Основы информационной безопасности

Лабораторная работа № 6. Мандатное разграничение прав в Linux

Сунгурова М.

Содержание

# 1 Цель работы

Целью данной работы является приобретение практических навыков администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinx на практике совместно с веб-сервером Apache.

# 2 Теоретические сведения

SELinux (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.

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.

# 3 Выполнение лабораторной работы

В конфигурационном файле /etc/httpd/httpd.conf зададим параметр ServerName. Также необходимо проследить, чтобы пакетный фильтр был отключён или в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола tcp. Отключим фильтр командами(рис. fig. 1)

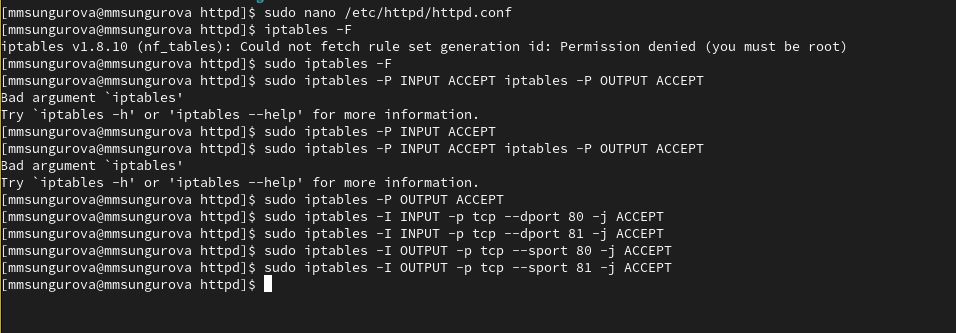


Рис. 1: Подготовка лабораторного стенда

Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus(рис. fig. 2).

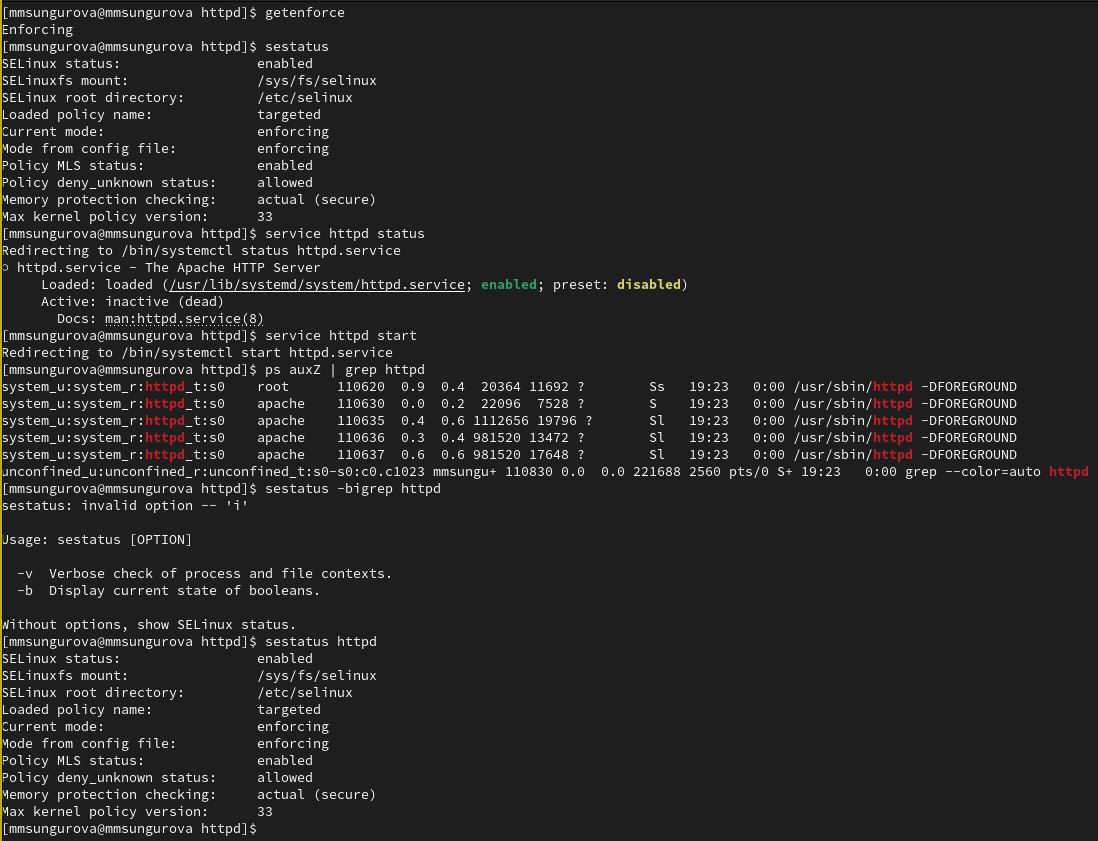


Рис. 2: Проверка статуса SELinux

Обратимся с помощью браузера к веб-серверу, запущенному на нашем компьютере, и убедитесь, что последний работает(рис. fig. 2).

Найдите веб-сервер Apache в списке процессов, определим его контекст безопасности(рис. fig. 2)

Мы можем видеть контекст безопасности SELinux: system\_u:system\_r:httpd\_t.

Также просмотрим множество пользователей, ролей, типов(рис. fig. 3):



Рис. 3: Множества пользователей, ролей, типов

Определив тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды ls -lZ /var/www, увидим, что есть директория, содержащая cgi-скрипты, и директория /var/www/html, содержащая все скрипты httpd(в данный момент пустая)(рис. fig. 4):



Рис. 4: Просмотр типов директорий в /var/www

Можно увидеть, что создание файлов в директории /var/www/html разрешено только владельцу – root.

Создадим от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания(рис. fig. 5):

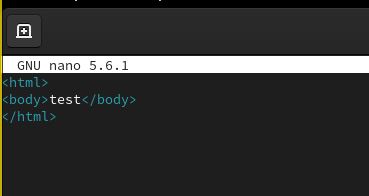


Рис. 5: Содержимое html-файла /var/www/html/test.html

Затем посмотрим контекст безопасности, который был задан по умолчанию этому файлу(fig. 6):

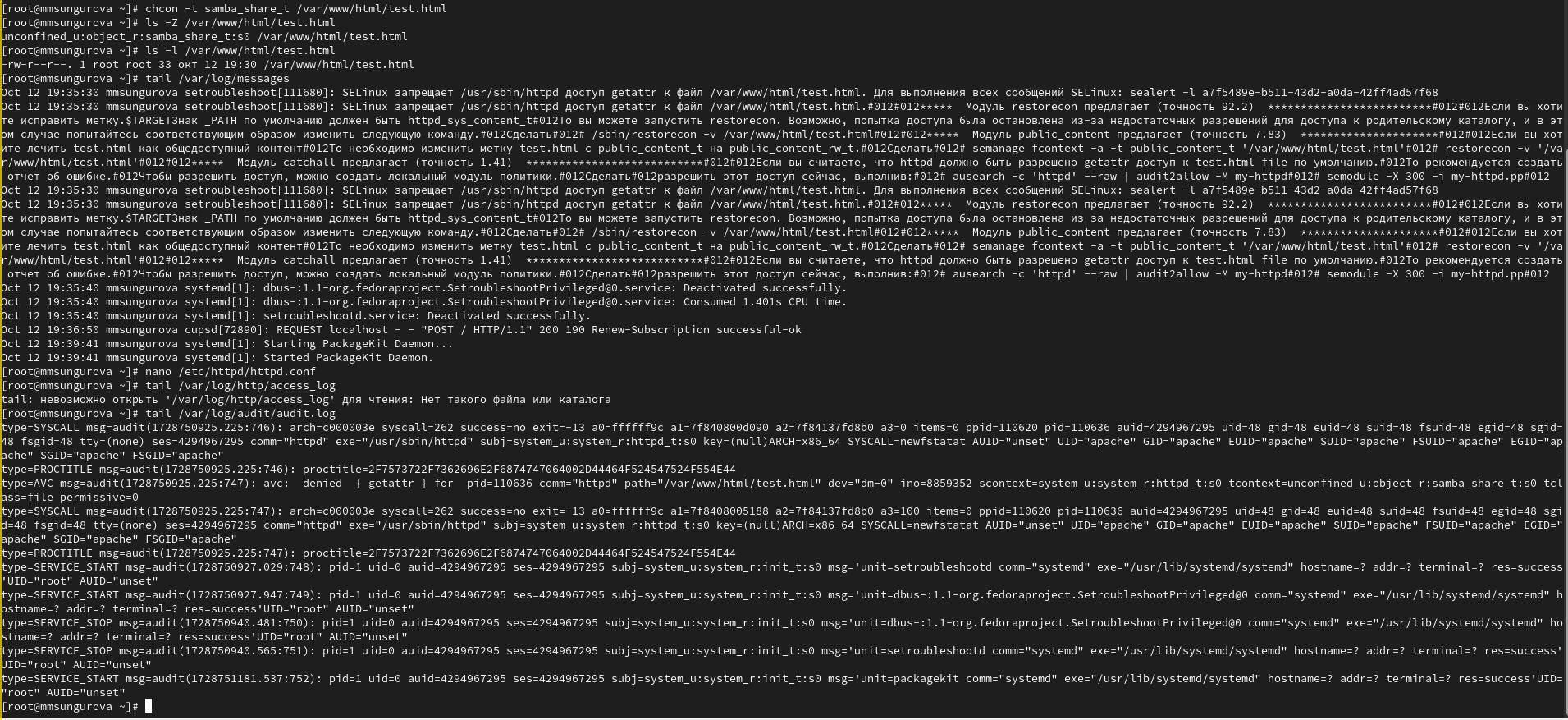


Рис. 6: Установка пароля для пользователя с правами администратора

Увидим, что файлам по умолчанию сопоставляется свободный пользователь SELinux unconfined\_u, указана роль object\_r используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах и тип httpd\_sys\_content\_t, который позволяет процессу httpd получить доступ к файлу

Обратимся к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html, убедимся, что файл был успешно отображён.(рис. fig. 7):

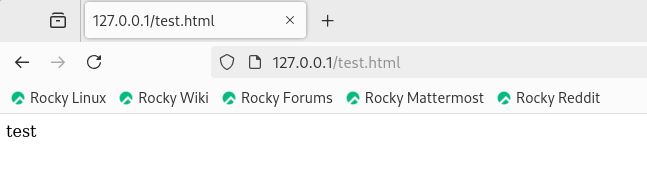


Рис. 7: Открытие html-страницы через браузер

Изучив справку man httpd\_selinux, выясним, какие контексты файлов определены для httpd. Сопоставив их с типом файла test.html увидим, что его контекст httpd\_sys\_content\_t для содержимого, которое должно быть доступно для всех скриптов httpd и для самого демона.

Изменим контекст файла /var/www/html/test.html с httpd\_sys\_content\_t на тот, к которому процесс httpd не должен иметь доступа – samba\_share\_t(рис. fig. 8):

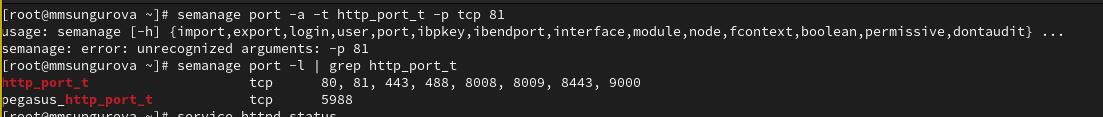


Рис. 8: Изменение контекста файла /var/www/html/test.html

Теперь снова попробуем получить доступ к файлу через браузер и получим отказ(рис. fig. 9):

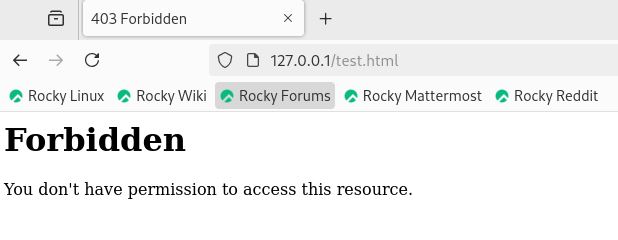


Рис. 9: Отказ в доступе к html-странице через браузер

Посмотрим log-файлы веб-сервера Apache и системный лог-файл и увидим, что отказ происходит, так как доступ запрещен SELinux именно к веб-серверу(на просто просмтр текстовых файлов это не влияет)(рис. fig. 10):

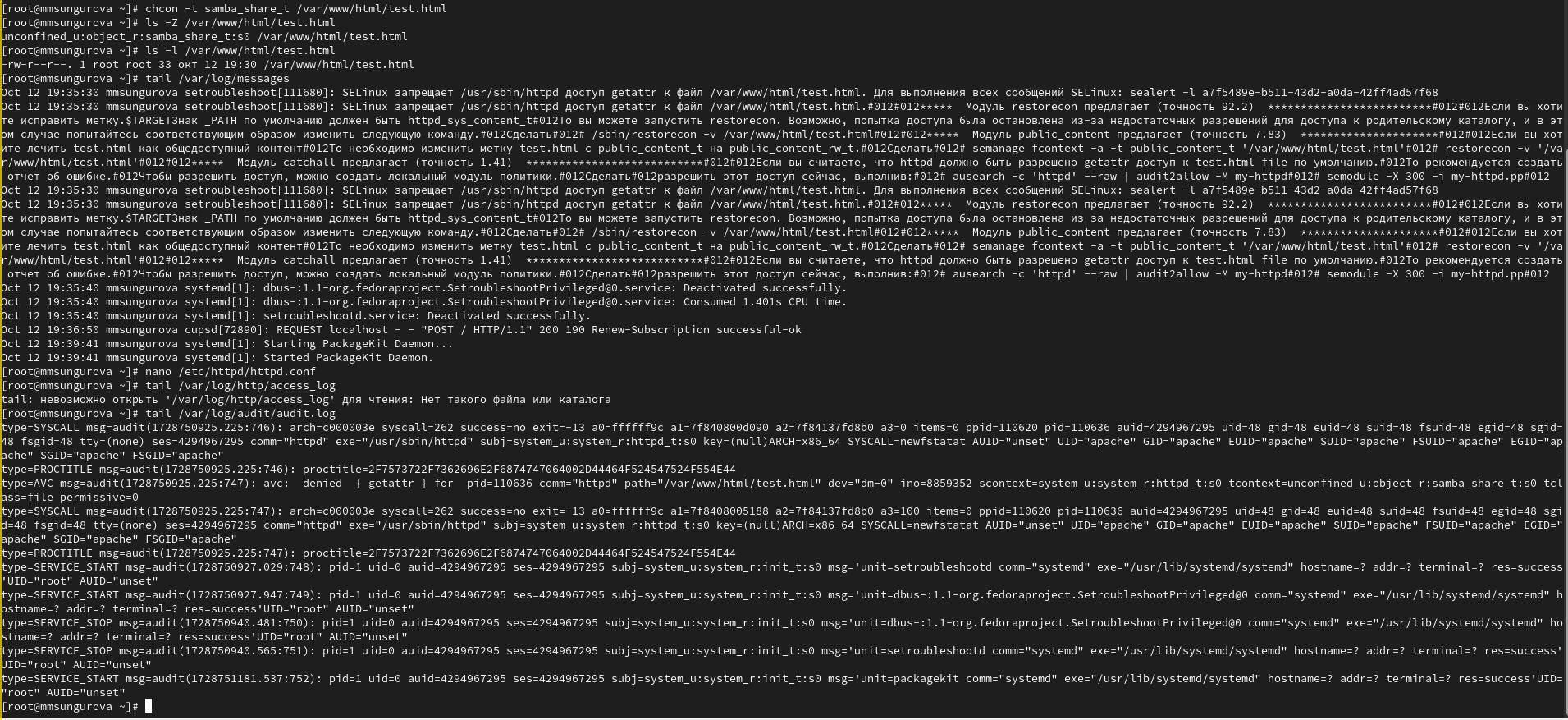


Рис. 10: Просмотр лог-файлов

Запустим веб-сервер Apache на прослушивание ТСР-порта 81. Для этого в файле /etc/httpd/httpd.conf найдем строчку Listen 80 и заменим её на Listen 81(рис. fig. 11):

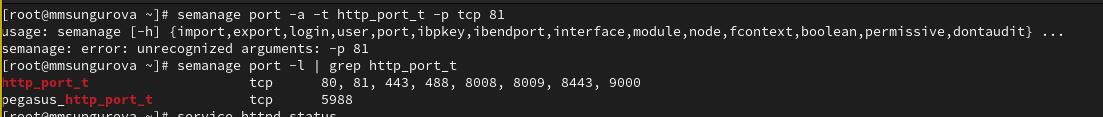


Рис. 11: Замена прослушиваемого порта

Просмотрев лог-файлы увидим, что порт для прослушивания был сменен(рис. fig. 12):

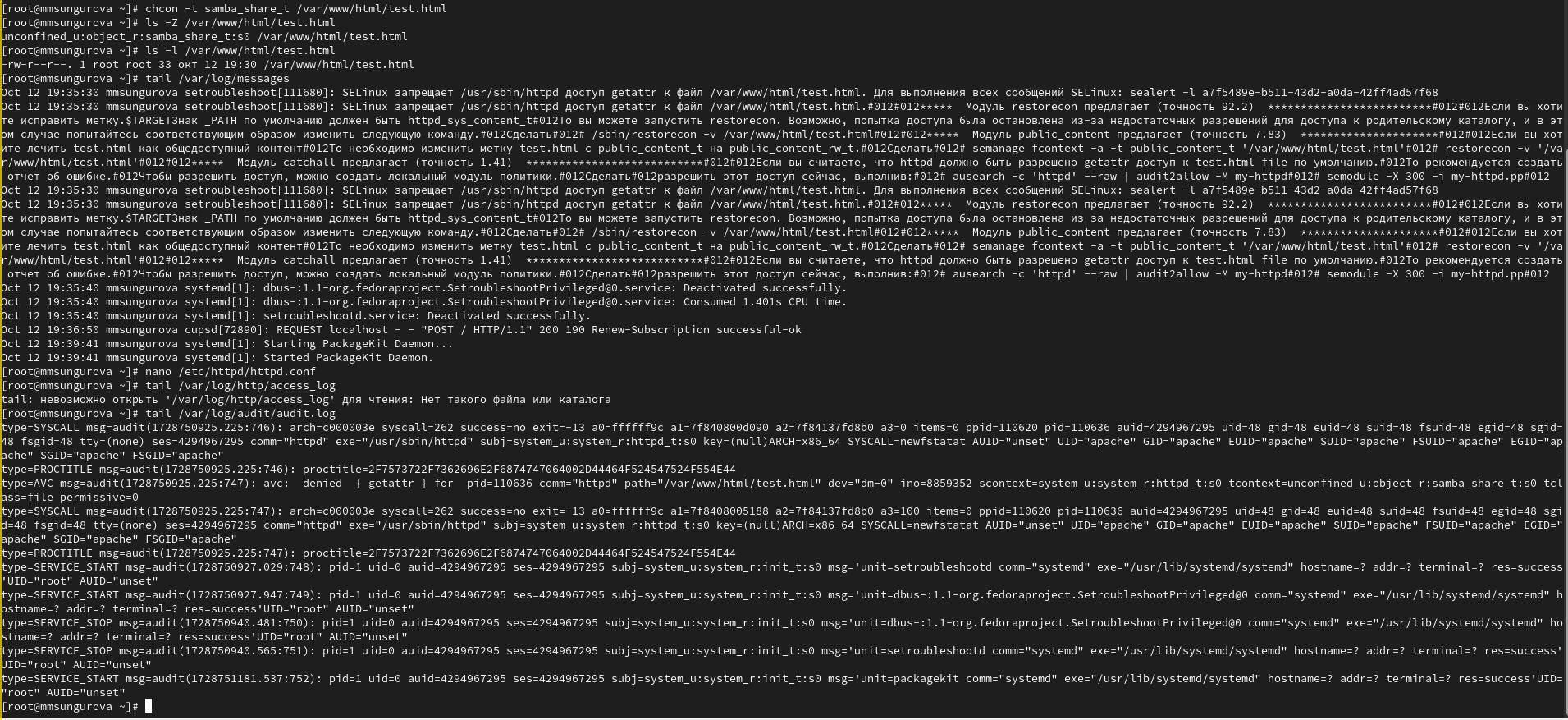


Рис. 12: Просмотр лог-файлов

Также этот порт мог быть отклчен, тогда мы бы совсем не видели страницу, добавлять порты и просматривать актуальные можно с помощью команды seamanage(рис. fig. **¿fig:018?**):

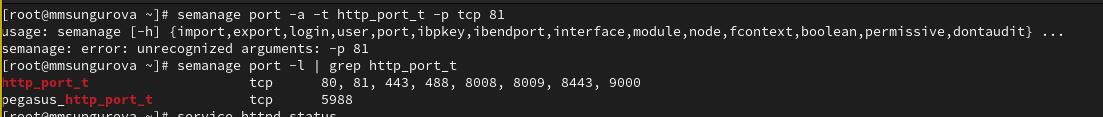


Рис. 13: Просмотр портов с помощью seamnage

В конце работы вернем все сделанные изменения в файлах конфигурации веб-сервера.

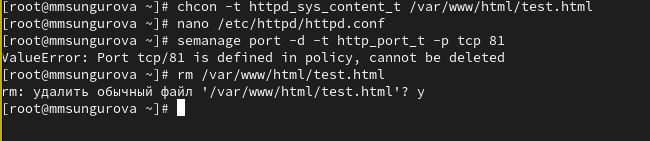


Рис. 14: Окончание работы

# 4 Выводы

В результате выполнения работы были приобретены практические навыки администрирования ОС Linux. Получено первое практическое знакомство с технологией SELinux. Проверена работа SELinx на практике совместно с веб-сервером Apache.

# Список литературы