Отчет по лабораторной работе №6

Мандатное разграничение прав в Linux

Бурдина Ксения Павловна

2022 Oct 11th

Содержание

# 1 Цель работы

Целью данной работы является развитие навыков администрирования ОС Linux, получение практического знакомства с технологией SELinux, а также проверка работы SELinux на практике совместно с веб-сервером Apache.

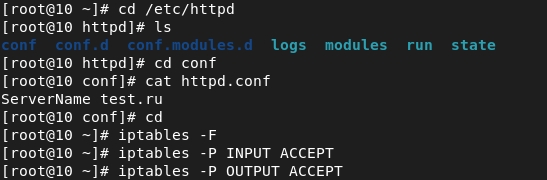
# 2 Теоретическое введение

При подготовке стенда нам необходима для работы политика targeted и режим enforcing, которые используются в данном дистрибутиве по умолчанию, т.е. каких-то специальных настроек не требуется. При этом пользователю следует убедиться, что политика и режим включены, особенно когда работа будет проводиться повторно и велика вероятность изменений при предыдущем использовании системы.

При необходимости администратор должен разбираться в работе SELinux и уметь как исправить конфигурационный файл /etc/selinux/config, так и проверить используемый режим и политику. Нам необходимо установить веб-сервер Apache. Причем следует учитывать, что при установке системы в конфигурации «рабочая станция» указанный пакет не ставится.

Перед началом работы в конфигурационном файле /etc/httpd/httpd.conf необходимо задать параметр ServerName, чтобы при запуске веб-сервера не выдавались лишние сообщения об ошибках, не относящихся к лабораторной работе. Также необходимо проследить, чтобы пакетный фильтр был отключён или в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола tcp [[1]](https://esystem.rudn.ru/pluginfile.php/1651889/mod_resource/content/2/005-lab_discret_sticky.pdf).

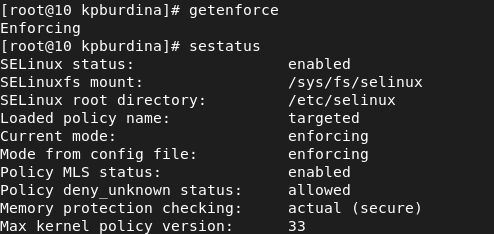
Выполним все действия для подготовки к работе. Установим httpd для работы, после чего проверим наличие необходимых файлов в каталоге и настроим фильтры. Контрольные команды для дальнейшей работы:



Подготовка к работе

# 3 Ход выполнения лабораторной работы

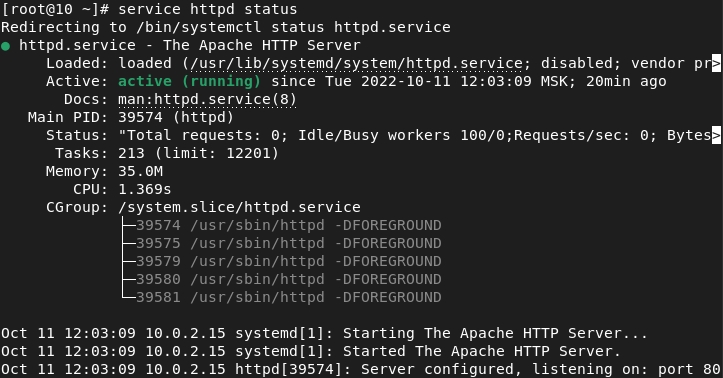
1. Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted:



Вызов команд getenforce и sestatus

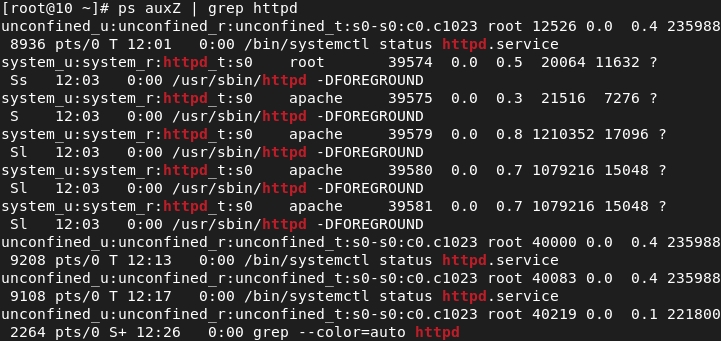
Видим, что у нас действительно все работает верно.

1. Обратимся с помощью браузера к веб-серверу, запущенному на нашем компьютере, и убедимся, что последний работает:



Обращение к веб-серверу

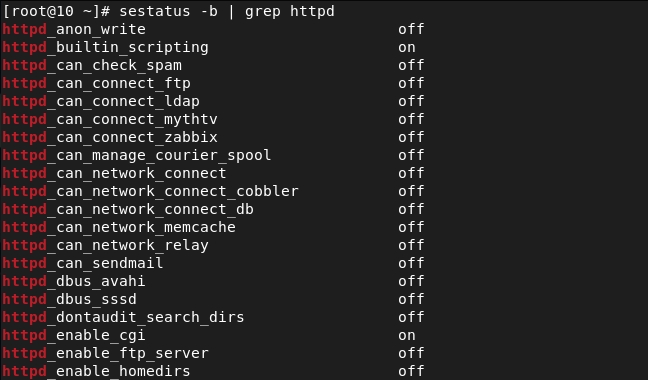
1. Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности:



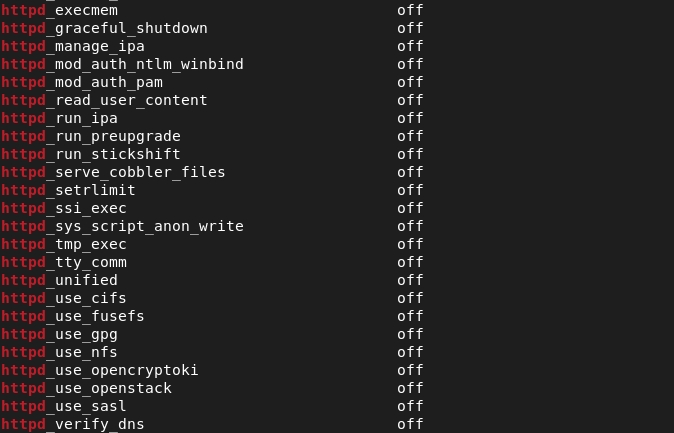
Нахождение Apache в списке процессоров

Контекстом безопасности будет system\_u:system\_r.

1. Посмотрим текущее состояние переключателей SELinux для Apache:



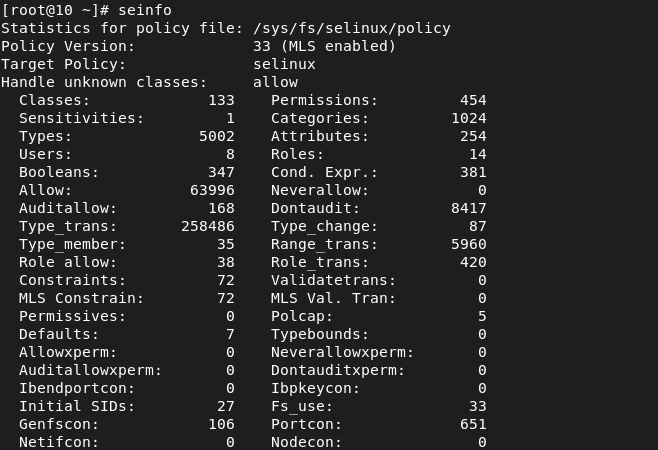
Состояние переключателей SELinux



Состояние переключателей SELinux

Обратим внимание на то, что многие из них находятся в положении “off”.

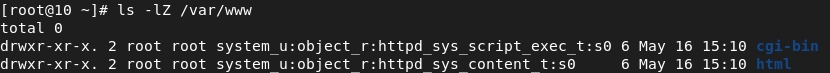
1. Посмотрим статистику по политике с помощью команды seinfo и определим множество пользователей, ролей и типов:



Статистика по политике

Видим, что у нас имеется 8 пользователей, 5002 типов и 14 ролей.

1. Определим тип файлов и поддиректорий, находящихся в директории /var/www:



Просмотр директории /var/www

Можем увидеть контекст файлов: system\_u, object\_r, httpd\_sys\_content\_t/httpd\_sys\_script\_exec\_t.

1. Определим тип файлов, находящихся в директории /var/www/html:

Просмотр директории /var/www/html

Просмотр директории /var/www/html

Видим, что у нас в данной директории отсутствуют какие-либо файлы.

1. Определяя круг пользователей, которым разрешено создание файлов в данной директории, можно сказать, что создание файлов разрешено всем пользователям.
2. Создадим от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html и запишем в него содержание:

Создание файла test.html

Создание файла test.html

Проверка создания файла

Проверка создания файла

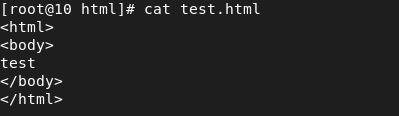
Открытие файла на редактирование

Открытие файла на редактирование



Запись содержимого в файл

1. Проверим контекст созданного нами файла:



Чтение содержимого файла

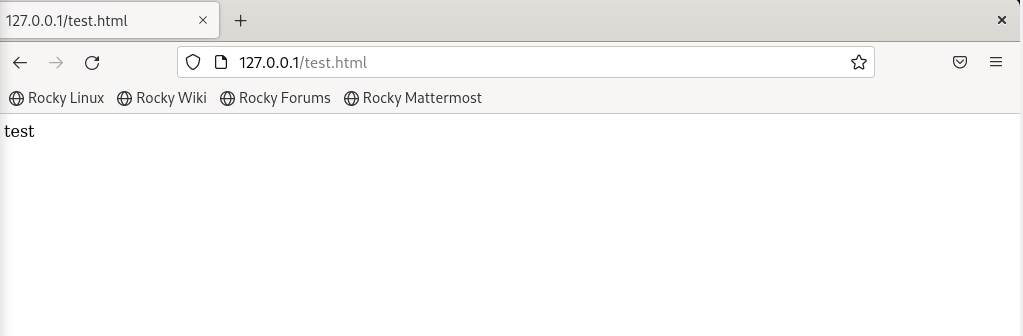
Контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html, выглядит следующим образом:

Контекст созаваемых файлов

Контекст созаваемых файлов

Всем создаваемым файлам присваивается контекст: unconfined\_u:object\_r:httpd\_sys\_content\_t:s0.

1. Обратимся к файлу через веб-браузер, введя в браузере адрес http://127.0.0.1/test.html:



Открытие файла через браузер

Видим, что наш файл был успешно передан на сервер, поскольку на открывшейся странице отобразился текст нашего файла.

1. Изучим справку по httpd\_selinux:

Открытие справки по httpd_selinux

Открытие справки по httpd\_selinux

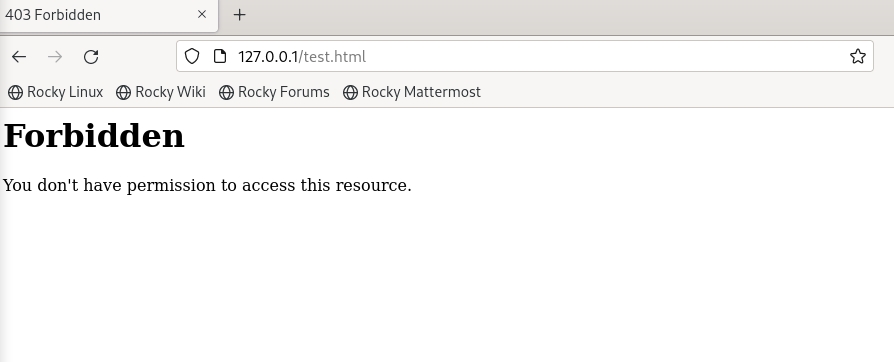
К сожалению, нам невозможно изучить справку, однако посмотреть, какие контексты файлов определены для httpd, мы можем, и они аналогичны тем, что отобразились при просмотре файла test.html.

1. 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, после чего проверим правильность изменения контекста:

Изменение контекста файла

Изменение контекста файла

1. Попробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. На этот раз мы получим следующее сообщение об ошибке:



Сообщение об ошибке доступа

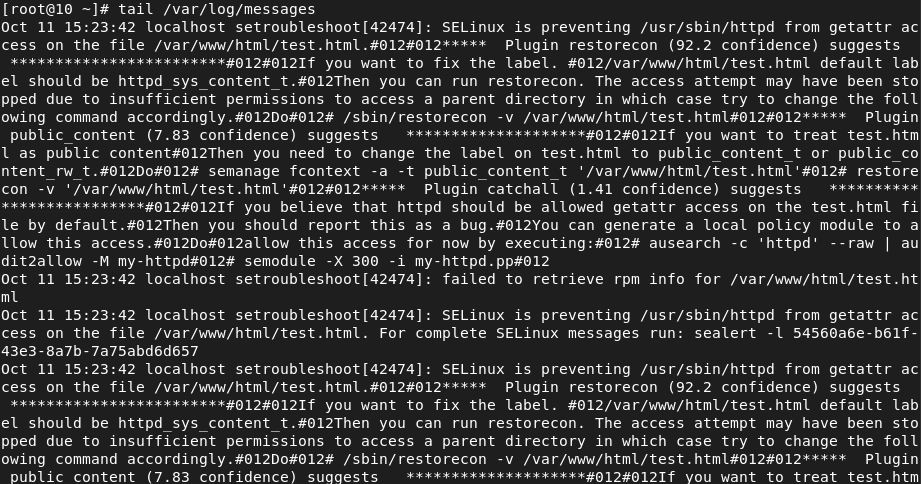
1. Проанализируем ситуацию. Наш файл не был открыт, поскольку на сервер загружены данные по контексту те, которые назначаются всем файлам по умолчанию. Поэтому при попытке перейти на сайт с измененными данными, мы получим отказ в выполнении действия.

Но несмотря на это, права доступа позволяют читать файл любому пользователю:

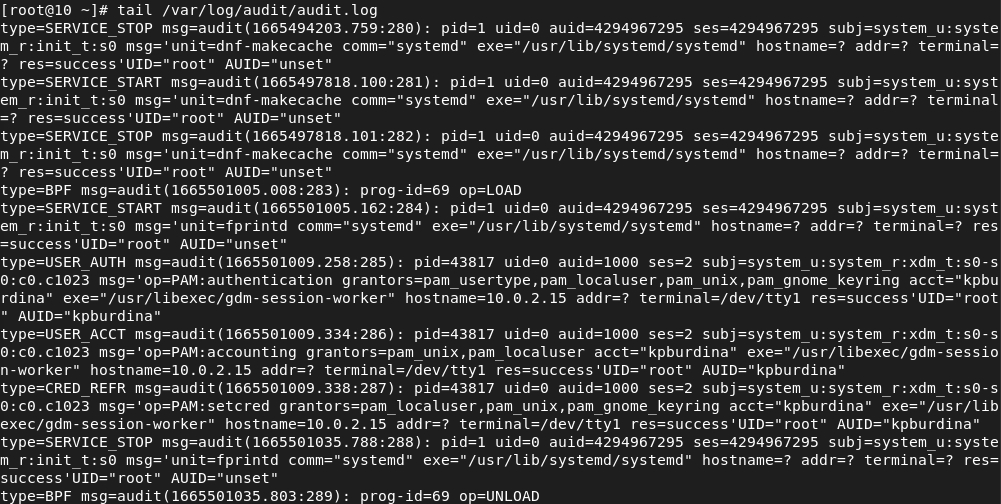
Проверка прав на чтение файла

Проверка прав на чтение файла

Теперь посмотрим log-файлы веб-сервера Apache, после чего посмотрим системный log-файл:



Просмотр log-файла Apache



Просмотр системного log-файла

Видим, что в системе есть запущенные процессы setroubleshootd и audtd, поэтому можно увидеть ошибки, аналогичные указанным выше в файле.

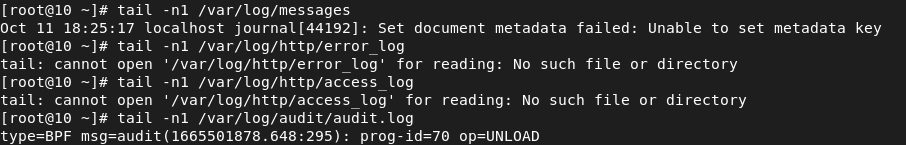
1. Теперь попробуем запустить веб-сервер Apache на прослушивание ТСР-порта. Откроем файл на просмотр:

Открытие файла на редактирование

Открытие файла на редактирование

Однако не сможем изменить данные, поскольку информация из файла нам не была доступна.

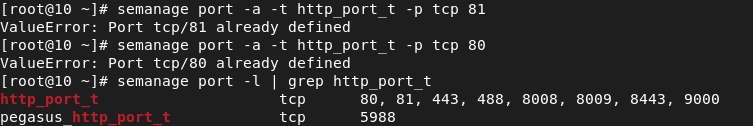
1. Если после смены строки с прослушиванием выполнить перезапуск сервера, то он не будет работать, поскольку сервер настроен на обределенную частоту прослушки и не сможет быть подключен с другими параметрами.
2. Проанализируем log-файлы:



Анализ log-файла messages

После чего посмотрим еще некоторые файлы, однако в силу невозможности работы при текущих критериях, можем понять, что записи не появились и в одном из файлов.

1. Выполним конанду “semanage port” для проверки списка команд, вводимых портом:



Команда semanage port

Видим, что порт 81 присутстсвует в нашем списке.

1. При попытке запустить веб-сервер Apache ещё раз, все стало работать. То есть на данный момент все добавлено и загружено для возможности пользоваться доступом к файлу.
2. Вернем контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html:

Возвращение контекста

Возвращение контекста

При попытке получить доступ к исходному файлу, мы снова видим в окне слово test.

1. Вернем обратно конфигурационный файл Apache, исправив Listen 80.
2. Удалим привязку http\_port\_t к 81 порту:

Удаление привязки к порту 81

Удаление привязки к порту 81

1. Удалим файл /var/www/html/test.html для завершения полной работы:



Удаление файла test.html и проверка содержимого каталога

# 4 Выводы

В ходе работы мы развили навыки администрирования ОС Linux; получили практическое знакомство с технологией SELinux; проверили работу SELinux на практике совместно с веб-сервером Apache.

# 5 Список литературы

1. Методические материалы курса [[1]](https://esystem.rudn.ru/pluginfile.php/1651891/mod_resource/content/2/006-lab_selinux.pdf)