Лабораторная работа № 6

Мандатное разграничение прав в Linux

Пак Мария НБИбд-01-18 1032184261"

Содержание

# Лабораторная работа №6

[Лабораторная работа №6](#лабораторная-работа-№6)  
 [Цель работы](#цель-работы)  
 [Задание](#задание)  
 [Теоретическое введение](#теоретическое-введение)  
 [Оборудование](#оборудование)  
[Выполнение лабораторной работы](#выполнение-лабораторной-работы)  
 [Выводы](#выводы)  
 [Список литературы](#список-литературы)

## Цель работы

Развить навыки администрирования ОС Linux. Получить первое прак-  
тическое знакомство с технологией SELinux1.  
Проверить работу SELinx на практике совместно с веб-сервером  
Apache.

## Задание

1. Войдите в систему с полученными учётными данными и убедитесь, что  
   SELinux работает в режиме enforcing политики targeted с помощью ко-  
   манд getenforce и sestatus.
2. Обратитесь с помощью браузера к веб-серверу, запущенному на вашем  
   компьютере, и убедитесь, что последний работает:  
   service httpd status  
   или  
   /etc/rc.d/init.d/httpd status  
   Если не работает, запустите его так же, но с параметром start.
3. Найдите веб-сервер Apache в списке процессов, определите его контекст  
   безопасности и занесите эту информацию в отчёт. Например, можно ис-  
   пользовать команду  
   ps auxZ | grep httpd  
   или  
   ps -eZ | grep httpd
4. Посмотрите текущее состояние переключателей SELinux для Apache с  
   помощью команды  
   sestatus -bigrep httpd  
   Обратите внимание, что многие из них находятся в положении «off»
5. Посмотрите статистику по политике с помощью команды seinfo, также  
   определите множество пользователей, ролей, типов.
6. Определите тип файлов и поддиректорий, находящихся в директории  
   /var/www, с помощью команды  
   ls -lZ /var/www
7. Определите тип файлов, находящихся в директории /var/www/html:  
   ls -lZ /var/www/html
8. Определите круг пользователей, которым разрешено создание файлов в  
   директории /var/www/html.
9. Создайте от имени суперпользователя (так как в дистрибутиве по-  
   сле установки только ему разрешена запись в директорию) html-файл  
   /var/www/html/test.html следующего содержания:  
     
   test
10. Проверьте контекст созданного вами файла. Занесите в отчёт контекст,  
    присваиваемый по умолчанию вновь созданным файлам в директории  
    /var/www/html.
11. Обратитесь к файлу через веб-сервер, введя в браузере адрес  
    <http://127.0.0.1/test.html>. Убедитесь, что файл был успеш-  
    но отображён.
12. Изучите справку man httpd\_selinux и выясните, какие контек-  
    сты файлов определены для httpd. Сопоставьте их с типом файла  
    test.html. Проверить контекст файла можно командой ls -Z.  
    ls -Z /var/www/html/test.html  
    Рассмотрим полученный контекст детально. Обратите внимание, что так  
    как по умолчанию пользователи CentOS являются свободными от типа  
    (unconfined в переводе с англ. означает свободный), созданному нами  
    файлу test.html был сопоставлен SELinux, пользователь unconfined\_u.  
    Это первая часть контекста.  
    Далее политика ролевого разделения доступа RBAC используется про-  
    цессами, но не файлами, поэтому роли не имеют никакого значения для  
    файлов. Роль object\_r используется по умолчанию для файлов на «по-  
    стоянных» носителях и на сетевых файловых системах. (В директории  
    /ргос файлы, относящиеся к процессам, могут иметь роль system\_r.  
    Если активна политика MLS, то могут использоваться и другие роли,  
    например, secadm\_r. Данный случай мы рассматривать не будем, как и  
    предназначение :s0).  
    Тип httpd\_sys\_content\_t позволяет процессу httpd получить доступ к фай-  
    лу. Благодаря наличию последнего типа мы получили доступ к файлу  
    при обращении к нему через браузер.
13. Измените 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.После этого проверьте, что контекст поменялся.
14. Попробуйте ещё раз получить доступ к файлу через веб-сервер, введя в  
    браузере адрес <http://127.0.0.1/test.html>. Вы должны получить  
    сообщение об ошибке:  
    Forbidden  
    You don't have permission to access /test.html on this server.
15. Проанализируйте ситуацию. Почему файл не был отображён, если права  
    доступа позволяют читать этот файл любому пользователю?  
    ls -l /var/www/html/test.html  
    Просмотрите log-файлы веб-сервера Apache. Также просмотрите си-  
    стемный лог-файл:  
    tail /var/log/messages  
    Если в системе окажутся запущенными процессы setroubleshootd и  
    audtd, то вы также сможете увидеть ошибки, аналогичные указанным  
    выше, в файле /var/log/audit/audit.log. Проверьте это утвержде-  
    ние самостоятельно.
    1. Попробуйте 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 найдите строчку Listen 80 и  
       замените её на Listen 81.
    2. Выполните перезапуск веб-сервера Apache. Произошёл сбой? Поясните  
       почему?
    3. Проанализируйте лог-файлы:  
       tail -nl /var/log/messages  
       Просмотрите файлы /var/log/http/error\_log,  
       /var/log/http/access\_log и /var/log/audit/audit.log и  
       выясните, в каких файлах появились записи.
    4. Выполните команду  
       semanage port -a -t http\_port\_t -р tcp 81  
       После этого проверьте список портов командой  
       semanage port -l | grep http\_port\_t  
       Убедитесь, что порт 81 появился в списке.
    5. Попробуйте запустить веб-сервер Apache ещё раз. Поняли ли вы, почему  
       он сейчас запустился, а в предыдущем случае не смог?
    6. Верните 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 попробуйте получить доступ к файлу через веб-сервер, вве-  
       дя в браузере адрес <http://127.0.0.1:81/test.html>.  
       Вы должны увидеть содержимое файла — слово «test».
    7. Исправьте обратно конфигурационный файл apache, вернув Listen 80.
    8. Удалите привязку http\_port\_t к 81 порту:  
       semanage port -d -t http\_port\_t -p tcp 81  
       и проверьте, что порт 81 удалён.
    9. Удалите файл /var/www/html/test.html:  
       rm /var/www/html/test.html

## Теоретическое введение

В операционной системе Linux есть много отличных функций безопасности, но она из самых важных - это система прав доступа к файлам. Linux, как последователь идеологии ядра Linux в отличие от Windows, изначально проектировался как многопользовательская система, поэтому права доступа к файлам в linux продуманы очень хорошо. И это очень важно, потому что локальный доступ к файлам для всех программ и всех пользователей позволил бы вирусам без проблем уничтожить систему [[1]](#X9a11f83b1ef3906df5df95cff9123f28b5a945b).

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. [[2]](#X9a11f83b1ef3906df5df95cff9123f28b5a945b).

По сути, в традиционной модели избирательного управления доступом (DAC), хорошо реализованы только два уровня доступа — пользователь и суперпользователь. Нет простого метода, который позволил бы устанавливать для каждого пользователя необходимый минимум привилегий.  
Конечно, есть множество методов обхода этих проблем в рамках классической модели безопасности, но ни один из них не является универсальным.

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. [[3]](#X9a11f83b1ef3906df5df95cff9123f28b5a945b)

**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. [[3]](#X9a11f83b1ef3906df5df95cff9123f28b5a945b)

**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. [[3]](#X9a11f83b1ef3906df5df95cff9123f28b5a945b)

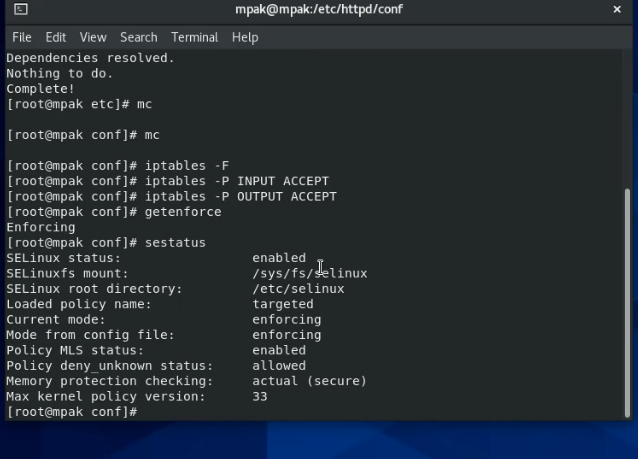
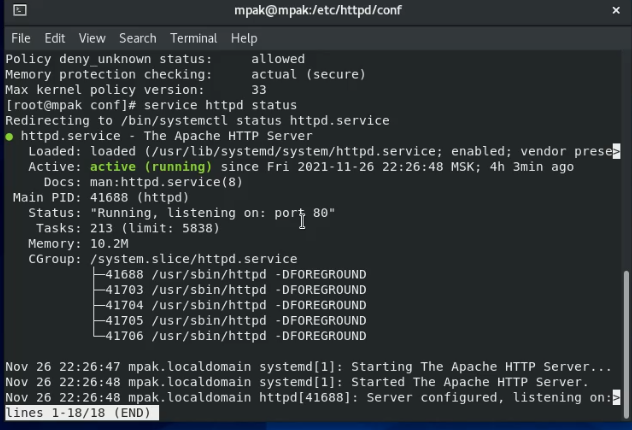
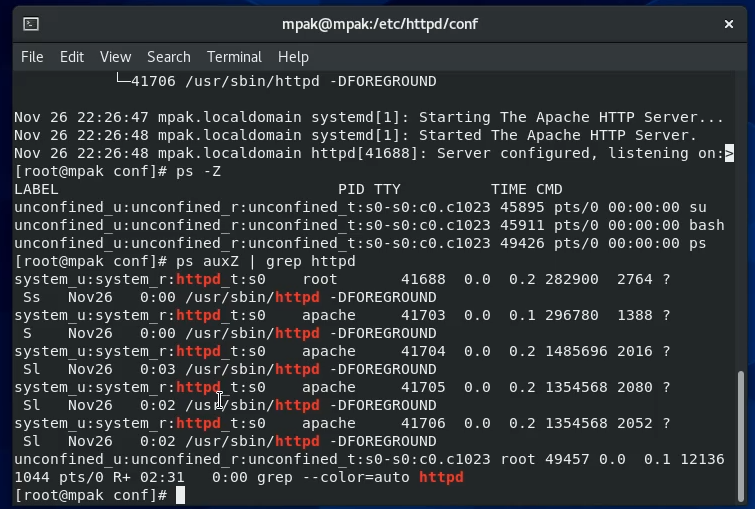
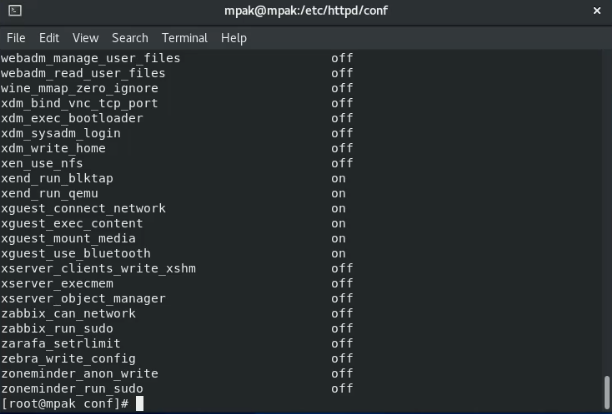
**Disabled**: Полное отключение системы принудительного контроля доступа. [[3]](#X9a11f83b1ef3906df5df95cff9123f28b5a945b).

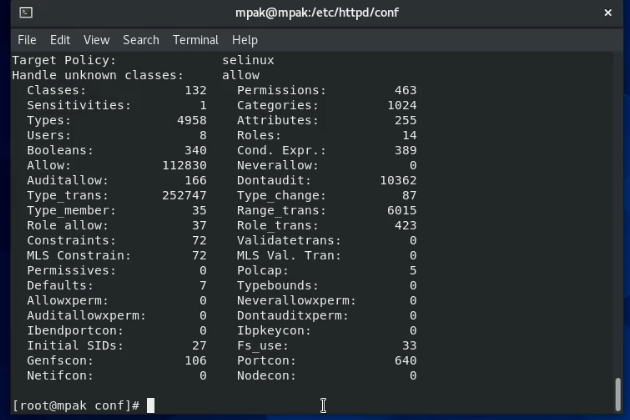
## Оборудование

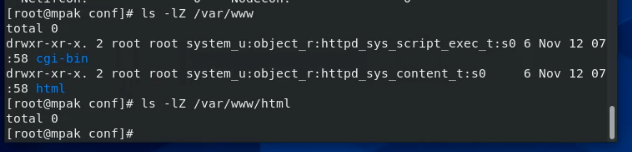
Лабораторная работа выполнялась дома со следующими характеристиками техники:

– Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz 2.81GHz  
– ОС Майкрософт Windows 10  
– VirtualBox верс. 6.1.26

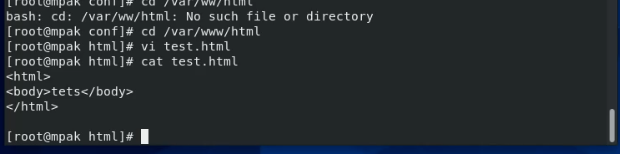
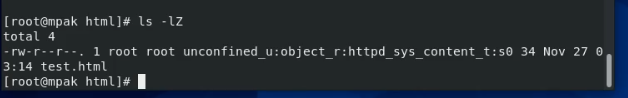
# Выполнение лабораторной работы

1. Вошла в систему с полученными учётными данными и убедилась, что  
   SELinux работает в режиме enforcing политики targeted с помощью ко-  
   манд getenforce и sestatus.  
   (рис. -@fig:001)  
   
2. C помощью команды service httpd status видим, что наш сервер работает.  
   (рис. -@fig:002)  
   
3. Нашла веб-сервер Apache в списке процессов, его контекст безопасности - httpd\_t   
   (рис. -@fig:003)  
   
4. Посмотрела текущее состояние переключателей SELinux для Apache. Большинство из них находятся в положении «off».  
   (рис. -@fig:004)  
   

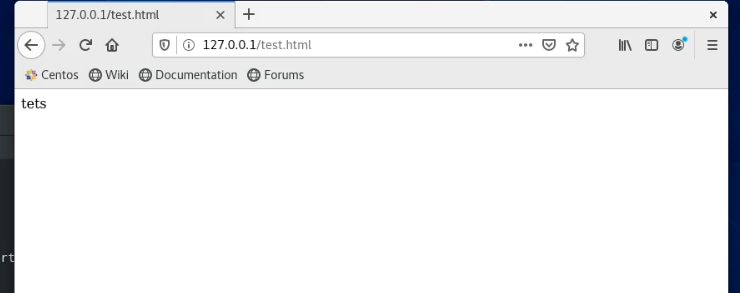
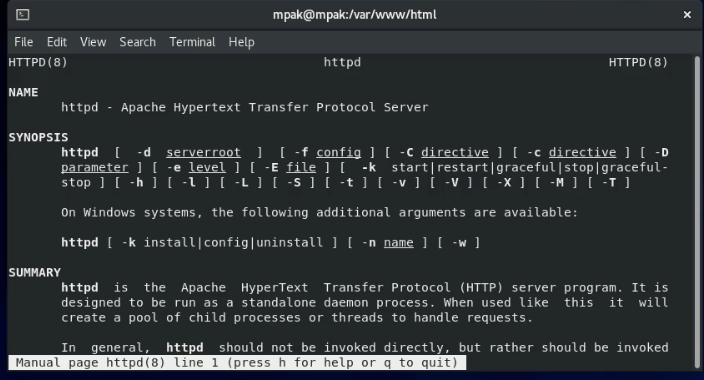
5.Посмотрела статистику по политике с помощью команды seinfo. Определите множество пользователей - 8, ролей - 37, типов - 4958.  
(рис. -@fig:005)  


6-8. Определила тип файлов и поддиректорий, находящихся в директории /var/www. Там находятся директории html и cgi-bin, где будут храниться файлы соответсвующей категории. В директории html пока нет файлов, так как я не создавала пока странички сайтов сервера.  
(рис. -@fig:006)  


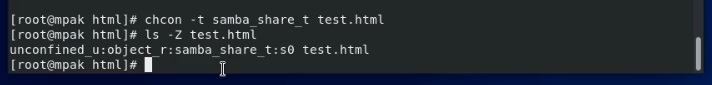
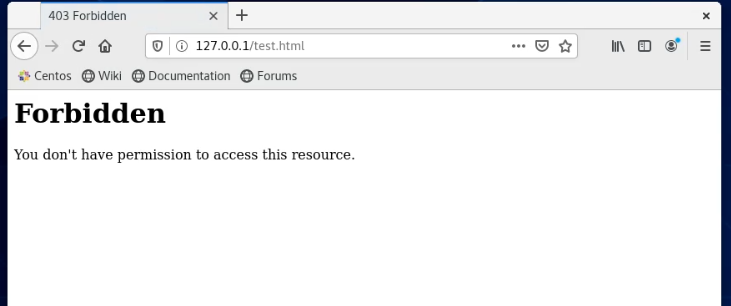
Как видно по информации директории html, создание файлов разрешено там только пользователю рут.

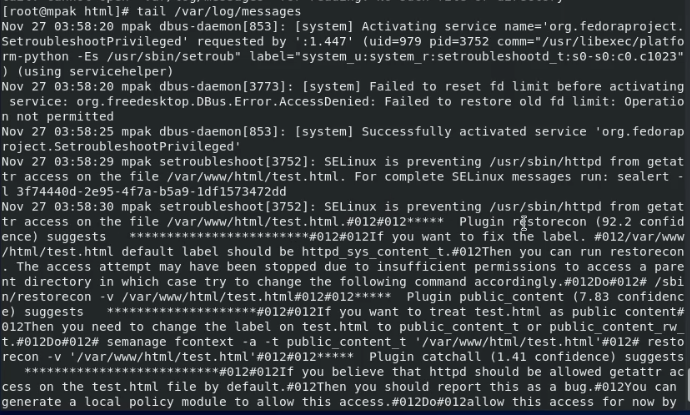
1. Создала от имени суперпользователя html-файл test.html.  
   (рис. -@fig:007)  
   
2. Проверила конкеткст только что созданного файла test.html.  
   Занесите в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html.   
   (рис. -@fig:008)  
   

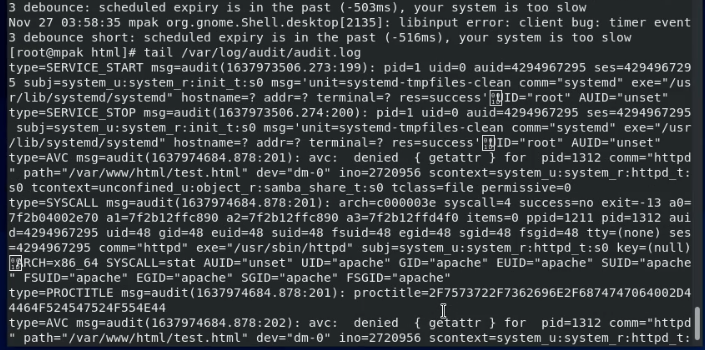
По умолчанию вновь созданным файлам в директории /var/www/html присваивается контекст httpd\_sys\_content\_t. Этот контекст дает права процессам Апаче обрабатывать файлы.

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. -@fig:009)  
   
2. Изучите справку man httpd\_selinux и выясните, какие контексты файлов определены для httpd. Сопоставьте их с типом файла test.html.  
   (рис. -@fig:010)  
   

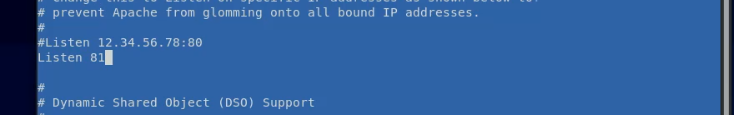
Контекст имеет схему: Тип, роль, домен  
Тип unconfined\_u - создал пользователь, свободный от типа (типично для пользователей CentOS )  
Роль - object\_r используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах.  
Домен - httpd\_sys\_content\_t позволяет процессу httpd получить доступ к файлу

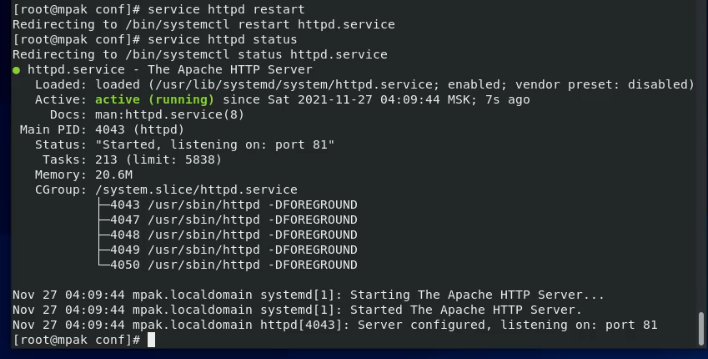
1. Изменила контекст файла /var/www/html/test.html с httpd\_sys\_content\_t на samba\_share\_t:  
   (рис. -@fig:011)  
     
   Контекст действительно поменялся.
2. Попробовала ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html>. И получила сообщение об ошибке.  
   (рис. -@fig:012)  
   
3. Файл не был отображён, потому что то процесс апаче, который должен был отобразить страничку html, не смог получить доступ к файлу из-за измененного контекста. Это специальное свойство контекста, которое защищает систему от взлома злоумышленниками.

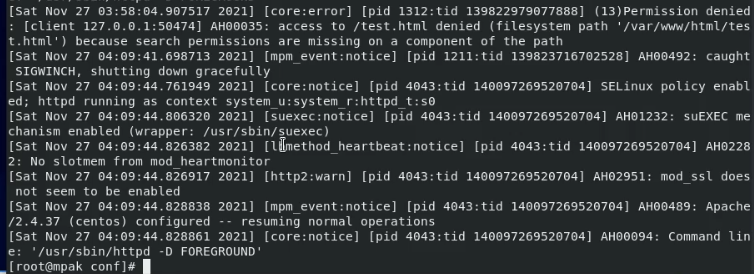
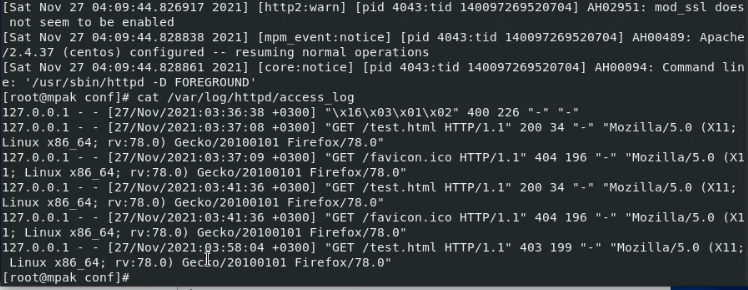
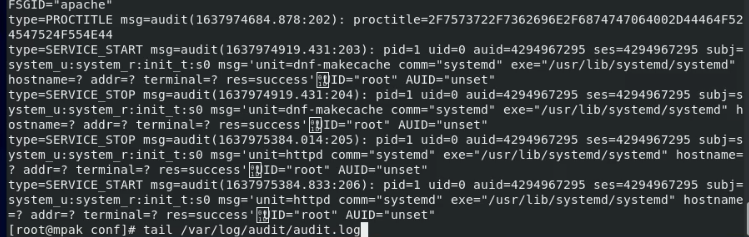
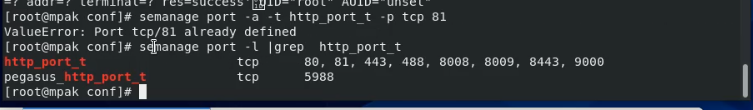
Просмотрела log-файлы веб-сервера Apache.   
/var/log/messages   
(рис. -@fig:013)  


/var/log/audit/audit.log.  
(рис. -@fig:014)  


В обоих файлах можно найти записи процессы setroubleshootd и audtd, где можно увидеть ошибки, аналогичные указанным выше.

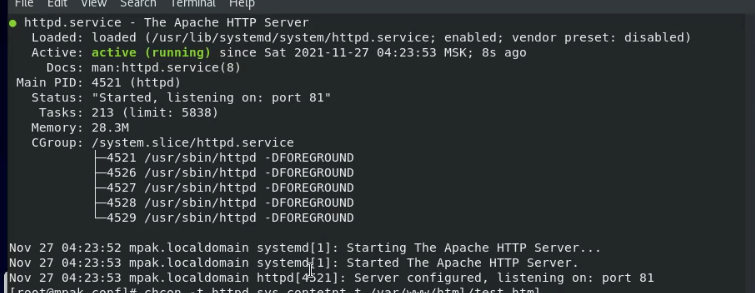
16-17. Попробовала запустить веб-сервер Apache на прослушивание ТСР-порта 81 . Для этого в файле /etc/httpd/httpd.conf заменила строчку на Listen 81.  
(рис. -@fig:015)  


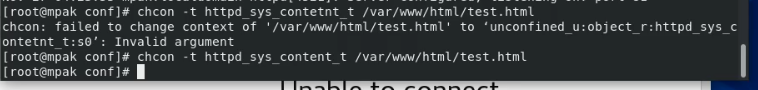
Выполнила перезапуск веб-сервера Apache. Должен был произойти сбой, но он не произошел, так как у меня порт 81 был уже записан в системе как используемый.  
(рис. -@fig:016)  


1. Проанализировала лог-файлы:  
   tail -nl /var/log/messages   
   (рис. -@fig:017)  
   fig:  
   /var/log/http/error\_log  
   (рис. -@fig:018)  
     
   /var/log/http/access\_log   
   (рис. -@fig:019)  
     
   /var/log/audit/audit.log   
   (рис. -@fig:020)  
     
   Так как у меня все же запустился сервер, то я просто назову, где бы повились сообщения об ошибках. Ошибки будут в : /var/log/http/error\_log, /var/log/audit/audit.log и /var/log/messages
2. Добавила 81 порт командой.  
   (рис. -@fig:021)  
   

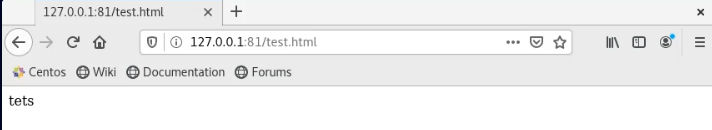
Убедилась, что порт 81 появился в списке.

1. Попробовала запустить веб-сервер Apache ещё раз. Он запустился сейчас, так как мы добавили порт к используемым в сервисе Апаче.

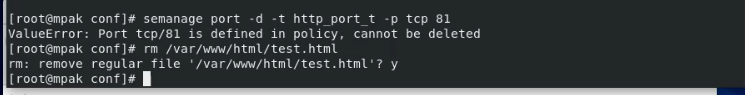
(рис. -@fig:022)  


1. Вернула контекст httpd\_sys\_cоntent\_t  
   (рис. -@fig:023)  
   

Теперь веб-сервер открывается, введя в браузере адрес <http://127.0.0.1:81/test.html>.

(рис. -@fig:024)  


22-24. Исправила обратно конфигурационный файл apache, вернув Listen 80.  
Удалила привязку http\_port\_t к 81 порту  
Удалите файл /var/www/html/test.html

(рис. -@fig:025)  


Не забыла выйти из режима суперпользователя.

## Выводы

Развила навыки администрирования ОС Linux. Получила первое практическое знакомство с технологией SELinux1. Проверила работу SELinx на практике совместно с веб-сервером Apache.

## Список литературы

1.SELinux – описание и особенности работы с системой. Часть 1 // Habr URL: <https://habr.com/ru/company/kingservers/blog/209644/> (дата обращения: 26.11.2021).

2.Контексты SELinux // fedora URL: <https://docs.fedoraproject.org/ru-RU/Fedora/13/html/Security-Enhanced_Linux/chap-Security-Enhanced_Linux-SELinux_Contexts.html> (дата обращения: 25.11.2021).

3.Безопасная эксплуатация Apache, часть 1: базовые понятия // rus-linux URL: <http://rus-linux.net/MyLDP/server/securing-apache-part-1.html> (дата обращения: 26.11.2021).