Лабораторная работа №6

Информационная безопасность

Худицкий Василий Олегович"

Содержание

# Цель работы

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

# Задание

Выполнить задания лабораторной работы и проанализировать полученные результаты.

# Теоретическое введение

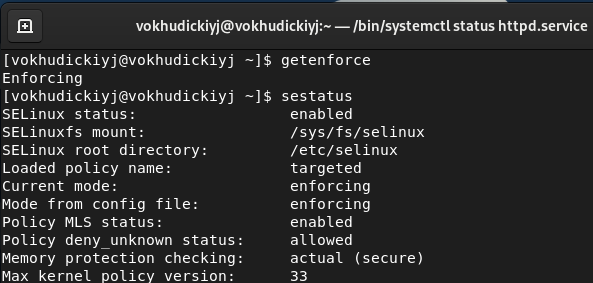
Мандатное управление доступом (англ. Mandatory access control, MAC) — разграничение доступа субъектов к объектам, основанное на назначении метки конфиденциальности для информации, содержащейся в объектах, и выдаче официальных разрешений (допуска) субъектам на обращение к информации такого уровня конфиденциальности. Также иногда переводится как Принудительный контроль доступа. Это способ, сочетающий защиту и ограничение прав, применяемый по отношению к компьютерным процессам, данным и системным устройствам и предназначенный для предотвращения их нежелательного использования.

Мандатная модель управления доступом, помимо дискреционной и ролевой, является основой реализации разграничительной политики доступа к ресурсам при защите информации ограниченного доступа. При этом данная модель доступа практически не используется «в чистом виде», обычно на практике она дополняется элементами других моделей доступа. Для файловых систем оно может расширять или заменять дискреционный контроль доступа и концепцию пользователей и групп. Самое важное достоинство заключается в том, что пользователь не может полностью управлять доступом к ресурсам, которые он создаёт.

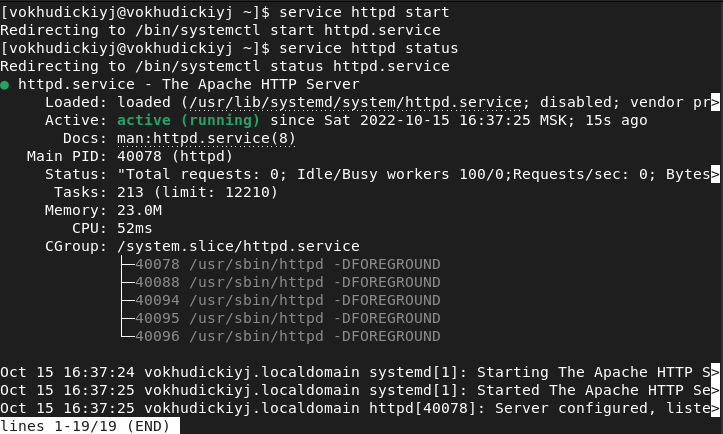
Политика безопасности системы, установленная администратором, полностью определяет доступ, и обычно пользователю не разрешается устанавливать более свободный доступ к своим ресурсам, чем тот, который установлен администратором пользователю. Системы с дискреционным контролем доступа разрешают пользователям полностью определять доступность своих ресурсов, что означает, что они могут случайно или преднамеренно передать доступ неавторизованным пользователям. Такая система запрещает пользователю или процессу, обладающему определённым уровнем доверия, получать доступ к информации, процессам или устройствам более защищённого уровня. Тем самым обеспечивается изоляция пользователей и процессов, как известных, так и не известных системе (неизвестная программа должна быть максимально лишена доверия, и её доступ к устройствам и файлам должен ограничиваться сильнее).

# Выполнение лабораторной работы

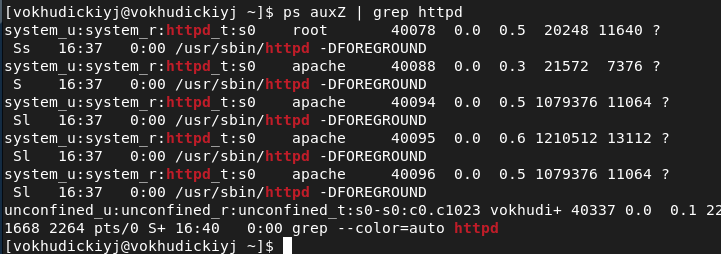
Вошёл в систему с полученными учётными данными и убедился, что SELinux работает в режиме enforcing политики targeted с помощью команд *getenforce* и *sestatus*:



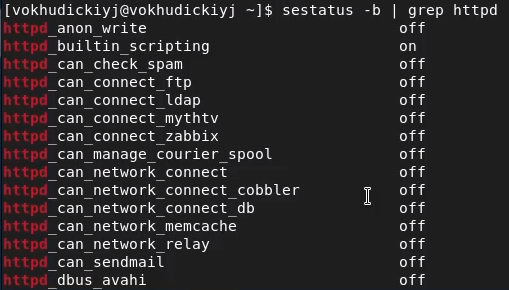
Обратился с помощью браузера к веб-серверу, запущенному на компьютере, и убедился, что последний работает командой *service httpd status*. Он не работал, поэтому пришлось запустить его так же, но с параметром командой *service httpd start*:



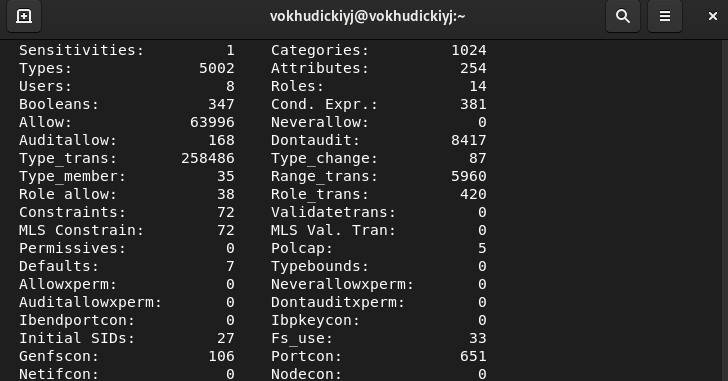
Нашёл веб-сервер Apache в списке процессов и определил его контекст безопасности, используя команду *ps auxZ | grep httpd*:



Посмотрел текущее состояние переключателей SELinux для Apache с помощью команды *sestatus -b | grep httpd*:



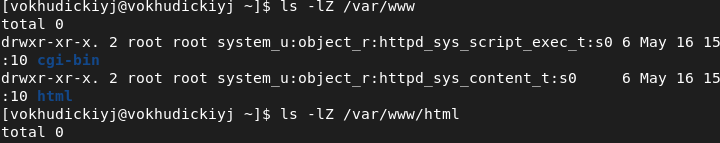
Посмотрел статистику по политике с помощью команды seinfo:



Определил тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды *ls -lZ /var/www*.

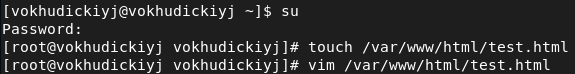
Определил тип файлов, находящихся в директории /var/www/html с помощью команды *ls -lZ /var/www/html*.

Определил круг пользователей, которым разрешено создание файлов в директории /var/www/html. Разрешено только владельцу.

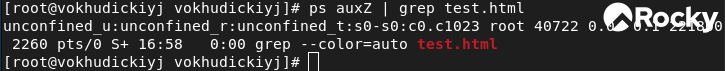


Создал от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл  
/var/www/html/test.html следующего содержания:

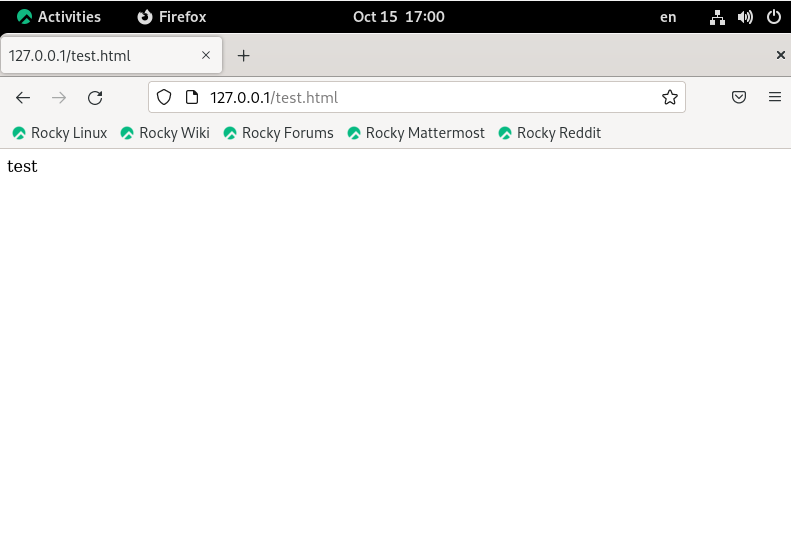
<html>
  
<body>test</body>
  
</html>



Проверил контекст созданного мной файла:



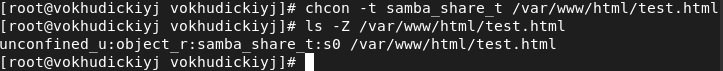
Обратился к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html>. Убедился, что файл был успешно отображён:

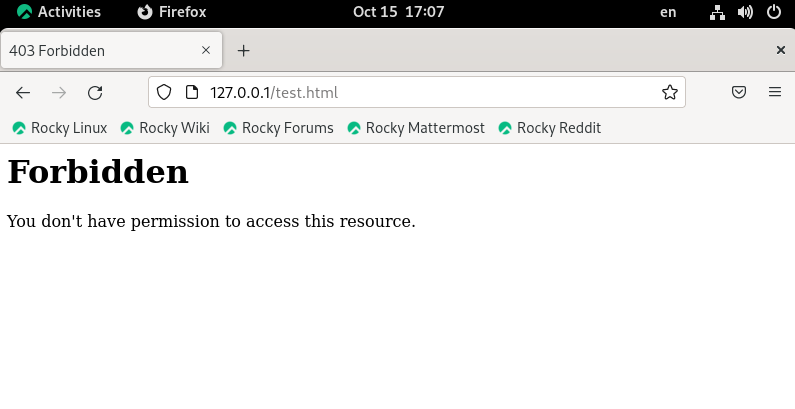


Изучил справку man httpd\_selinux и выяснил, какие контексты файлов определены для httpd. Сопоставил их с типом файла test.html. Проверил контекст файла командой *ls -Z /var/www/html/test.html*:

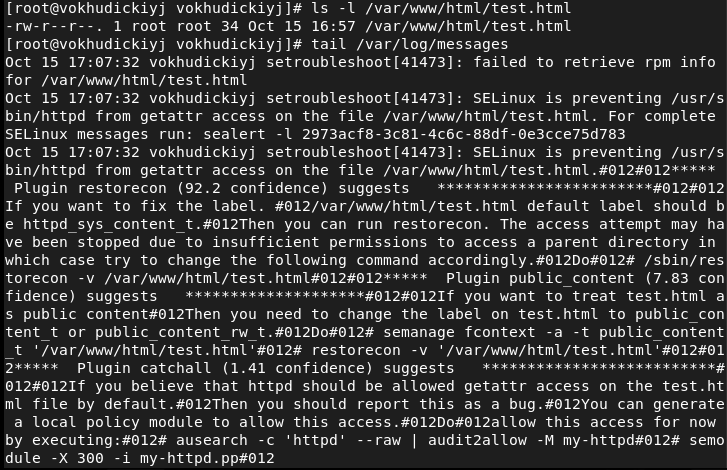


Изменил контекст файла /var/www/html/test.html с httpd\_sys\_content\_t на samba\_share\_t командами *chcon -t samba\_share\_t /var/www/html/test.html* и *ls -Z /var/www/html/test.html*. После этого проверил, что контекст поменялся:

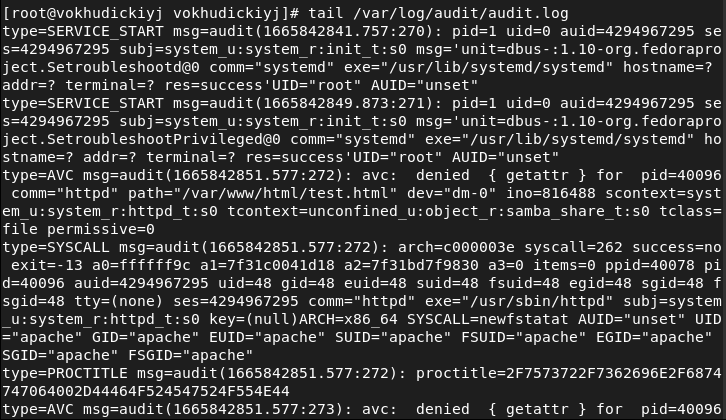


Попробовал ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html>. Получил сообщение об ошибке:  


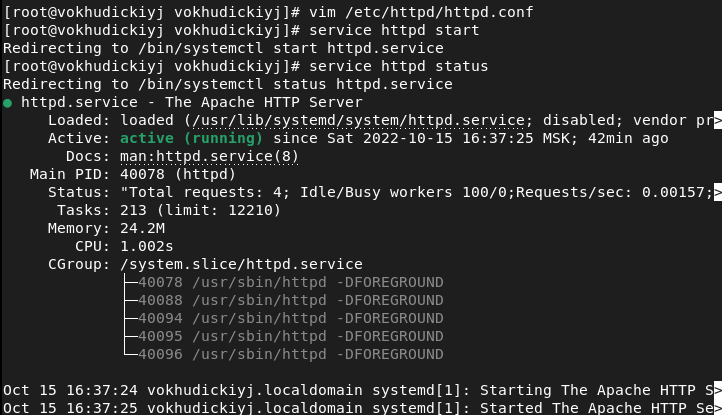
Проверил, действительно ли права доступа позволяют читать этот файл любому пользователю, с помощью команды *ls -l /var/www/html/test.html*. Просмотрел системный лог-файл командой *tail /var/log/messages*:



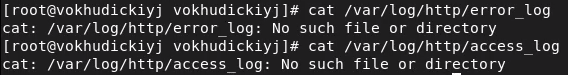
Увидел ошибки, аналогичные указанным выше, в файле /var/log/audit/audit.log:



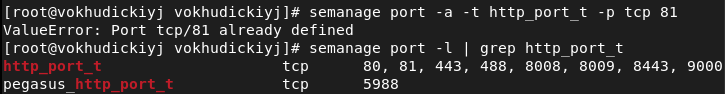
Попробовал запустить веб-сервер Apache на прослушивание ТСР-порта 81 (а не 80, как рекомендует IANA и прописано в /etc/services). Для этого в файле /etc/httpd/httpd.conf нашёл строчку Listen 80 и заменил её на Listen 81. Выполнил перезапуск веб-сервера Apache. Сбоя не произошло.



Проанализировал лог-файлы /var/log/http/error\_log, /var/log/http/access\_log и /var/log/audit/audit.log. Так как сбоя не было, ни в каких файлах не появились записи:



Выполнил команду *semanage port -a -t http\_port\_t -р tcp 81*. Порт 81 уже был определён. После этого проверил список портов командой *semanage port -l | grep http\_port\_t*. Убедился, что порт 81 в списке:



Вернул контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html командой *chcon -t httpd\_sys\_content\_t /var/www/html/test.html*.  
После этого попробовал получить доступ к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1:81/test.html>. Я увидел содержимое файла — слово «test».



Исправил обратно конфигурационный файл apache, вернув Listen 80. Попытался удалить привязку http\_port\_t к 81 порту командой *semanage port -d -t http\_port\_t -p tcp 81*. Совершить удаление не удалось:



Удалил файл /var/www/html/test.html командой *rm /var/www/html/test.html*:



# Выводы

В ходе выполнения работы я развил навыки администрирования ОС Linux, получил первое практическое знакомство с технологией SELinux, проверил работу SELinx на практике совместно с веб-сервером Apache.

# Список литературы

* [Кулябов Д. С., Королькова А. В., Геворкян М. Н Лабораторная работа №6. Мандатное разграничение прав в Linux](https://esystem.rudn.ru/pluginfile.php/1651755/mod_resource/content/2/006-lab_selinux.pdf)