Лабораторная работа №6

Мандатное разграничение прав в Linux

Парфенова Елизавета Евгеньевна

Содержание

# 1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux

# 2 Задание

Проверить работу SELinx на практике совместно с веб-сервером Apache.

# 3 Теоретическое введение

**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.

Основные термины, использующиеся в SELinux:

* *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 отношения 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.

SELinux имеет три основных режим работы, при этом по умолчанию установлен режим Enforcing.

Режимы работы SELinux:

1. 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.
2. 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.
3. Disabled: Полное отключение системы принудительного контроля доступа [1].

Для просмотра контекста безопасности используется команда ls -Z с указанием адреса. Например: *ls -Z /var/www/html/index.html*

# 4 Выполнение лабораторной работы

Подготовим лабораторный стенд. Для этого проверим утсановку сервера Apache. У меня он установлен не был, поэтому я заранее провела его установку. Затем в конфигурационном файле /etc/httpd/conf/httpd.conf необходимо зададим параметр ServerName: *ServerName test.ru*. Также было необходимо проследить, чтобы пакетный фильтр был либо отключён, либо в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола tcp. Мной было решено отключить фильтр следующими командами

iptables -F  
iptables -P INPUT ACCEPT   
iptables -P OUTPUT ACCEPT

(рис. 1).

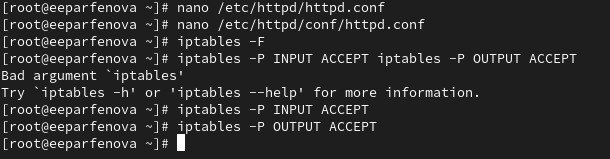


Рис. 1: Подготовка лабораторного стенда

Убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд *getenforce* и *sestatus*. Видим, что этой действительно так (рис. 2)



Рис. 2: Просмотр режима и политики SELinux

Запустим сервер Apache командой *sudo systemctl start httpd* и с помощью браузера обратимся к нему, убедившись, что сервер работает командой *sydo systemctl status httpd* (рис. 3).

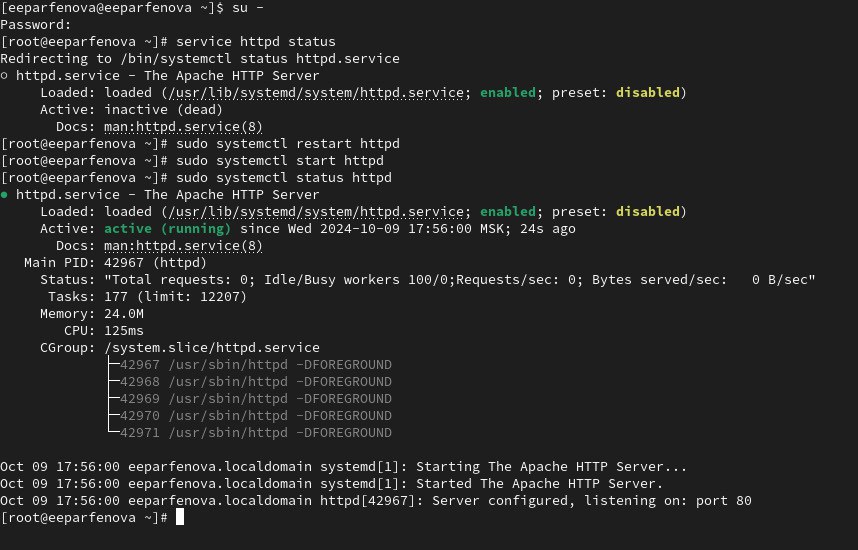


Рис. 3: Запуск и проверка работы сервера Apache

С помощью команд *ps auxZ | grep httpd* и *ps -eZ | grep httpd* найдем веб-сервер Apache в списке процессов, определив его контекст безопасности. Его контекст безопасности: system\_u:system\_r:httpd\_t:s0. (рис. 4)

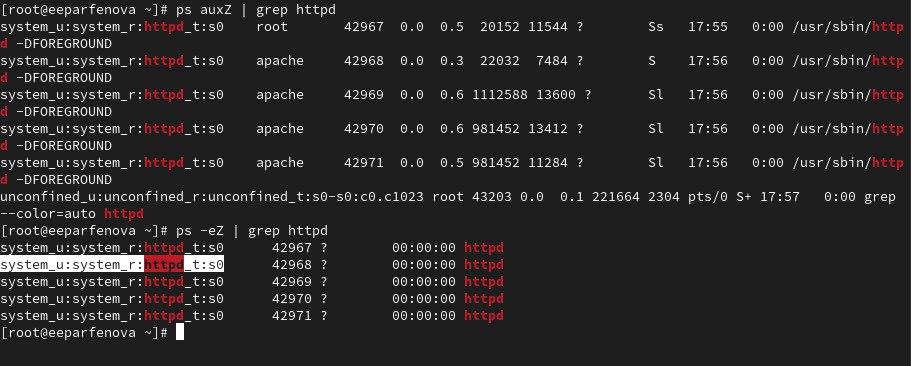


Рис. 4: Контекст безопасности сервера Apache

Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды *sestatus -b | grep httpd*. Обратим внимание, что многие из них находятся в положении «off». (рис. 5)

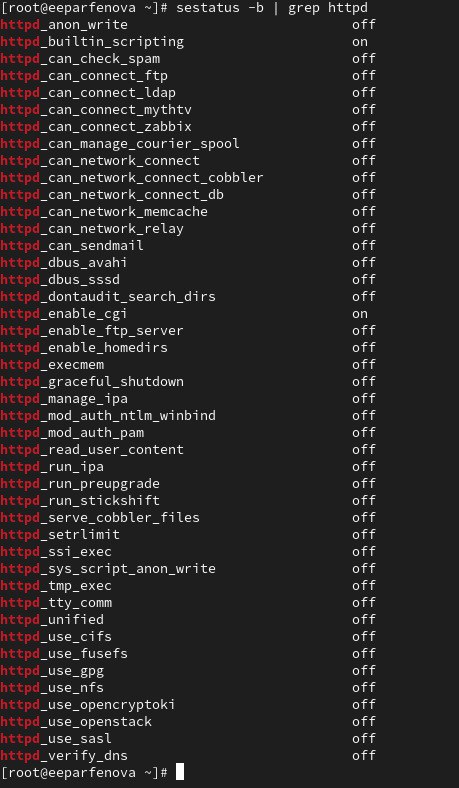


Рис. 5: Текущее состояние переключателей SELinux для Apache

Посмотрим статистику по политике с помощью команды *seinfo* (рис. 6), также определим множество пользователей, ролей, типов с помощью команд *seinfo -u*, *seinfo -r*, *seinfo -t* соответственно (рис. 7). Оно соотвсевтенно равно 8, 15, 5145.

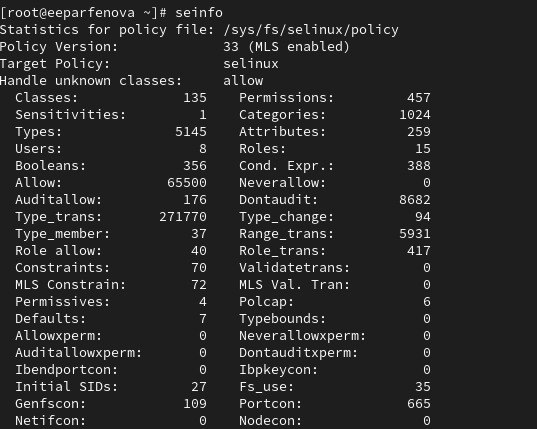


Рис. 6: Статистика по политике



Рис. 7: Множество пользователей, ролей, типов

Определим тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды *ls -lZ /var/www*. Видим, что все файлы и поддиректории принадлежат пользователю root, а также имеют некоторый контекст безопасностиОпределим тип файлов, находящихся в директории /var/www/html *ls -lZ /var/www/html*. Она оказалась пуста. (рис. 8)

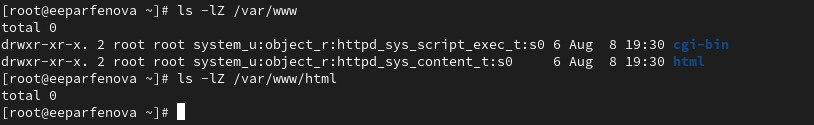


Рис. 8: Файлы и поддиректори в /var/www

Как уже было указано, создание файлов в директории /var/www/html доступно только владельцу, то есть пользователю root.

Создадим от имени суперпользователя html-файл */var/www/html/test.html* следующего содержания:

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

Проверим его контекст, присовенный по умолчанию созданным файлам в директории /var/www/html: unconfined\_u:object\_r:httpd\_sys\_content\_r:s0 (рис. 9).

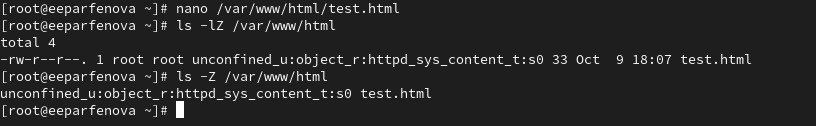


Рис. 9: Создание файла test.html и его контекст

Обратимся к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html убедимся, что файл был успешно отображён (рис. 10).

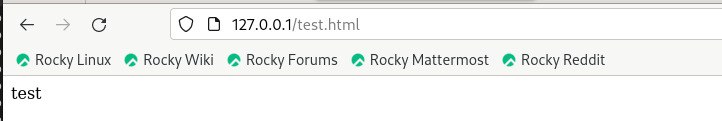


Рис. 10: Отображение файла через веб-сервер

Изучив справку man httpd\_selinux, увидим, что для файла test.html был определен один из контекстов безопасности httpd. Проверить контекст файла можно командой. *ls -Z /var/www/html/test.html* (мы уже делали это ранее, на рис. 9)

Рассмотрим полученный контекст детально. - Обратим внимание, что так как по умолчанию пользователи являются свободными от типа, 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. - 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.

Изменим 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. Сдлаем это с помощью команды *chcon -t samba\_share\_t /var/www/html/test.html*. Затем проверим успешность действия командой *ls -Z /var/www/html/test.html* (рис. 11).

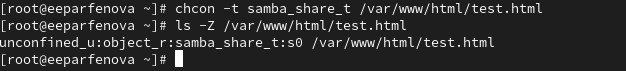


Рис. 11: Изменение контекста файла

Попробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Но получаем следующее сообщение об ошибке (рис. 12):

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

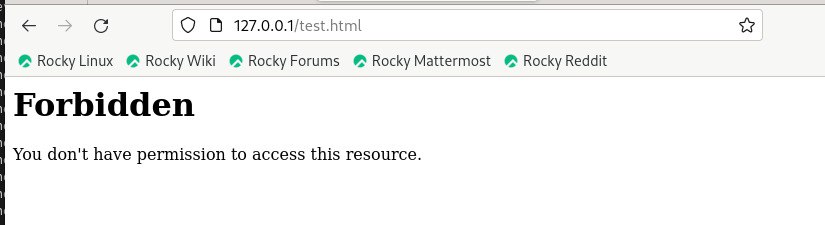


Рис. 12: Сообщение об ошибке после изменения контекста файла

Данная ситуация произошла, так как конекст безопасности был изменен на то, к которому httpd не имеет доуступа, то есть читать файл по прежнему могут все, но отображать его через браузер запрещено.

Просмотрим log-файлы веб-сервера Apache. Просмотрим системный лог-файл командой *tail /var/log/messages* (рис. 13) и файл /var/log/audit/audit.log с помощью команды *tail /var/log/audit/audit.log* (рис. 14). Видим в системе запущены процессы setroubleshootd и audtd, которые свидетельствуют об ошибке доступа - пишется, что SELinux препятствует серверу в получении полного доступа к вызываемому файлу (это и является причиной, почему мы не можем прочесть его через браузер).

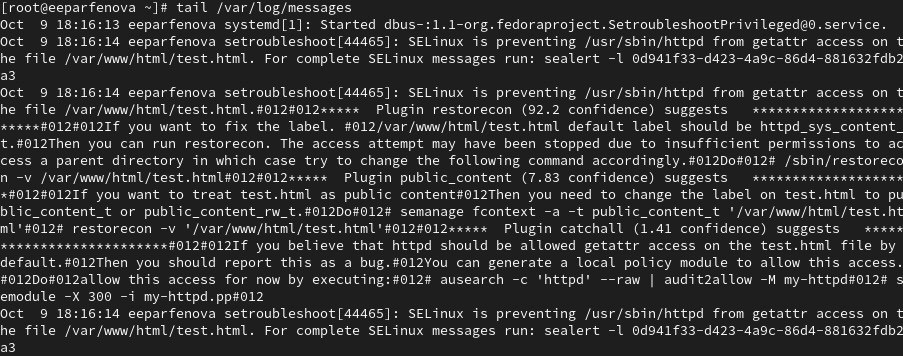


Рис. 13: Файл /var/log/messages\*

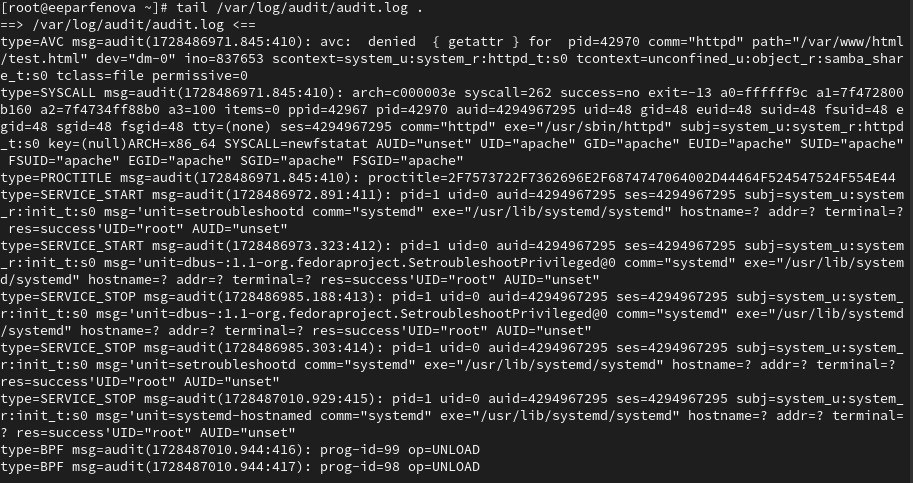


Рис. 14: Файл /var/log/audit/audit.log

Попробуем запустить веб-сервер Apache на прослушивание ТСР-порта 81 (а не 80). Для этого в файле /etc/httpd/httpd.conf найдем строчку *Listen 80* и заменим её на Listen 81 (рис. 15).

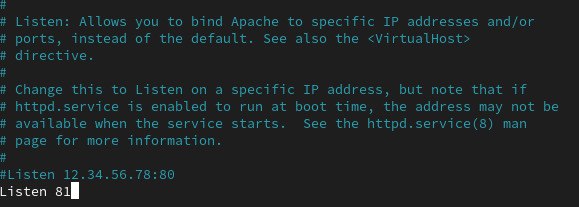


Рис. 15: Замена порта в config файле

Выполним перезапуск веб-сервера Apache. У нас произошел сбой, так как порт настроен на 80 порт, а мы пытаемся запустить его с 81 (рис. 16).

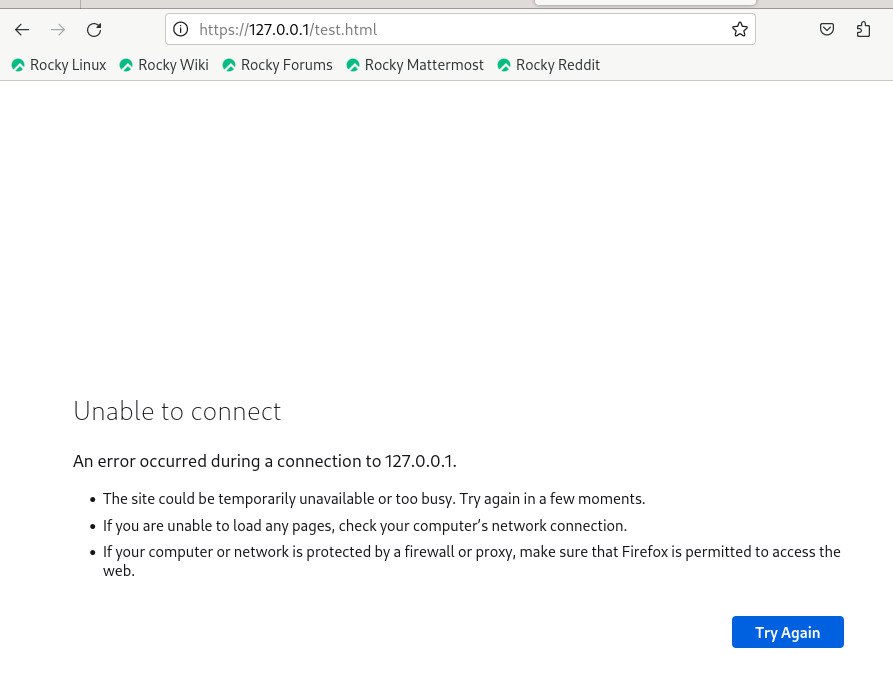


Рис. 16: Сбой сервера при запуске с 81 порта

Проанализируйте лог-файлы с помощью команд *tail -l /var/log/messages* (рис. 17), *tail -l /var/log/http/error\_log* и *tail -l /var/log/http/access\_log*..

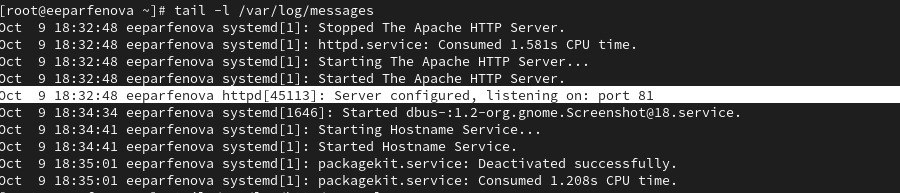


Рис. 17: Файл /var/log/messages

Видим, что есть сообщение, что файл сконфигурирован и прослушивается с 81 порта. Также у нас появилась новая запись в лог-файла ошибок о невозможности загрузки, и не появилось новой записи в лог-файле доступа.

Выполним команду *semanage port -a -t http\_port\_t -р tcp 81*, добавив 81 порт в список подключенных портов. После этого проверим список портов командой *semanage port -l | grep http\_port\_t* Убедимся, что порт 81 появился в списке (он у нас еще и повторяется дважды) (рис. 18).

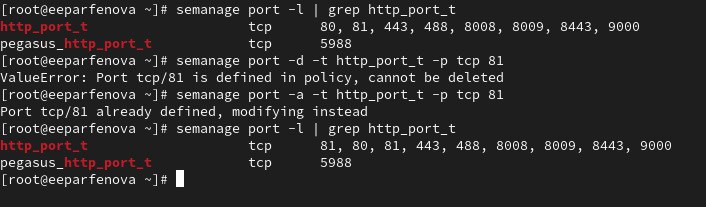


Рис. 18: Добавление 81 порта

Попробуем запустить веб-сервер Apache ещё раз, но и в этот раз у меня произошел сбой, так как указан не тот контекст. Вернем контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html командой *chcon -t httpd\_sys\_content\_t /var/www/html/test.html* (рис. 19). После этого попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html, и на этот раз мы действительно смогли прочитать файл через браузер, увидев содержимое файла — слово «test». (рис. 20)

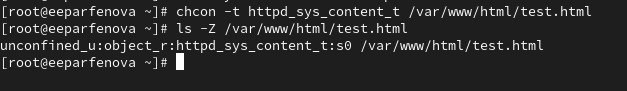


Рис. 19: Изменение контекста обратно

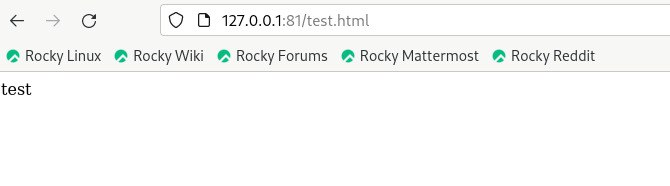


Рис. 20: Содержимое файла по новому адресу

В конце вернем все, что изменили на место: исправим обратно конфигурационный файл apache, вернув Listen 80; удалим привязку http\_port\_t к 81 порту командой *semanage port -d -t http\_port\_t -p tcp 81*; проверим, что порт 81 удалён; удалим файл /var/www/html/test.html командой *rm /var/www/html/test.html* (рис. 21)

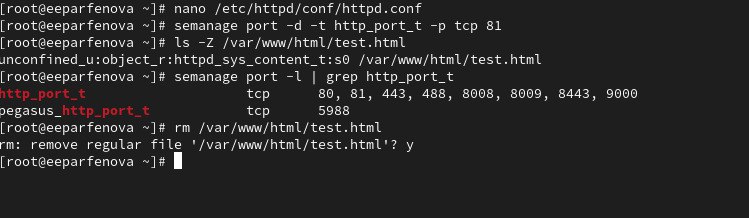


Рис. 21: Возвращение всех изменений

# 5 Выводы

Мы развили навыки администрирования ОС Linux и получили первое практическое знакомство с технологией SELinux. Для этого мы проверили работу SELinx на практике совместно с веб-сервером Apache.

# Список литературы

1. SELinux – описание и особенности работы с системой. Часть 1 [Электронный ресурс]. © 2006–2024, Habr, 2014. URL: <https://habr.com/ru/companies/kingservers/articles/209644/>.