Лабораторная работа №6

Дисциплина: Информационная безопасность

Дорофеева Алёна Тимофеевна

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.[**01?**]

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.[**02?**]

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.[**03?**]

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).

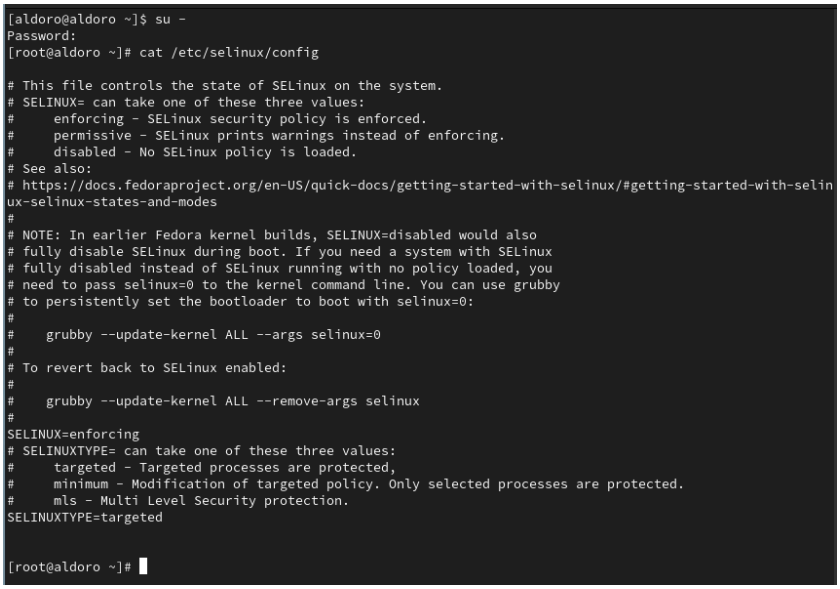

Рис. 4.1: Проверка политики и режима

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

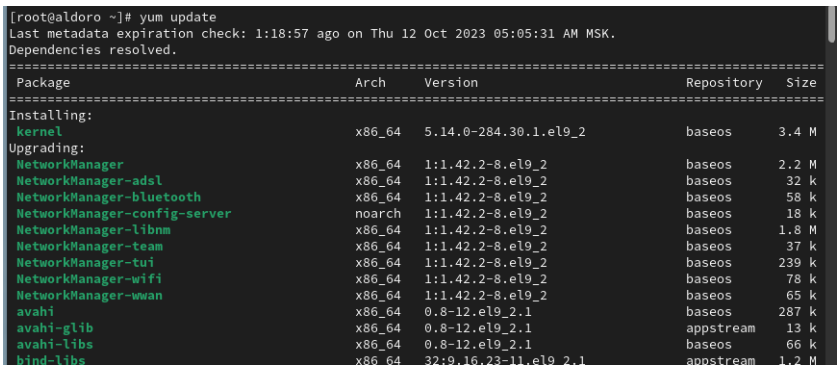

Рис. 4.2: Обновление

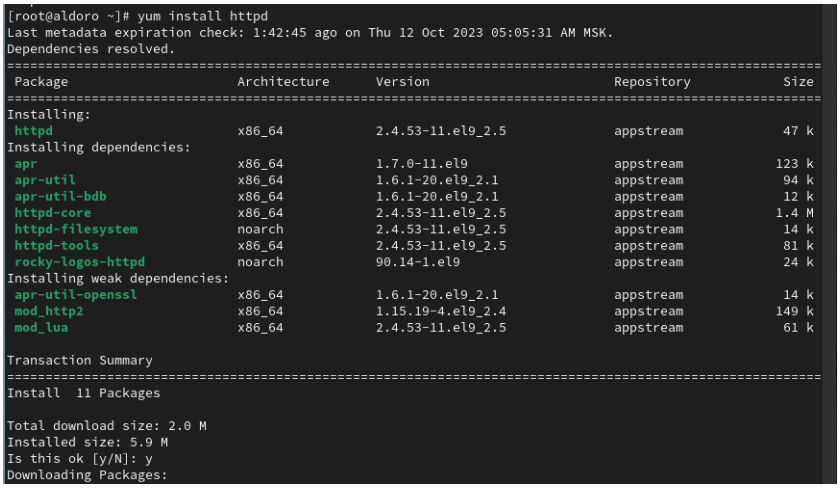

Рис. 4.3: Загрузка Apache

Далее зададим `ServerName test.ru` в конфигурационной файле httpd (рис. 4.5), для этого сперва найдем, где он находится (рис. 4.4).

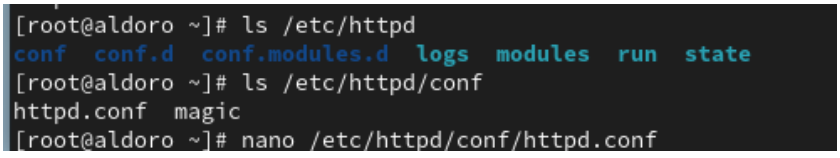

Рис. 4.4: Расположение конфигурационного файла httpd

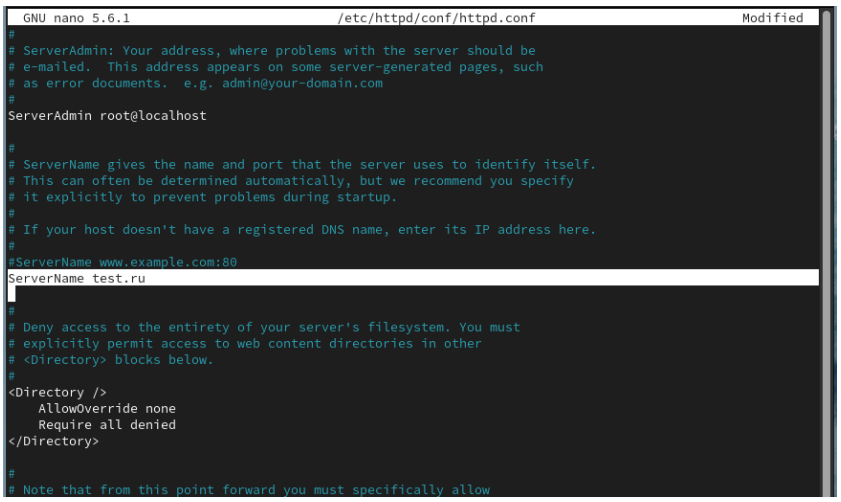

Рис. 4.5: Задаем ServerName

Чтобы пакетный фильтр в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола tcp добавим разрешающие правила (рис. 4.6):

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

Рис. 4.6: 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. 4.7).

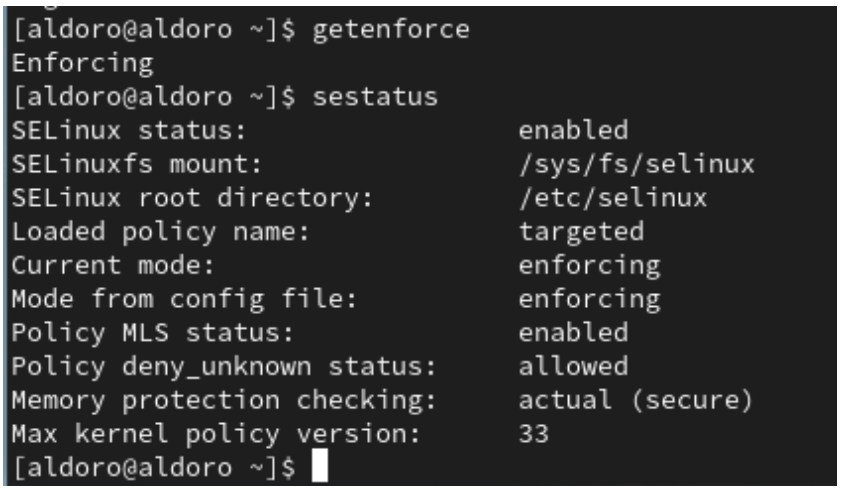

Рис. 4.7: Режим enforcing политики targeted

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).

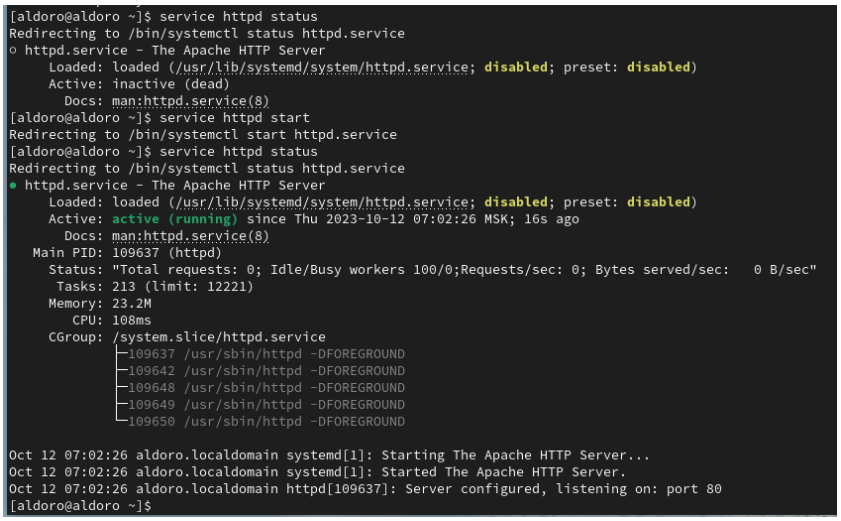

Рис. 4.8: Запуск

3. Найдите веб-сервер Apache в списке процессов, определите его контекст безопасности и занесите эту информацию в отчёт. Например, можно ис-

пользовать команду `ps auxZ | grep httpd` (4.9).

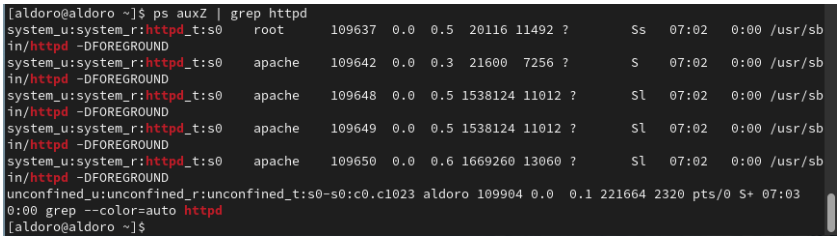

Рис. 4.9: Контекст безопасности

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

4. Посмотрите текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -b httpd` (рис. 4.10). Многие из них находятся в положении «off».

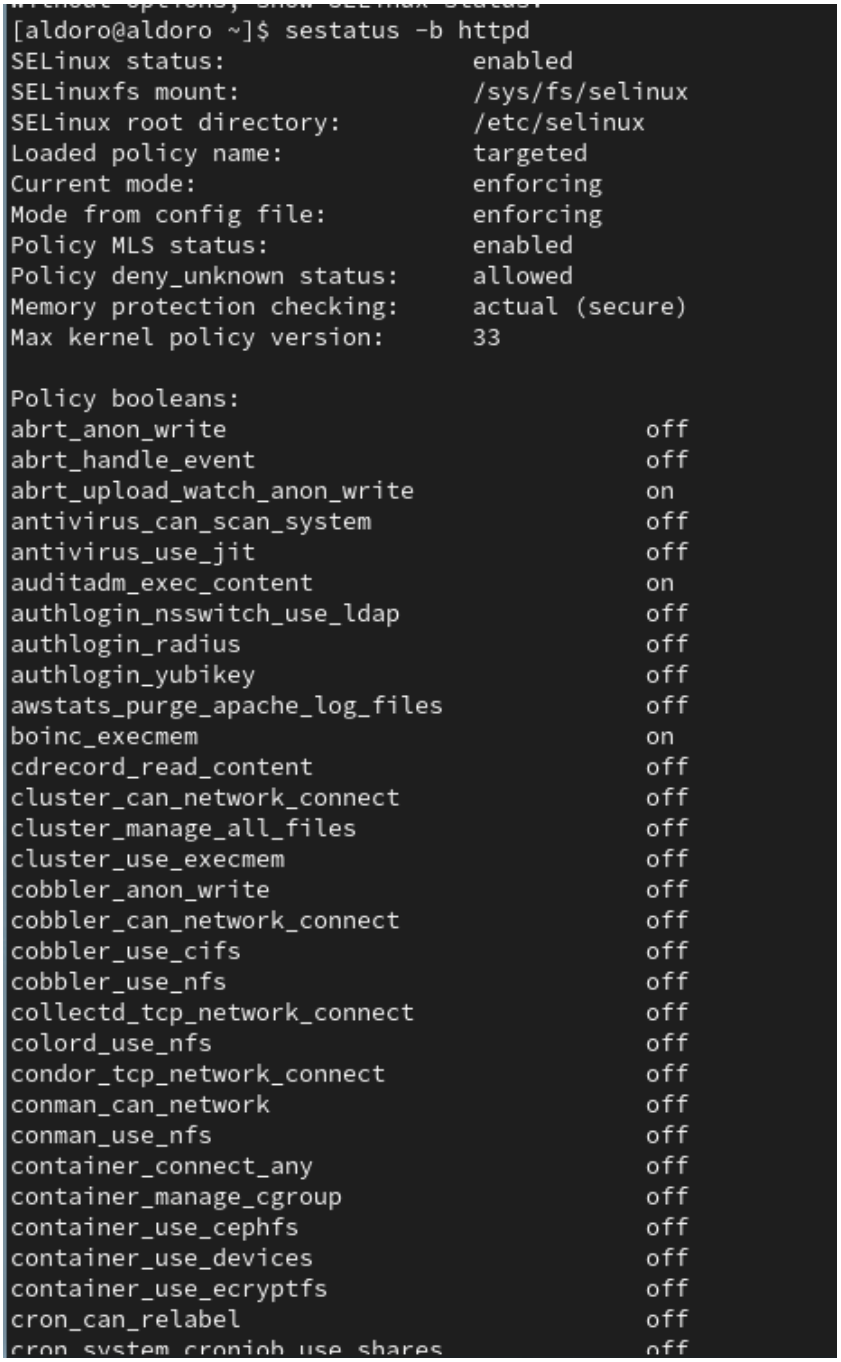

Рис. 4.10: Переключатели SELinux для Apache

5. Посмотрите статистику по политике с помощью команды `seinfo`, также определите множество пользователей, ролей, типов (рис. 4.11).

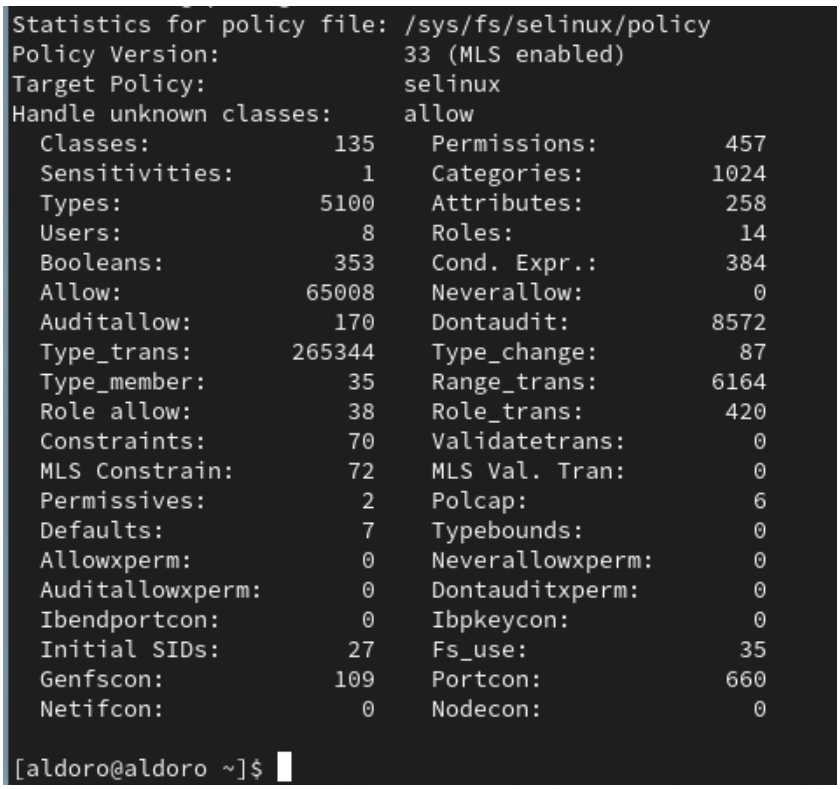

Рис. 4.11: Статистика по политике

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

6. Определите тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды `ls -lZ /var/www` (рис. 4.12).

Рис. 4.12: Типы файлов

В каталоге находятся только директории.

7. Определите тип файлов, находящихся в директории /var/www/html: `ls -lZ /var/www/html` (рис. 4.13).

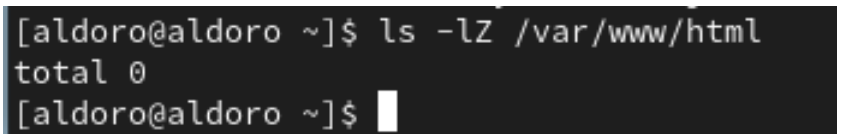

Рис. 4.13: Типы файлов

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. 4.14):

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

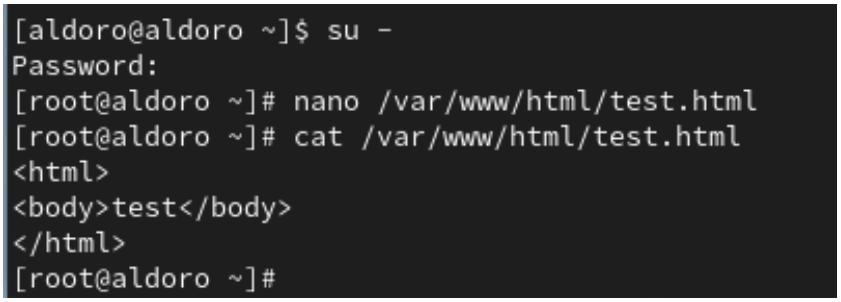

Рис. 4.14: Создание html файла

10. Проверьте контекст созданного вами файла. Занесите в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html (рис. 4.15).

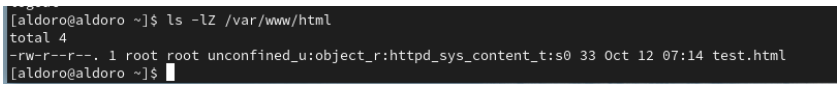

Рис. 4.15: Создание html файла

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. 4.16). Файл был успешно отображён.

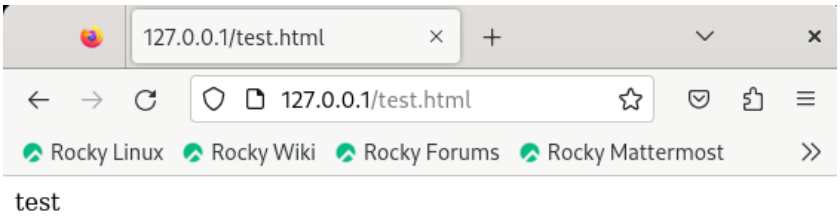

Рис. 4.16: Обращение к файлу через веб-сервер

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 /var/www/html/test.html` (рис. 4.17). 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.

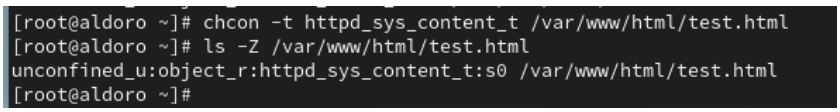

Рис. 4.17: Контекст безропасности html файла

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. 4.18):

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

Рис. 4.18: 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. 4.19).

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

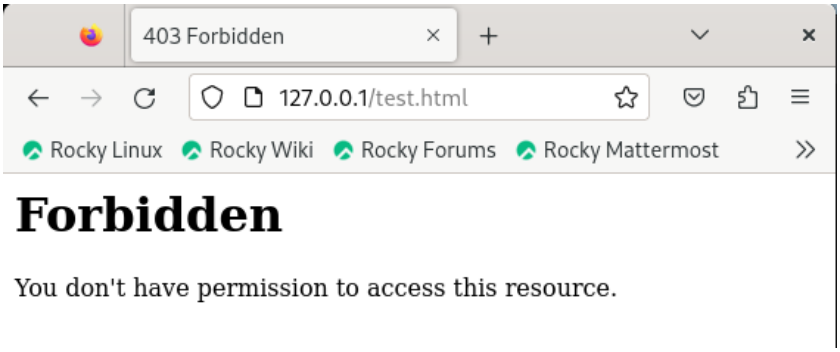

Рис. 4.19: Попытка получения доступа к файлу через веб-сервер

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. **??**) - 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. **??**).

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

Рис. 4.20: tail /var/log/messages

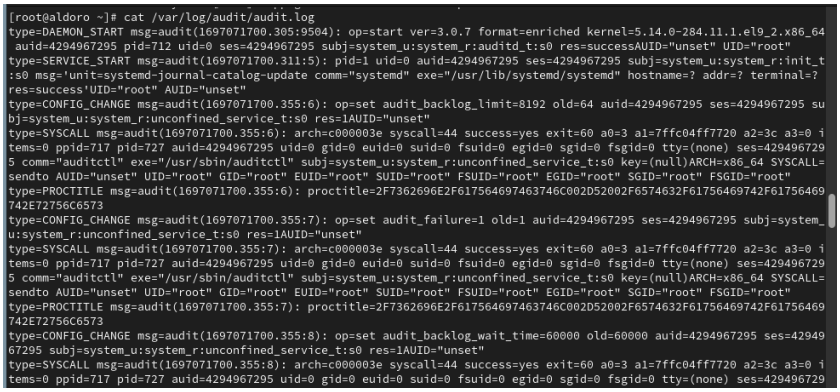

Рис. 4.21: /var/log/audit/audit.log

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. 4.22).

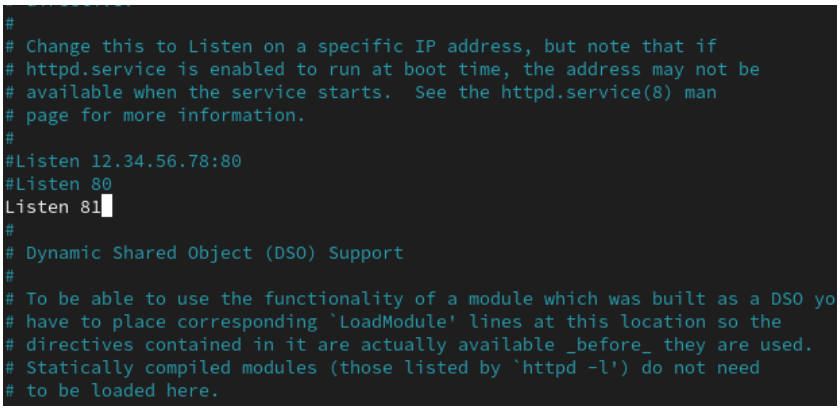

Рис. 4.22: Смена прослушиваемого порта

17. Выполните перезапуск веб-сервера Apache (рис. 4.23). Произошёл сбой? Поясните почему?

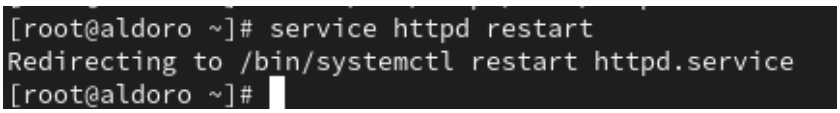

Рис. 4.23: Перезапуск Apache

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. 4.24). Просмотрите файлы `/var/log/http/error_log` (рис. 4.25), `/var/log/http/access_log` (рис. 4.26) и `/var/log/audit/audit.log` (рис. 4.27) и выясните, в каких файлах появились записи - нет записей об ошибках, т.к. нет ошибок.

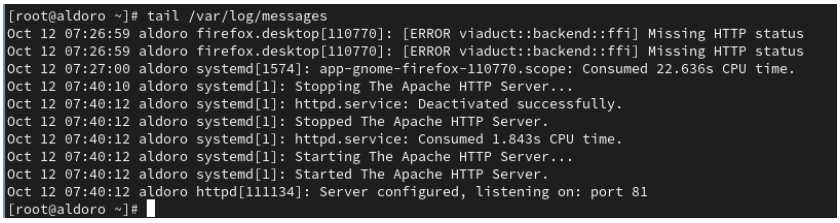

Рис. 4.24: tail -nl /var/log/messages

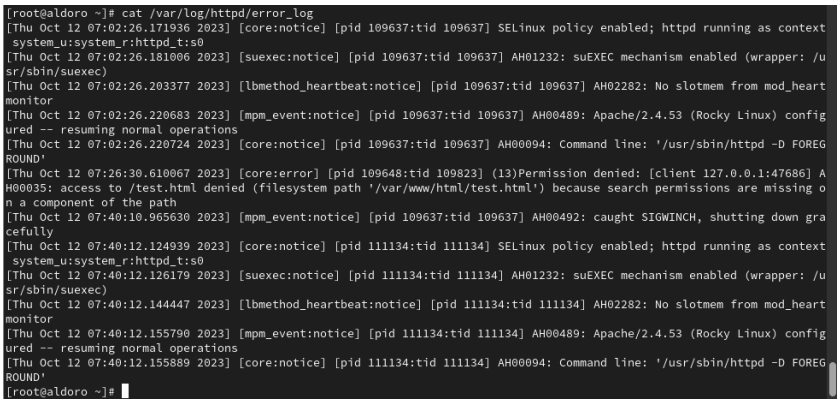

Рис. 4.25: /var/log/http/error_log

Рис. 4.26: /var/log/http/access_log

Рис. 4.27: /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. 4.28). Добавление порта не производим, т.к. нам известно, что он и так уже добвлен - сразу смотрим список.

Порт 81 есть в списке.

Рис. 4.28: Список портов

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

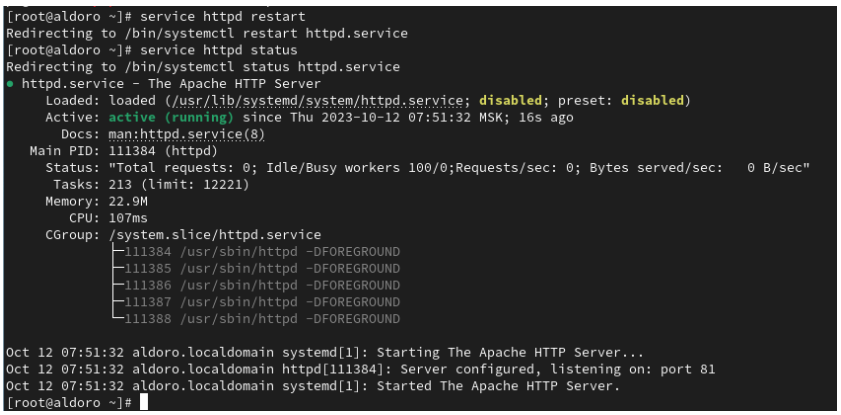

Рис. 4.29: Запуск веб-сервера

Сервер перезапустился 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. 4.30).

После этого попробуйте получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html (рис. 4.31).

Видим содержимое файла — слово «test».

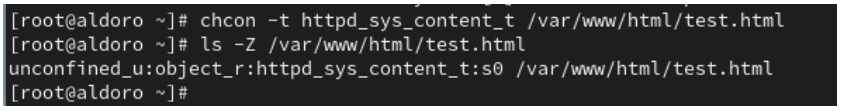

Рис. 4.30: Возвращаем контекст безопасности

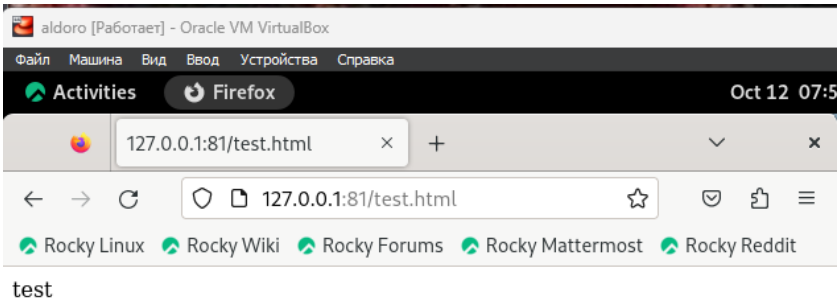

Рис. 4.31: Получение доступа к файлу через веб-сервер

22. Исправьте обратно конфигурационный файл apache, вернув Listen 80 (рис. 4.32).

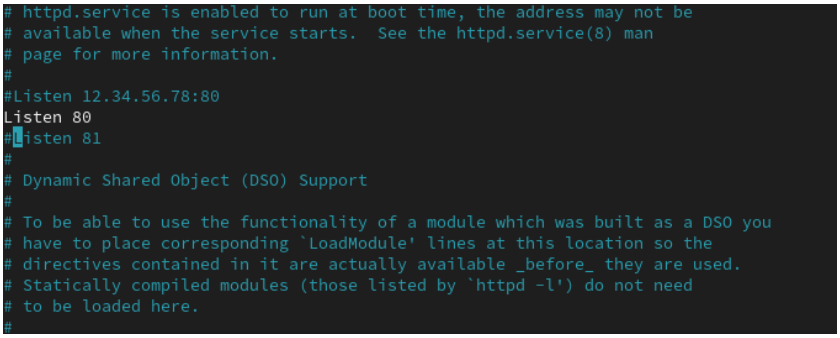

Рис. 4.32: Изменение прослушиваемого порта

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. (рис. 4.33).

24. Удалите файл /var/www/html/test.html (рис. 4.33): `rm /var/www/html/test.html`

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

Рис. 4.33: 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