Лабораторная работа №6

Информационная безопасность

Николаев Дмитрий Иванович

Содержание

# 1 Цель работы

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

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

# 2 Теоретическое введение

**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.

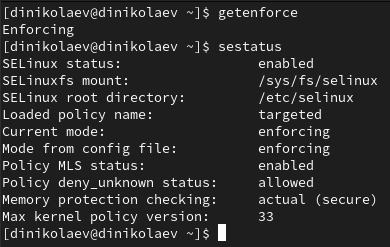
**Тип** — набор действий, которые допустимы по отношения к объекту. Тип отличается от домена тем, что он может применяться к пайпам, каталогам и файлам, в то время как домен применяется к процессам.

*Контекст безопасности* — все атрибуты SELinux — роли, типы и домены [1].

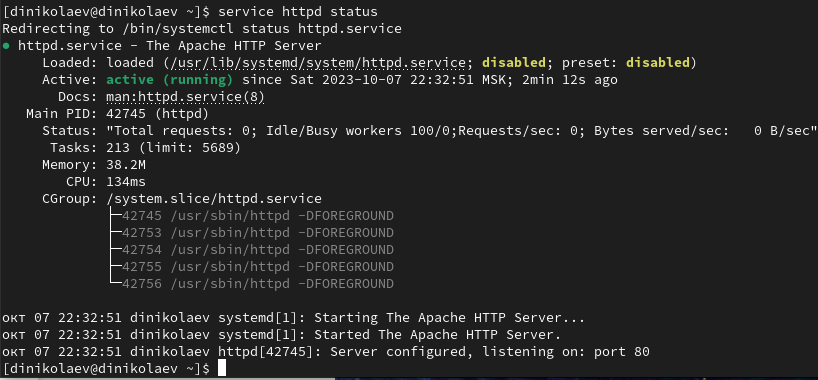
# 3 Выполнение лабораторной работы

Установим веб-сервер Apache с помощью команды yum install httpd, после чего следуем согласно [2].

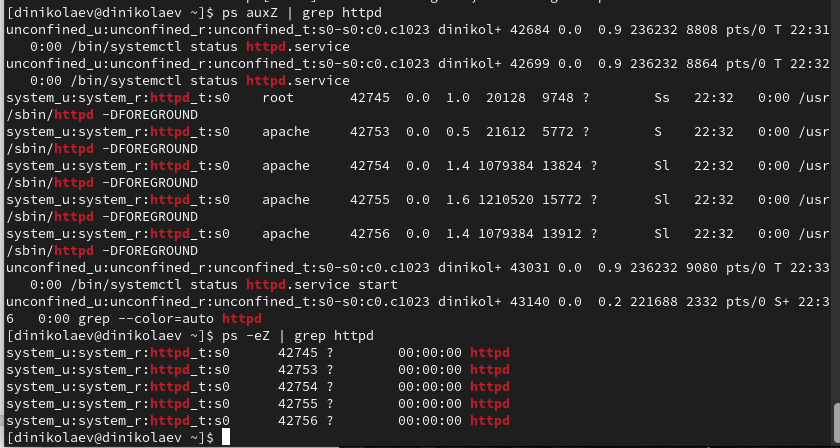
1. Войдем в систему и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus. ([??]).

* 
* Режим работы SELinux

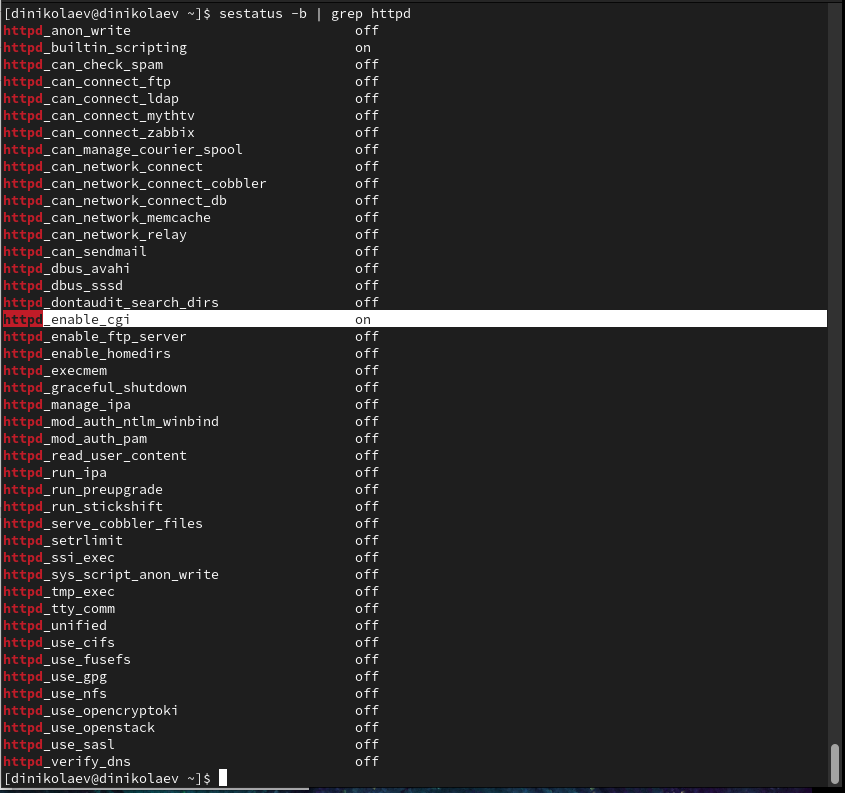
1. Обратимся с помощью браузера к веб-серверу, запущенному на нашем компьютере, и убедимся, что последний работает: service httpd status ([??]).

* 
* Обращение к веб-серверу

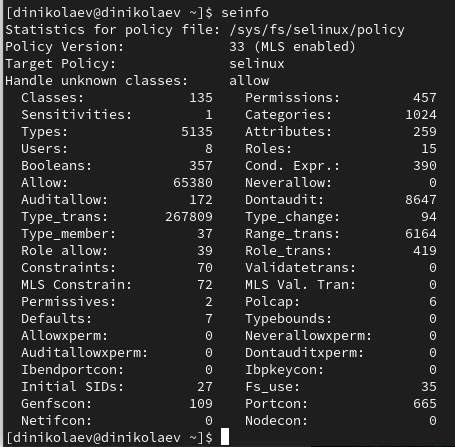
1. Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности командами ps auxZ | grep httpd и ps -eZ | grep httpd. Имеем следующий контекст: system\_u:system\_r:httpd\_t (пользователь:роль:тип) ([??]).

* 
* Проверка контекста безопасности процессов

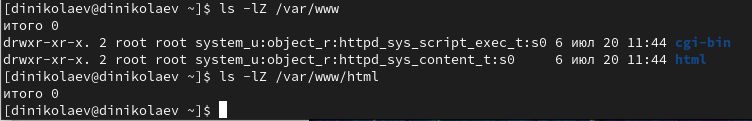
1. Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды sestatus -b | grep httpd ([??]):

* 
* Состояние переключателей SELinux

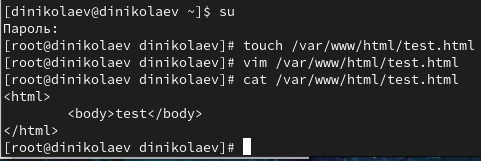
1. Посмотрим статистику по политике с помощью команды seinfo. Как видим, имеется 8 пользователей, 15 ролей и 5135 типов ([??]).

* 
* Статистика по политике

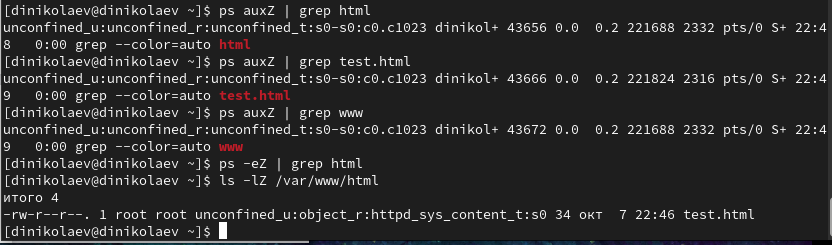
1. Определим тип файлов и поддиректорий, находящихся в директории /var/www с помощью команды ls -lZ /var/www — в ней находится две поддиректории. Определим тип файлов, находящихся в директории /var/www/html командой ls -lZ /var/www/html — директория пуста. Определим круг пользователей, которым разрешено создание файлов в директории /var/www/html — разрешено только владельцу директории ([??]).

* 
* Информация о файлах и поддиректориях директории /var/www

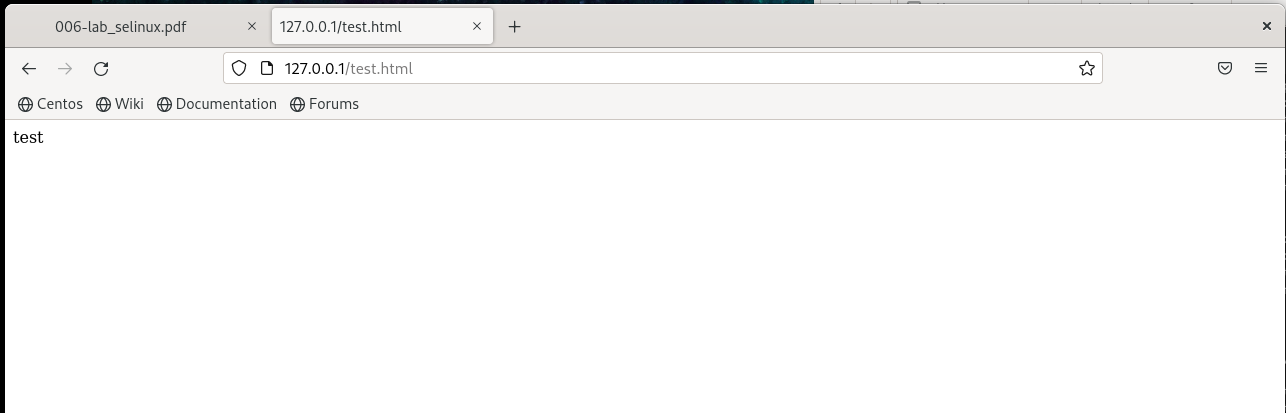
1. Создадим от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания ([??]):

* <html>  
  <body>test</body>  
  </html>
* 
* Создание html-файла

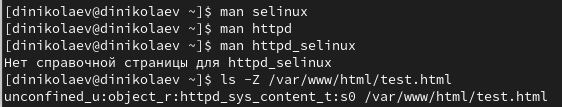
1. Проверим контекст созданного нами файла. По умолчанию устанавливается следующий контекст: пользователь — unconfined\_u (несвязанный), роль — object\_r, тип — httpd\_sys\_content\_t ([??]).

* 
* Проверка контекста созданного файла

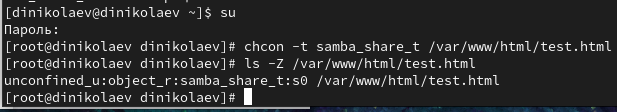
1. Обратимся к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Убедимся, что файл был успешно отображён ([??]).

* 
* Обращение к файлу через веб-сервер

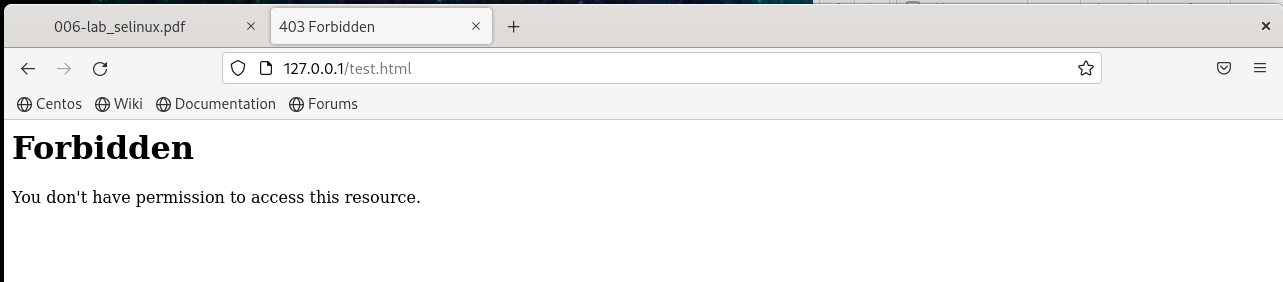
1. Изучим справку man httpd\_selinux. Справки по httpd\_selinux нет, а справки по httpd и selinux не содержат информации о возможных контекстах. Проверим контекст нашего файла командой ls -Z /var/www/html/test.html ([??]).

* 
* Вызов справки и проверка контекста созданного файла

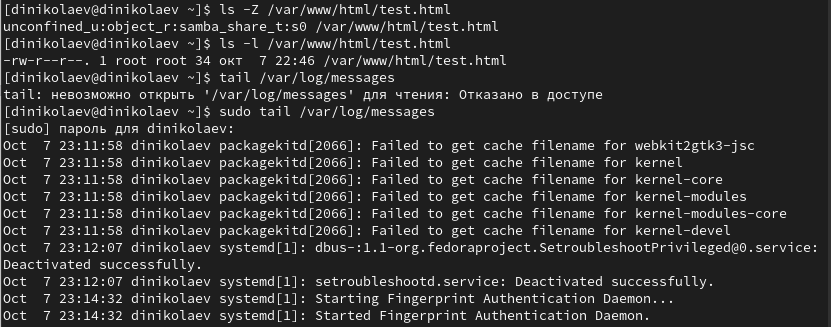
1. Изменим контекст файла /var/www/html/test.html с httpd\_sys\_content\_t на любой другой, к которому процесс httpd не должен иметь доступа, например, на samba\_share\_t ([??]):
   * chcon -t samba\_share\_t /var/www/html/test.html
   * ls -Z /var/www/html/test.html

* 
* Смена контекста файла

1. Попробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Получили сообщение об ошибке ([??]):

* Forbidden You don’t have permission to access /test.html on this server
* 
* Попытка получения доступа к файлу через веб-сервер

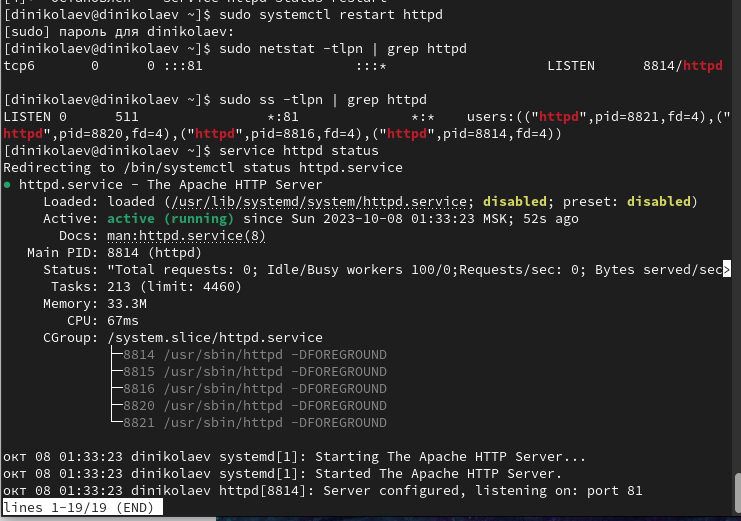
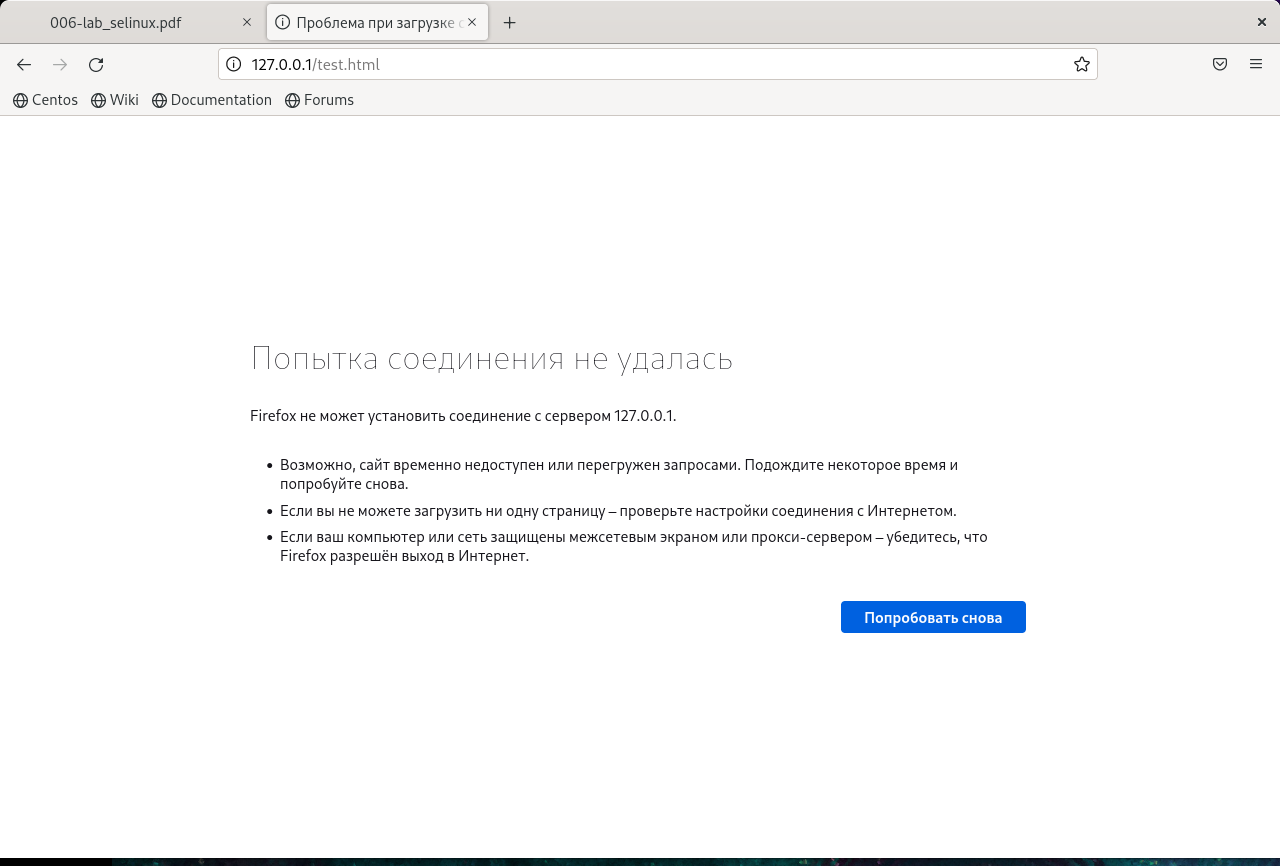
1. Проанализируем ситуацию командой ls -l /var/www/html/test.html. Просмотрим log-файлы веб-сервера Apache. Также просмотрим системный лог-файл: tail /var/log/messages ([??]).

* 
* Последние сообщения log-файла
* Не смогли получить доступ к кэшу и сообщение о деактивации от setroubleshootd.service.

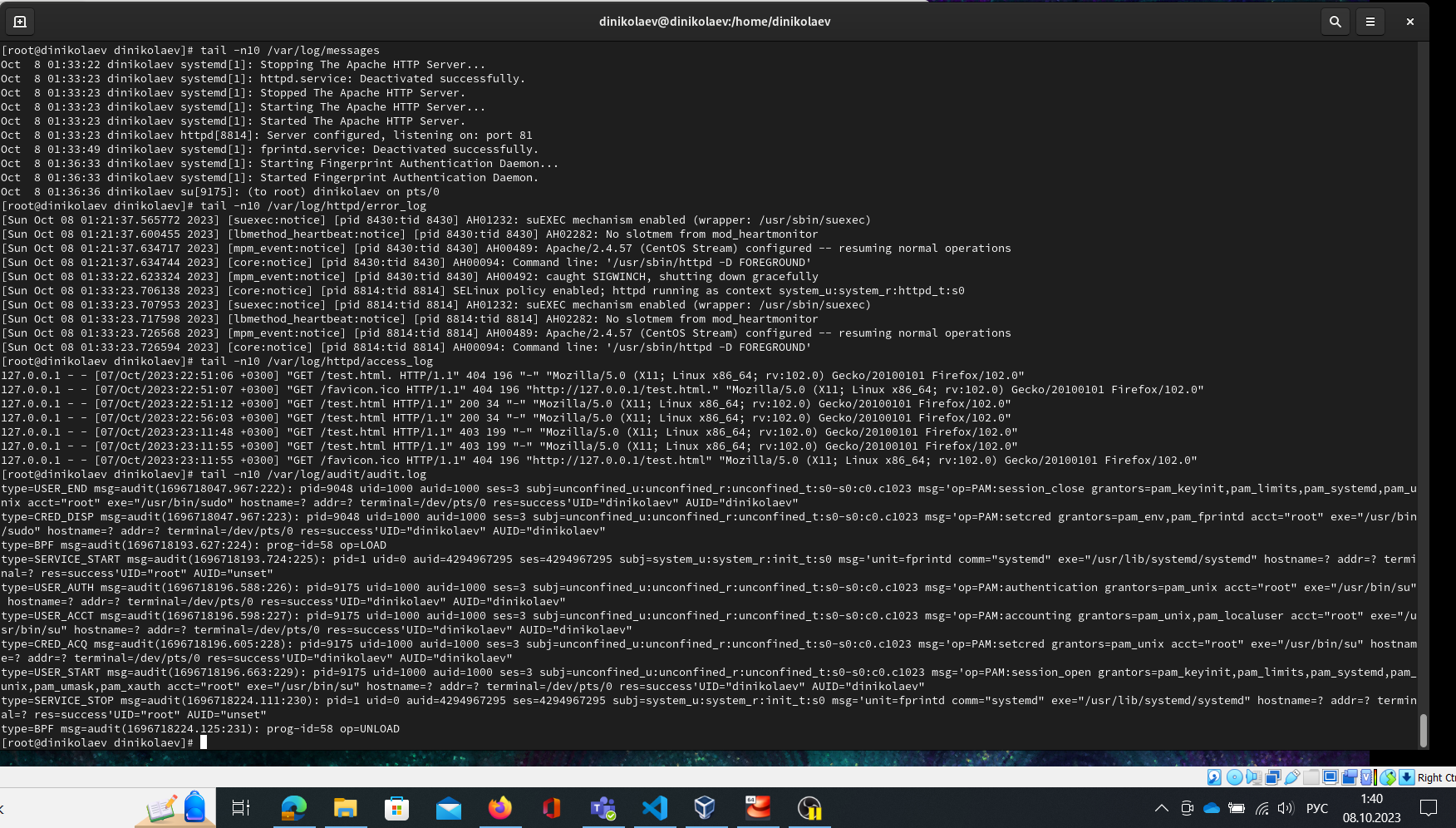
1. Попробуем запустить веб-сервер Apache на прослушивание ТСР-порта 81 (а не 80, как рекомендует IANA и прописано в /etc/services). Для этого в файле /etc/httpd/conf/httpd.conf найдем строчку Listen 80 и заменим её на Listen 81 ([??]).

* Смена порта прослушивания с 80 на 81
* Смена порта прослушивания с 80 на 81

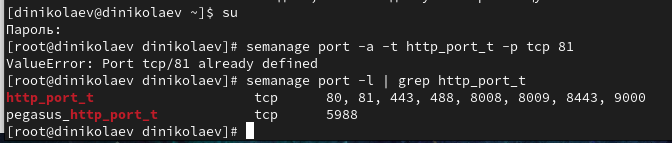
1. Выполним перезапуск веб-сервера Apache ([??]).

* 
* Перезапуск веб-сервера Apache
* Снова попытавшись получить доступ через веб-сервер, введя адрес в браузере увидим следующую картину ([??]).
* 
* Попытка подключения к веб-серверу через браузер

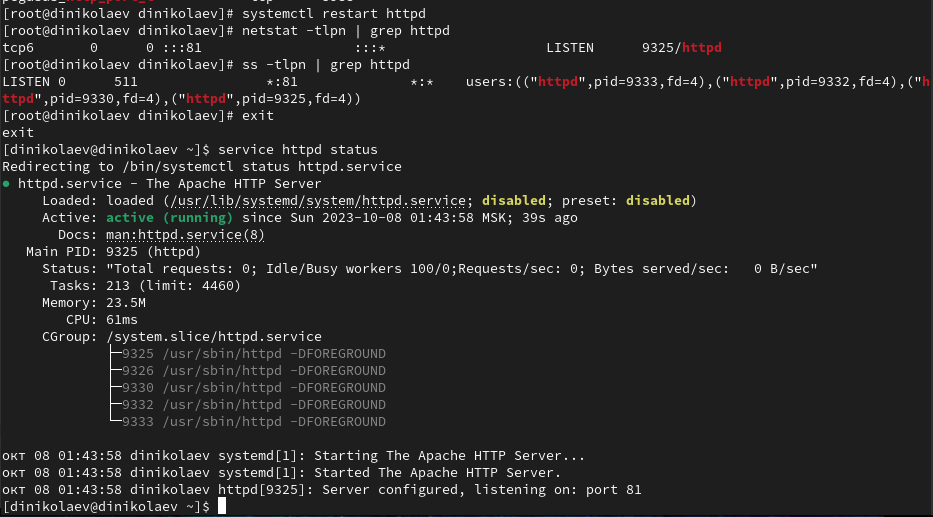
1. Проанализируем лог-файлы: tail -n10 /var/log/messages. Также просмотрим файлы /var/log/httpd/error\_log, /var/log/httpd/access\_log и /var/log/audit/audit.log/ Новые записи появились в файлах /var/log/messages и /var/log/httpd/error\_log. ([??]).

* 
* Анализ сообщений лог-файлов

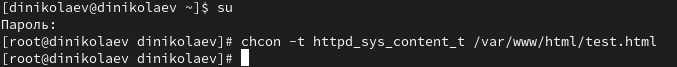
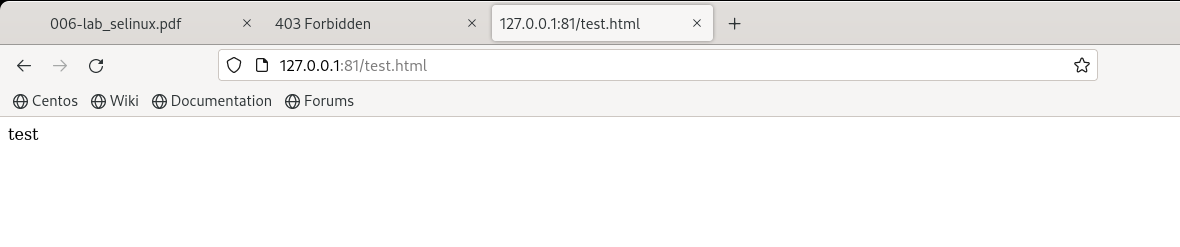
1. Выполним команду semanage port -a -t http\_port\_t -р tcp 81. После проверим список портов командой semanage port -l | grep http\_port\_t. Убедимся, что порт 81 появился в списке ([??]).

* 
* Установка 81 порта и проверка списка всех портов

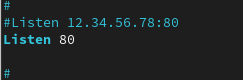
1. Попробуем запустить веб-сервер Apache ещё раз ([??]).

* 
* Повторный запуск веб-сервера Apache

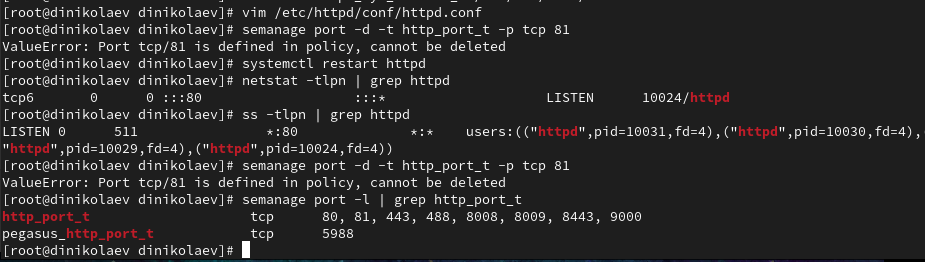
1. Вернем контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/test.html: chcon -t httpd\_sys\_content\_t /var/www/html/test.html ([??]). После этого попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html. Увидим содержимое файла — слово «test» ([??]).

* 
* Возвращение контекста httpd\_sys\_content\_t нашему файлу
* 
* Получение доступа к файлу через браузер

1. Исправим обратно конфигурационный файл apache, вернув Listen 80 ([??]).

* 
* Смена порта прослушивания с 81 на 80

1. Удалим привязку http\_port\_t к 81 порту: semanage port -d -t http\_port\_t -p tcp 81. Порт 81 не был удален, так как в начале 80 и 81 порты были настроены по умолчанию ([??]).

* 
* Попытка удаления привязки к 81 порту

1. Удалим файл /var/www/html/test.html: rm /var/www/html/test.html ([??]).

* Удаление файла test.html
* Удаление файла test.html

# 4 Выводы

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

# Список литературы

1. SELinux [Электронный ресурс]. URL: <https://habr.com/ru/companies/kingservers/articles/209644/>.

2. Кулябов Д. С., Королькова А. В., Геворкян М. Н Лабораторная работа №6 [Электронный ресурс]. RUDN, 2023. URL: <https://esystem.rudn.ru/pluginfile.php/2090210/mod_resource/content/2/006-lab_selinux.pdf>.