]# exit
exit
[dlatihpova@user ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
```

Рис. 4.11: Пункт 13

1 Попробовала ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Вышла ошибка (рис. 4.12):

Forbidden

You don't have permission to access /test.html on this server.

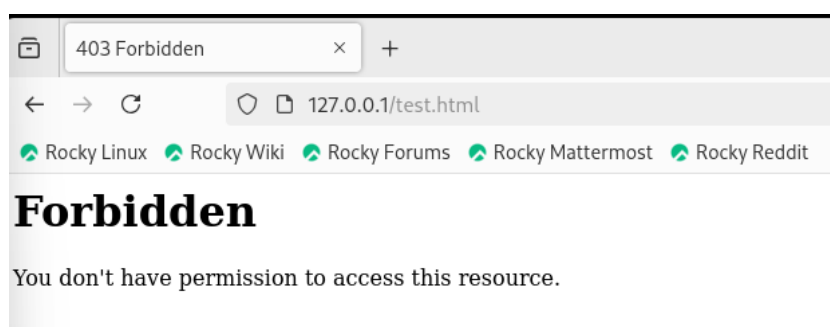
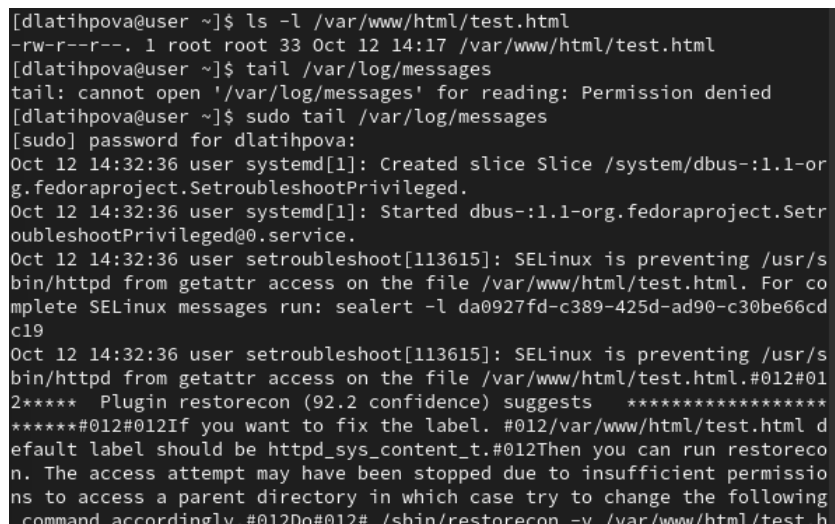


Рис. 4.12: Пункт 14

Просмотрела log-файлы веб-сервера Apache. Также просмотрела системный лог-файл(рис. 4.13):

```
tail /var/log/messages
```



```
[dlatihpova@user ~]$ ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 33 Oct 12 14:17 /var/www/html/test.html
[dlatihpova@user ~]$ tail /var/log/messages
tail: cannot open '/var/log/messages' for reading: Permission denied
[dlatihpova@user ~]$ sudo tail /var/log/messages
[sudo] password for dlatihpova:
Oct 12 14:32:36 user systemd[1]: Created slice Slice /system/dbus-:1.1-org.fedoraproject.SetroubleshootPrivileged.
Oct 12 14:32:36 user systemd[1]: Started dbus-:1.1-org.fedoraproject.SetroubleshootPrivileged@0.service.
Oct 12 14:32:36 user setroubleshoot[113615]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www/html/test.html. For complete SELinux messages run: sealert -l da0927fd-c389-425d-ad90-c30be66cd c19
Oct 12 14:32:36 user setroubleshoot[113615]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www/html/test.html.#012#012***** Plugin restorecon (92.2 confidence) suggests *****
*****#012#012If 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 attempt may have been stopped due to insufficient permissions to access a parent directory in which case try to change the following command accordingly.#012Do#012# /sbin/restorecon -v /var/www/html/test.h
```

Рис. 4.13: Пункт 15

Попробовала запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в /etc/services). Для этого в файле /etc/httpd/httpd.conf Listen 80 заменила её на Listen 81 (рис. 4.14):

```
GNU nano 5.6.1      httpd.conf      Modified
ServerRoot "/etc/httpd"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on a specific IP address, but note that if
# httpd.service is enabled to run at boot time, the address may not be
# available when the service starts. See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
Listen 81

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO
# you have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
Include conf.modules.d/*.conf

#
# If you wish httpd to run as a different user or group, you must run
```

Рис. 4.14: Замена Listen 80 на Listen 81

Выполнила перезапуск веб-сервера Apache (рис. 4.15):

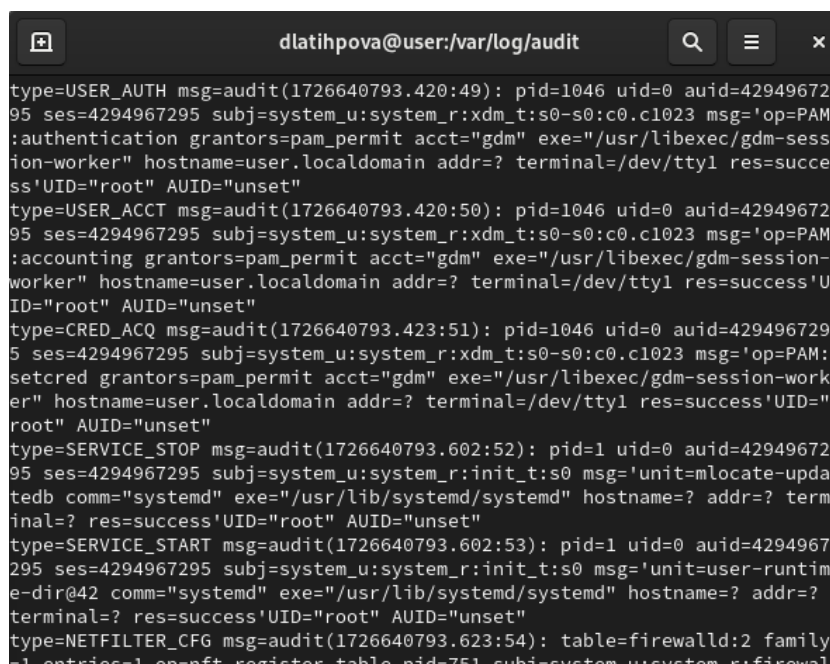
```
[dlatihpova@user conf]$ su
Password:
[root@user conf]# nano /etc/httpd/conf/httpd.conf
[root@user conf]# exit
exit
[dlatihpova@user conf]$ systemctl restart httpd
```

Рис. 4.15: Пункты 16-17

Проанализировла лог-файлы:

```
tail -n1 /var/log/messages
```

Просмотрела файлы /var/log/http/error_log, /var/log/http/access_log и /var/log/audit/audit.log. В последнем появились записи (рис. 4.16):



```
dlatihpova@user:/var/log/audit
type=USER_AUTH msg=audit(1726640793.420:49): pid=1046 uid=0 auid=42949672
95 ses=4294967295 subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 msg='op=PAM
:authentication grantors=pam_permit acct="gdm" exe="/usr/libexec/gdm-sess
ion-worker" hostname=user.localdomain addr=? terminal=/dev/tty1 res=succe
ss'UID="root" AUID="unset"
type=USER_ACCT msg=audit(1726640793.420:50): pid=1046 uid=0 auid=42949672
95 ses=4294967295 subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 msg='op=PAM
:accounting grantors=pam_permit acct="gdm" exe="/usr/libexec/gdm-session-
worker" hostname=user.localdomain addr=? terminal=/dev/tty1 res=success'U
ID="root" AUID="unset"
type=CRED_ACQ msg=audit(1726640793.423:51): pid=1046 uid=0 auid=429496729
5 ses=4294967295 subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 msg='op=PAM:
setcred grantors=pam_permit acct="gdm" exe="/usr/libexec/gdm-session-work
er" hostname=user.localdomain addr=? terminal=/dev/tty1 res=success'UID="
root" AUID="unset"
type=SERVICE_STOP msg=audit(1726640793.602:52): pid=1 uid=0 auid=42949672
95 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=mlocate-upda
tedb comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? term
inal=? res=success'UID="root" AUID="unset"
type=SERVICE_START msg=audit(1726640793.602:53): pid=1 uid=0 auid=4294967
295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=user-runtim
e-dir@42 comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=?
terminal=? res=success'UID="root" AUID="unset"
type=NETFILTER_CFG msg=audit(1726640793.623:54): table=firewalld:2 family
=1 entries=1 op=nft register table pid=751 subj=system_u:system_r:firewal
```

Рис. 4.16: Пункт 18

Выполнила команду:

```
semanage port -a -t http_port_t -p tcp 81
```

После этого проверила список портов командой:

```
semanage port -l | grep http_port_t
```

Порт 81 появился в списке. Попробовала запустить веб-сервер Apache ещё раз (рис. 4.17):

```

[dlatihpova@user ~]$ sudo semanage port -a -t http_port_t -p tcp 81
Port tcp/81 already defined, modifying instead
[dlatihpova@user ~]$ semanage port -l | grep http_port_t
ValueError: SELinux policy is not managed or store cannot be accessed.
[dlatihpova@user ~]$ sudo semanage port -l | grep http_port_t
http_port_t                tcp      81, 80, 81, 443, 488, 8008, 8009,
8443, 9000
pegasus_http_port_t        tcp      5988
[dlatihpova@user ~]$ systemctl restart httpd
[dlatihpova@user ~]$ curl ifconfig.me
37.18.92.241[dlatihpova@user ~]$ status httpdctl status httpd
bash: status: command not found...
[dlatihpova@user ~]$ curl ifconfig.me
37.18.92.241[dlatihpova@user ~]$ systemctl status httpdctl status httpd
Unit httpdctl.service could not be found.
Unit status.service could not be found.
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; pre>
   Active: active (running) since Sat 2024-10-12 14:58:52 MSK; 2min 20>
     Docs: man:httpd.service(8)
   Main PID: 114725 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0>
    Tasks: 177 (limit: 12208)
   Memory: 21.7M
      CPU: 96ms
   CGroup: /system.slice/httpd.service
           └─114725 /usr/sbin/httpd -DFOREGROUND
             └─114726 /usr/sbin/httpd -DFOREGROUND
               └─114733 /usr/sbin/httpd -DFOREGROUND
                 └─114734 /usr/sbin/httpd -DFOREGROUND
                   └─114735 /usr/sbin/httpd -DFOREGROUND
Oct 12 14:58:51 user.localdomain systemd[1]: Starting The Apache HTTP Ser

```

Рис. 4.17: Пункты 19-20

Вернула 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 попробовала получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html (рис. 4.18):

```

[dlatihpova@user ~]$ sudo chcon -t httpd_sys_content_t /var/www/html/test
.html
[sudo] password for dlatihpova:
[dlatihpova@user ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html

```

Рис. 4.18: Пункт 21

Увидела содержимое файла — слово «test» (рис. 4.19):

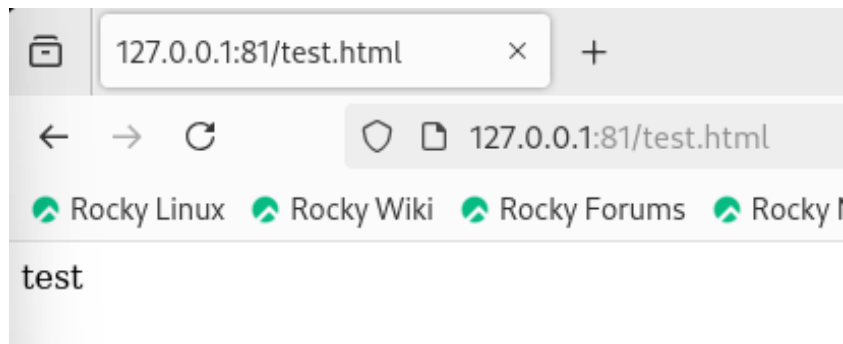


Рис. 4.19: Проверка сайта

Исправила обратно конфигурационный файл apache, вернув Listen 80 (рис. 4.20):



Рис. 4.20: Пункт 22

Удалила привязку http_port_t к 81 порту:

```
semanage port -d -t http_port_t -p tcp 81
```

и проверила, что порт 81 удалён (рис. 4.21):

```

[root@user dlatihpova]# semanage port -d -t http_port_t -p tcp 81
[root@user dlatihpova]# cat /etc/httpd/conf/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', whereas '/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.
#
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used. If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/etc/httpd"
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on a specific IP address, but note that if
# httpd.service is enabled to run at boot time, the address may not be
# available when the service starts. See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
Listen 80

```

Рис. 4.21: Пункт 23

Удалила файл /var/www/html/test.html (рис. 4.22):

```
rm /var/www/html/test.html
```

```

[root@user dlatihpova]# rm /var/www/html/test.html
rm: remove regular file '/var/www/html/test.html'? y
[root@user dlatihpova]# ls /var/www/html/test.html
ls: cannot access '/var/www/html/test.html': No such file or directory

```

Рис. 4.22: Пункт 24

5 Выводы

Я развила навыки администрирования ОС Linux и получила первое практическое знакомство с технологией SELinux. А также проверила работу SELinux на практике совместно с веб-сервером Apache.

Список литературы

1. SELinux [Электронный ресурс]. Wikipedia®, 2024. URL: <https://ru.wikipedia.org/wiki/SELinux>.
2. SELinux – описание и особенности работы с системой. Часть 1 [Электронный ресурс]. Habr, 2024. URL: <https://habr.com/ru/companies/kingservers/articles/209644/>.
3. Apache HTTP Server [Электронный ресурс]. Wikipedia®, 2024. URL: https://ru.wikipedia.org/wiki/Apache_HTTP_Server.
4. Серверы Linux. Часть I. Серверы Apache и Squid [Электронный ресурс]. Автор: Paul Cobbaut, Перевод: А.Панин, 2015. URL: <https://rus-linux.net/MyLDP/BOOKS/Linux-Servers/ch01.html>.