Лабораторная работа № 6

Мандатное разграничение прав в Linux

Алибаева Данагуль НБибд-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.

# Задание

1. Войти в систему с полученными учётными данными и убедится, что SELinux работает в режиме enforcing политики targeted.
2. Обратится с помощью браузера к веб-серверу, запущенному на компьютере, и убедится, что последний работает. Если не работает, запустить его так же, но с параметром start.
3. Найти веб-сервер Apache в списке процессов, определить его контекст безопасности и занести эту информацию в отчёт.
4. Посмотреть текущее состояние переключателей SELinux для Apache. Обратить внимание, что многие из них находятся в положении «off».
5. Посмотреть статистику по политике с помощью команды seinfo, также определить множество пользователей, ролей, типов.
6. Определить тип файлов и поддиректорий, находящихся в директории /var/www.
7. Определить тип файлов, находящихся в директории /var/www/html.
8. Определить круг пользователей, которым разрешено создание файлов в директории /var/www/html.
9. Создать от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания: Test
10. Проверить контекст созданного файла. Занести в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html.
11. Обратиться к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Убедится, что файл был успешно отображён.
12. Изучить справку man httpd\_selinux и выяснить, какие контексты файлов определены для httpd. Сопоставить их с типом файла test.html.
13. Изменить 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. После этого проверить, что контекст поменялся.
14. Попробовать ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Должны получить сообщение об ошибке: Forbidden You don’t have permission to access /test.html on this server.
15. Проанализировать ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю? Просмотреть log-файлы веб-сервера Apache. Также просмотреть системный лог-файл. Если в системе окажутся запущенными процессы setroubleshootd и audtd, то можно увидеть ошибки, аналогичные указанным выше, в файле /var/log/audit/audit.log.
16. Попробовать 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 найти строчку Listen 80 и заменить её на Listen 81.
17. Выполнить перезапуск веб-сервера Apache. Произошёл сбой? Пояснить почему?
18. Проанализирвать лог-файлы. Просмотреть файлы /var/log/http/error\_log, /var/log/http/access\_log и /var/log/audit/audit.log и выяснить, в каких файлах появились записи.
19. Проверить список портов. Убедиться, что порт 81 появился в списке.
20. Попробовать запустить веб-сервер Apache ещё раз.
21. Вернуть контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html. После этого попробовать получить доступ к файлу через веб-сервер. Должны увидеть содержимое файла — слово «test».
22. Исправить обратно конфигурационный файл apache, вернув Listen 80.
23. Удалить привязку http\_port\_t к 81 порту и проверить, что порт 81 удалён.
24. Удалить файл /var/www/html/test.html.

# Теоретическое введение

В Linux дискреционные механизмы разграничения доступа (DAC, discretionary access control) являются основными и всегда активны. Их использование предполагает, что владельцы объектов правильно распоряжаются правами доступа к находящимся в их владении объектам. [1]

Например, пользовательские закрытые ключи, используемые службой W:[SSH], в каталоге ~/.ssh или ключи W:[GnuPG] в каталоге ~/.gnupg, и прочие секретные данные (подобные ключи доступа в банковские информационные системы)- должны быть недоступны никому, кроме их владельца. Запускаемые пользователем программы выполняются от лица запустившего их пользователя и имеют доступ к файлам согласно установленным режимам или спискам доступа. [1]

В примере из листинга ниже клиент ssh, браузер firefox и коммуникатор skype имеют абсолютно равные возможности по чтению и модификации пользовательского закрытого ключа ~/.ssh/id\_rsa, тогда как настоящим «владельцем» ключей является только ssh. [1]

Абсолютно естественно предполагать, что программы firefox и skype не имеют никаких намерений доступа к пользовательским ключам SSH. [1]

Можно даже доверять программе firefox, штатно установленной из доверенного источника (дистрибутива), где она была изготовлена из открытых исходных текстов, подлежащих верификации. Однако нет никаких оснований доверять закрытому skype, поставляемому в бинарном виде.[1]

Более того, предоставлять доступ программам firefox и skype к SSH-ключам пользователя нет никакой необходимости, во-первых, просто потому, что это выходит за рамки набора минимально необходимых условий их целевого функционирования. [1]

Во-вторых, практически в любой программе есть ошибки, используя которые злоумышленник может осуществлять непреднамеренные действия в свою пользу. Таким воздействиям особенно подвержены программы, использующие сетевой обмен с недоверенной внешней средой — клиенты и серверы сетевых служб операционной системы. [1]

Тем временем, дискреционный подход и механизмы служат для разграничения доступа разных пользователей к файлам, но никак не предназначены для разграничения доступа программ одного и того же пользователя к разным файлам этого пользователя. [1]

Для разграничения доступа субъектов — программ к объектам — файлам дерева каталогов используют так называемый мандатный (от англ, mandatory — обязательный или принудительный) подход (MAC, mandotary access control), предполагающий следование обязательным правилам доступа к файлам, назначаемым администраторами системы. [1]

Правила доступа строятся на основе знания о внутреннем устройстве программ и представляют собой описание набора минимально необходимых условий их целевого функционирования. [1]

Таким образом, в мандатных правилах, ограничивающих доступ к SSH-ключам пользователя, только программе ssh должен быть разрешен доступ для непосредственного выполнения своих прямых функций, а программам firefox и skype в доступе к SSH-ключам должно быть отказано. [1]

# Выполнение лабораторной работы

1.Лабораторная работа выполнялась дома со следующими характеристиками техники:

– Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz, 2304 МГц, ядер: 4, логических процессоров: 8

– ОС Майкрософт Windows 10 Pro

– VirtualBox верс. 6.1.26

1. Вошла в систему с полученными учётными данными и убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce (рис 1.1) и sestatus (рис 1.2).

Figure 1: 1.1. SELinux в режиме enforcing

Figure 1: 1.1. SELinux в режиме enforcing



Figure 2: 1.2. Политика targeted

1. Обратилась с помощью браузера к веб-серверу, запущенному на компьютере, и убедилась, что последний работает: service httpd status или /etc/rc.d/init.d/httpd status (рис 1.3), (рис 1.4).

Figure 3: 1.3. Обращение к веб-серверу

Figure 3: 1.3. Обращение к веб-серверу

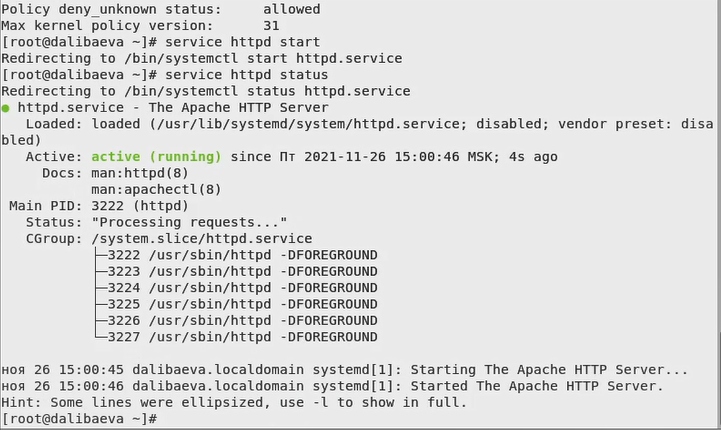


Figure 4: 1.4. Обращение к веб-серверу (2)

1. Нашла веб-сервер Apache в списке процессов, определила его контекст безопасности и занесла эту информацию в отчёт с помощью команды ps auxZ | grep httpd или ps -eZ | grep httpd (рис 1.5).

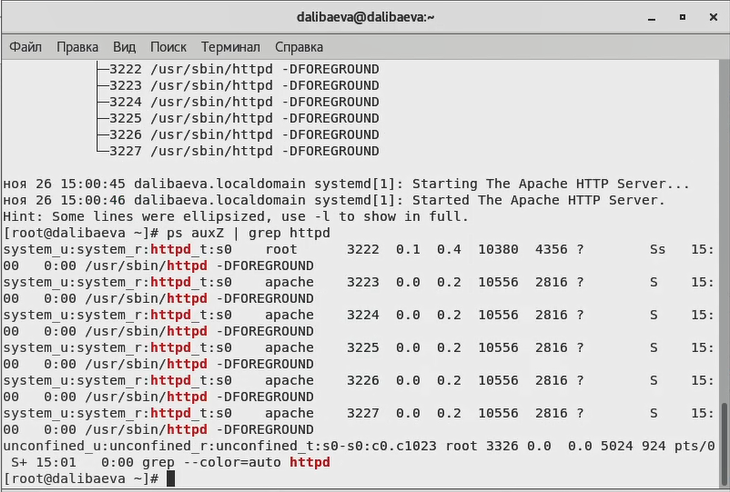


Figure 5: 1.5. Занесение контекста в отчет

1. Посмотрела текущее состояние переключателей SELinux для Apache с помощью команды sestatus -bigrep httpd. Обратила внимание, что многие из них находятся в положении «off» (рис 1.6).

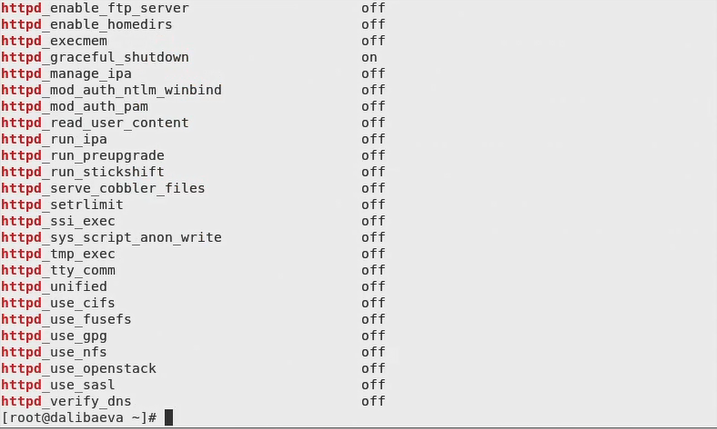


Figure 6: 1.6. Просмотр текущего состояния переключателей

1. Посмотрела статистику по политике с помощью команды seinfo, также определила множество пользователей, ролей, типов. (рис 1.7).

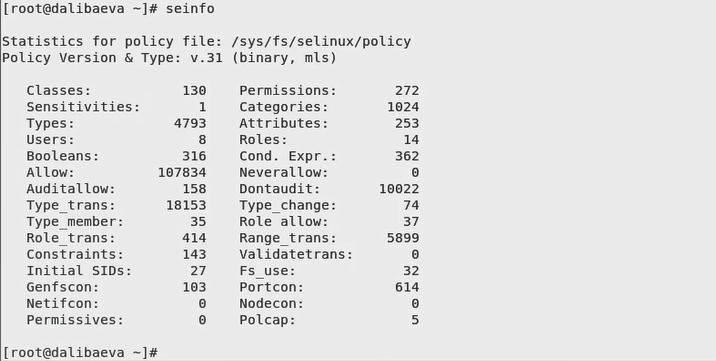


Figure 7: 1.7. Просмотр статистики

1. Определила тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды ls -lZ /var/www (рис 1.8).



Figure 8: 1.8. Определение типа файлов и поддиректорий

1. Определила тип файлов, находящихся в директории /var/www/html: ls -lZ /var/www/html (рис 1.9).



Figure 9: 1.9. Определение типа файлов

1. Создала от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html (рис 1.10)

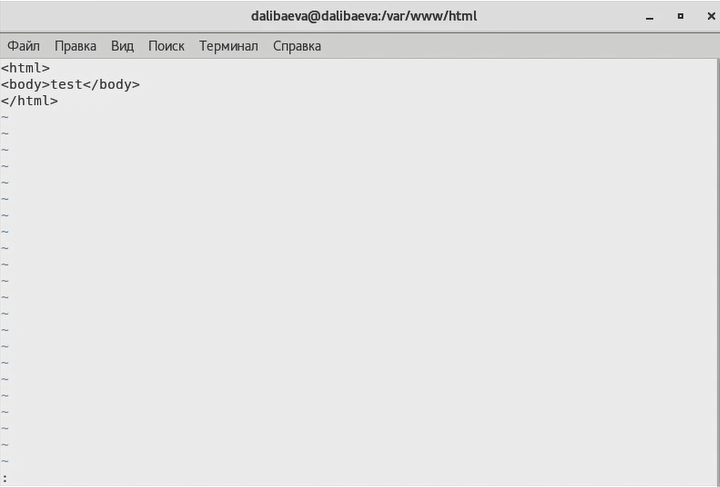


Figure 10: 1.10. Создание html-файла

1. Проверила контекст созданного файла. Занесла в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html. (рис 1.11).



Figure 11: 1.11. Проверка контекста созданного файла

1. Обратилась к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Убедилась, что файл был успешно отображён. (рис 1.12).

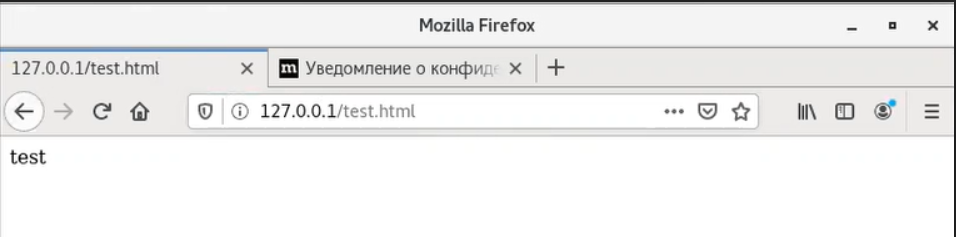


Figure 12: 1.12. Обращение к файлу через веб-сервер

1. Изучила справку man httpd\_selinux и выяснила, какие контексты файлов определены для httpd с помощью команды ls -Z. ls -Z /var/www/html/test.html. Сопоставила их с типом файла test.html (рис 1.13), (рис 1.14).

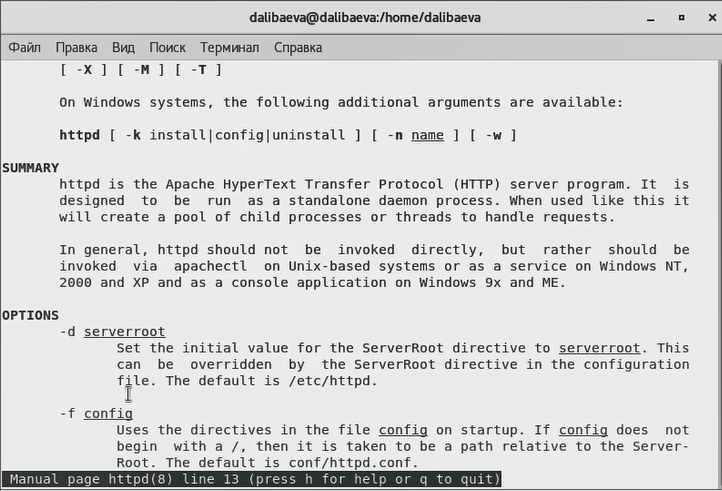


Figure 13: 1.13. Справка selinux

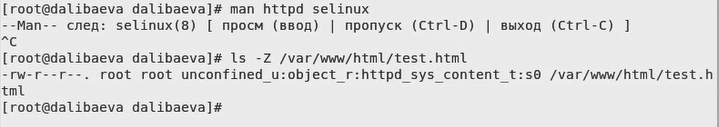


Figure 14: 1.14. Определение контекстов файлов

1. Изменила контекст файла /var/www/html/test.html с httpd\_sys\_content\_t на samba\_share\_t: chcon -t samba\_share\_t /var/www/html/test.html ls -Z /var/www/html/test.html (рис 1.15).



Figure 15: 1.15. Изменение контекста файла

1. Попробовала ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Получила сообщение об ошибке: Forbidden You don’t have permission to access /test.html on this server. (рис 1.16).

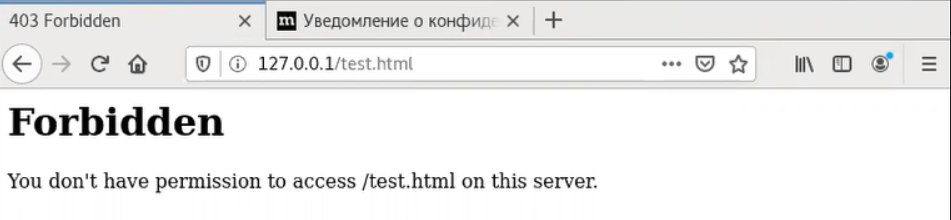


Figure 16: 1.16. Получение сообщения об ошибке

1. Проанализировала ситуацию. Просмотрела log-файлы веб-сервера Apache. Также просмотрела системный лог-файл: tail /var/log/messages. (рис 1.17), (рис 1.18).



Figure 17: 1.17. Просмотр системного лог-файла

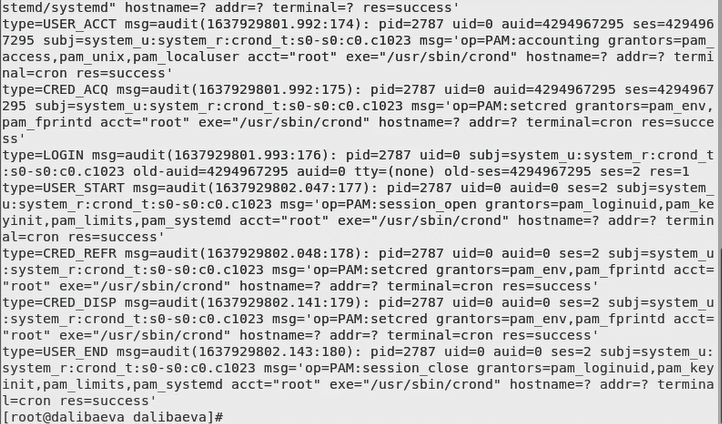


Figure 18: 1.18. Просмотр системного лог-файла (2)

1. Попробовала 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 нашла строчку Listen 80 и заменила её на Listen 81. (рис 1.19), (рис 1.20).

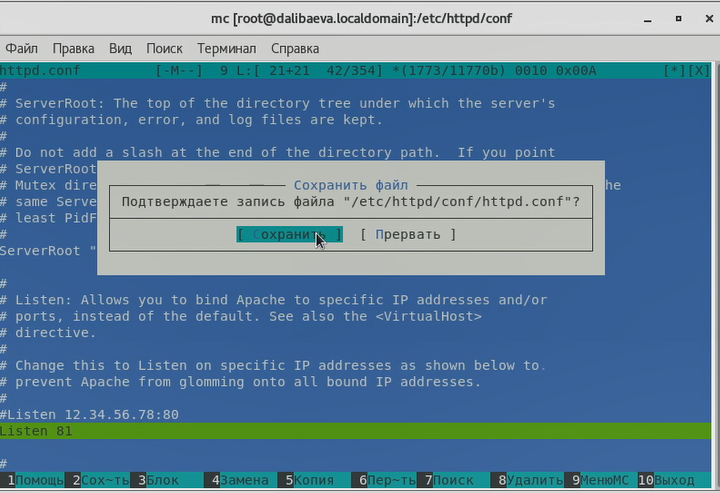


Figure 19: 1.19. Замена строчки Listen 80

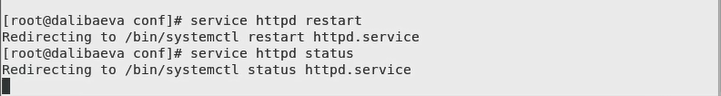


Figure 20: 1.20. Запуск веб-сервера Apache

1. Проанализировала лог-файлы: tail -nl /var/log/messages (рис 1.21). Просмотрела файлы /var/log/http/error\_log, /var/log/http/access\_log (рис 1.22) и /var/log/audit/audit.log и выяснила, в каких файлах появились записи. (рис 1.23).

Figure 21: 1.21. Анализ лог-файла messages

Figure 21: 1.21. Анализ лог-файла messages

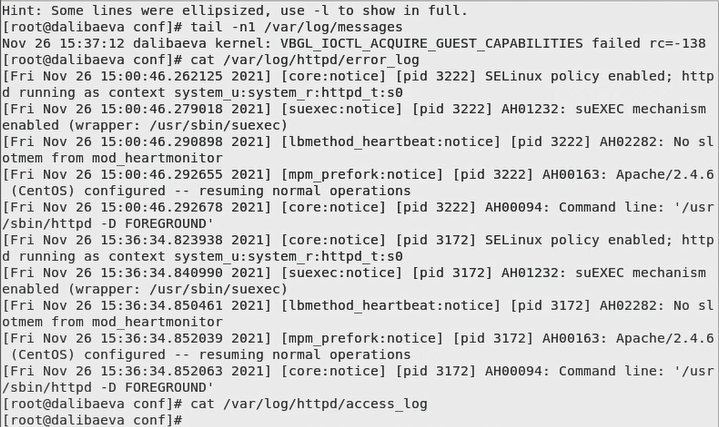


Figure 22: 1.22. Просмотр файлов error\_log и access\_log

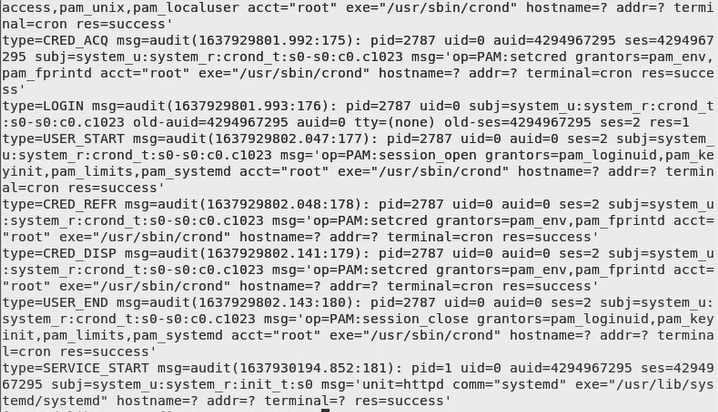


Figure 23: 1.23. Просмотр файла audit.log

1. Выполнила команду semanage port -a -t http\_port\_t -р tcp 81 (рис 1.24). После этого проверила список портов командой semanage port -l | grep http\_port\_t. Убедилась, что порт 81 появился в списке. (рис 1.25).

Figure 24: 1.24. Выполнение команды semanage

Figure 24: 1.24. Выполнение команды semanage

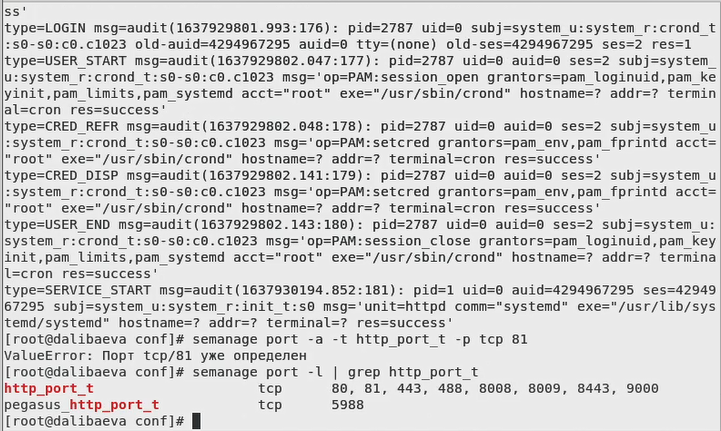


Figure 25: 1.25. Проверка списка портов

1. Вернула контекст httpd\_sys\_cоntent\_\_t к файлу /var/www/html/ test.html: chcon -t httpd\_sys\_content\_t /var/www/html/test.html (рис 1.26). После этого попробовала получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html (рис 1.27).

Figure 26: 1.26. Возвращение контекста

Figure 26: 1.26. Возвращение контекста

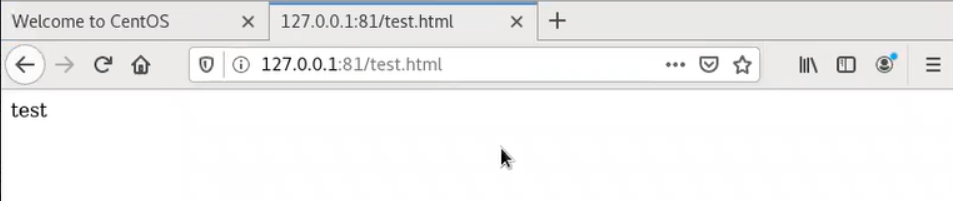


Figure 27: 1.27. Получение доступа к веб-серверу

1. Исправила обратно конфигурационный файл apache, вернув Listen 80 (рис 1.28).

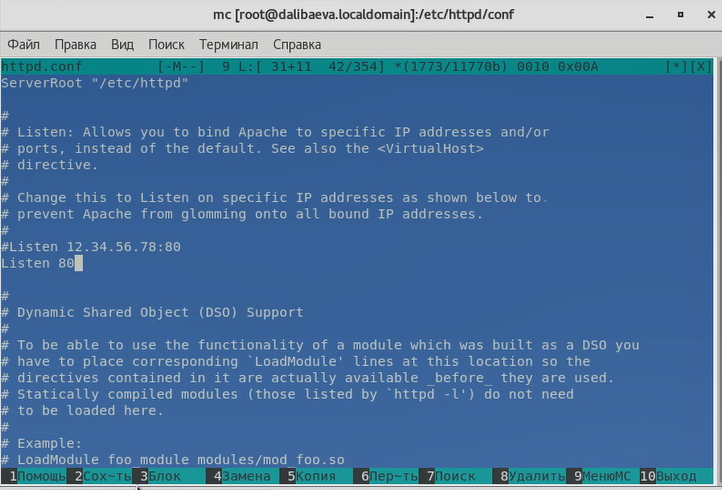


Figure 28: 1.28. Исправление конфигурационного файла

1. Удалила привязку http\_port\_t к 81 порту: semanage port -d -t http\_port\_t -p tcp 81 и проверила, что порт 81 удалён. (рис 1.29).

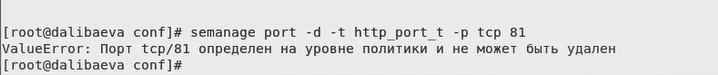


Figure 29: 1.29. Удаление привязки к порту

1. Удалила файл /var/www/html/test.html: rm /var/www/html/test.html (рис 1.30).

Figure 30: 1.30. Удаление файла

Figure 30: 1.30. Удаление файла

# Выводы

Развила навыки администрирования ОС Linux. Получила первое практическое знакомство с технологией SELinux. Проверила работу SELinx на практике совместно с веб-сервером Apache.

# Список литературы

1. Мандатное разграничение прав в Linux// URL: https://debianinstall.ru/mandatnoe-prinuditelnoe-razgranichenie-dostupa-linux/ (дата обращения: 26.11.2021).