Отчёт по лабораторной работе №6

Мандатное разграничение прав в Linux

Ханина Ирина Владимировна, НБИбд-02-18

Содержание

# Цель работы

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

# Задание

Формулировка задания представлена в разделе 6.4 “Порядок выполнения работы” в файле “Лабораторная работа № 6. Описание”.

# Теоретическое введение

Linux с улучшенной безопасностью (SELinux) - это механизм безопасности с мандатной моделью контроля доступа (MAC), реализованный в ядре. Это разграничение контроля доступа внедряется поверх того, что уже есть в каждом дистрибутиве Linux, DAC (Discretionary Access Control). Можно сказать, что SELinux расширяет возможности стандартной системы безопасности. Первоначально он был разработан Агентством национальной безопасности США для защиты компьютерных систем от вторжения злоумышленников и взлома. Со временем SELinux появился в открытом доступе, и тогда различные дистрибутивы включили его в свой код. Он был впервые представлен в CentOS 4 и значительно улучшен в более поздних выпусках CentOS. [[1]](https://wiki.centos.org/HowTos/SELinux)

Система SELinux – это средство для точной настройки требований контроля доступа. С помощью SELinux можно определить, что позволено делать пользователю или процессу. Она ограничивает каждый процесс своим собственным доменом, поэтому процесс может взаимодействовать только с определенными типами файлов и другими процессами из разрешенных доменов. Это предотвращает взлом любого процесса и получение хакерами общесистемного доступа. [[3]](https://www.8host.com/blog/vvedenie-v-selinux-v-centos-7-bazovye-ponyatiya/)

SELinux имеет три основных режим работы:

* Enforcing: Режим по-умолчанию. При выборе этого режима все действия, которые каким-то образом нарушают текущую политику безопасности, будут блокироваться, а попытка нарушения будет зафиксирована в журнале.
* Permissive: В случае использования этого режима, информация о всех действиях, которые нарушают текущую политику безопасности, будут зафиксированы в журнале, но сами действия не будут заблокированы.
* Disabled: Полное отключение системы принудительного контроля доступа. [[2]](https://habr.com/ru/company/kingservers/blog/209644/)

Для просмотра текущего режима и других настроек SELinux используется команда sestatus. Узнать статус SELinux можно при помощи команды getenforce. Команда setenforce позволяет быстро переключаться между режимами Enforcing и Permissive, изменения вступают в силу без перезагрузки. Но если вы включаете или отключаете SELinux, требуется перезагрузка, ведь нужно заново устанавливать метки безопасности в файловой системе. [[3]](https://www.8host.com/blog/vvedenie-v-selinux-v-centos-7-bazovye-ponyatiya/)

# Выполнение лабораторной работы

1. Выполнила подготовку лабораторного стенда: вошла в систему от имени суперпользователя и запустила команду для установки веб-сервера Apache: yum install httpd ([рис. 1](image/1.png)), задала параметр ServerName в конфигурационном файле /etc/httpd/httpd.conf ([рис. 2](image/2.png)), отключила пакетный фильтр командами iptables -F, iptables -P INPUT ACCEPT и iptables -P OUTPUT ACCEPT. ([рис. 3](image/3.png))

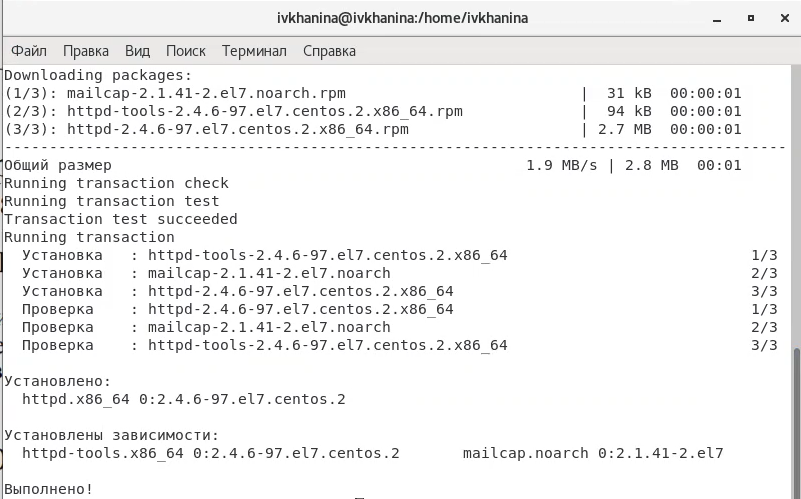


Рис 1. Установка веб-сервера

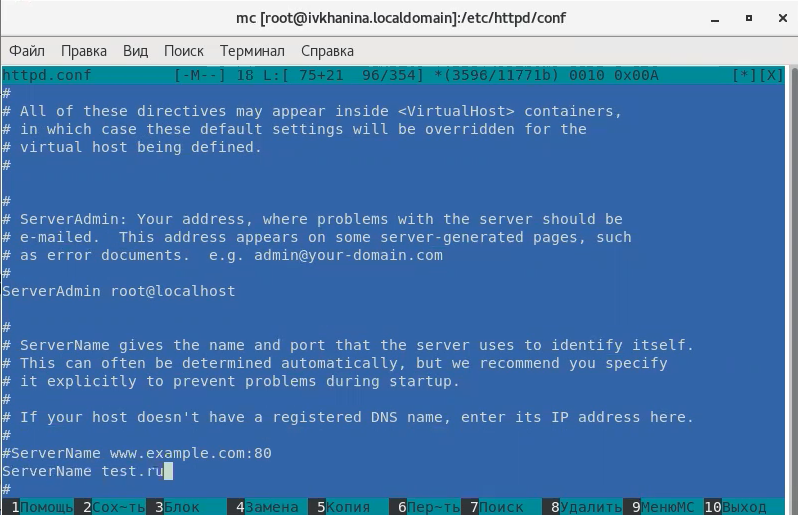


Рис 2. Изменение конфигурационного файла /etc/httpd/httpd.conf

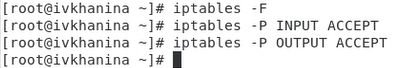


Рис 3. Отключение пакетного фильтра

1. Я вошла в систему и убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus. ([рис. 4](image/4.png)). Далее с помощью команды service httpd status я обратилась с помощью браузера к веб-серверу, запущенному на компьютере, и убедилась, что он работает. ([рис. 5](image/5.png))



Рис 4. SELinux работает в режиме enforcing политики targeted

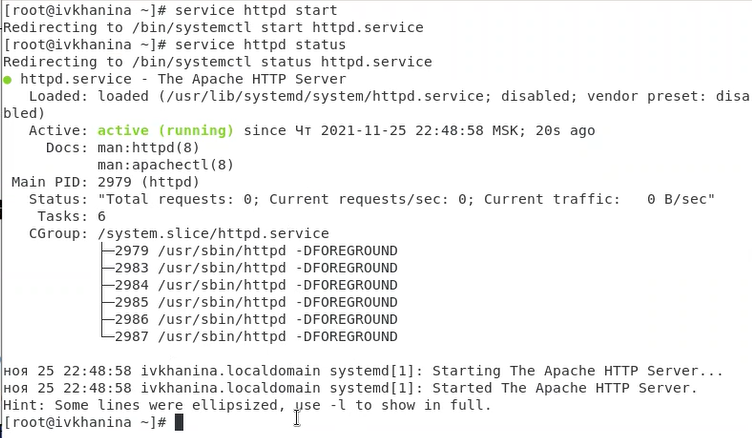


Рис 5. Команда service httpd status

1. Затем я нашла веб-сервер Apache в списке процессов и определила его контекст безопасности, используя команду ps auxZ | grep httpd. ([рис. 6](image/6.png))

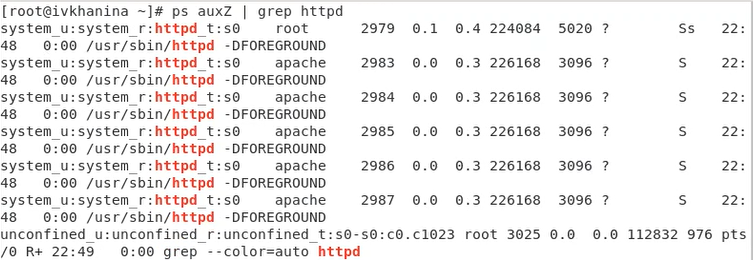


Рис 6. Команда ps auxZ | grep httpd

1. Я посмотрела текущее состояние переключателей SELinux для Apache с помощью команды sestatus -bigrep httpd и обратила внимание, что многие из них находятся в положении «off». ([рис. 7](image/7.png))

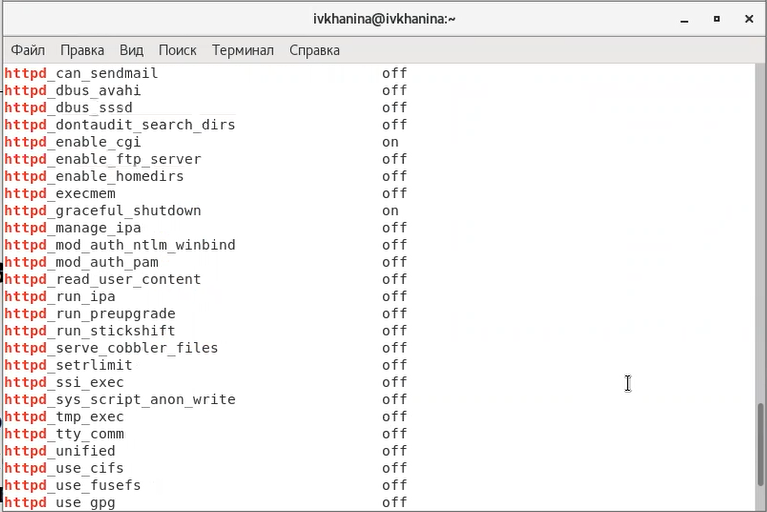


Рис 7. Команда sestatus -bigrep httpd

1. Я посмотрела статистику по политике с помощью команды seinfo, также определила множество пользователей, ролей, типов. Затем я определила тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды ls -lZ /var/www. А потом определила тип файлов, находящихся в директории /var/www/html, введя команду ls -lZ /var/www/html. Определила круг пользователей, которым разрешено создание файлов в директории /var/www/html. ([рис. 8](image/8.png))

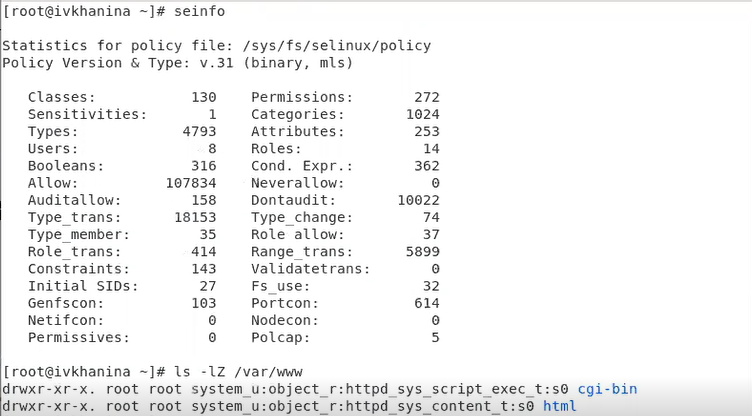


Рис 8. Команды seinfo, ls -lZ /var/www и ls -lZ /var/www/html

1. Далее от имени суперпользователя я создала html-файл /var/www/html/test.html. Проверила контекст созданного файла. ([рис. 9](image/9.png))

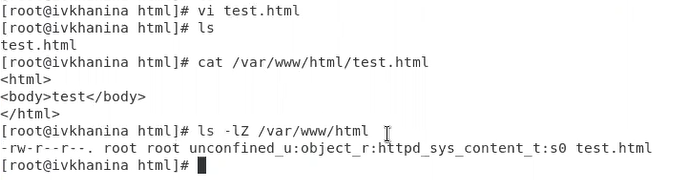


Рис 9. Создание файла /var/www/html/test.html

1. Обратилась к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Убедилась, что файл был успешно отображён. ([рис. 10](image/10.png))

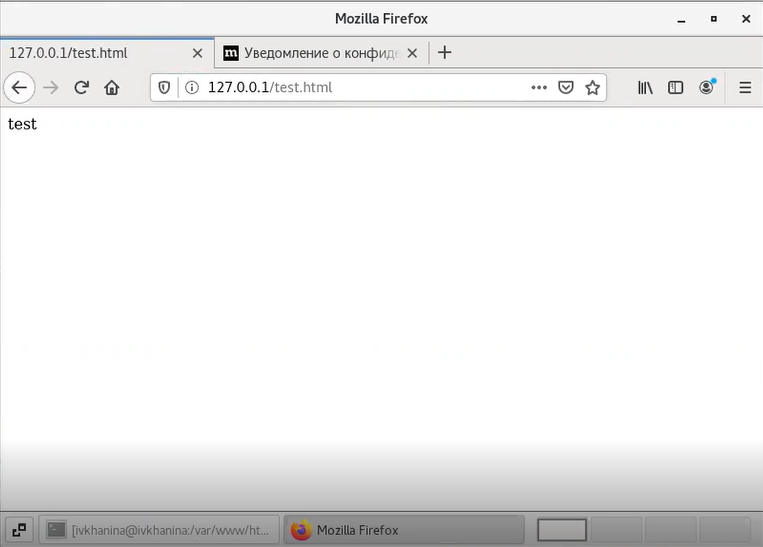


Рис 10. Обращение к файлу через сервер

1. Я изучила справку man httpd\_selinux и выяснила, какие контексты файлов определены для httpd. Затем я сопоставила их с типом файла test.html. Проверила контекст файла с помощью команды ls -Z /var/www/html/test.html. Я изменила контекст файла /var/www/html/test.html с httpd\_sys\_content\_t на samba\_share\_t командами chcon -t samba\_share\_t /var/www/html/test.html и ls -Z /var/www/html/test.html. Проверила, контекст поменялся. ([рис. 11](image/11.png))



Рис 11. Изменение контекста файла /var/www/html/test.html

1. Попробовала ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html и получила сообщение об ошибке. Из-за смены контекста доступ к файлу запрщен. ([рис. 12](image/12.png))

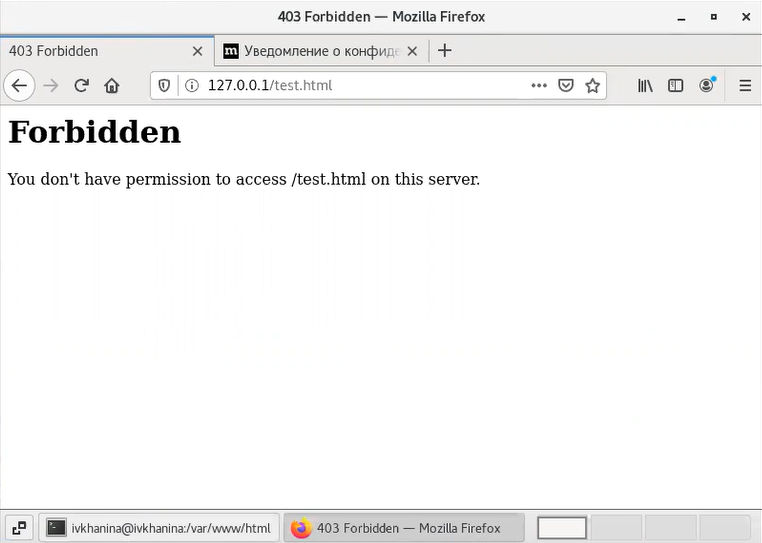


Рис 12. Доступ к файлу запрещен

1. Я просмотрела log-файлы веб-сервера Apache, а также системный лог-файл: tail /var/log/messages. ([рис. 13](image/13.png))

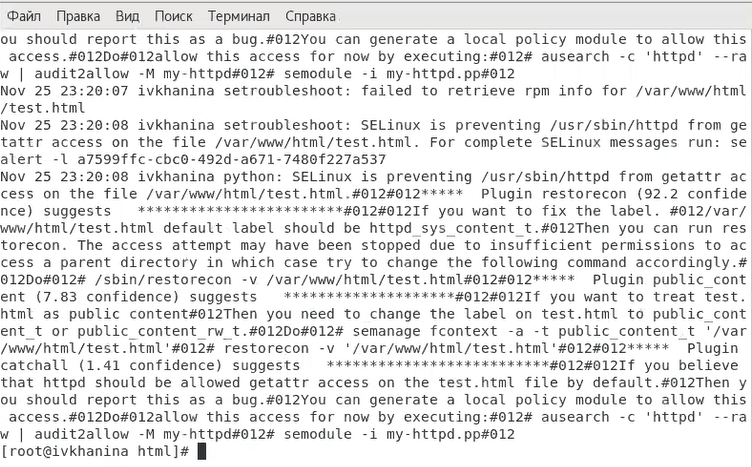


Рис 13. Команда tail /var/log/messages

1. Попробовала запустить веб-сервер Apache на прослушивание ТСР-порта 81. Для этого в файле /etc/httpd/httpd.conf нашла строчку Listen 80 и заменила её на Listen 81. ([рис. 14](image/14.png))

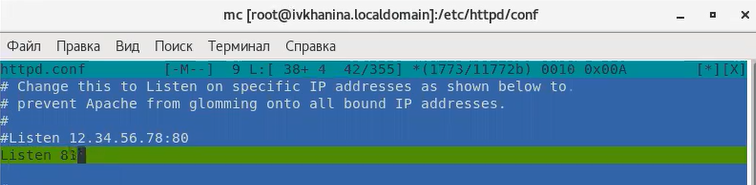


Рис 14. Изменение Listen 80 на Listen 81

1. Перезапустила веб-сервер Apache. Проанализировала лог-файлы: tail -nl /var/log/messages. Просмотрела файлы /var/log/http/error\_log, /var/log/http/access\_log и /var/log/audit/audit.log и выяснила, в каких файлах появились записи. ([рис. 15](image/15.png))

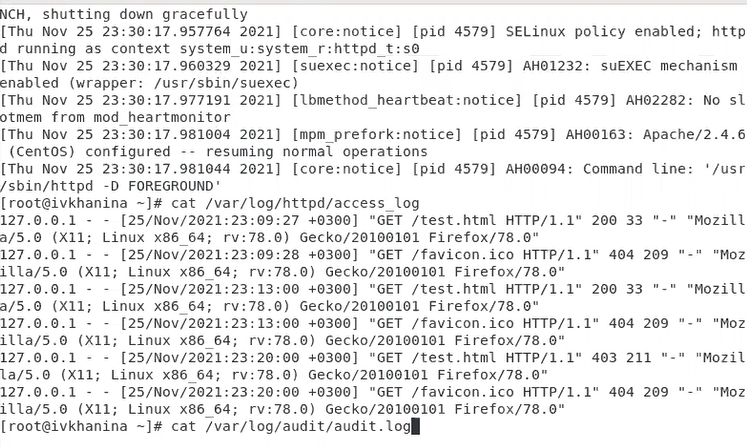


Рис 15. Просмотр log файлов

1. Выполнила команду semanage port -a -t http\_port\_t -р tcp 81. После этого проверила список портов командой semanage port -l | grep http\_port\_t. Убедителась, что порт 81 появился в списке. Запустила веб-сервер Apache ещё раз. ([рис. 16](image/16.png))



Рис 16. Команда semanage port -a -t http\_port\_t -р tcp 81

1. Вернула контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/test.html командами chcon -t httpd\_sys\_content\_t /var/www/html/test.html. ([рис. 17](image/17.png))

Рис 17. Изменение контекста файла /var/www/html/test.html

Рис 17. Изменение контекста файла /var/www/html/test.html

1. После этого попробовала получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html. Я увидела содержимое файла — слово «test». ([рис. 18](image/18.png))

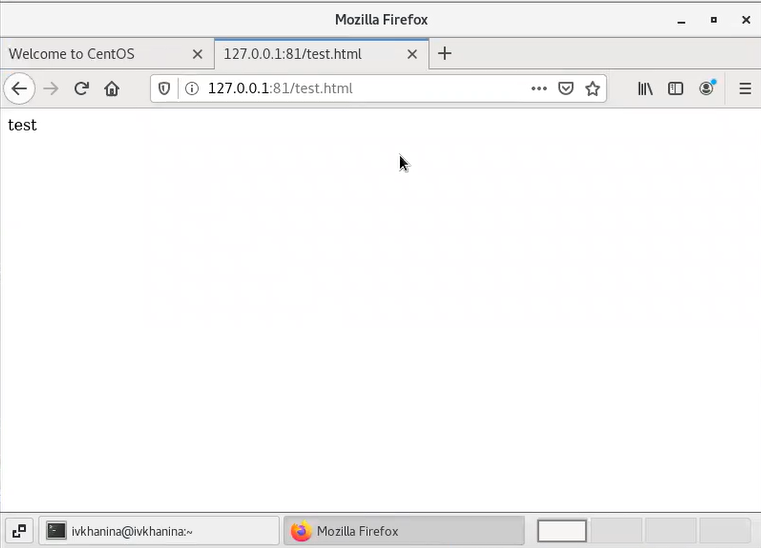


Рис 18. Обращение к файлу через сервер

1. Исправила обратно конфигурационный файл apache, вернув Listen 80. ([рис. 19](image/19.png))

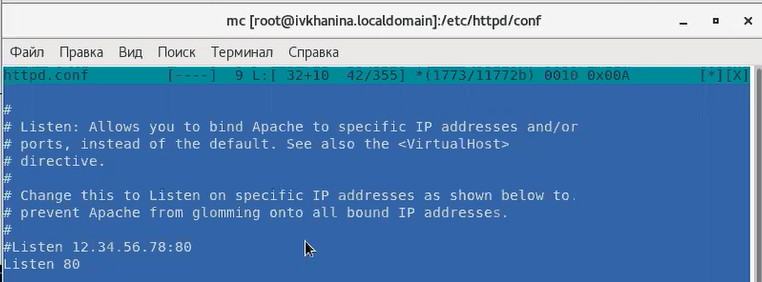


Рис 19. Изменение конфигурационного файла /etc/httpd/httpd.conf

1. Удалила привязку http\_port\_t к 81 порту с помощью команды semanage port -d -t http\_port\_t -p tcp 81 и проверила, что порт 81 удалён. Удалила файл /var/www/html/test.html командой /rm /var/www/html/test.html. ([рис. 20](image/20.png))

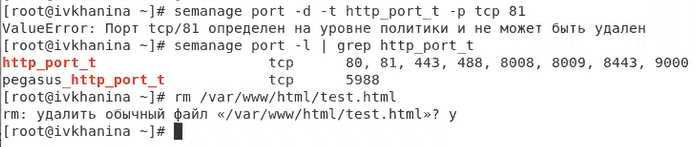


Рис 20. Удаление файла /var/www/html/test.html

# Выводы

В результате выполнения лабораторной работы я развила навыки администрирования ОС Linux, получила первое практическое знакомство с технологией SELinux, а также проверить работу SELinux на практике совместно с веб-сервером Apache.

# Список литературы

1. [SELinux](https://wiki.centos.org/HowTos/SELinux)
2. [SELinux – описание и особенности работы с системой](https://habr.com/ru/company/kingservers/blog/209644/)
3. [ВВЕДЕНИЕ В SELINUX В CENTOS 7: БАЗОВЫЕ ПОНЯТИЯ](https://www.8host.com/blog/vvedenie-v-selinux-v-centos-7-bazovye-ponyatiya/)