

Лабораторная работа № 6

Мандатное разграничение прав в Linux

Сухарев Кирилл

Содержание

Цель работы	5
Условные обозначения и термины	6
Теоретические вводные данные	7
Техническое оснащение и выбранные методы проведения работы	9
Выполнение работы	10
Выводы	34
Библиография	35

List of Figures

0.1	Проверка режима SELinux	10
0.2	Запуск и проверка веб-сервера	11
0.3	Определение контекста безопасности	12
0.4	Проверка текущего состояния переключателей SELinux	13
0.5	Проверка статистики по политике	14
0.6	Проверка файлов в var/www	15
0.7	Проверка файлов в var/www/html	16
0.8	Проверка владельцев var/www/html	17
0.9	Создание test.html	18
0.10	Контекст test.html	19
0.11	Проверка test.html	20
0.12	Изучение справки httpd_selinux	21
0.13	Изменение контекста test.html	22
0.14	Попытка получить доступ к файлу	23
0.15	Проверка логов	24
0.16	Смена порта	25
0.17	Перезапуск веб-сервера	26
0.18	Анализ лог-файлов	27
0.19	Добавление порта	28
0.20	Повторный запуск сервера	29
0.21	Попытка доступ к файлу через веб-сервер	30
0.22	Возврат порта 80	31
0.23	Удаление привязки	32
0.24	Удаление test.html	33

List of Tables

Цель работы

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

Условные обозначения и термины

Утилита - сервисная программа, облегчающая пользование другими программами, работу с компьютером.

Учетная запись - хранимая в компьютерной системе совокупность данных о пользователе, необходимая для его опознавания (аутентификации) и предоставления доступа к его личным данным и настройкам.

Директория - объект в файловой системе, упрощающий организацию файлов.

Теоретические вводные данные

Мандатная модель управления доступом (Mandatory Access Control, MAC) — способ разграничения доступа с фиксированным набором полномочий. Обычно настоящий MAC используется в системах с повышенными требованиями к безопасности и стоит на службе всевозможных силовых ведомств и организаций, связанных с государственной или служебной тайной.

Модель MAC по своей сути является «электронной» реализацией бумажного «секретного» документооборота. В MAC имеются следующие «действующие лица»:

- Иерархия уровней доступа, которые обрабатываются в системе (обычно регистрируются в ОС). Для удобства часто задается в виде беззнаковых чисел (от 0 до значения, ограниченного реализацией). В этом случае для сравнения уровней доступа (выше/ниже/равно) используются простейшие арифметические операции (равно, меньше, больше).
- Объект с уровнем секретности. Любой файл, каталог в файловой системе, ячейка или запись в таблице БД, таблица в БД, сама БД, сетевой пакет и т.д. Объекту присваивается любое значение из иерархии уровней доступа. Для объекта допускается повышение уровня секретности (изменение до большего значения уровня, чем текущий). Понижение уровня секретности категорически не допускается (хотя вполне реализуемо при помощи определенных уловок).
- Субъект с уровнем доступа. Процесс какого-либо приложения либо сеанс пользователя (по сути тоже процесс приложения). Метка уровня доступа наследуется от субъекта всеми создаваемыми данным субъектом объектами.

Значение уровня доступа субъекта или уровня секретности объекта обычно называют термином «мандатный уровень», «мандатная метка» или просто «метка» (в STCSEC данный термин называется «hierarchical classification level»). Просто, емко и почти однозначно.

Проверка полномочий осуществляется при каждом факте доступа субъекта к объекту, защищаемому MAC. При этом мандатная модель управления доступом обычно используется совместно с другими механизмами контроля доступа, например, DAC (UNIX-моделью и POSIX ACL). При этом MAC проверяется в последнюю очередь. Сперва проверяется доступ по DAC (как наименее защищенный), а затем уже MAC.

При проверке правомочности доступа субъекта к объекту согласно мандатной модели возможны следующие комбинации:

1. Мандатная метка субъекта равна мандатной метке объекта. В этом случае субъекту разрешено читать и изменять объект.
2. Мандатная метка субъекта выше мандатной метки объекта. Субъекту разрешено только читать объект: он его видит, но не может изменить.
3. Мандатная метка субъекта ниже мандатной метки объекта. Субъекту формально разрешено создать объект с более высокой мандатной меткой (так называемое «повышение уровня секретности объекта»). На практике у субъекта нет технической возможности для выполнения данной операции (он просто «не видит» изменяемый объект, например, файл или каталог с файлами).

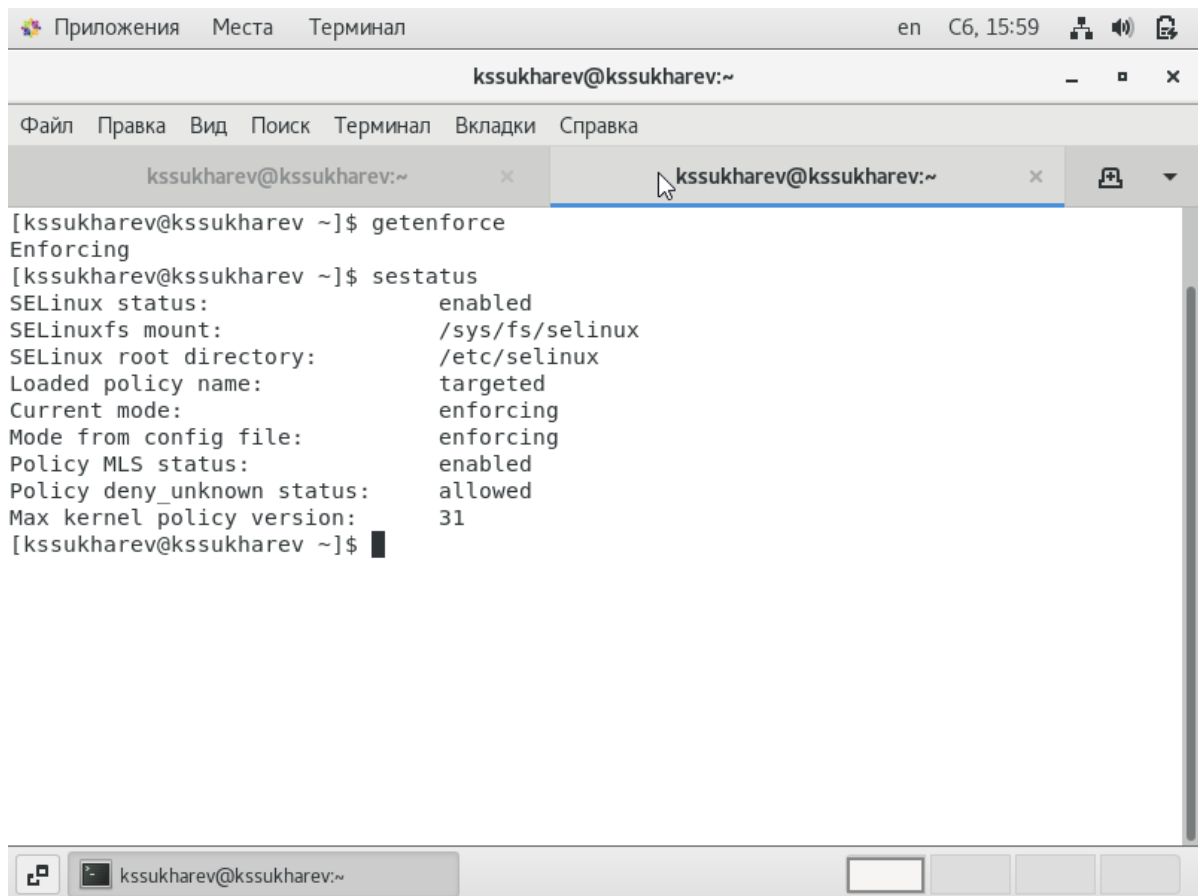
Также в MAC существует такое понятие, как «категория» (в терминологии STCSEC данный термин называется «non-hierarchical categories»). Категории в MAC являются опциональными к применению. В практике реализации MAC категории используются для «горизонтального» разграничения доступа между различными подразделениями организации. В этом случае сотрудники, несмотря на один мандатный уровень, будут получать доступ только к тем категориям объектов, к которым для них открыт доступ согласно их метке.

Техническое оснащение и выбранные методы проведения работы

В качестве среды выполнения лабораторной работы используется менеджер виртуальных машин VirtualBox и установленная с его помощью ОС Centos 7 на базе Linux.

Выполнение работы

1. Войдем в систему под пользователем `guest` выполним команды `getenforce` и `sestatus`. Видим, что SELinux работает в режиме enforcing политики targeted (fig. 0.1).

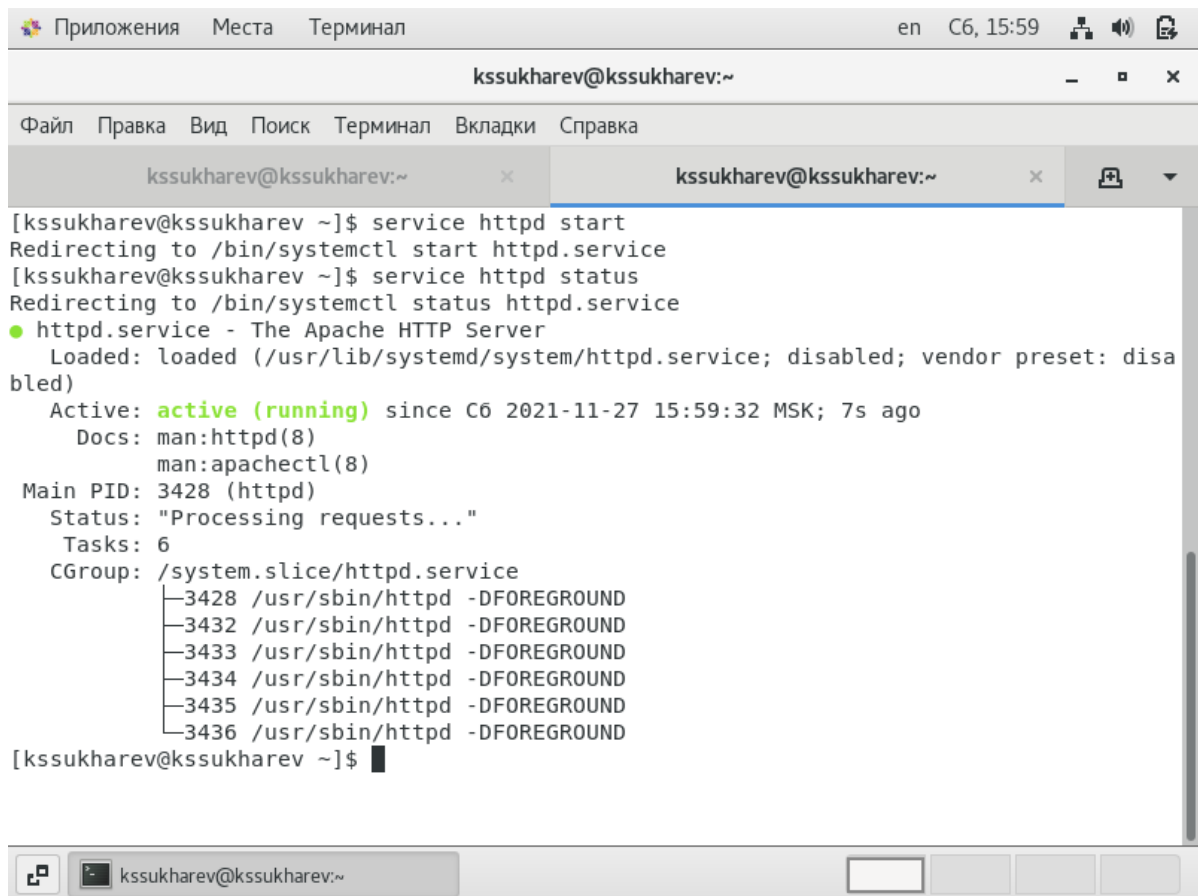
A screenshot of a Linux terminal window. The window has a title bar with 'Приложения', 'Места', and 'Терминал' on the left, and 'en C6, 15:59' and system icons on the right. The terminal title is 'kssukharev@kssukharev:~'. The menu bar includes 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', 'Вкладки', and 'Справка'. The terminal shows the following commands and output:

```
[kssukharev@kssukharev ~]$ getenforce
Enforcing
[kssukharev@kssukharev ~]$ sestatus
SELinux status:                enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:              targeted
Current mode:                    enforcing
Mode from config file:           enforcing
Policy MLS status:               enabled
Policy deny_unknown status:      allowed
Max kernel policy version:       31
[kssukharev@kssukharev ~]$
```

Figure 0.1: Проверка режима SELinux

2. Запустим веб-сервер командой `service httpd start` и обратимся к нему, при

помощи браузера, затем проверим, что сервер работает командой `service httpd status` (fig. 0.2).



```
[kssukharev@kssukharev ~]$ service httpd start
Redirecting to /bin/systemctl start httpd.service
[kssukharev@kssukharev ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since C6 2021-11-27 15:59:32 MSK; 7s ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Main PID: 3428 (httpd)
    Status: "Processing requests..."
     Tasks: 6
   CGroup: /system.slice/httpd.service
           └─3428 /usr/sbin/httpd -DFOREGROUND
             └─3432 /usr/sbin/httpd -DFOREGROUND
               └─3433 /usr/sbin/httpd -DFOREGROUND
                 └─3434 /usr/sbin/httpd -DFOREGROUND
                   └─3435 /usr/sbin/httpd -DFOREGROUND
                     └─3436 /usr/sbin/httpd -DFOREGROUND
[kssukharev@kssukharev ~]$
```

Figure 0.2: Запуск и проверка веб-сервера

3. Найдем веб-сервер Apache в списке процессов и определим его контекст безопасности командой `ps -eZ | grep httpd`. Получим следующий контекст `system_u:system)r:httpd_t:s0` (fig. 0.3).

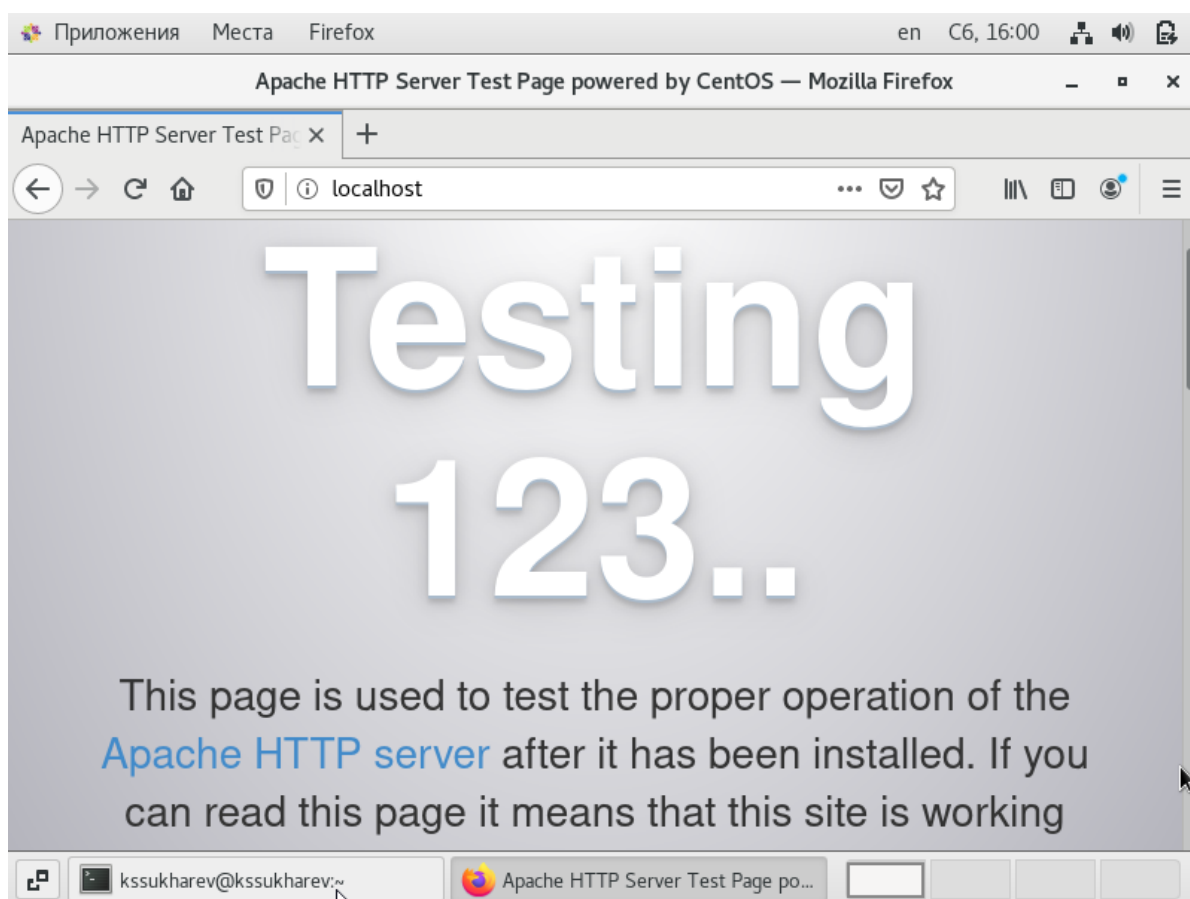


Figure 0.3: Определение контекста безопасности

4. Просмотрим текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -b | grep httpd`. Большинство из них находятся в положении "off" (fig. 0.4).

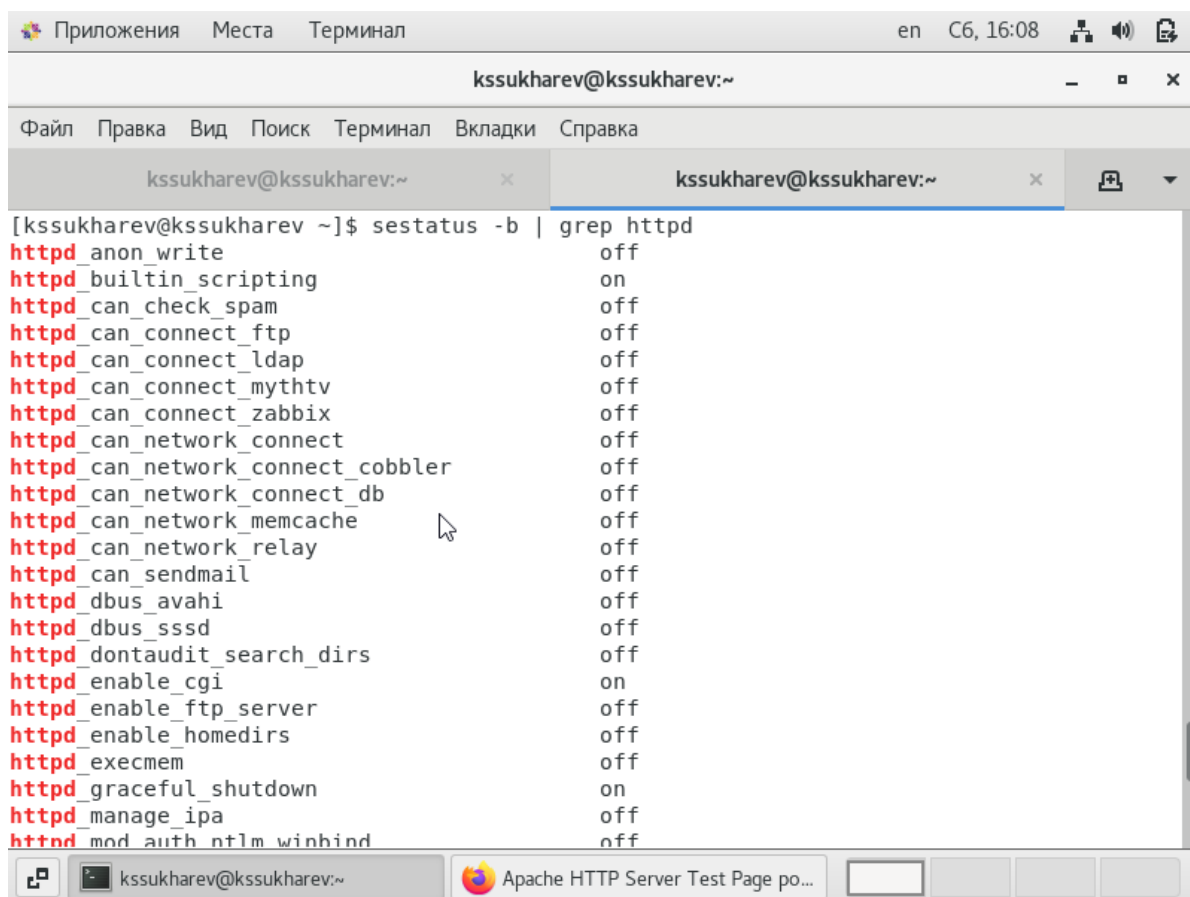


Figure 0.4: Проверка текущего состояния переключателей SELinux

5. Посмотрим на статистику по политике с помощью команды `seinfo`, также определим множество пользователей (ключ `-u`), ролей (ключ `-r`), типов (ключ `-t`) (fig. 0.5).

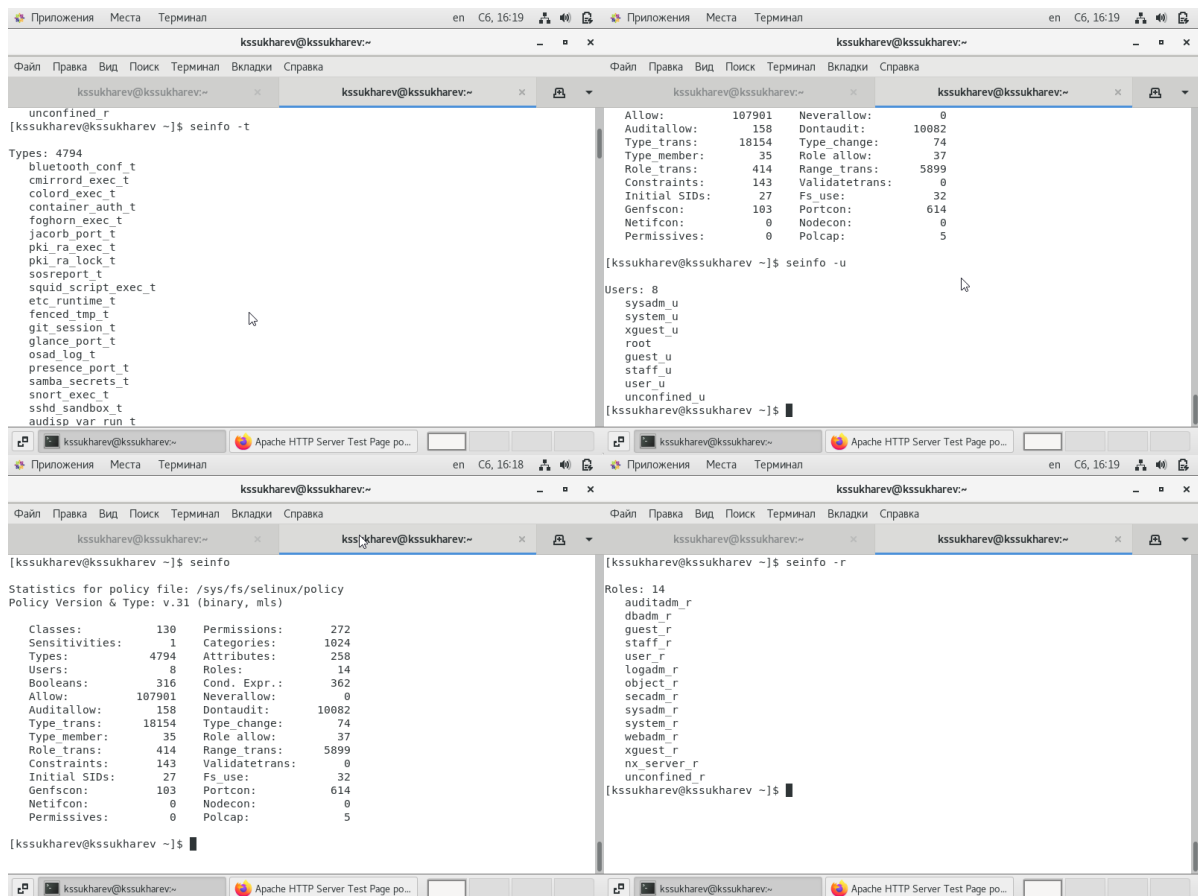


Figure 0.5: Проверка статистики по политике

6. Определим тип файлов, находящихся в директории `/var/www` при помощи команды `ls -lZ /var/www - httpd_sys_script_exec_t` и `httpd_sys_content_t` (fig. 0.6).

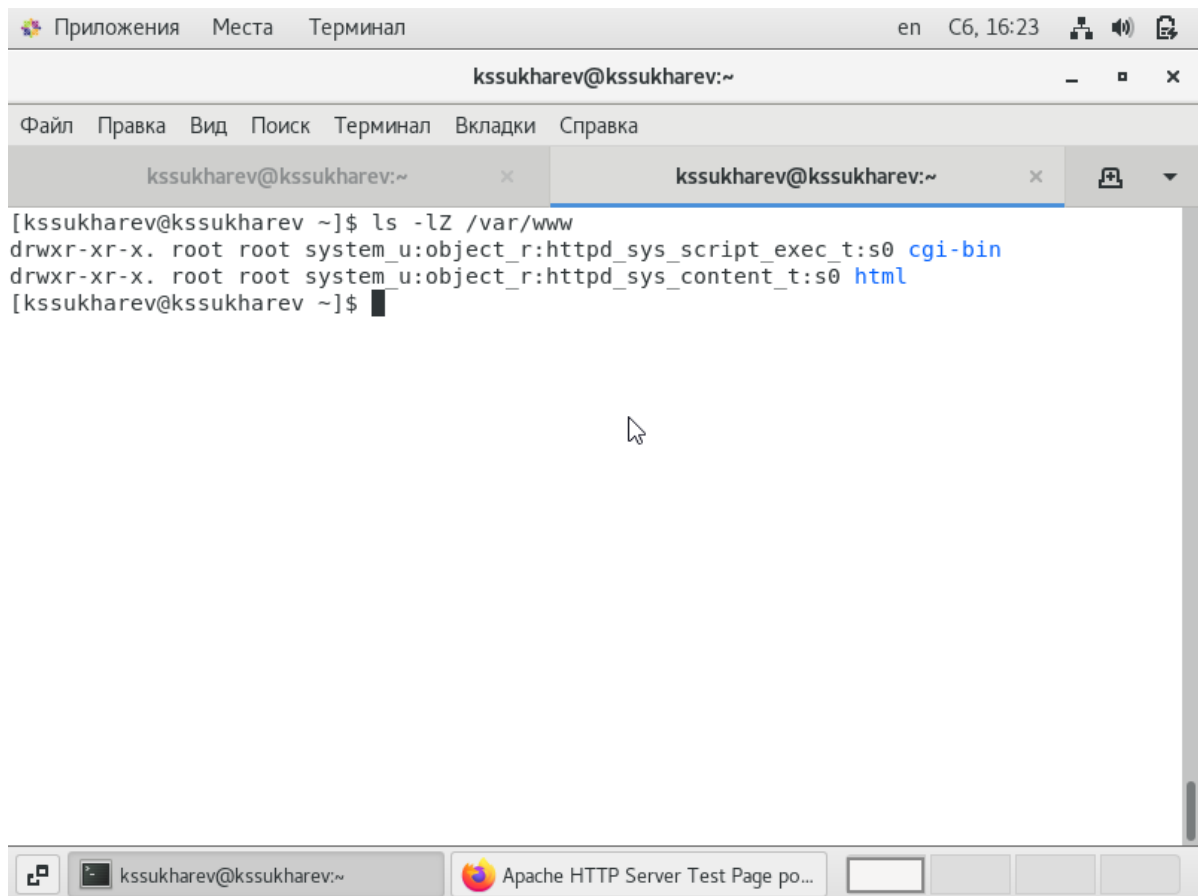


Figure 0.6: Проверка файлов в var/www

7. Аналогичным образом проверим директорию /var/www/html. Поскольку файлов в ней нет, то и типы определить не получится (fig. 0.7).

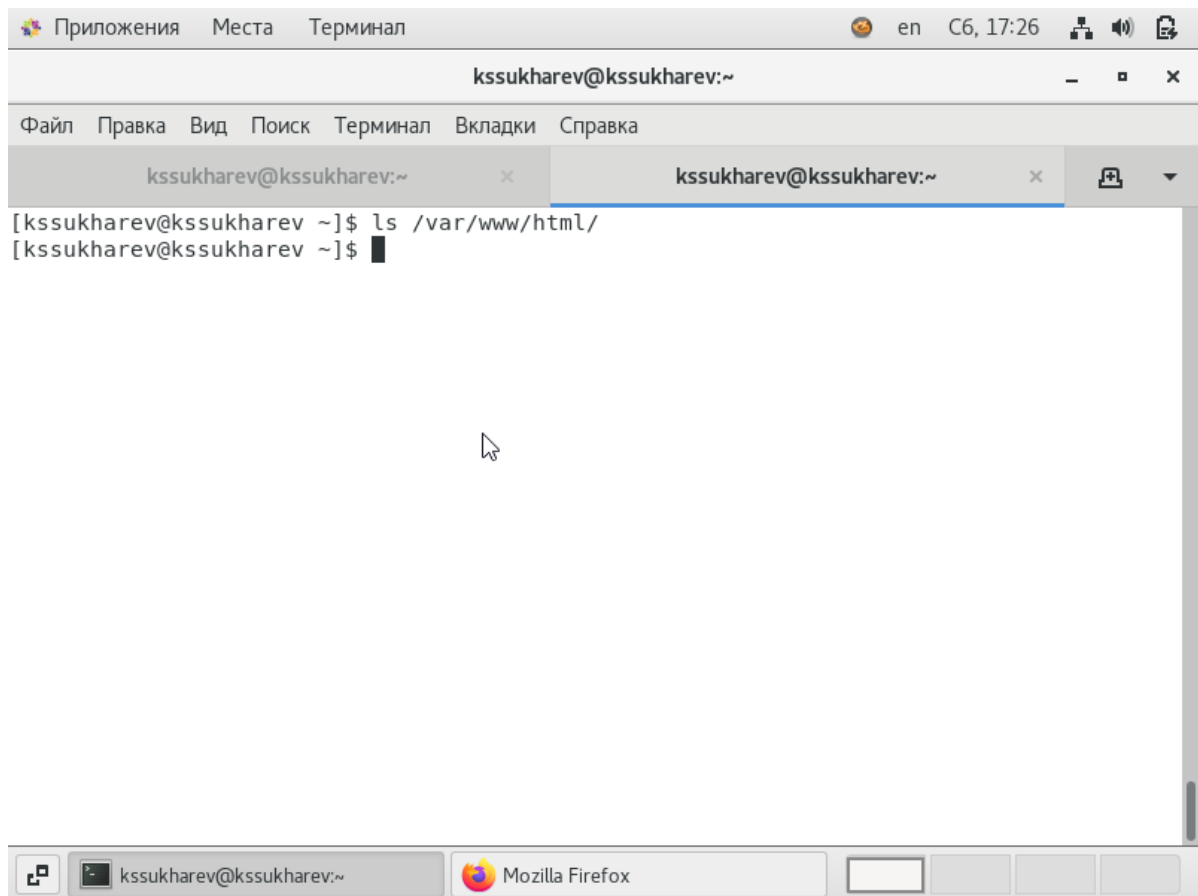


Figure 0.7: Проверка файлов в `var/www/html`

8. Определим круг пользователей, которым разрешено создание файлов в директории `/var/www/html` при помощи команды `ls -lZ /var/www`. Видим, что установлен пользователь `system_u`, что говорит о том, что создавать файлы в директории может только суперпользователь (fig. 0.8).

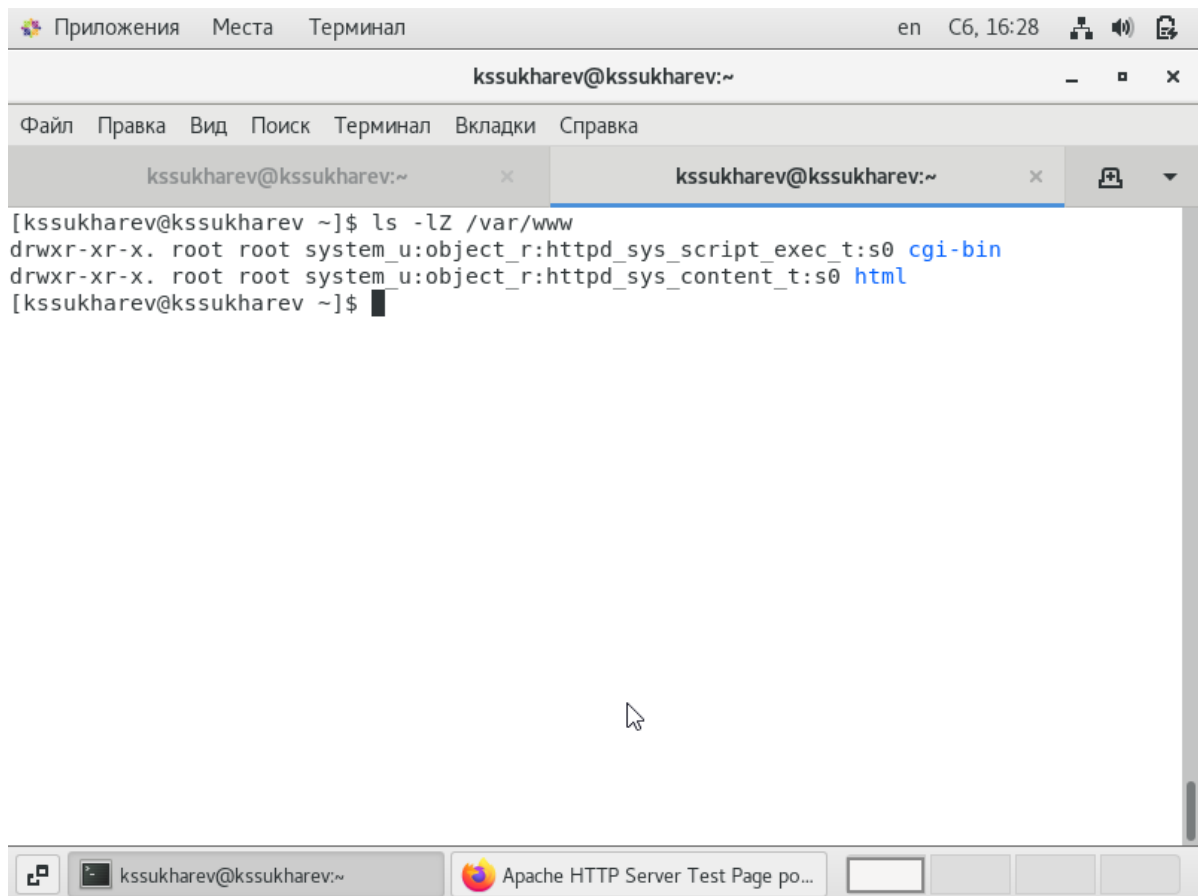


Figure 0.8: Проверка владельцев var/www/html

9. От имени суперпользователя создадим html-файл /var/www/html/test.html (fig. 0.9).

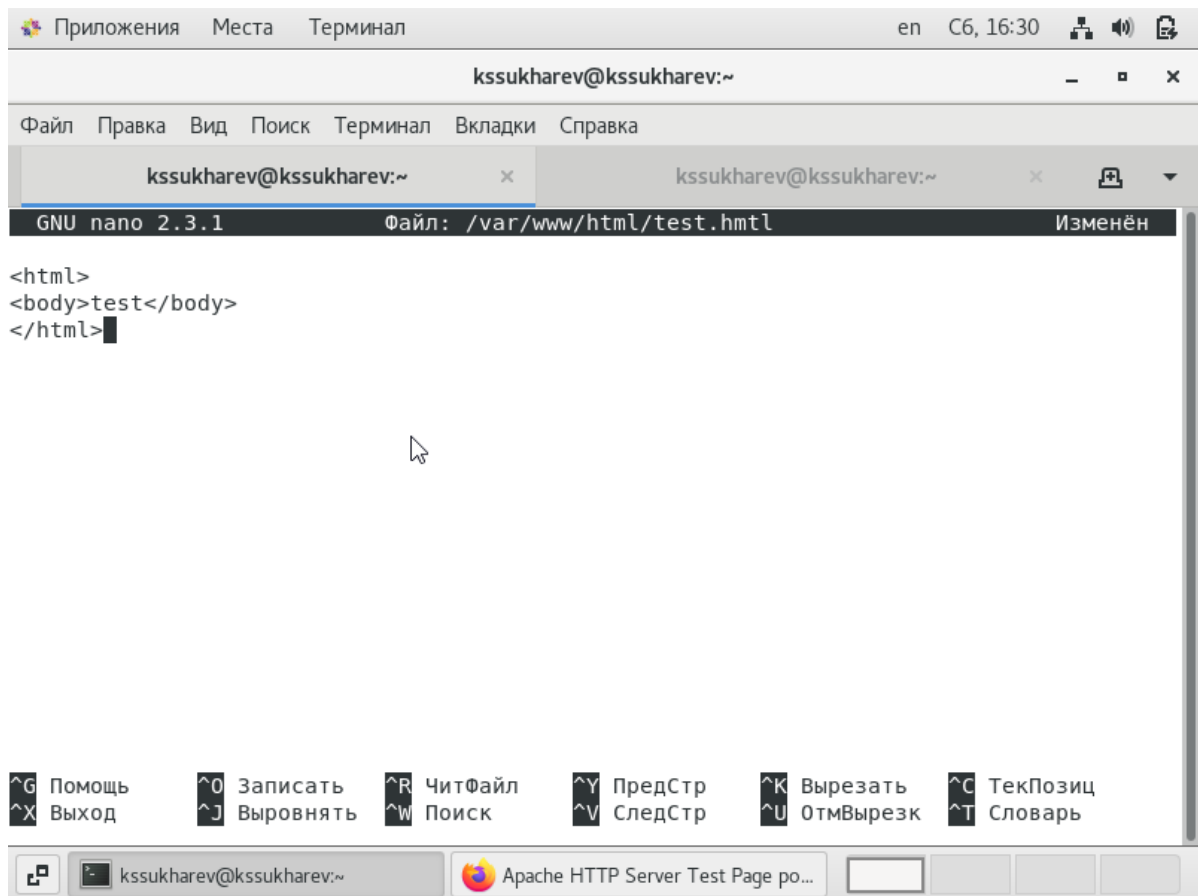


Figure 0.9: Создание test.html

10. Проверим контекст созданного файла командой `ls -lZ /var/www/html`. По умолчанию созданным файлам присваивается контекст `unconfined_u:object_r:httpd_sys_content_` (fig. 0.10).

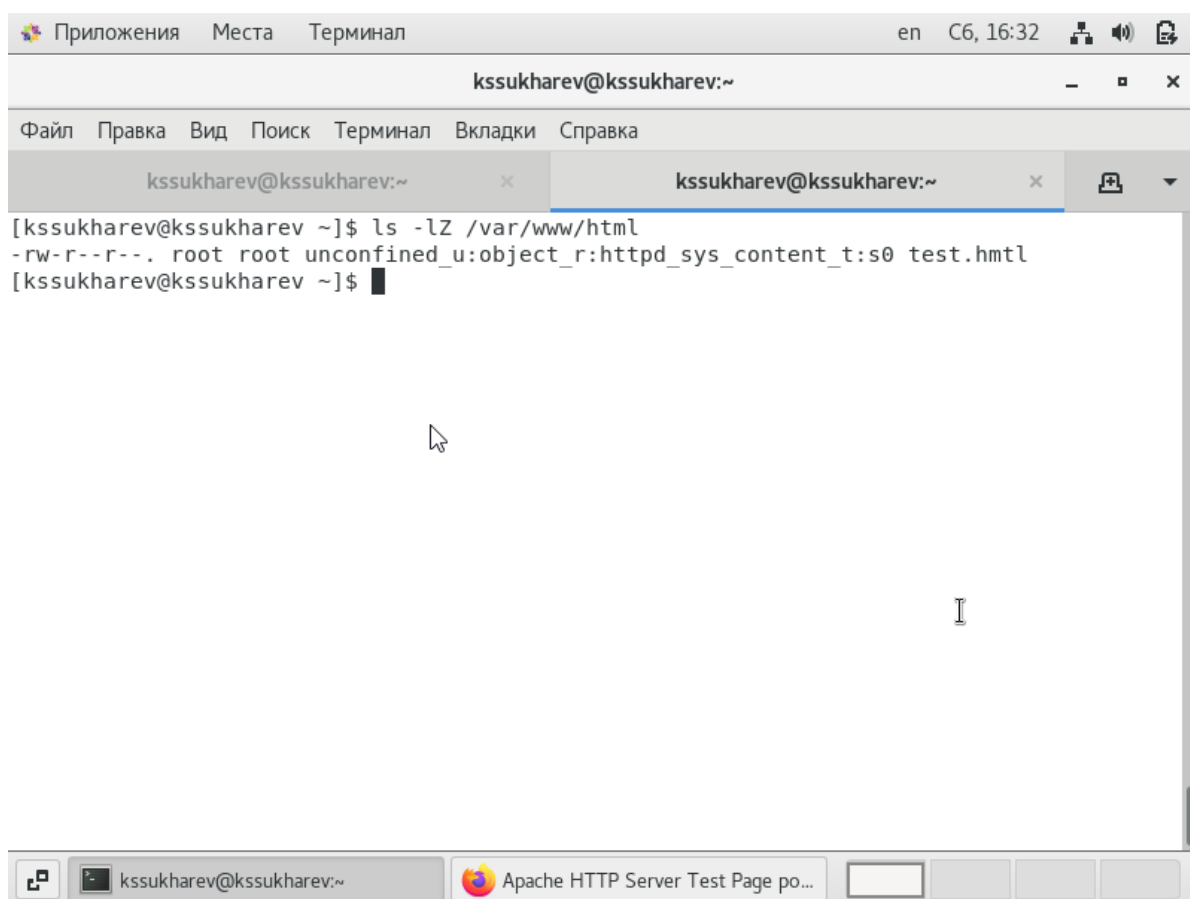


Figure 0.10: Контекст test.html

11. Обратимся к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html>.
Файл был успешно отображен (fig. 0.11).



Figure 0.11: Проверка test.html

12. Изучим справку командой `man httpd_selinux`. Выясняется, что контекст был выбран верно (fig. 0.12).

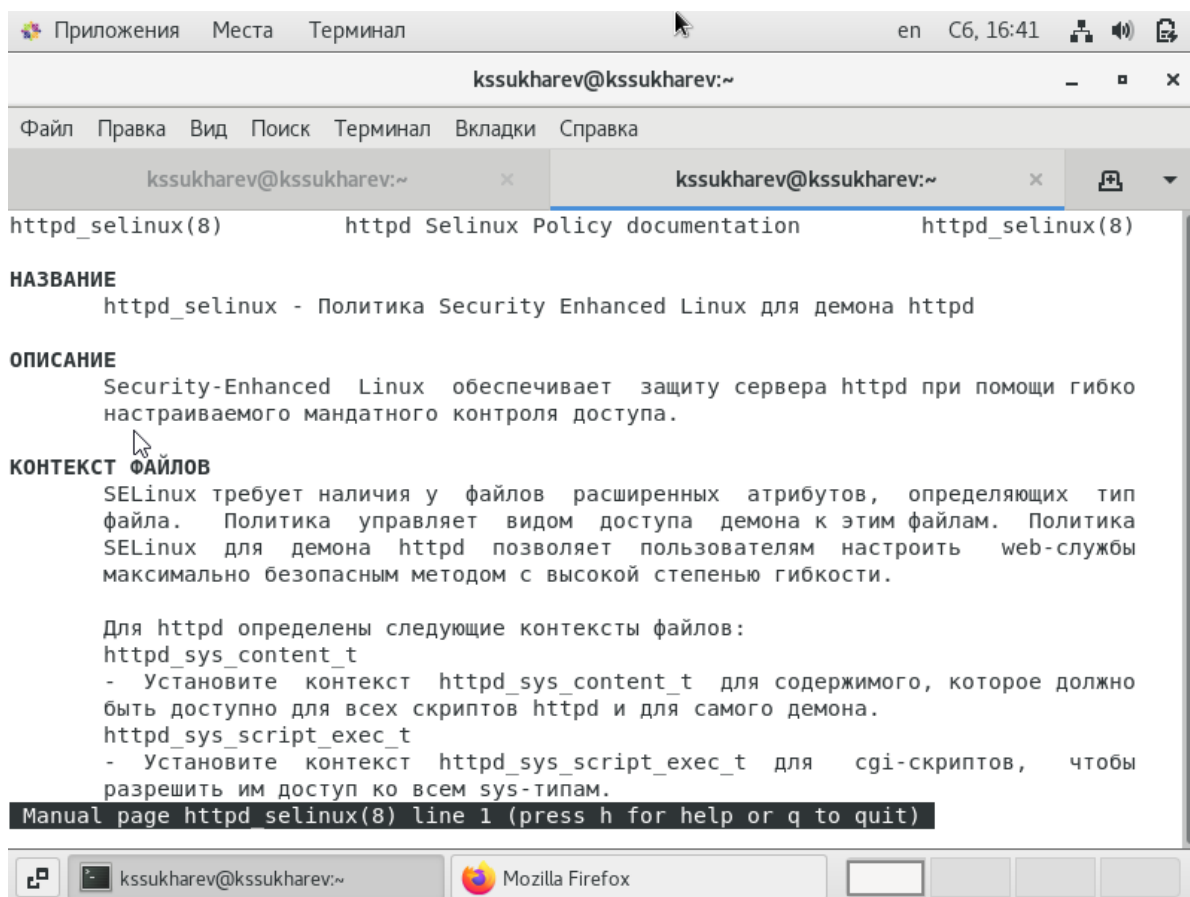


Figure 0.12: Изучение справки httpd_selinux

13. Изменим контекст файла test.html, например на samba_share_t командой `chcon -t samba_share_t /var/www/html/test.html`. Затем командой `ls -Z /var/www/html/test.html` убедимся, что контекст изменился (fig. 0.13).

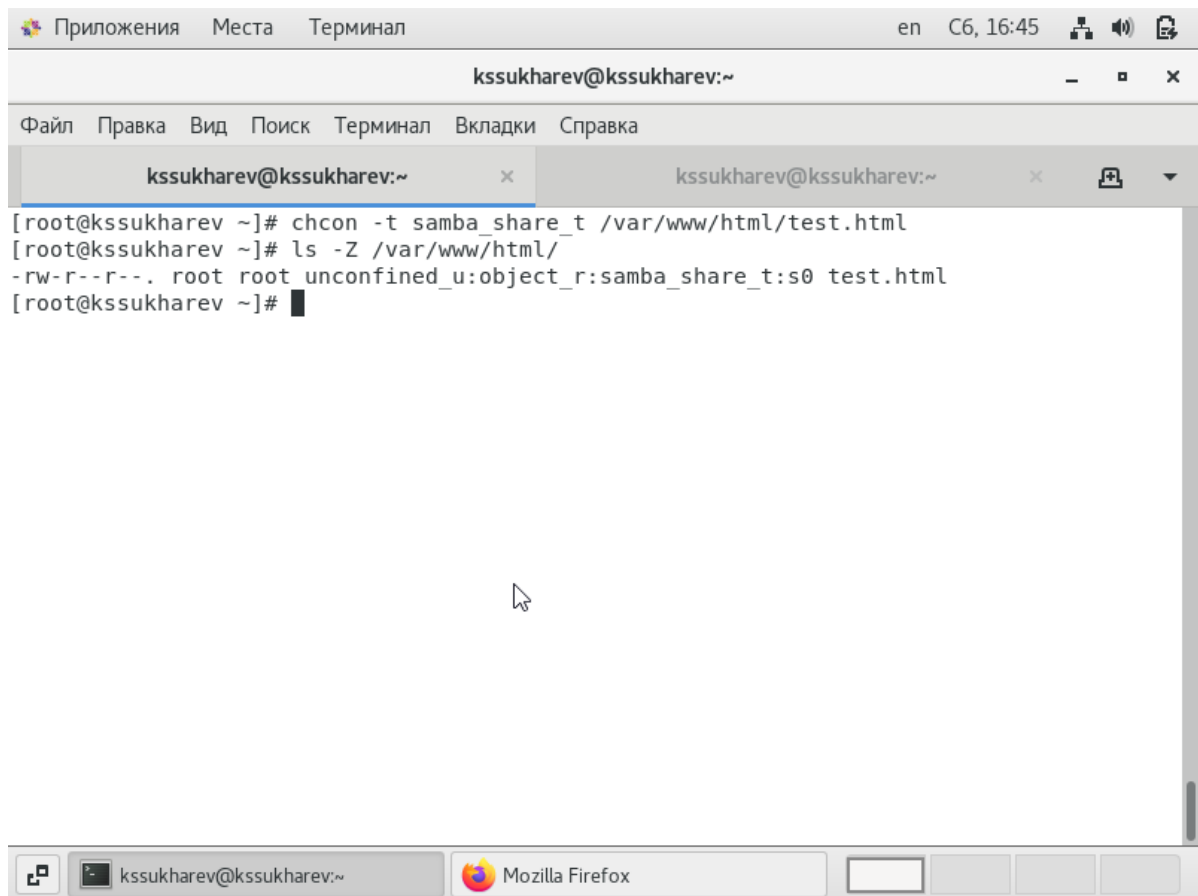
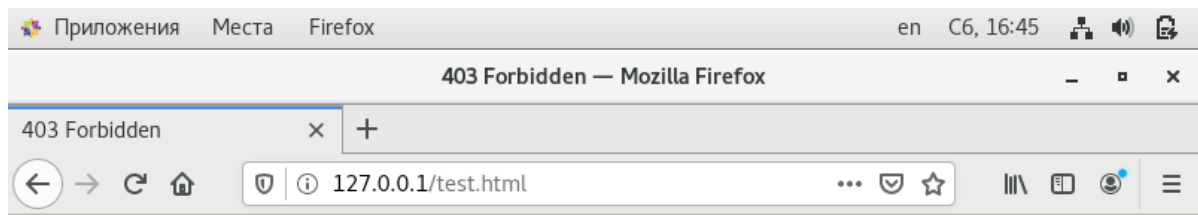


Figure 0.13: Изменение контекста test.html

14. Попробуем еще раз получить доступ к файлу через веб-сервер. Получим ошибку (fig. 0.14).



Forbidden

You don't have permission to access /test.html on this server.

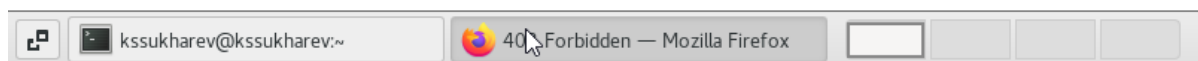


Figure 0.14: Попытка получить доступ к файлу

15. Посмотрим логи командой `tail /var/log/messages` и увидим, что ошибка возникла, поскольку служба `httpd` не имеет доступа к выбранному нами типу файлов из-за разницы контекстов (fig. 0.15).

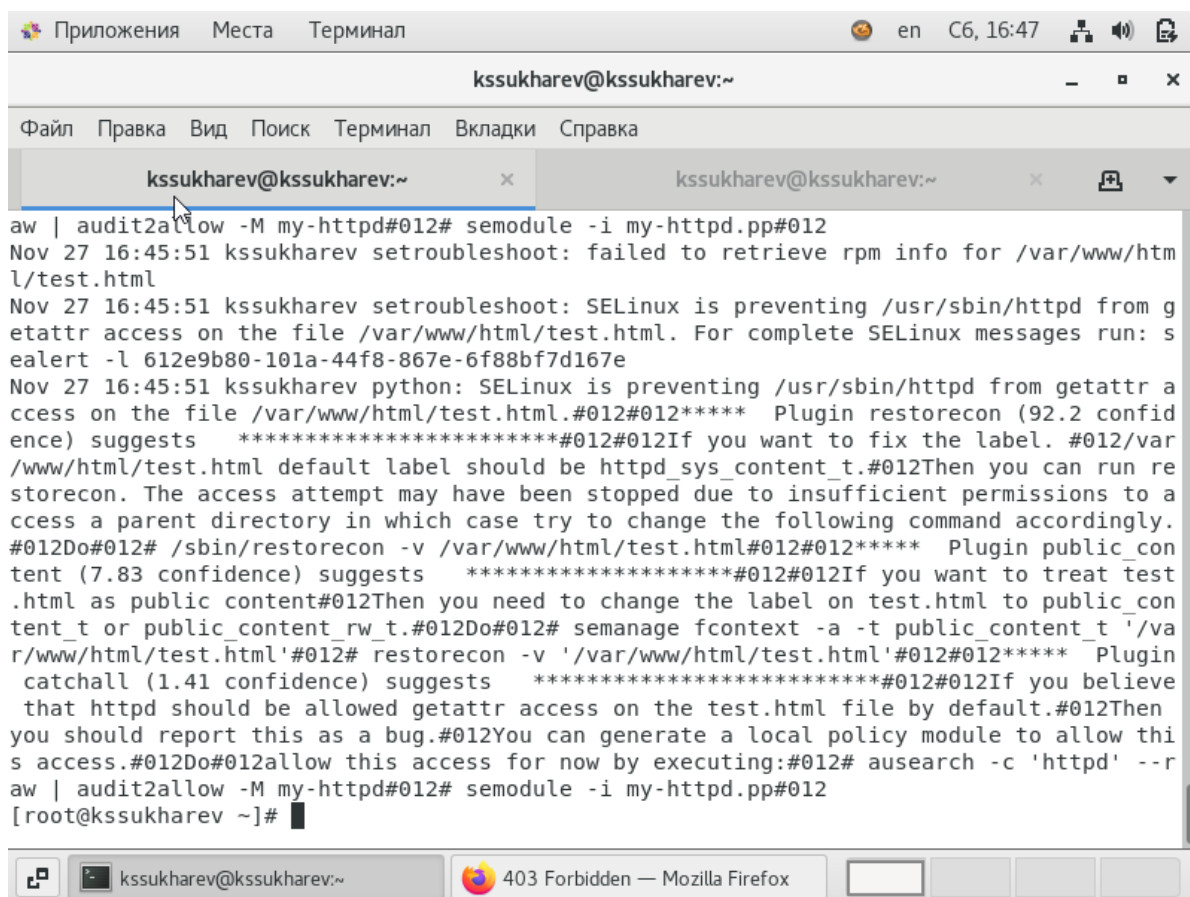


Figure 0.15: Проверка логов

16. Изменим в файле `/etc/httpd/httpd.conf` строчку `Listen 80` на `Listen 81` (fig. 0.16).

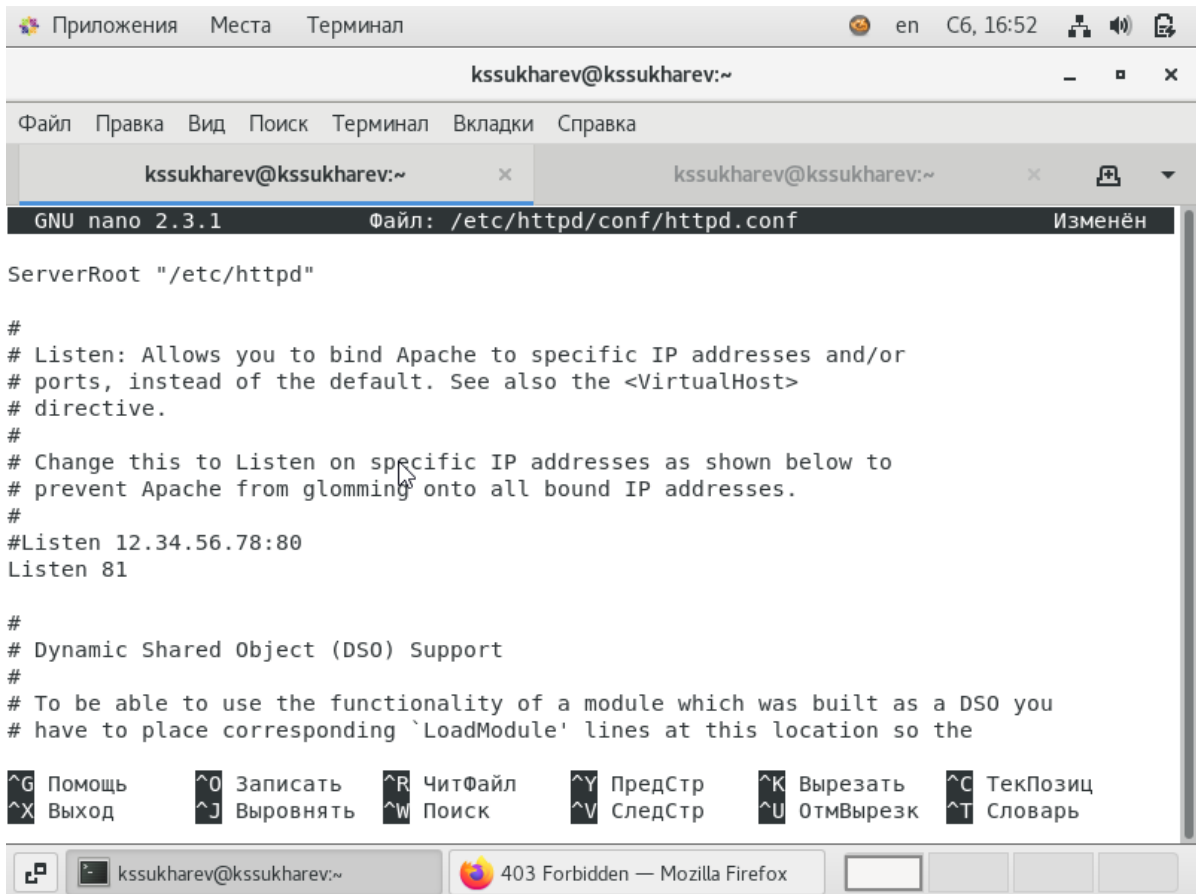


Figure 0.16: Смена порта

17. Попробуем перезапустить веб-сервер Apache командой `service httpd restart`. Сбоя не произошло (fig. 0.17).

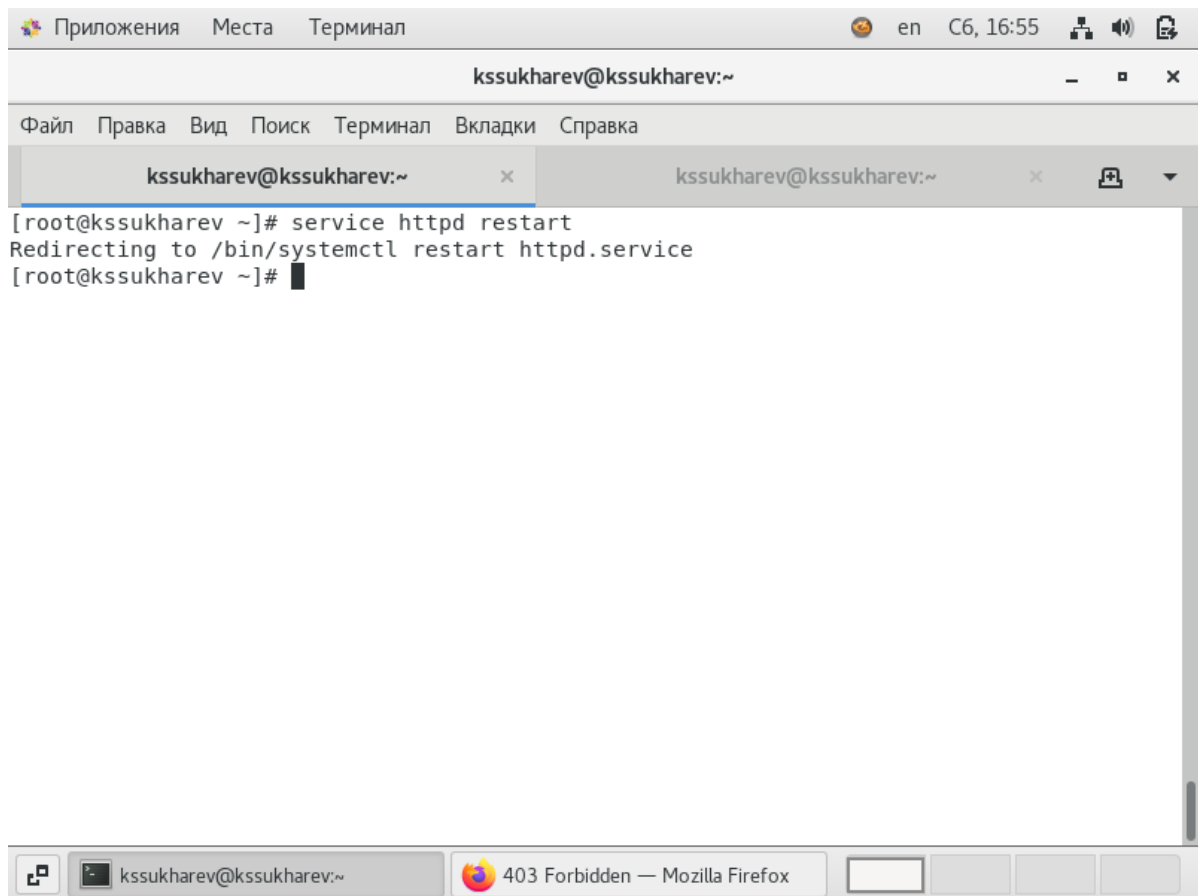


Figure 0.17: Перезапуск веб-сервера

18. Проанализируем лог-файлы командой `tail -n1 /var/log/messages`. Видим, что сервер был успешно запущен (fig. 0.18).

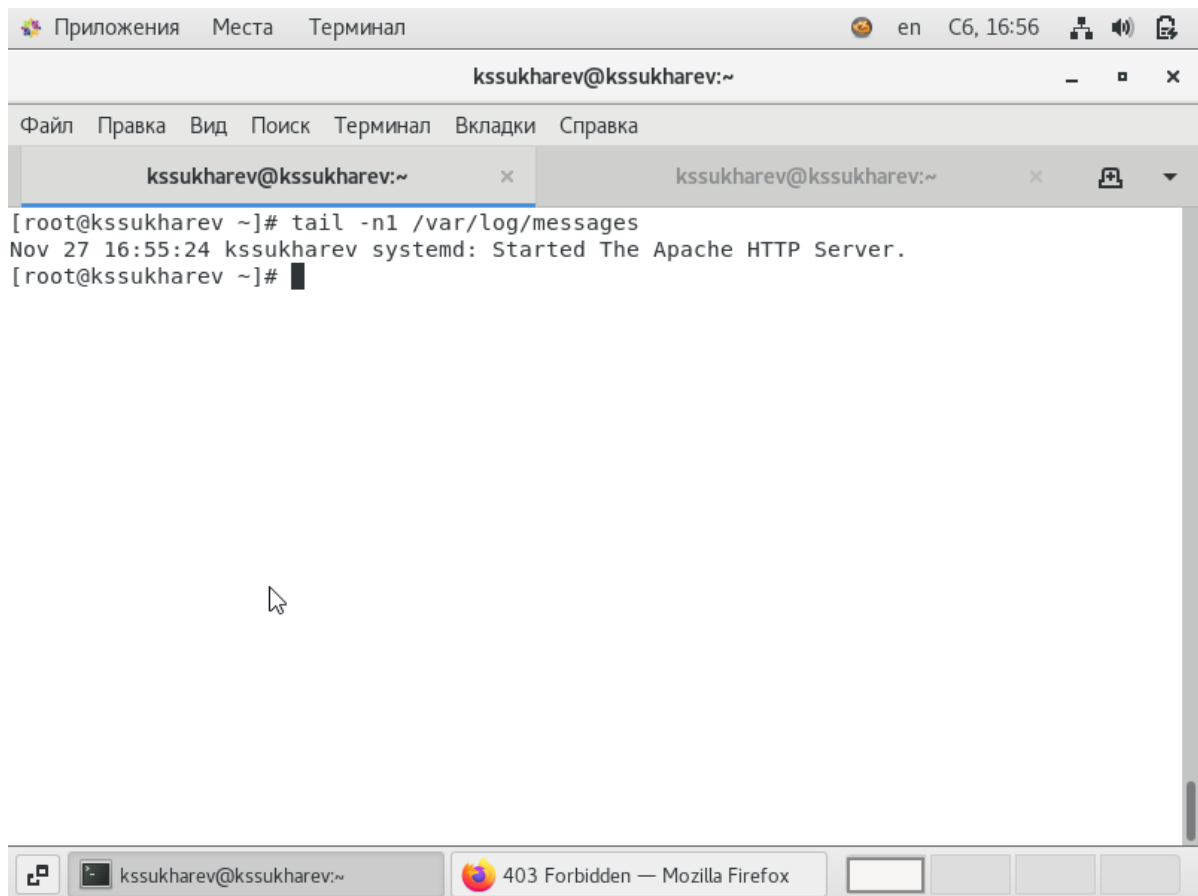


Figure 0.18: Анализ лог-файлов

19. Выполним команду `semanage port -a -t http_port_t -p tcp 81` и затем проверим, что порт 81 появился в выводе команды `semanage port -l | grep http_port_t`. Видим, что порт уже был в этом списке (fig. 0.19).

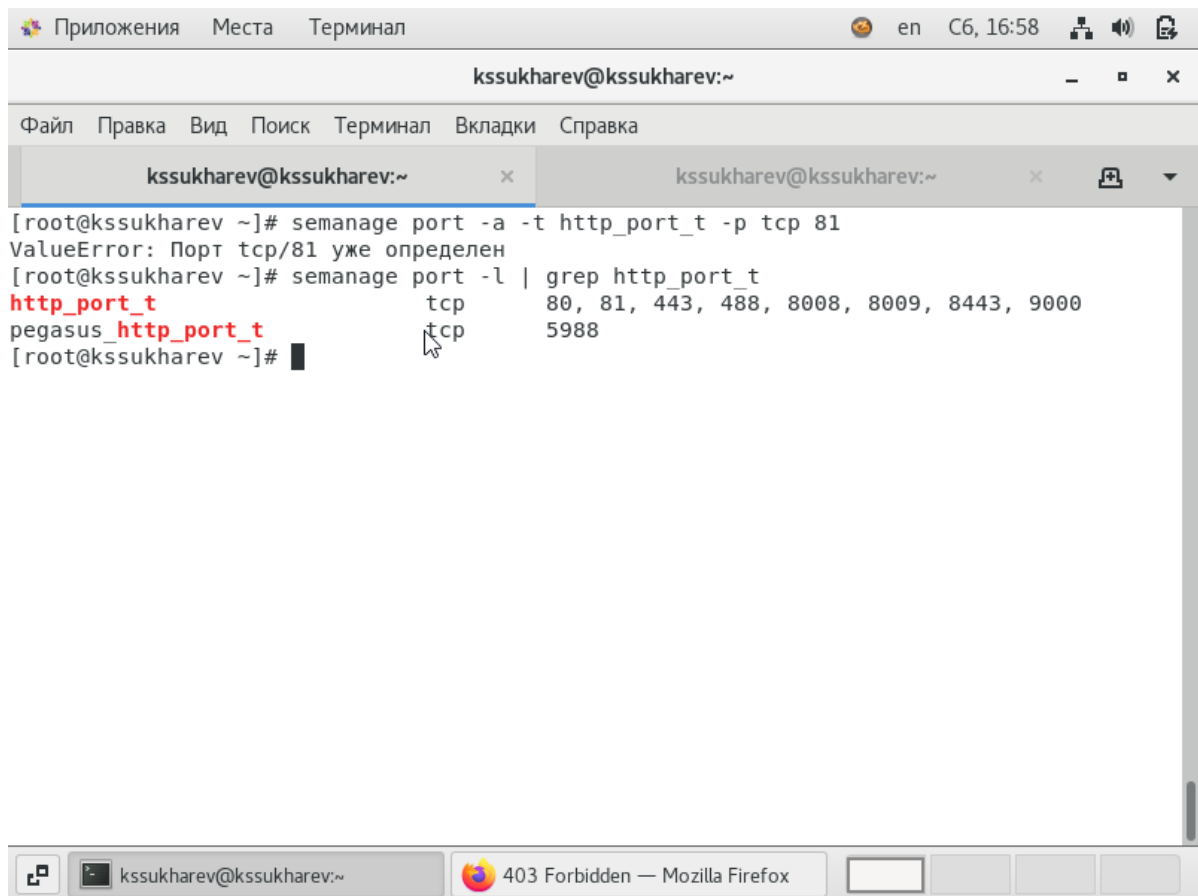


Figure 0.19: Добавление порта

20. Попробуем запустить веб-сервер еще раз. Как и в прошлый раз, он запустился, поскольку 81 порт уже был в политике (fig. 0.20).

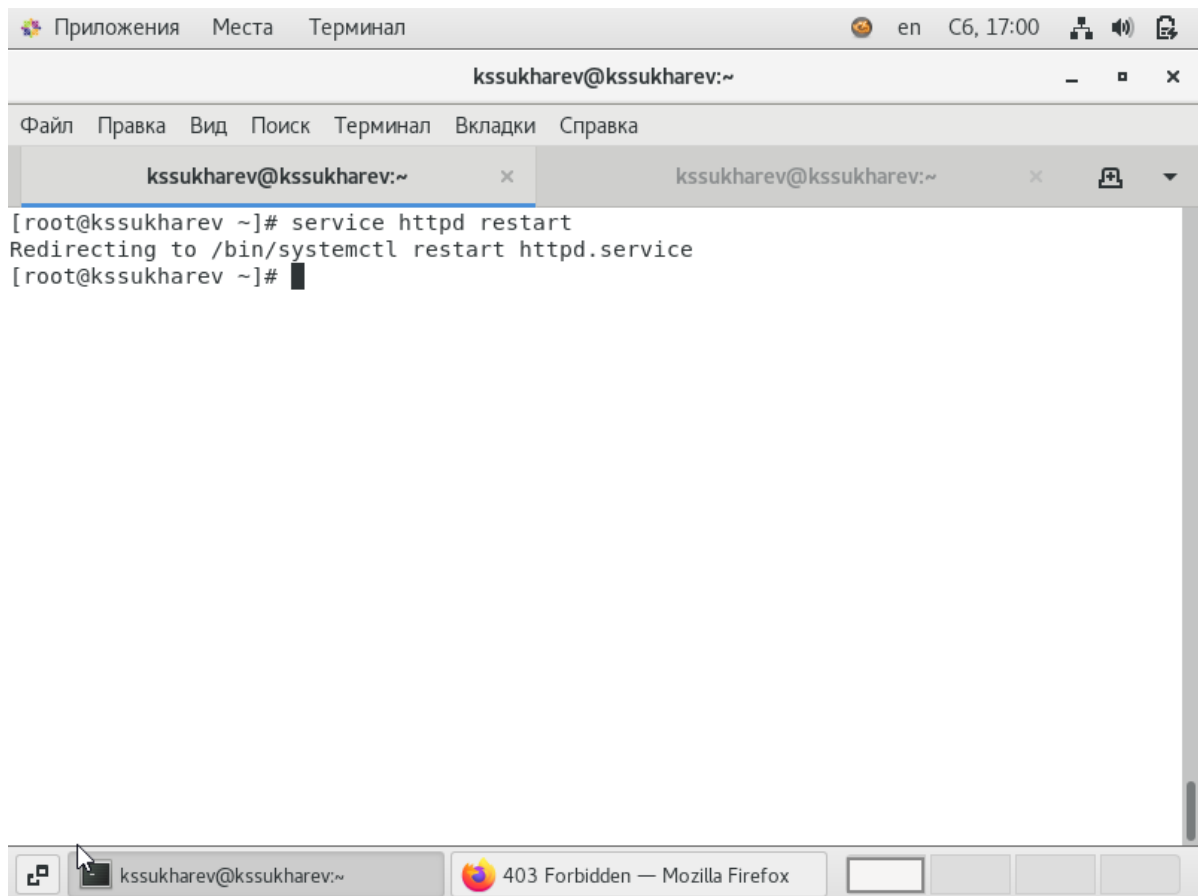
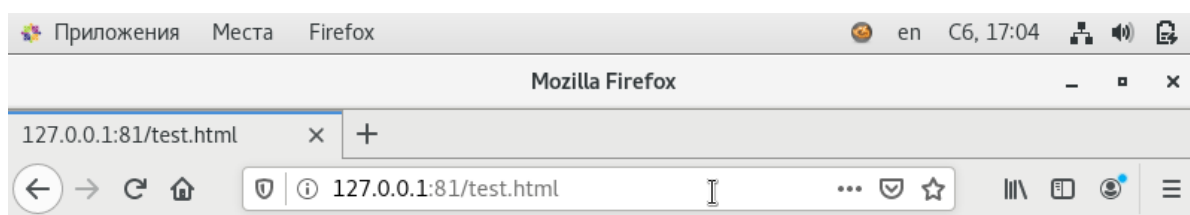


Figure 0.20: Повторный запуск сервера

21. Вернем нормальный контекст командой `chcon -t httpd_sys_content_t /var/www/html/test.html`. Затем снова попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html` (fig. 0.21).



test



Figure 0.21: Попытка доступ к файлу через веб-сервер

22. Вновь вернем порт 80 в конфигурационном файле `/etc/httpd/conf/httpd.conf` (fig. 0.22).

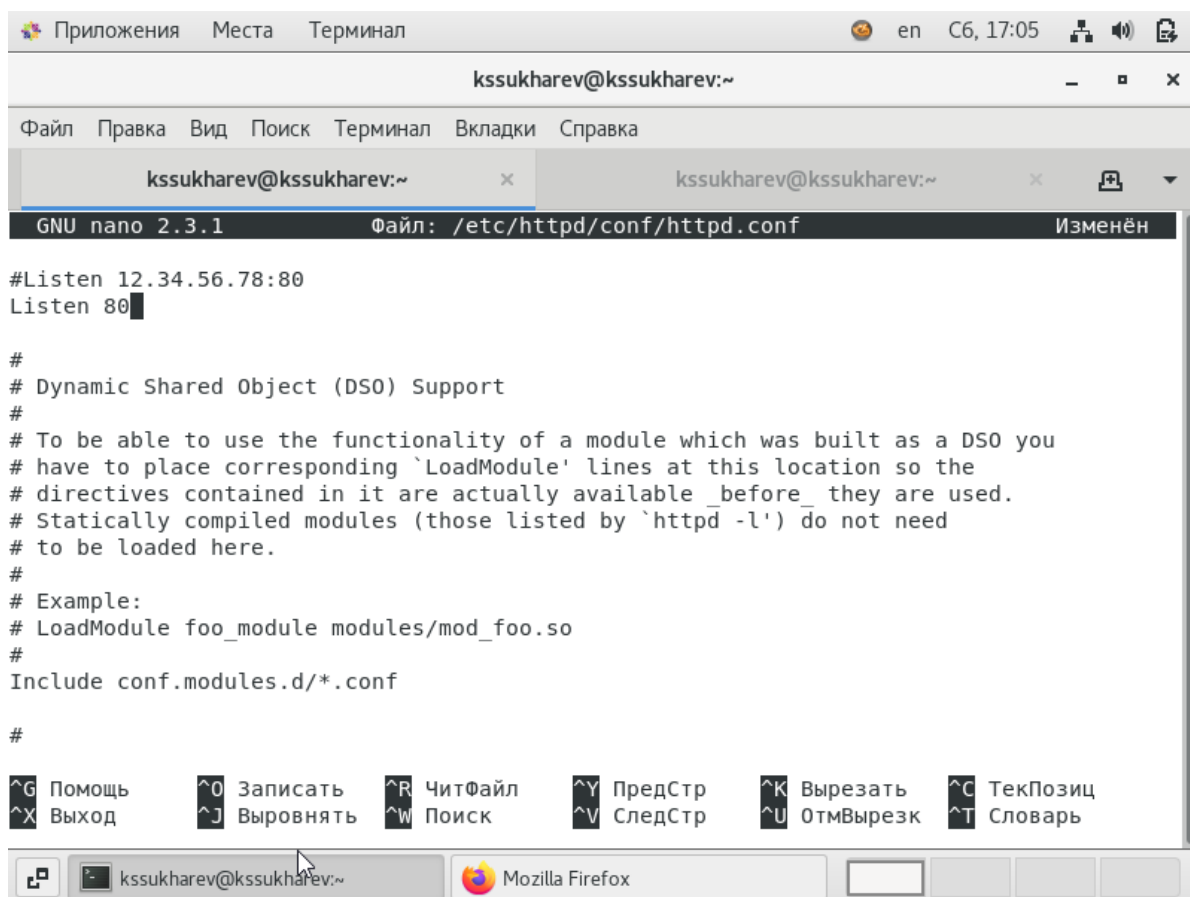


Figure 0.22: Возврат порта 80

23. Удалим привязку `http_port_t` к 81 порту командой `semanage port -d -t http_port_t -p tcp 81`. Сделать это не получится, поскольку порт определен политикой (fig. 0.23).

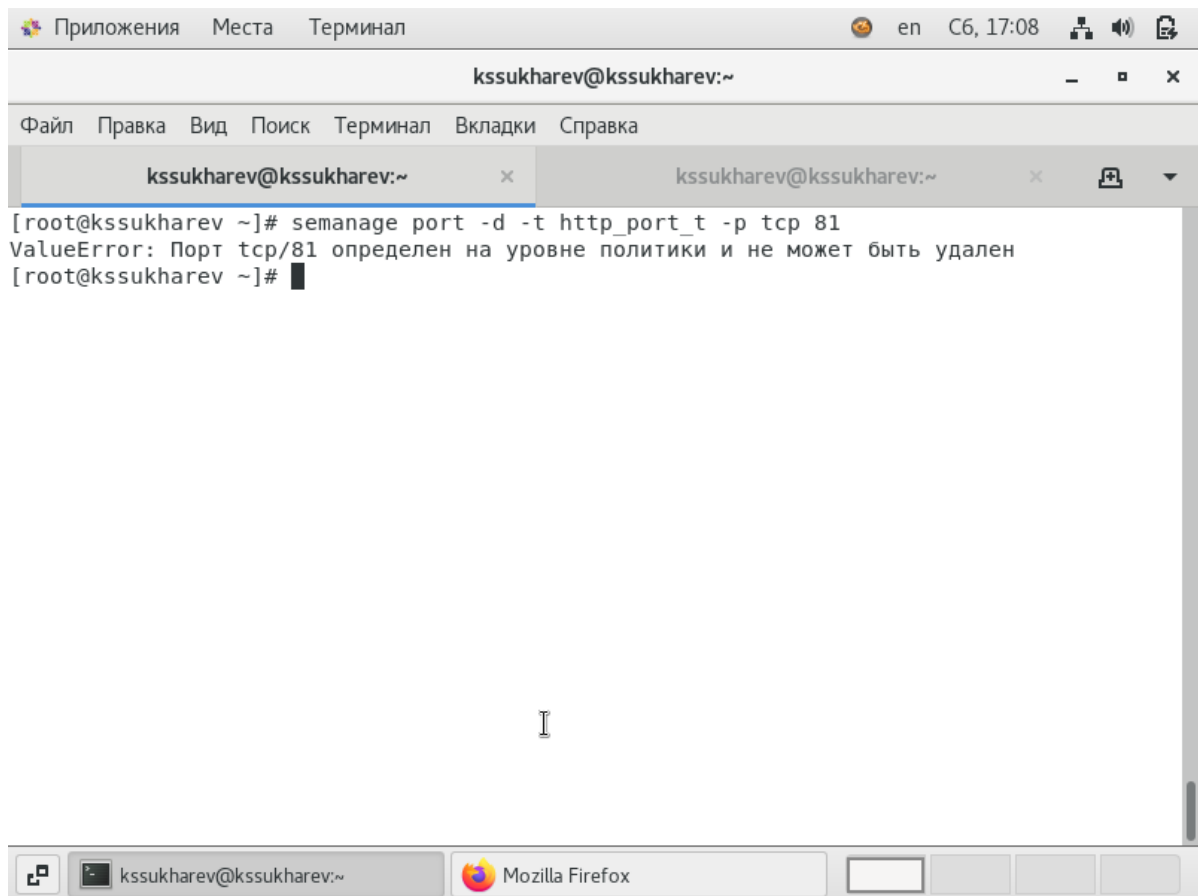


Figure 0.23: Удаление привязки

24. Удалим файл `/var/www/html/test.html` (fig. 0.24).

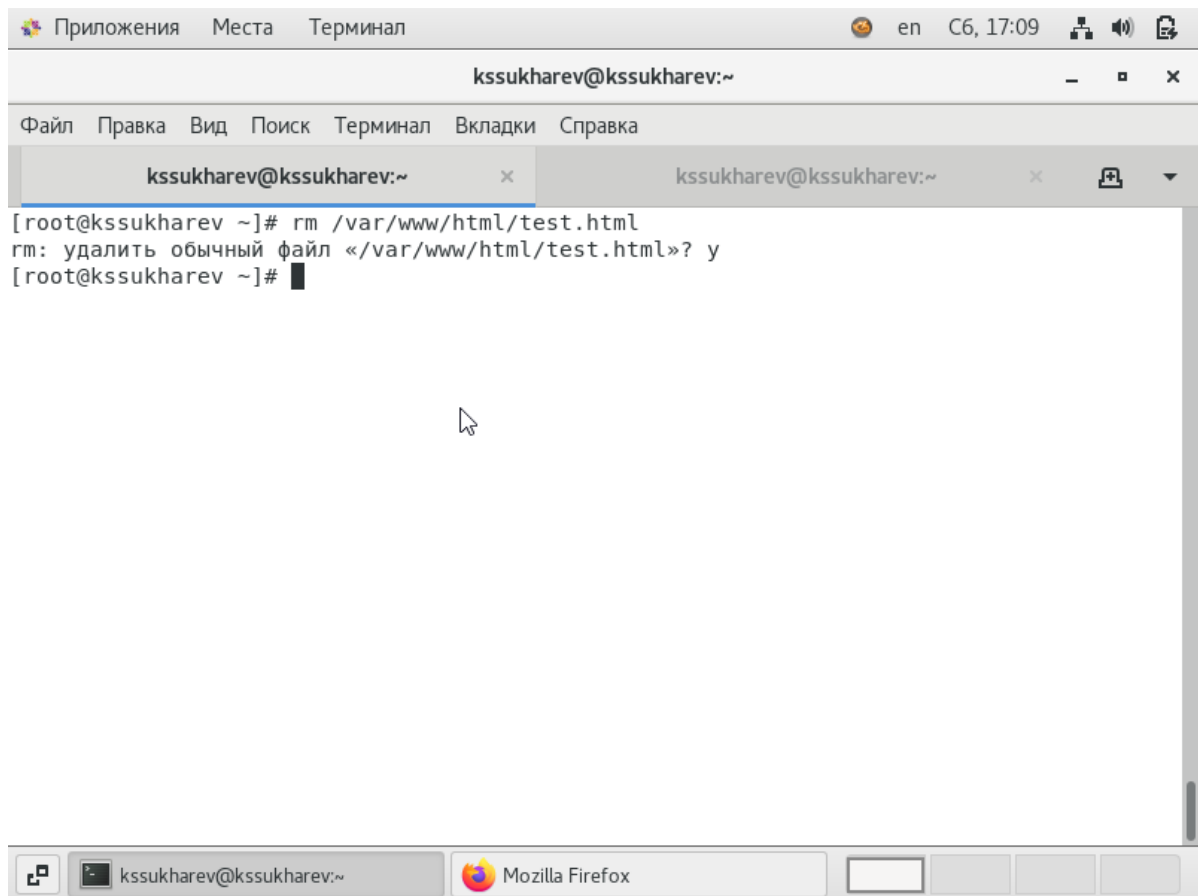


Figure 0.24: Удаление test.html

Выводы

Были развиты навыки администрирования ОС Linux, получено первое практическое знакомство с технологией SELinux. Также мы проверили работу SELinux на практике совместно с веб-сервером Apache.

Библиография

1. Мандатная модель управления доступом (MAC): обзор и применение в прикладных системах. URL: <https://habr.com/ru/company/avanpost/blog/482060/> (Дата обращения: 27.11.2021).
2. Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян. Информационная безопасность компьютерных сетей: лабораторные работы. // Факультет физико-математических и естественных наук. М.: РУДН, 2015. 64 с..