Отчёт по лабораторной работе 6

Мандатное разграничение прав в Linux

Гебриал Ибрам Есам Зекри НПИ-01-18

Содержание

# Цель работы

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

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

# Теоретические сведения

Linux с улучшенной безопасностью (SELinux) - это реализация мандатного управления доступом mandatory access control в ядре Linux, проверяющего разрешение операций после проверки стандартного дискреционного управления доступом discretionary access controls. SELinux создан Агенством Национальной Безопасности и вводит в действие правила для файлов и процессов в системе Linux, для совершаемых над ними действий, основываясь на установленной политике.

При использовании SELinux, файлы, включая директории и устройства являются объектами. Процессы, такие как, выполнение команды пользователем или приложение Mozilla® Firefox®, являются субъектами. Большинство операционных систем используют механизм дискретного контроля доступа (Discretionary Access Control (DAC), который контролирует каким образом субъекты взаимодействуют с объектами, и как субъекты взаимодействуют друг с другом. В операционных системах с использованием DAC, пользователи контролируют права доступа к файлам (объектам), для которых они являются собственниками. Например, в операционных системах Linux®, пользователи могут сделать свои домашние директории читаемыми для всех, предоставив пользователям и процессам (субъектам) доступ к потенциально конфеденциальной информации, без какой либо защиты от этого нежелательных действий. [1]

**Режимы работы SELinux**

SELinux имеет три основных режим работы, при этом по умолчанию установлен режим Enforcing. Это довольно жесткий режим, и в случае необходимости он может быть изменен на более удобный для конечного пользователя.

Enforcing: Режим по-умолчанию. При выборе этого режима все действия, которые каким-то образом нарушают текущую политику безопасности, будут блокироваться, а попытка нарушения будет зафиксирована в журнале.

Permissive: В случае использования этого режима, информация о всех действиях, которые нарушают текущую политику безопасности, будут зафиксированы в журнале, но сами действия не будут заблокированы.

Disabled: Полное отключение системы принудительного контроля доступа.[2]

# Выполнение лабораторной работы

**Подготовка лабораторного стенда**

1. При подготовке стенда обратил внимание, что необходимая для работы и указанная выше политика targeted и режим enforcing используются в данном дистрибутиве по умолчанию. (рис. 1)

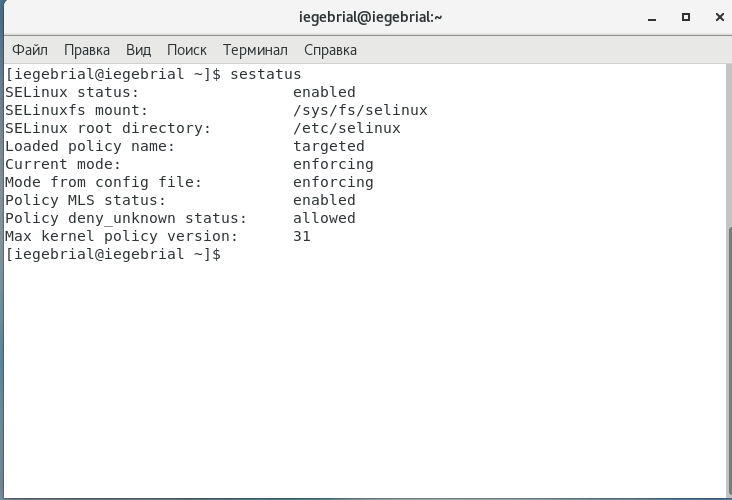


Figure 1: Проверка статус SELINUX

1. В конфигурационном файле /etc/httpd/httpd.conf задал параметр ServerName: (рис. 2)

Чтобы при запуске веб-сервера не выдавались лишние сообщения об ошибках, не относящихся к лабораторной работе.

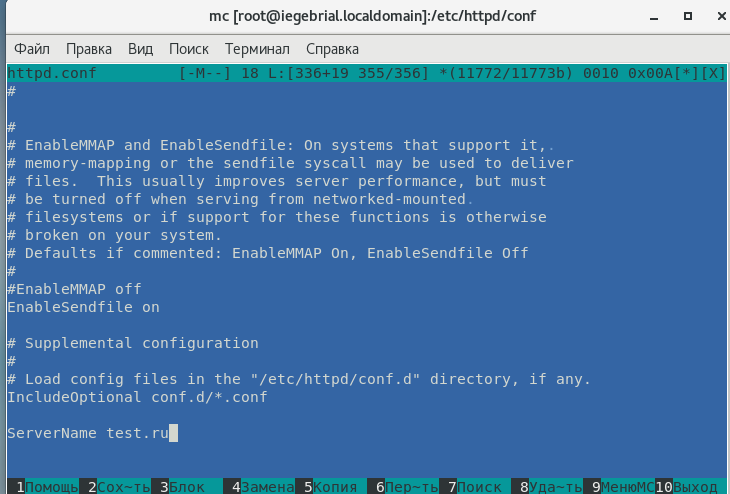


Figure 2: Установка параметра ServerName

1. Также необходимо проследить, чтобы пакетный фильтр был отключён или в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола tcp.

Отключил фильтр. (рис. 3)

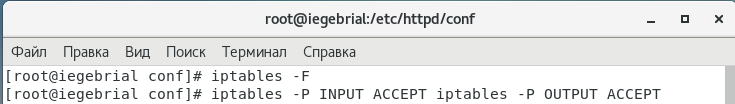


Figure 3: Отключение фильтра

1. Вошёл в систему с полученными учётными данными и убедитесь, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus. (рис. 4)

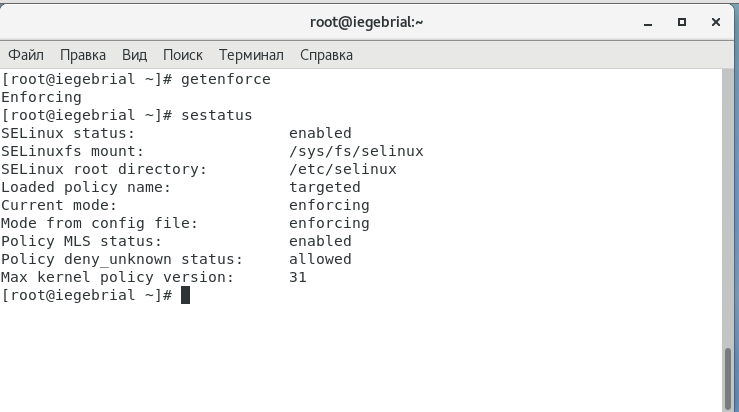


Figure 4: Проверка статуса

1. Обратился с помощью браузера к веб-серверу, запущенному на своем компьютере, и убедился, что последний работает: (рис. 5)

Он не работал, поэтому запустил его так же, но с параметром start.

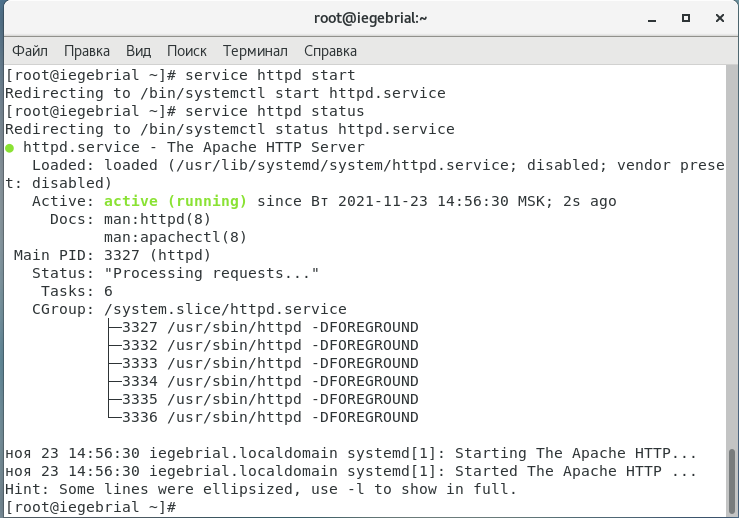


Figure 5: Запуск и проверка httpd

1. Нашёл веб-сервер Apache в списке процессов, определил его контекст безопасности. (рис. 6)

system\_u — системный пользователь;

system\_r — роль уровня системы, используемая для запуска системных процессов с указанием конкретного типа субъекта, определяемого типом объекта (файла).

httpd\_t- задан тип

s0- задан уровень

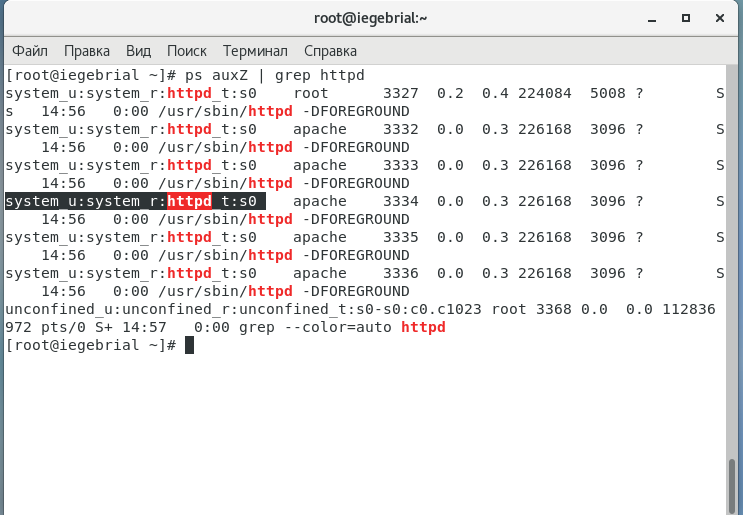


Figure 6: Контекст безопасности

1. Посмотрел текущее состояние переключателей SELinux для Apache. (рис. 7)

Можем заметить, что многие из них находятся в положении «off»

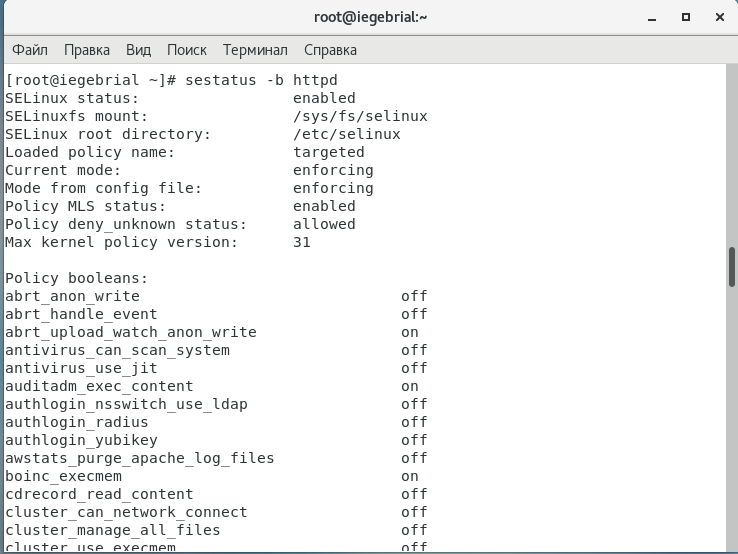


Figure 7: Просмотр состояния переключателей SELinux для Apache

1. Посмотрел статистику по политике с помощью команды seinfo (рис. 8)

пользователей 8

ролей 14

типов 4793

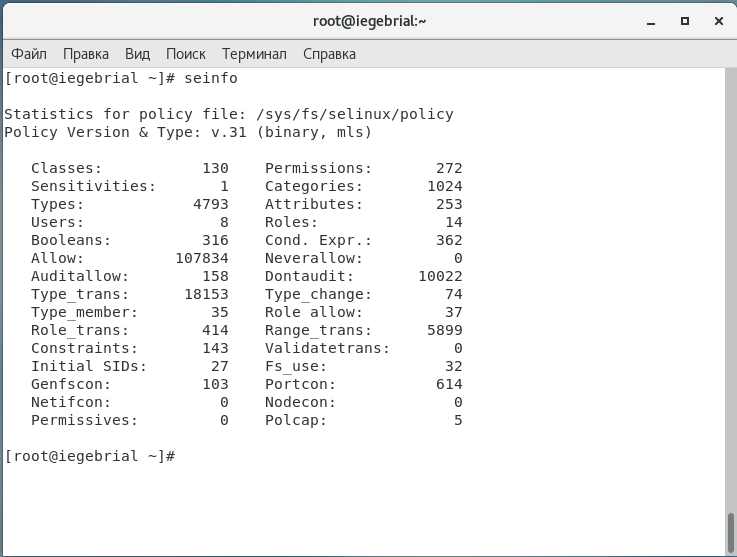


Figure 8: Просмотр статистики по политике с помощью команды seinfo

1. Определил тип файлов и поддиректорий, находящихся в директории /var/www (рис. 9)

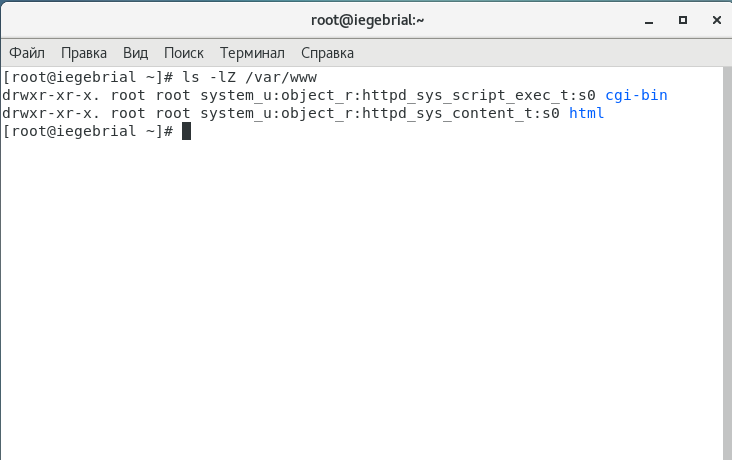


Figure 9: Определение типа файлов и поддиректорий, находящихся в директории /var/www

1. Создал от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания и проверял его конескст. (рис. 10) (рис. 11)

unconfined\_u — прочие пользователи;

object\_r — роль, указываемая для объектов типа файл или каталог;

httpd\_sys\_content\_t- тип

s0- уровень.

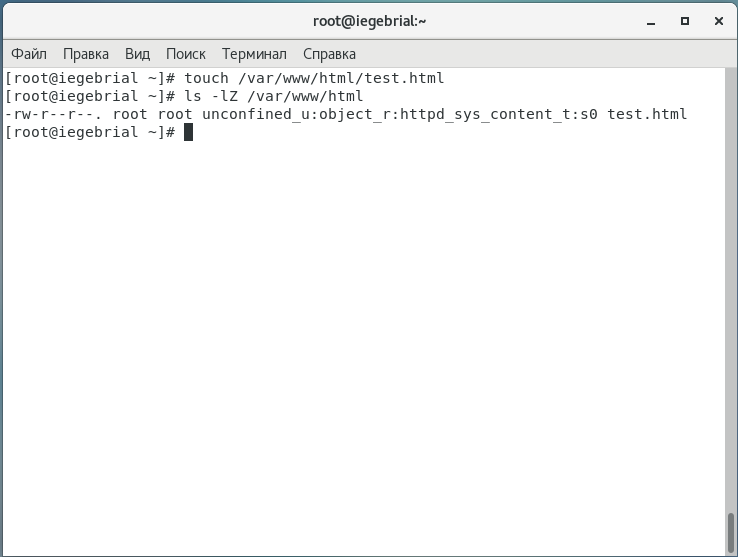


Figure 10: Создание html-файла и проверка его контекста

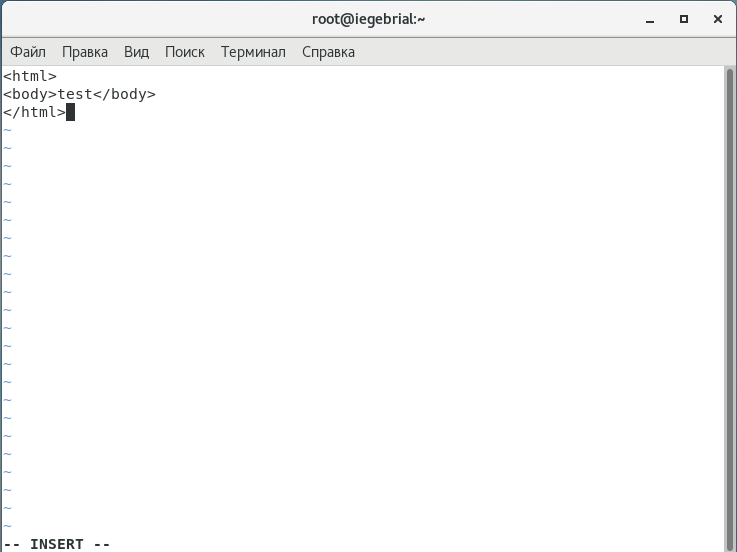


Figure 11: Содержание файла

1. Обратился к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html (рис. 12)

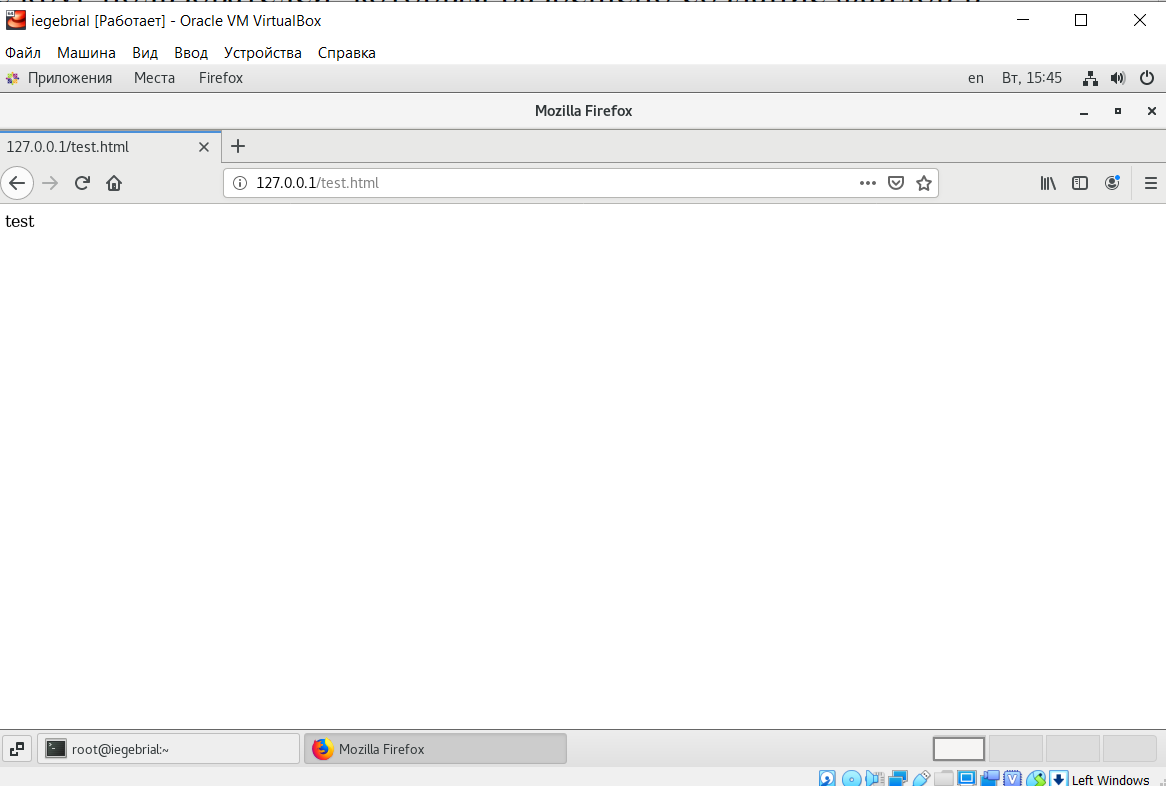


Figure 12: Проверка html-файла в браузере

1. Изменил 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: (рис. 13)

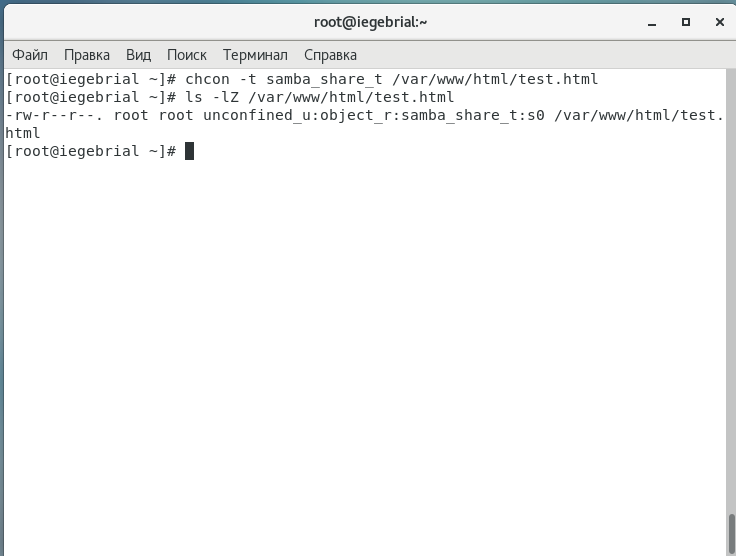


Figure 13: Изменение контекста файла /var/www/html/test.html и его проверка

1. Попробовал ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. (рис. 14)

Получил сообщение об ошибке: Forbidden You don’t have permission to access /test.html on this server.

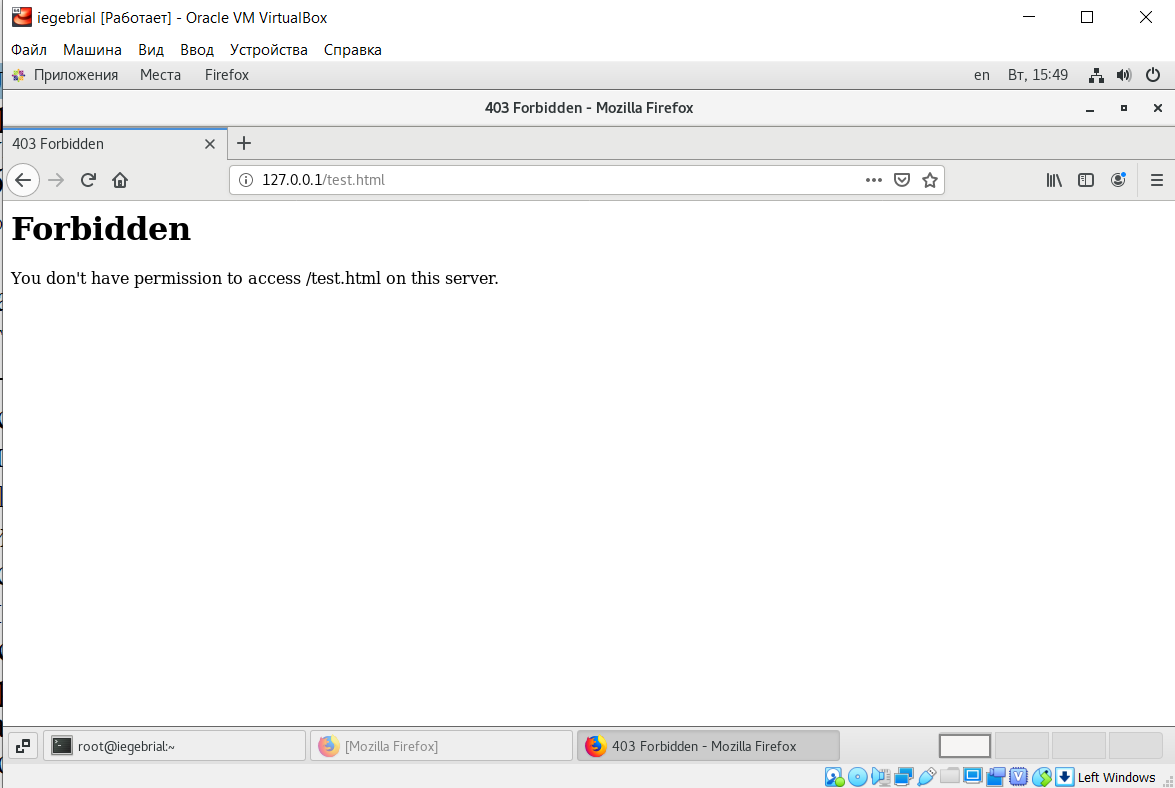


Figure 14: Проверка html-файла в браузере

1. Проанализировал ситуацию. (рис. 15) и просмотрел log-файлы веб-сервера Apache(рис. 16)

Заметил что не отображен потому что у httpd не доступа к типу файла. SeLinux запретил доступ из за разницы в контекстах.

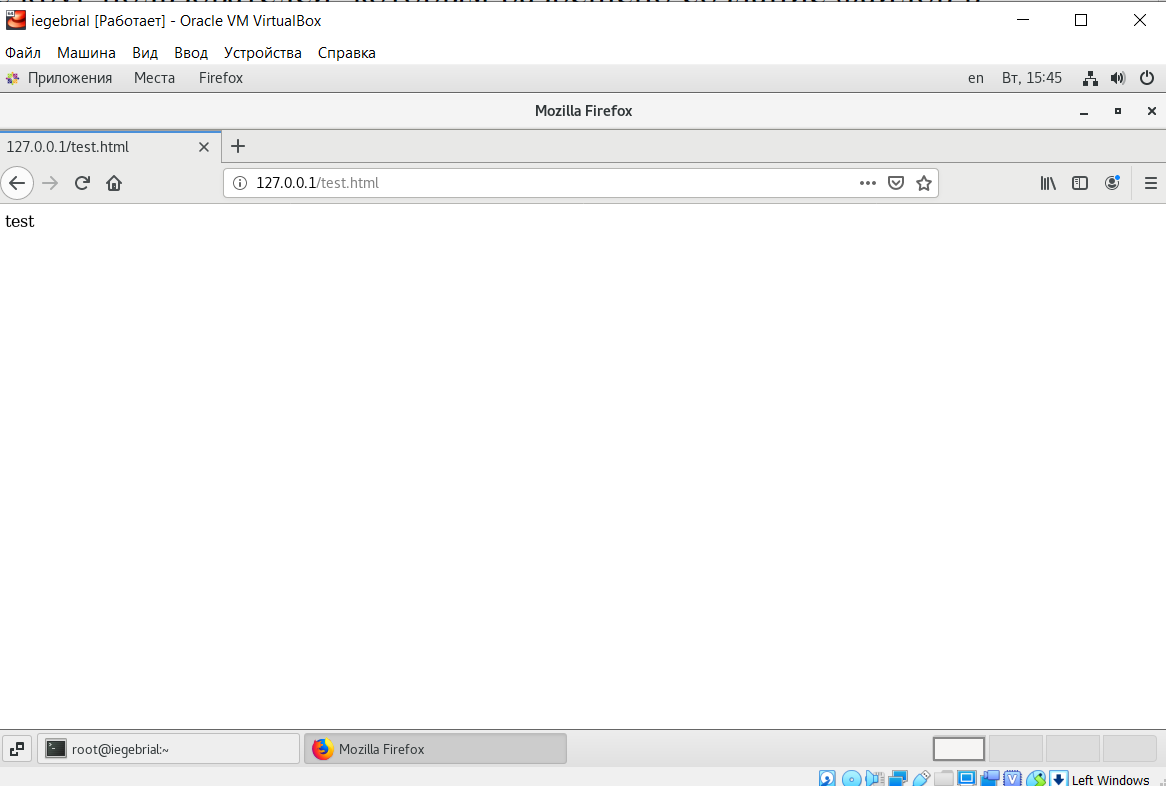


Figure 15: Анализ ситуации.

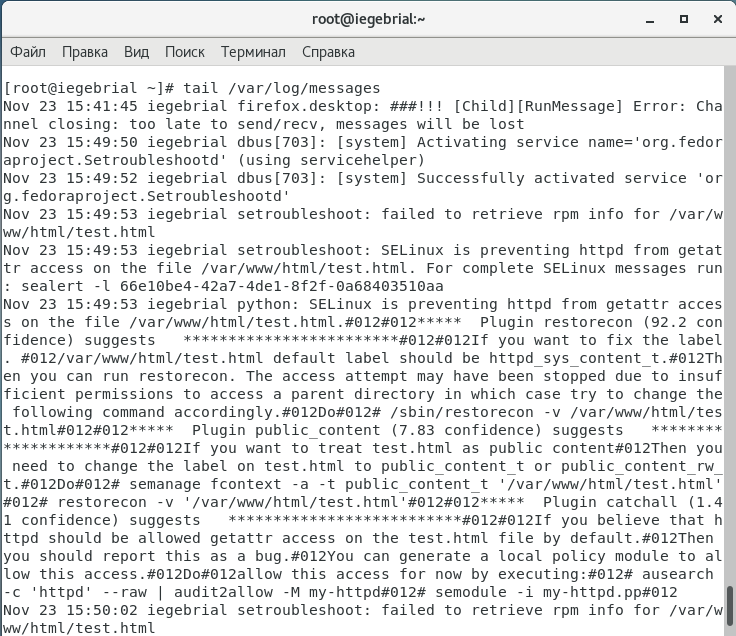


Figure 16: Проверка логи /var/log/messages

1. Попробовал 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/httpd.conf нашёл строчку Listen 80 и заменил её на Listen 81. (рис. 17)

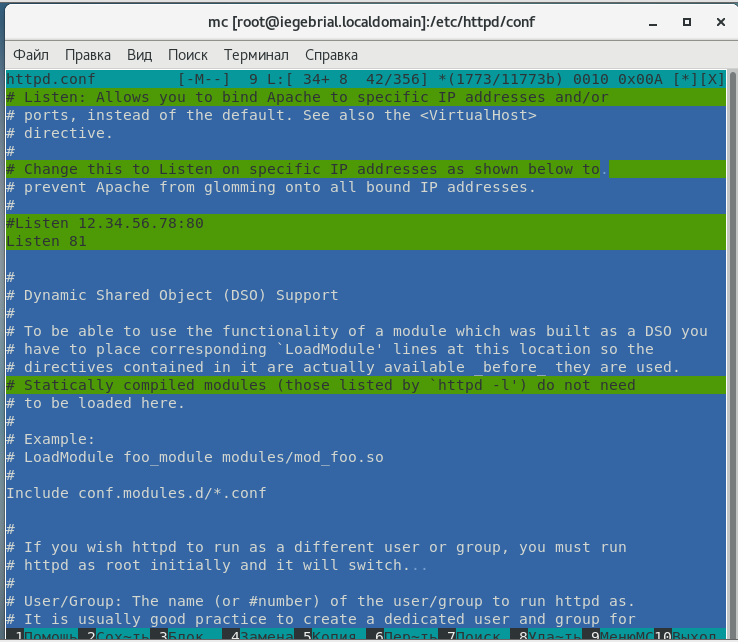


Figure 17: Замена порта прослушивание ТСР

1. Выполнил перезапуск веб-сервера Apache. (рис. 18)

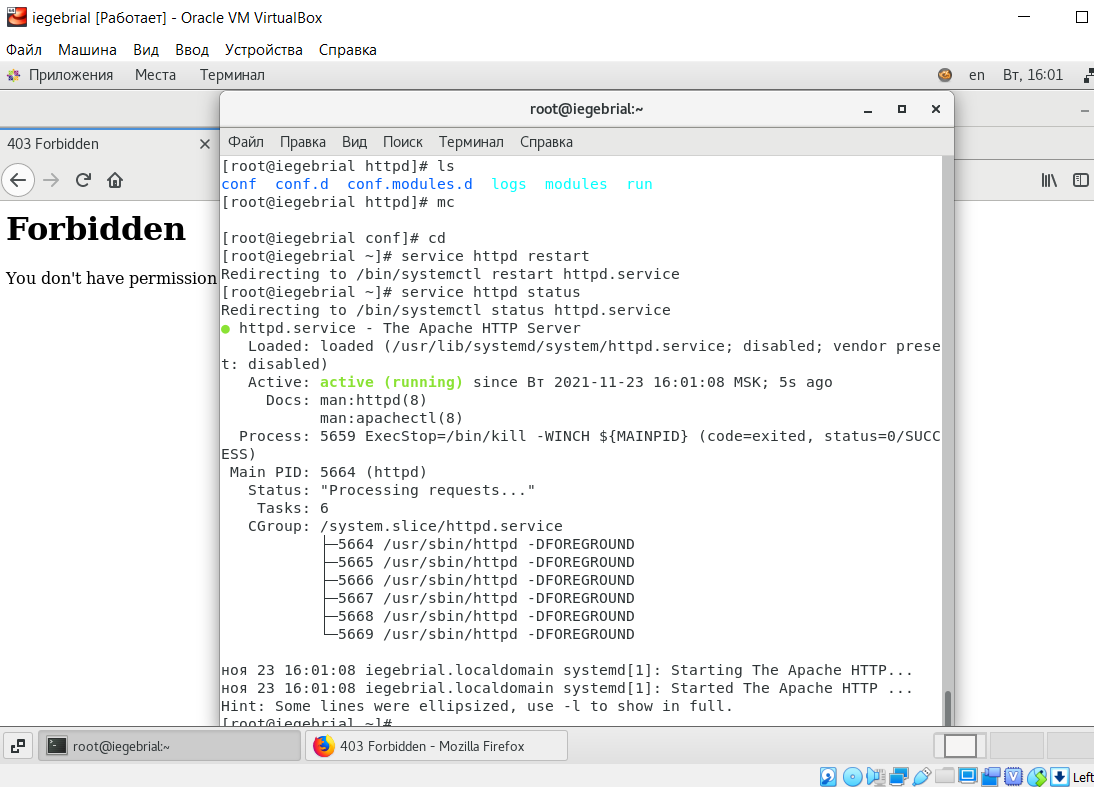


Figure 18: Выполнение перезапуска веб-сервера Apache

1. Проанализировал лог-файлы: (рис. 19)

Мы можем видить что нет ошибок.

Figure 19: Анализ лог-файлы

Figure 19: Анализ лог-файлы

1. Просмотел файлы /var/log/http/error\_log, /var/log/http/access\_log и /var/log/audit/audit.log (рис. 20)(рис. 21)(рис. 22)

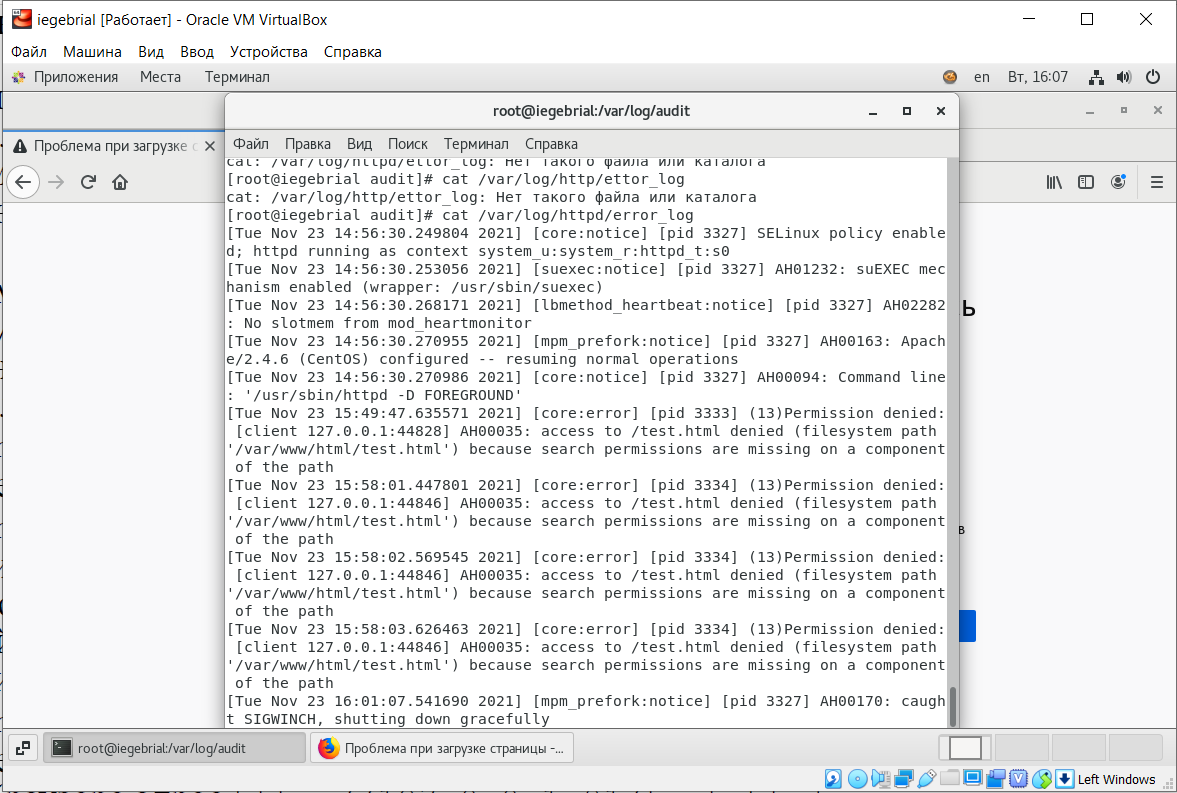


Figure 20: Просмотр файла /var/log/http/error\_log

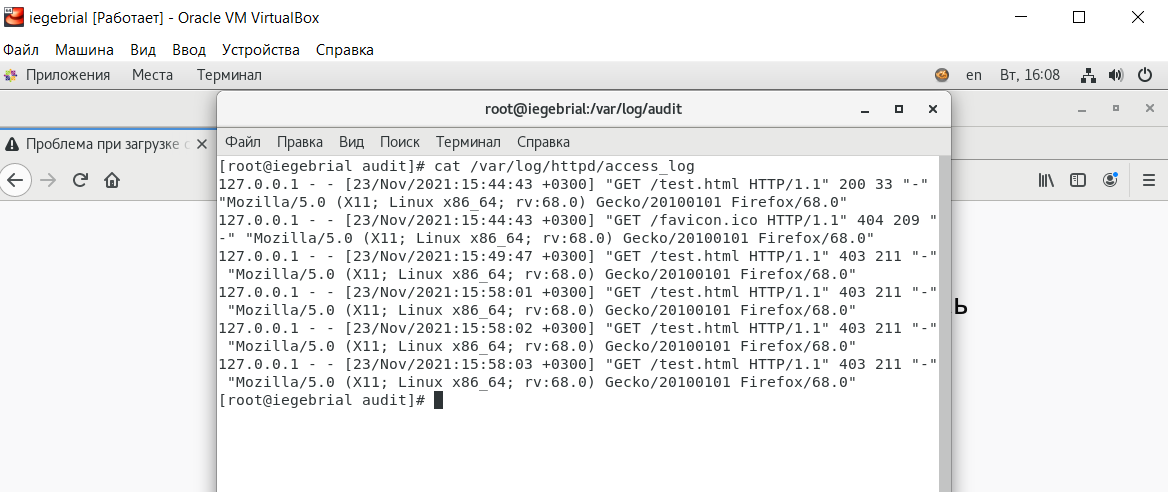


Figure 21: Просмотр файла /var/log/http/access\_log

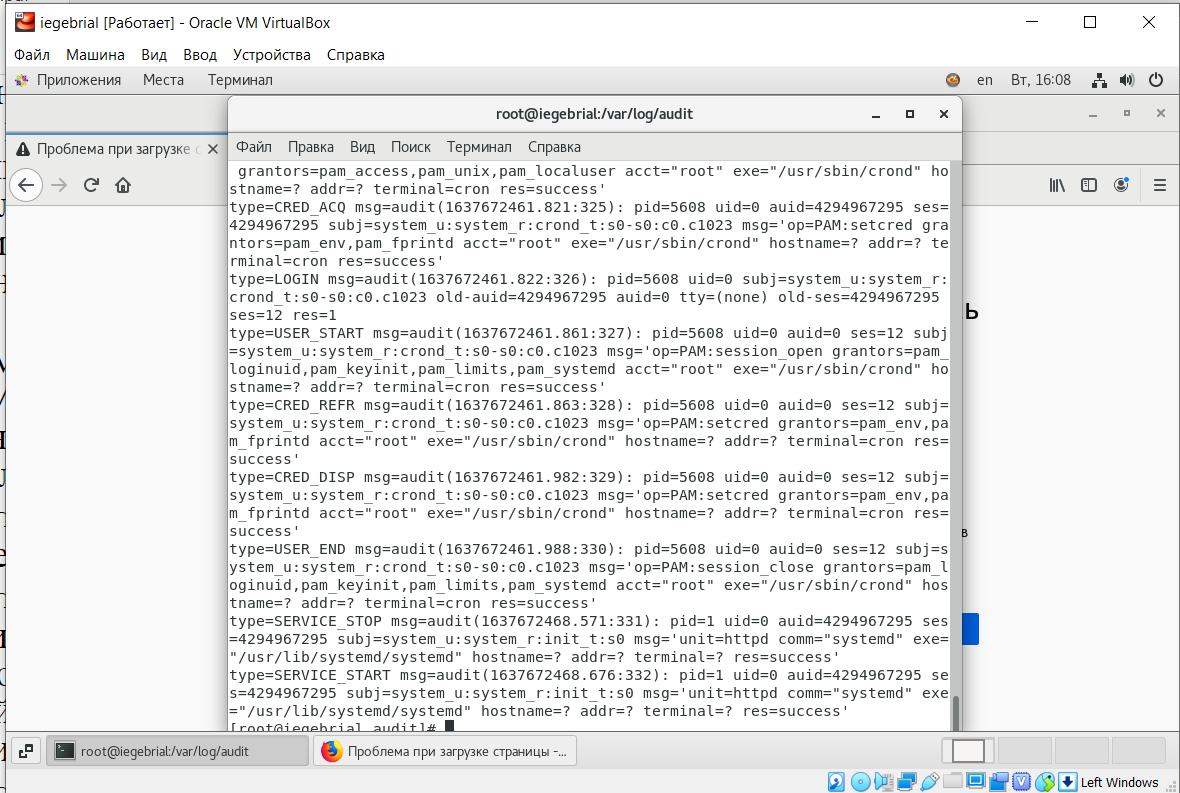


Figure 22: Просмотр файла /var/log/audit/audit.log

1. Выполнил команду semanage port -a -t http\_port\_t -р tcp 81. После этого проверил список портов командой (рис. 23)

Убедился, что порт 81 появился в списке.

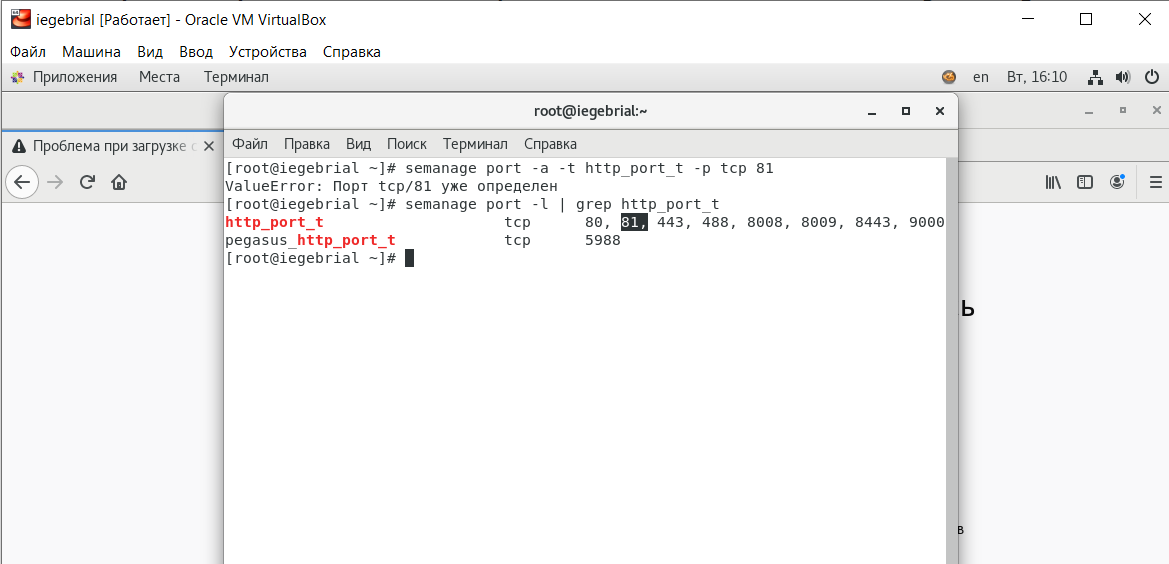


Figure 23: Добавление порта 81 и проверка список портов

1. Попробовал запустить веб-сервер Apache ещё раз. (рис. 24)

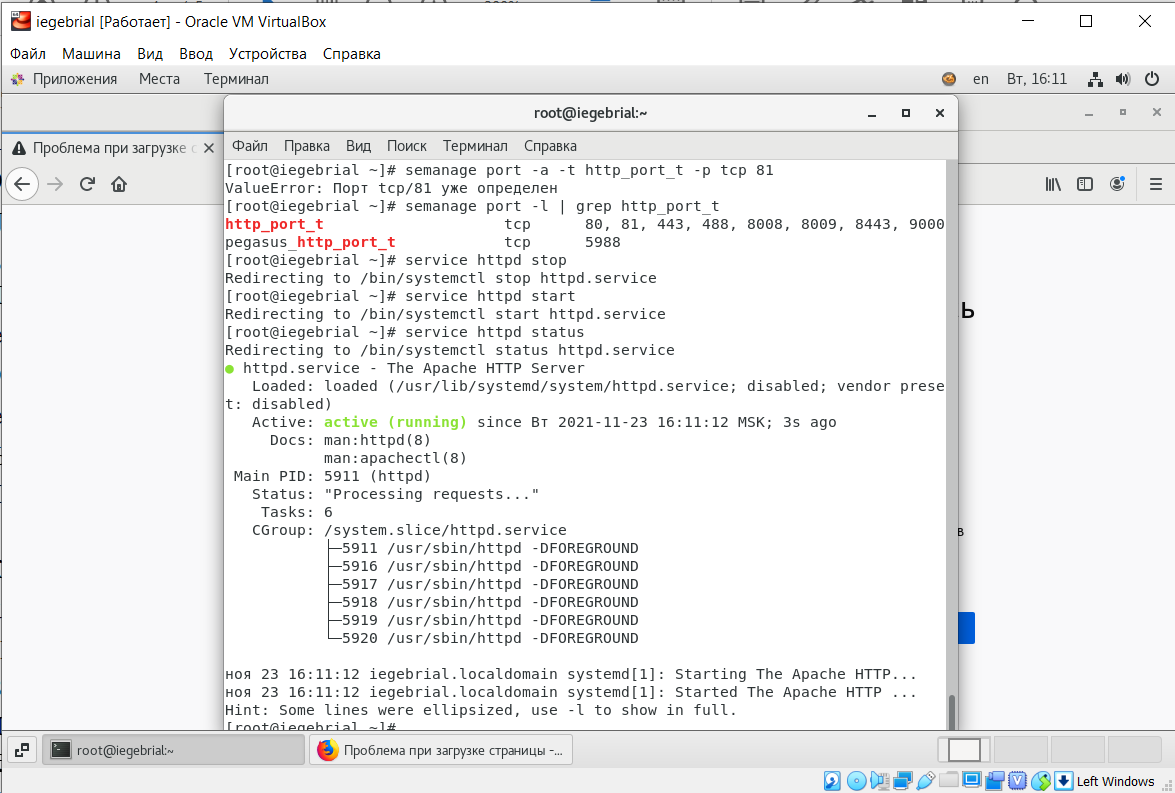


Figure 24: Попытка запуски веб-сервер Apache

1. Вернил контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html:(рис. 25)

chcon -t httpd\_sys\_content\_t /var/www/html/test.html.

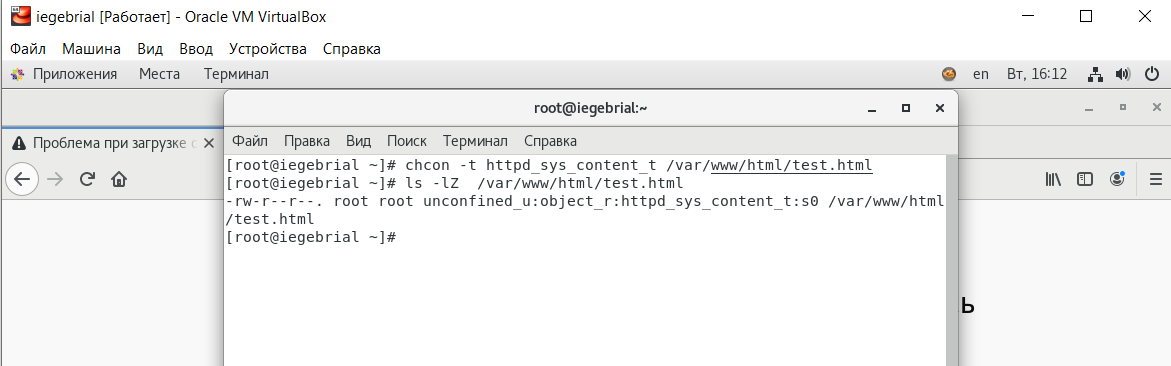


Figure 25: Возвращение конекста файла

1. Попробовал получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html. (рис. 26)

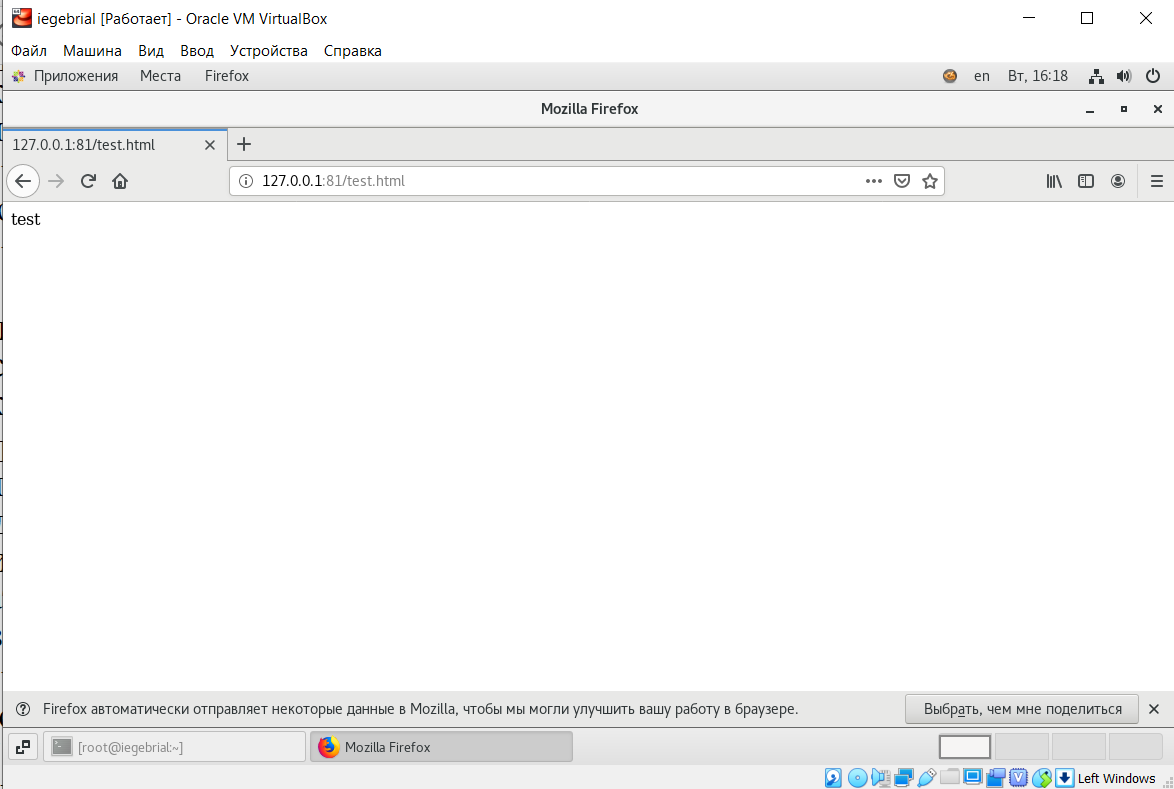


Figure 26: Попытка получения доступа к файлу через веб-сервер

1. Исправил обратно конфигурационный файл apache, вернув Listen 80. (рис. 27)

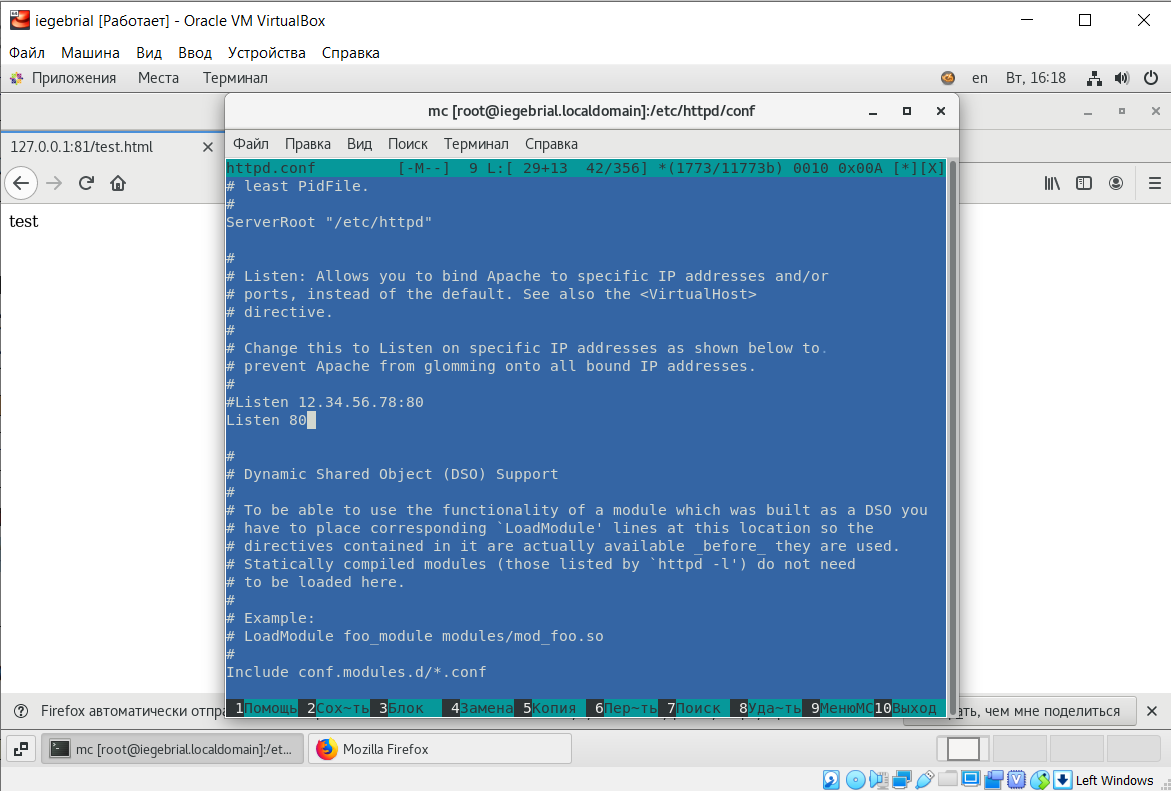


Figure 27: Исправления обратно конфигурационный файл apache

1. Удалил привязку http\_port\_t к 81 порту и удалил файл /var/www/html/test.html. (рис. 28)

Не получилось удалить привязку так как он определен на уровне политики.



Figure 28: Удаление привязки http\_port\_t к 81 порту и файла /var/www/html/test.html

# Выводы

Развил навыки администрирования ОС Linux. Получил первое практическое знакомство с технологией SELinux. Проверил работу SELinx на практике совместно с веб-сервером Apache.

# Список литературы

1. Security-Enhanced Linux. Linux с улучшенной безопасностью: руководство пользователя / M. McAllister, S. Radvan, D. Walsh, D. Grift, E. Paris, J. Morris. — URL: https://docs-old.fedoraproject.org/ru-RU/Fedora/13/html/Security-Enhanced\_Linux/index.html.
2. SELinux – описание и особенности работы с системой. Часть 1: автор / itNews. — URL: https://habr.com/ru/company/kingservers/blog/209644/. (дата обращения 20.01.2014).