Отчет по лабораторной работе №6

Мандатное разграничение прав в Linux

Татьяна Александровна Буллер

Содержание

Список иллюстраций

1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux1. Проверить работу SELinx на практике совместно с веб-сервером Apache.

2 Выполнение лабораторной работы

Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus.

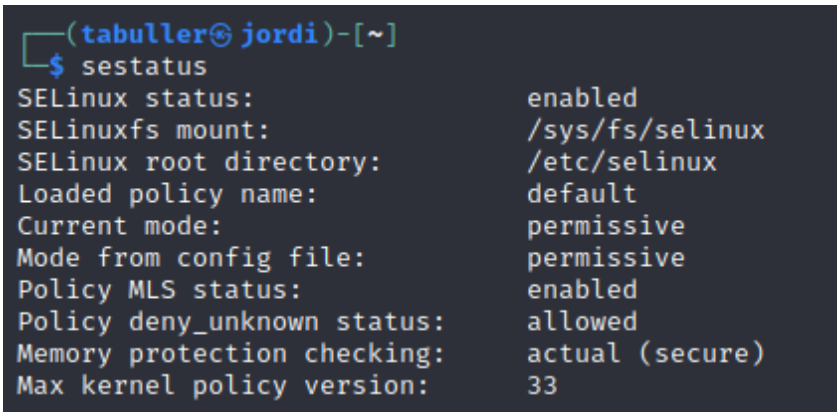

Рис. 2.1: sestatus

Предварительно запустив сервис apache2, проверим, работает ли httpd.

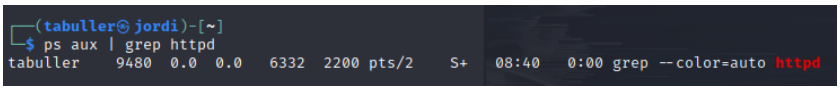

Рис. 2.2: httpd

Далее с помощью команды ‘ps auxZ | grep httpd’ определим контекст безопасности веб-сервера Apache. Строка unconfined говорит о том, что никакие дополнительные ограничения не установлены.

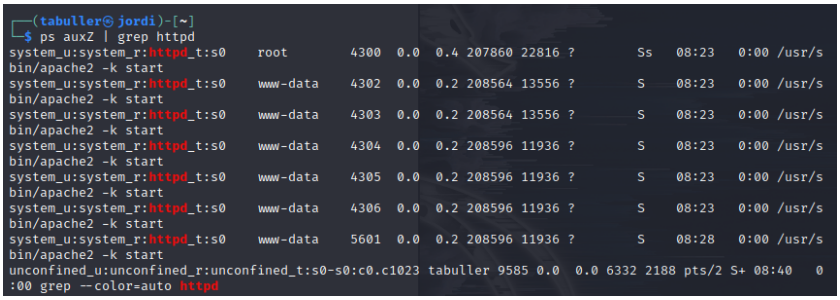

Рис. 2.3: Контекст безопасности веб-сервера Apache

Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды sestatus '-b | grep httpd'. Видим настройки по умолчанию: все переключатели разрешений для httpd в режиме off.

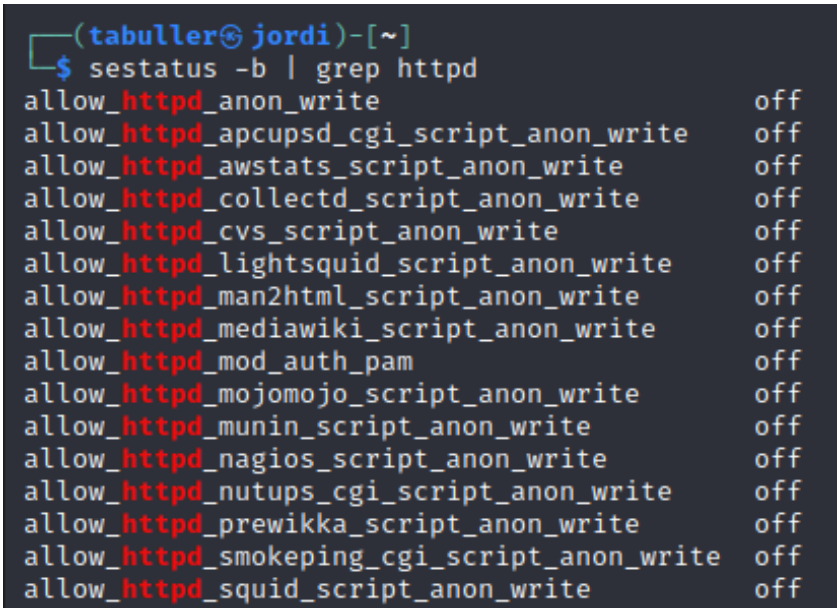

Рис. 2.4: sestatus

Команда seinfo позволяет просмотреть статистику по политике. Видим 3983 типа, 9 пользователей и 15 ролей.

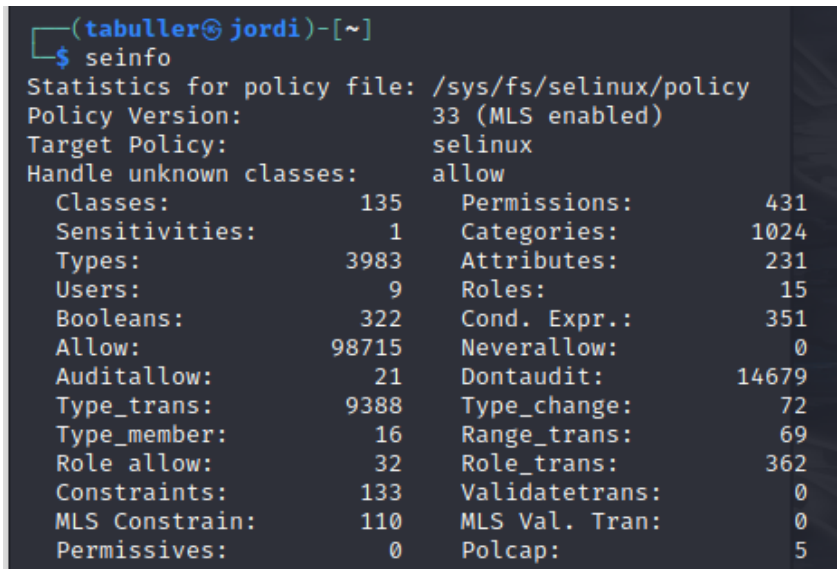

Рис. 2.5: seinfo

Командой ls -lZ определим типы файлов, находящихся в директории /var/www и то же самое для /var/www/html. Видим, что для всех файлов и поддиректорий установлены права, позволяющие только владельцу (суперпользователю) осуществлять запись в них, потому дальнейшую работу необходимо будет вести через sudo.

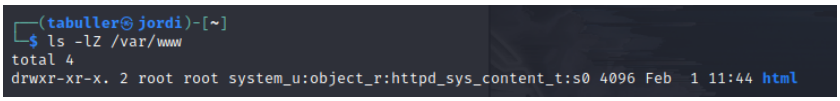

Рис. 2.6: Типы файлов, находящихся в директории /var/www

От имени суперпользователя пишем в директории /var/www/html короткий файл test.html, при компиляции которого на экран будет выведено только слово test. Этот файл должен быть доступен по адресу 127.0.0.1/test.html, так как на машине запущен сервис apache. Перейдя по адресу и не указывая имя файла мы встретим приветственную страницу apache (it works!), далее, уточнив имя файла test.html, увидим вывод созданного нами файла.

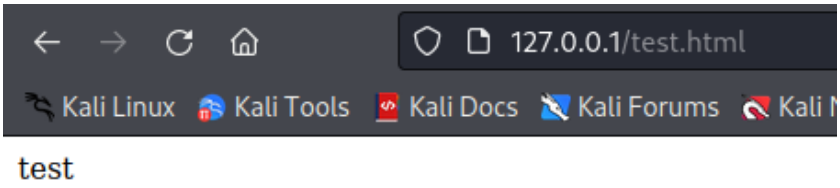

Рис. 2.7: test.html

В контексте созданного файла, который проверим снова командой ls -lZ, встречаем строку unconfined_u - свободный пользователь, роль object_r используемая по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах, и тип httpd_sys_content_t, позволяющий процессу httpd получить доступ к этому файлу.

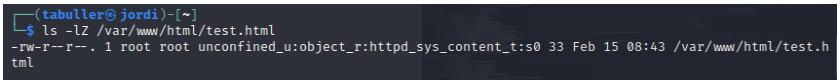

Рис. 2.8: Контекст test.html

Теперь командой chcon -t samba_share_t /var/www/html/test.html изменим контекст файла. Теперь процесс samba может получить доступ к файлу, а httpd - нет, поэтому при повторной попытке получить файл через браузер мы столкнемся с ошибкой. Соответствующее предупреждение можем видеть и в системных логах.

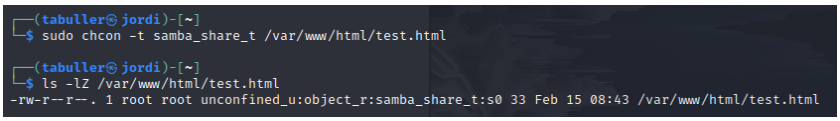

Рис. 2.9: Изменение контекста файла

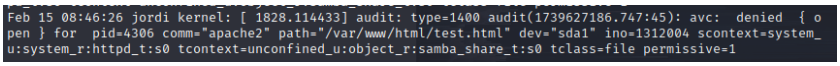

Рис. 2.10: Сообщение об ошибке из системного лога

Перезапустим веб-сервер на прослушивание TCP порта 81 вместо 80, стоящего по умолчанию. Для этого в файле /etc/apache/port.conf заменим строчку Listen

80 на Listen 81. При попытке перезапустить сервер после внесенных изменений, однако, снова столкнемся с ошибкой, потому что в конфигурацию SELinux соответствующие разрешения и изменения внесены не были.

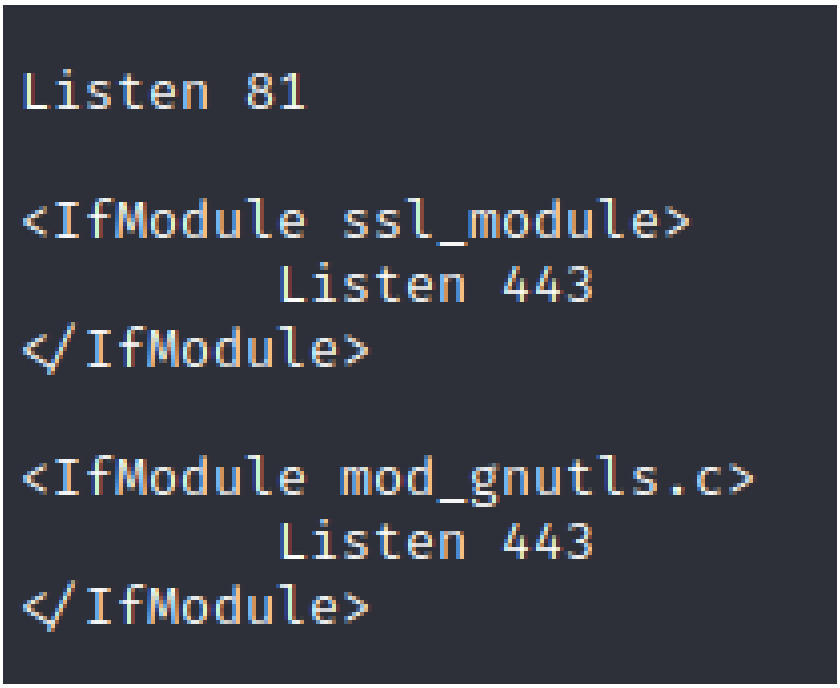

Рис. 2.11: Перевод сервера на прослушивание другого порта

Следующим шагом выполним semanage port -a -t http_port_t -p tcp 81. Эта команда внесет порт 81 в список доступных для сервиса httpd, в чем далее мы можем убедиться командой semanage port -l | grep 81. При попытке теперь запустить веб-сервер ошибок не возникнет.

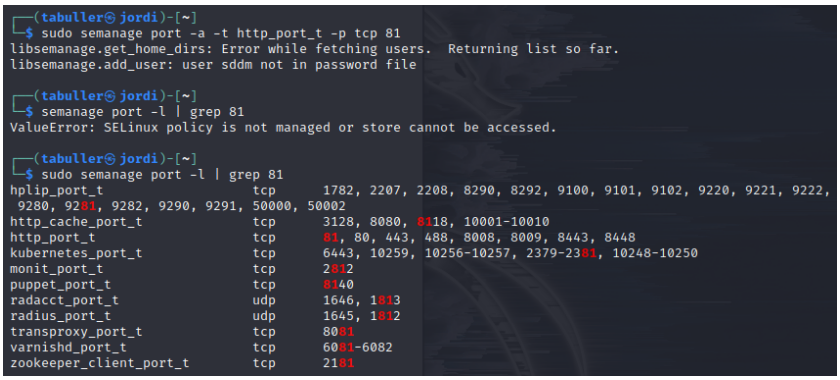

Рис. 2.12: Добавленный в конфигурацию порт

В завершение работы вернем систему к настройкам по умолчанию: удалим созданный файл и добавленные контексты, а также выключим сервис apache.

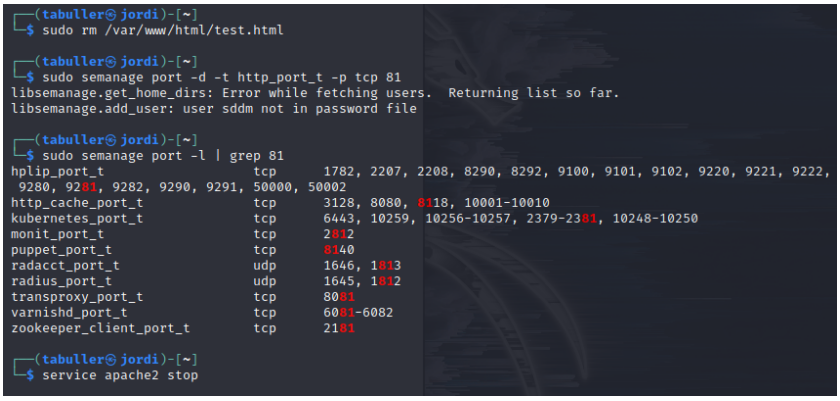

Рис. 2.13: Завершение работы

3 Выводы

Получено первое практическое знакомство с технологией SELinux. Проверена работу SELinx на практике совместно с веб-сервером Apache.