Лабораторная работа №6

Дисциплина: Информационная безопасность

Губина Ольга Вячеслвовна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

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

2 Задание

- Изучить на практике работу SELinx и Apache.

3 Теоретическое введение

SELinux (SELinux) — это система принудительного контроля доступа, реализованная на уровне ядра. Впервые эта система появилась в четвертой версии CentOS, а в 5 и 6 версии реализация была существенно дополнена и улучшена. Эти улучшения позволили SELinux стать универсальной системой, способной эффективно решать массу актуальных задач. Стоит помнить, что классическая система прав Unix применяется первой, и управление перейдет к SELinux только в том случае, если эта первичная проверка будет успешно пройдена.

Для того, чтобы понять, в чем состоит практическая ценность SELinux, рассмотрим несколько примеров, когда стандартная система контроля доступа недостаточна. Если SELinux отключен, то вам доступна только классическая дискреционная система контроля доступа, которая включает в себя DAC (избирательное управление доступом) или ACL(списки контроля доступа). То есть речь идет о манипулировании правами на запись, чтение и исполнение на уровне пользователей и групп пользователей, чего в некоторых случаях может быть совершенно недостаточно. Например:

- Администратор не может в полной мере контролировать действия пользователя. Например, пользователь вполне способен дать всем остальным пользователям права на чтение собственных конфиденциальных файлов, таких как ключи SSH.

- Процессы могут изменять настройки безопасности. Например, файлы, содержащие в себе почту пользователя должны быть доступны для чтения

только одному конкретному пользователю, но почтовый клиент вполне может изменить права доступа так, что эти файлы будут доступны для чтения всем.

- Процессы наследуют права пользователя, который их запустил. Например, зараженная трояном версия браузера Firefox в состоянии читать SSH-ключи пользователя, хотя не имеет для того никаких оснований.[1]

Apache – это свободное программное обеспечение для размещения веб-сервера. Он хорошо показывает себя в работе с масштабными проектами, поэтому заслуженно считается одним из самых популярных веб-серверов. Кроме того, Apache очень гибок в плане настройки, что даёт возможность реализовать все особенности размещаемого веб-ресурса.[2]

Установить веб-сервер Apache можно слеудющим образом. Откройте окно терминала и обновите списки пакетов репозитория, введя следующее: `sudo yum update`

Теперь вы можете установить Apache с помощью команды: `sudo yum -y install httpd`

httpd - это имя службы Apache в CentOS. Опция –y автоматически отвечает да на запрос подтверждения.[3]

4 Выполнение лабораторной работы

4.1 Подготовительный этап

Сперва проверим конфигурационный файл SELinux - видим, что политика targeted и режим enforcing используются в данном дистрибутиве по умолчанию, т.е. каких-то специальных настроек не требуется (рис. 4.1).

```
[root@ovgubina ~]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
# See also:
# https://docs.fedoraproject.org/en-US/quick-docs/getting-started-with-selinux/#getting-started-with-selinux-selinux-states-and-modes
#
# NOTE: In earlier Fedora kernel builds, SELINUX=disabled would also
# fully disable SELinux during boot. If you need a system with SELinux
# fully disabled instead of SELinux running with no policy loaded, you
# need to pass selinux=0 to the kernel command line. You can use grubby
# to persistently set the bootloader to boot with selinux=0:
#
#    grubby --update-kernel ALL --args selinux=0
#
# To revert back to SELinux enabled:
#
#    grubby --update-kernel ALL --remove-args selinux
#
SELINUX=enforcing
# SELINUXTYPE= can take one of these three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Рис. 4.1: Проверка политики и режима

Дальше потребуется установить менеджер Apache, для этого предварительно обновим систему (рис. 4.2), только после этого устанавливаем Apache (httpd) (рис. 4.3-4.4).

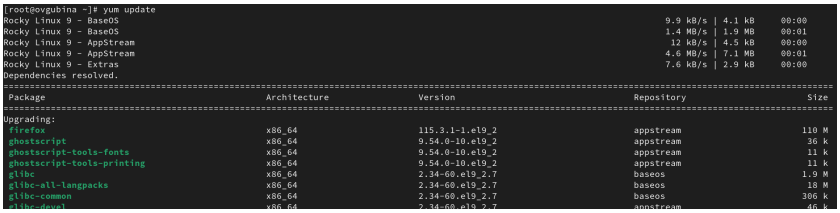

Рис. 4.2: Обновление

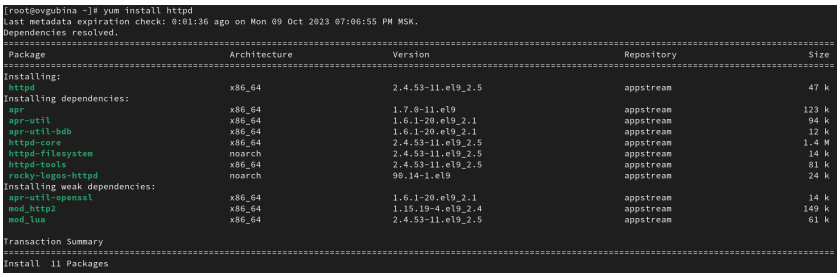

Рис. 4.3: Загрузка Apache

Рис. 4.4: Конец загрузки Apache

Далее зададим `ServerName test.ru` в конфигурационной файле httpd (рис. 4.6), для этого сперва найдем, где он находитсЯ (рис. 4.5).

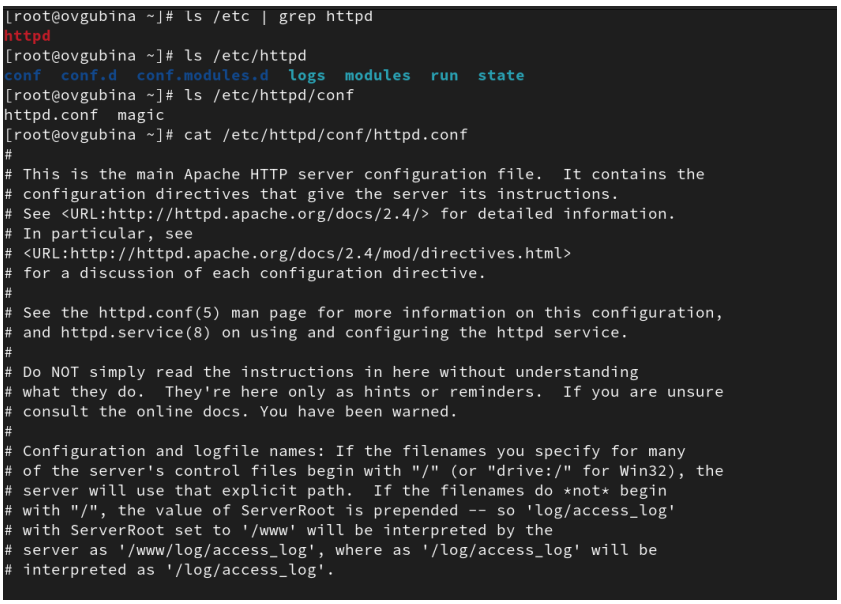

Рис. 4.5: Расположение конфигурационного файла httpd

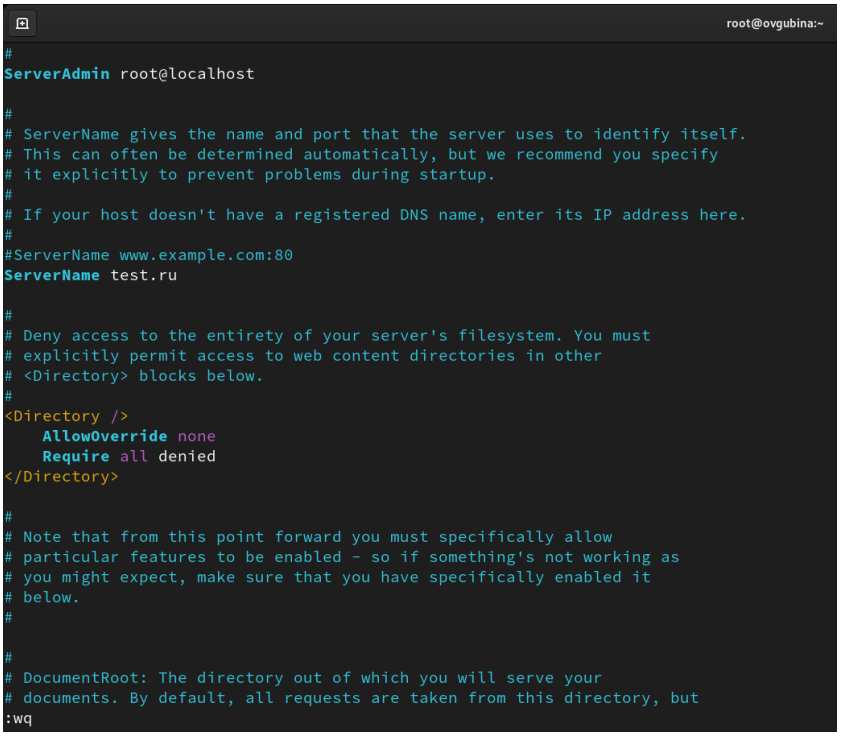

Рис. 4.6: Задаем ServerName

Чтобы пакетный фильтр в своей рабочей конфигурации позволял подклю-

чаться к 80-у и 81-у портам протокола tcp добавим разрешающие правила (рис. 4.7):

```
[root@ovgubina ~]#
[root@ovgubina ~]# iptables -I INPUT -p tcp --dport 80 -j ACCEPT
[root@ovgubina ~]# iptables -I INPUT -p tcp --dport 81 -j ACCEPT
[root@ovgubina ~]# iptables -I OUTPUT -p tcp --sport 80 -j ACCEPT
[root@ovgubina ~]# iptables -I OUTPUT -p tcp --sport 81 -j ACCEPT
[root@ovgubina ~]#
```

Рис. 4.7: Задаем ServerName

4.2 Порядок выполнения работы

1. Войдите в систему с полученными учётными данными и убедитесь, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus` (рис. 4.8).

```
[ovgubina@ovgubina ~]$ getenforce
Enforcing
[ovgubina@ovgubina ~]$ 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
[ovgubina@ovgubina ~]$
```

Рис. 4.8: Режим enforcing политики targeted

2. Обратитесь с помощью браузера к веб-серверу, запущенному на вашем компьютере, и убедитесь, что последний работает: `service httpd status`. Видим, что он неактивен, поэтому запускаем его командой `service httpd start`, после чего снова проверяем, в этот раз серсвис активен (рис. 4.9).

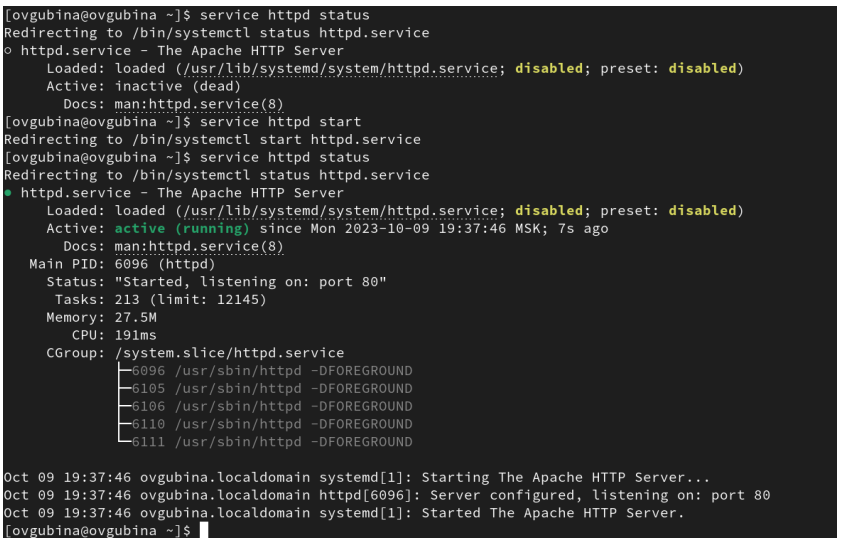

Рис. 4.9: Запуск

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` (4.10).

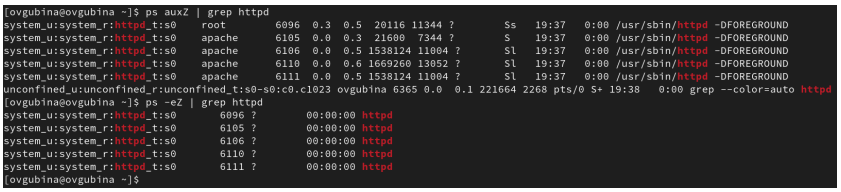

Рис. 4.10: Контекст безопасности

Видим, что веб-сервер имеет контекст безопасности `httpd_t`.

4. Посмотрите текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -b httpd` (рис. 4.11). Многие из них находятся в положении «off».

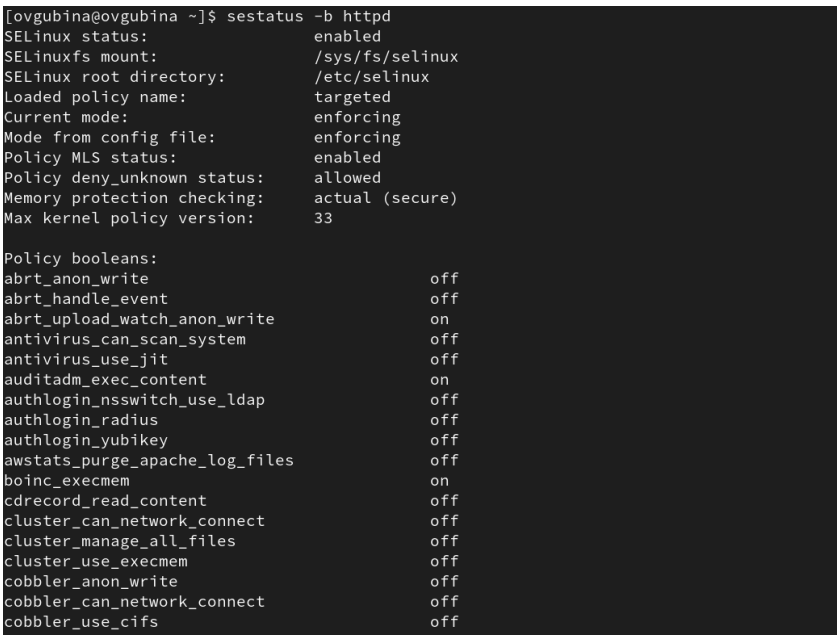

Рис. 4.11: Переключатели SELinux для Apache

5. Посмотрите статистику по политике с помощью команды `seinfo`, также определите множество пользователей, ролей, типов (рис. 4.12).

```
[ovgubina@ovgubina ~]$ 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:             65008    Neverallow:            0
  Auditallow:          170    Dontaudit:          8572
  Type_trans:       265344    Type_change:          87
  Type_member:          35    Range_trans:        6164
  Role allow:           38    Role_trans:          420
  Constraints:          70    Validatetrans:         0
  MLS Constrain:        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
[ovgubina@ovgubina ~]$
```

Рис. 4.12: Статистика по политике

Число типов = 5100, ролей = 14, пользователей = 8.

6. Определите тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды `ls -lZ /var/www` (рис. 4.13).

```
[ovgubina@ovgubina ~]$ 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
```

Рис. 4.13: Типы файлов

В каталоге находятся только директории.

7. Определите тип файлов, находящихся в директории /var/www/html: `ls -lZ /var/www/html` (рис. 4.14).

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

Рис. 4.14: Типы файлов

Директория пуста.

8. Определите круг пользователей, которым разрешено создание файлов в директории /var/www/html - это только пользователь root.

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 (рис. 4.15-4.16):

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

```
[root@ovgubina ~]# vim /var/www/html/test.html
[root@ovgubina ~]#
```

Рис. 4.15: Создание html файла

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

Рис. 4.16: Создание html файла

10. Проверьте контекст созданного вами файла. Занесите в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html (рис. 4.17).

Рис. 4.17: Создание html файла

Контекст безопасности - **httpd_sys_content_t**.

11. Обратитесь к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html (рис. 4.18). Файл был успешно отображён.

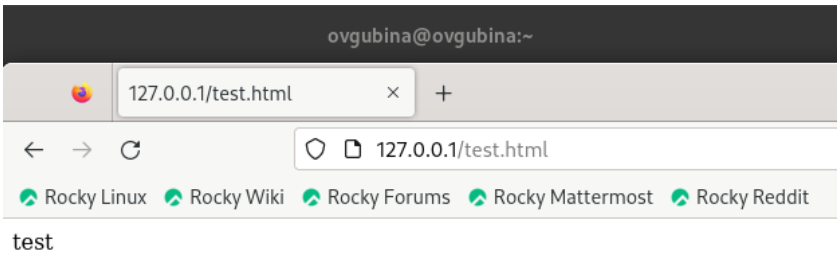

Рис. 4.18: Обращение к файлу через веб-сервер

12. Изучите справку man httpd_selinux и выясните, какие контексты файлов определены для httpd.

Вызвать данную справку неудалось, контексты безопасновти httpd были просмотрены через интернет.

Сопоставьте их с типом файла test.html. Проверить контекст файла можно командой `ls -Z /var/www/html/test.html` (рис. 4.19). Контекст безопасности

файла - **httpd_sys_content_t**. Данный контекст входит в перечень контекстов безопасности httpd.

Роль **object_r** используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах. 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.

Рис. 4.19: Контекст безропасности html файла

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 (рис. 4.20):

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

Рис. 4.20: Смена контекста безропасности html файла

Видим, что контекст безопасности действительно изменился.

14. Попробуйте ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html (рис. 4.21).

Мы получили сообщение об ошибке: **Forbidden You don't have permission to access /test.html on this server.**

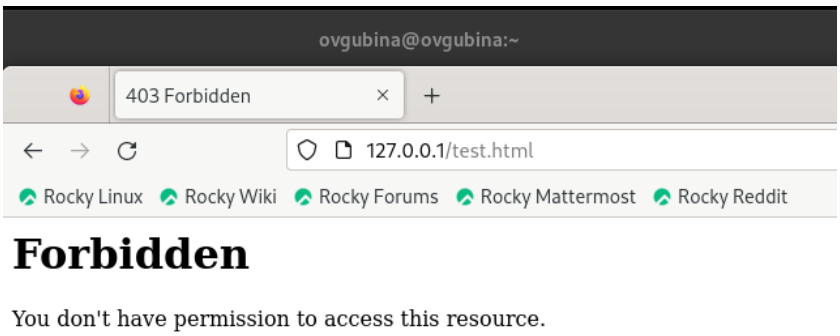

Рис. 4.21: Попытка получения доступа к файлу через веб-сервер

15. Проанализируйте ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю? `ls -l /var/www/html/test.html` (рис. 4.22) - нет доступа к файлу из-за недопустимого контекста безопасности для httpd.

Просмотрите log-файлы веб-сервера Apache. Также просмотрите системный лог-файл: `tail /var/log/messages` (рис. 4.22).

Если в системе окажутся запущенными процессы setroubleshootd и audtd (рис. 4.22), то вы также сможете увидеть ошибки, аналогичные указанным выше, в файле /var/log/audit/audit.log (рис. 4.23).

Рис. 4.22: tail /var/log/messages

Рис. 4.23: /var/log/audit/audit.log

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 (рис. 4.24).

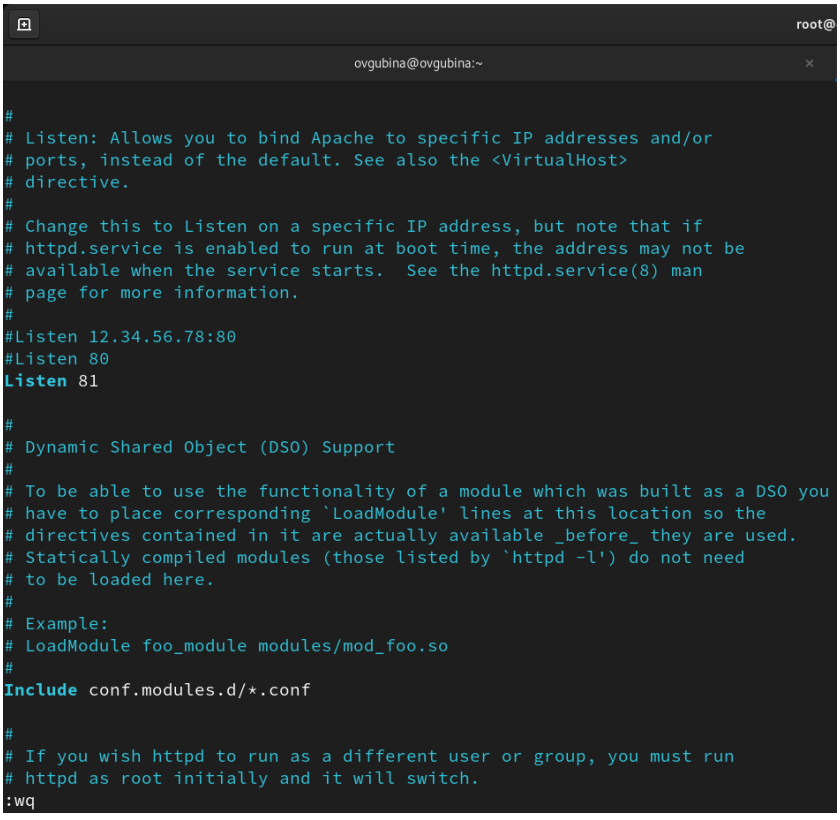

Рис. 4.24: Смена прослушиваемого порта

17. Выполните перезапуск веб-сервера Apache (рис. 4.25). Произошёл сбой? Поясните почему?

```
[root@ovgubina ~]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@ovgubina ~]#
```

Рис. 4.25: Перезапуск Apache

Никакой ошибки не возникает, поскольку в изначальных настройках системы порт 81 уже был прописан в рекомендациях системы.

18. Проанализируйте лог-файлы: `tail -nl /var/log/messages` - нет никаких ошибок (рис. 4.26). Просмотрите файлы `/var/log/http/error_log` (рис. 4.27), `/var/log/http/access_log` (рис. 4.28) и `/var/log/audit/audit.log` (рис. 4.29) и выясните, в каких файлах появились записи - нет записей об ошибках, т.к. нет ошибок.

```
[root@ovgubina ~]# tail /var/log/messages
Oct  9 20:14:08 ovgubina systemd[1]: Starting The Apache HTTP Server...
Oct  9 20:14:08 ovgubina systemd[1]: Started The Apache HTTP Server.
Oct  9 20:14:08 ovgubina httpd[7156]: Server configured, listening on: port 81
Oct  9 20:14:27 ovgubina systemd[1]: Stopping The Apache HTTP Server...
Oct  9 20:14:28 ovgubina systemd[1]: httpd.service: Deactivated successfully.
Oct  9 20:14:28 ovgubina systemd[1]: Stopped The Apache HTTP Server.
Oct  9 20:14:28 ovgubina systemd[1]: Starting The Apache HTTP Server...
Oct  9 20:14:28 ovgubina httpd[7400]: Server configured, listening on: port 81
Oct  9 20:14:28 ovgubina systemd[1]: Started The Apache HTTP Server.
Oct  9 20:14:34 ovgubina systemd[1]: fprintd.service: Deactivated successfully.
[root@ovgubina ~]#
```

Рис. 4.26: tail -nl /var/log/messages

```
[root@ovgubina ~]# cat /var/log/httpd/error_log
[Mon Oct 09 19:37:46.335070 2023] [core:notice] [pid 6096:tid 6096] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Mon Oct 09 19:37:46.337081 2023] [suexec:notice] [pid 6096:tid 6096] AH01232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Mon Oct 09 19:37:46.364554 2023] [lbmethod_heartbeat:notice] [pid 6096:tid 6096] AH02282: No slotmem from mod_heartmonitor
[Mon Oct 09 19:37:46.381672 2023] [mpm_event:notice] [pid 6096:tid 6096] AH00489: Apache/2.4.53 (Rocky Linux) configured -- resuming normal operations
[Mon Oct 09 19:37:46.381834 2023] [core:notice] [pid 6096:tid 6096] AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'
[Mon Oct 09 20:06:00.204611 2023] [core:error] [pid 6106:tid 6235] (13)Permission denied: [client 127.0.0.1:46850] AH00035: access to /test.html denied (files
ystem path '/var/www/html/test.html') because search permissions are missing on a component of the path
[Mon Oct 09 20:14:07.381718 2023] [mpm_event:notice] [pid 6096:tid 6096] AH00492: caught SIGWINCH, shutting down gracefully
[Mon Oct 09 20:14:08.472088 2023] [core:notice] [pid 7156:tid 7156] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Mon Oct 09 20:14:08.475792 2023] [suexec:notice] [pid 7156:tid 7156] AH01232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Mon Oct 09 20:14:08.497138 2023] [lbmethod_heartbeat:notice] [pid 7156:tid 7156] AH02282: No slotmem from mod_heartmonitor
[Mon Oct 09 20:14:08.530943 2023] [mpm_event:notice] [pid 7156:tid 7156] AH00489: Apache/2.4.53 (Rocky Linux) configured -- resuming normal operations
[Mon Oct 09 20:14:08.531003 2023] [core:notice] [pid 7156:tid 7156] AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'
[Mon Oct 09 20:14:27.553478 2023] [mpm_event:notice] [pid 7156:tid 7156] AH00492: caught SIGWINCH, shutting down gracefully
[Mon Oct 09 20:14:28.641464 2023] [core:notice] [pid 7400:tid 7400] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Mon Oct 09 20:14:28.642912 2023] [suexec:notice] [pid 7400:tid 7400] AH01232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Mon Oct 09 20:14:28.663487 2023] [lbmethod_heartbeat:notice] [pid 7400:tid 7400] AH02282: No slotmem from mod_heartmonitor
[Mon Oct 09 20:14:28.678642 2023] [mpm_event:notice] [pid 7400:tid 7400] AH00489: Apache/2.4.53 (Rocky Linux) configured -- resuming normal operations
[Mon Oct 09 20:14:28.678690 2023] [core:notice] [pid 7400:tid 7400] AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'
[root@ovgubina ~]#
```

Рис. 4.27: /var/log/http/error_log

Рис. 4.28: /var/log/http/access_log

Рис. 4.29: /var/log/audit/audit.log

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` (рис. 4.30). Добавление порта не производим, т.к. нам известно, что он и так уже добвлен - сразу смотрим список.

Порт 81 есть в списке.

Рис. 4.30: Список портов

20. Попробуйте запустить веб-сервер Apache ещё раз. Поняли ли вы, почему он сейчас запустился, а в предыдущем случае не смог?

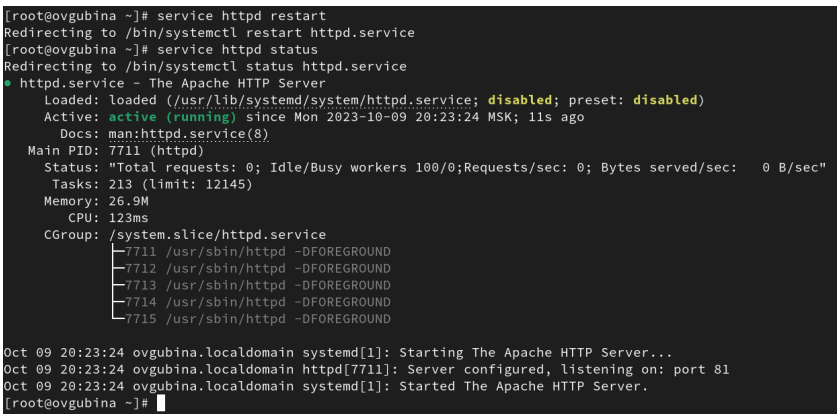

Рис. 4.31: Запуск веб-сервера

Сервер запустился также успешно, как и в тот раз, посколько оба раза порт 81 был в списке портов.

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` (рис. 4.32).

После этого попробуйте получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html (рис. 4.33).

Втидим содержимое файла — слово «test».

Рис. 4.32: Возвращаем контекст безопасности

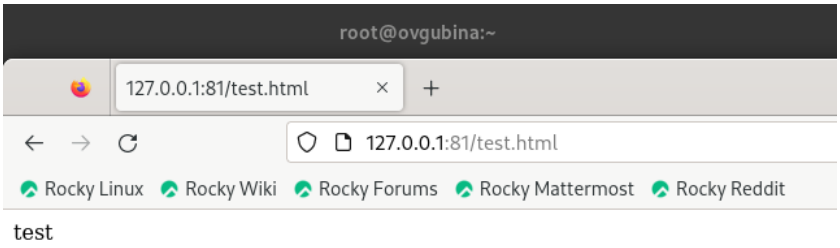

Рис. 4.33: Получение доступа к файлу через веб-сервер

22. Исправьте обратно конфигурационный файл apache, вернув Listen 80 (рис. 4.34).

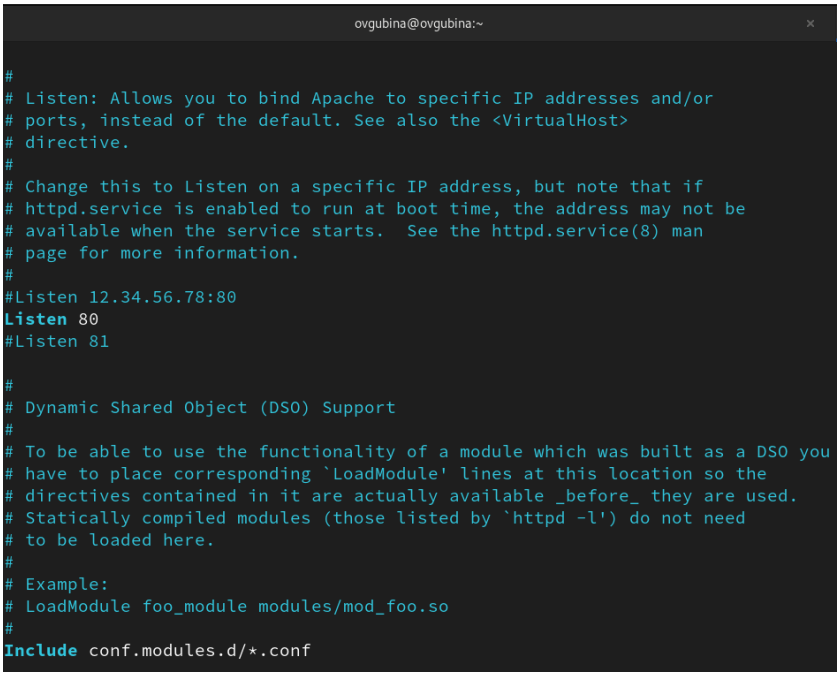

Рис. 4.34: Изменение прослушиваемого порта

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.

Мы не можем этого сделать, поскольку это была изначальная настройка системы, работает - не трогай, поэтому мы пока не станем этого делать (рис. 4.35).

24. Удалите файл /var/www/html/test.html (рис. 4.35): `rm /var/www/html/test.html`

```
[root@ovgubina ~]# semanage port -d -t http_port_t -p tcp 81
ValueError: Port tcp/81 is defined in policy, cannot be deleted
[root@ovgubina ~]# vim /etc/httpd/conf/httpd.conf
[root@ovgubina ~]# rm /var/www/html/test.html
rm: remove regular file '/var/www/html/test.html'? y
[root@ovgubina ~]# ls /var/www/html
[root@ovgubina ~]#
```

Рис. 4.35: Удаление файла

5 Выводы

Развила навыки администрирования ОС Linux. Получила первое практическое знакомство с технологией SELinux1. Проверила работу SELinx на практике совместно с веб-сервером Apache.

Список литературы

1. SELinux – описание и особенности работы с системой [Электронный ресурс]. 2023. URL: https://habr.com/ru/companies/kingservers/articles/209644/.
2. Что такое Apache [Электронный ресурс]. 2023. URL: https://eternalhost.net/blog/hosting/web-server-apache.
3. Установка веб-сервера Apache на Linux [Электронный ресурс]. 2023. URL: https://wiki.merionet.ru/articles/ustanovka-apache-v-linux.