**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**по лабораторной работе № 6**

*Дисциплина: Основы информационной безопасности  
Название работы: Мандатное разграничение прав в Linux*

Студент: Невзоров Дмитрий

**МОСКВА**

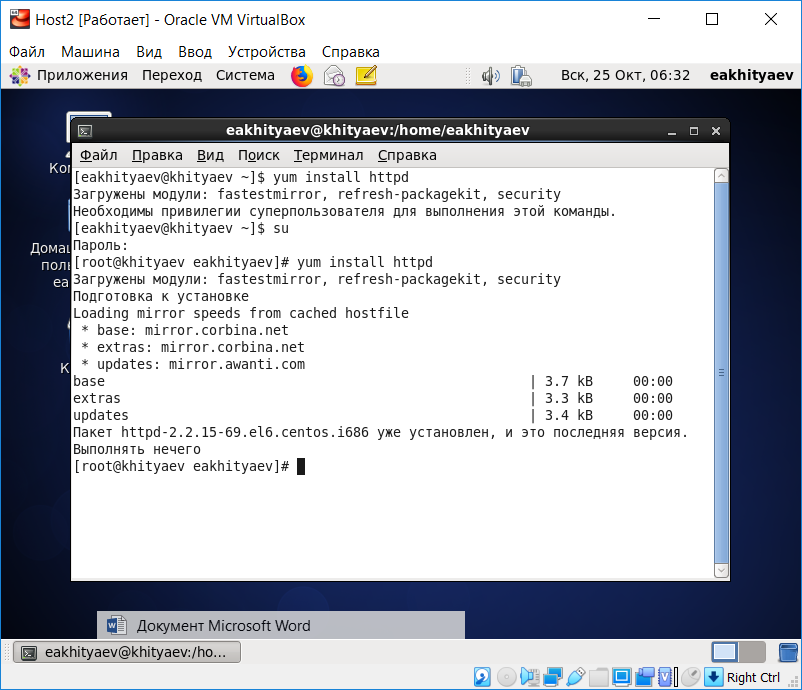
2021 г.

**Цель работы**

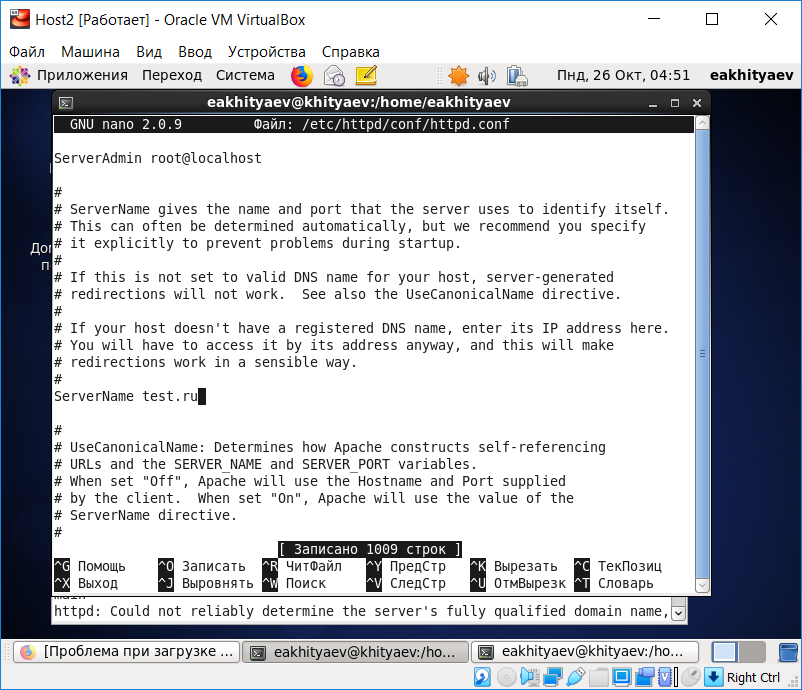
Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinux на практике совместно с веб-сервером Apache.

**Подготовка лабораторного стенда**

1. Установим/обновим (за суперпользователя) веб-сервер Apache с помощью команды *yum install httpd*



1. В конфигурационном файле */etc/httpd/httpd.conf* зададим параметр ServerName: *ServerName test.ru* чтобы при запуске веб-сервера не выдавались лишние сообщения об ошибках, не относящихся к лабораторной работе.



1. Также необходимо проследить, чтобы пакетный фильтр был отключен или в своей рабочей конфигурации позволял подключаться к 80-му и 81-му портам протокола tcp. Добавим разрешающие правила с помощью команд:

*iptables -I INPUT -p tcp --dport 80 -j ACCEPT*

*iptables -I INPUT -p tcp --dport 81 -j ACCEPT*

*iptables -I OUTPUT -p tcp --sport 80 -j ACCEPT*

*iptables -I OUTPUT -p tcp --sport 81 -j ACCEPT*

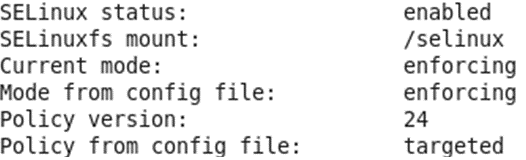
Можно также отключить фильтр командами:

*iptables -F*

*iptables -P INPUT ACCEPT iptables -P OUTPUT ACCEPT*

**Порядок выполнения работы**

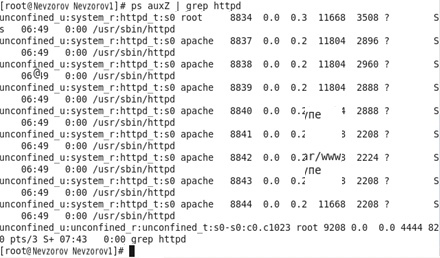
1. Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд *getenforce* и *sestatus*



1. Обратимся к веб-серверу, запущенному на нашем компьютере, и убедимся, что последний работает: *service httpd status*

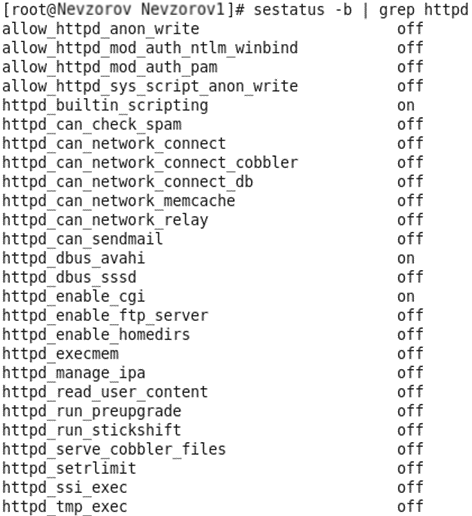
https://sun9-38.userapi.com/impg/kqltBqWkEzjJQX-db2hxKfJV-OyUfGPj-0qnYg/j49yBE4cHU0.jpg?size=424x57&quality=96&sign=5e2216d96d859459bc5af23a2214597d&type=album

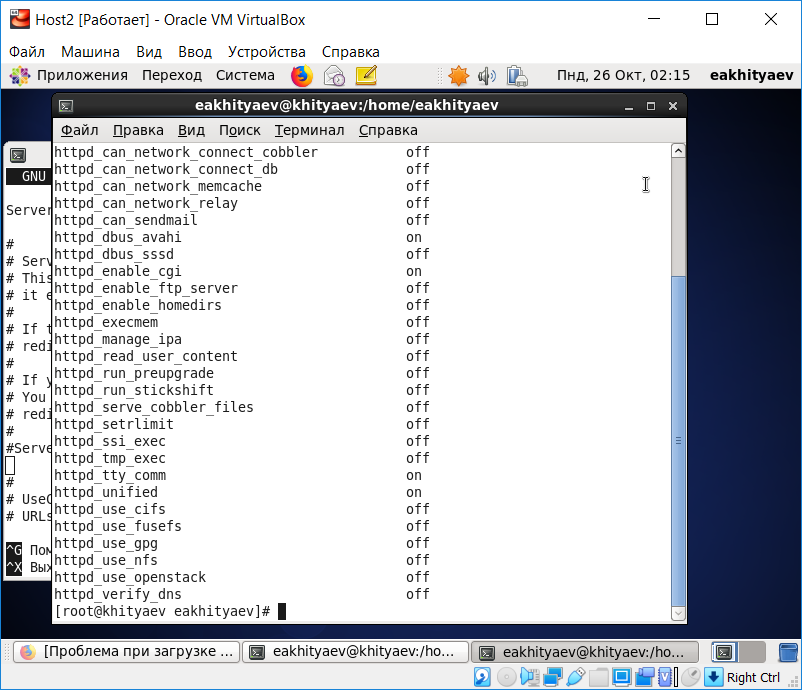
1. Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности, используем команду *ps auxZ | grep httpd*



В нашем случае контекст безопасности unconfined\_u:system\_r:httpd\_t

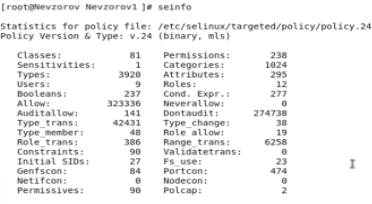
1. Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды *sestatus –b | grep httpd*





Многие из переключателей находятся в положении «off».

1. Посмотрим статистику по политике с помощью команды *seinfo*, также определим множество пользователей, ролей и типов.



Пользователей: 9, ролей: 12, типов: 3920.

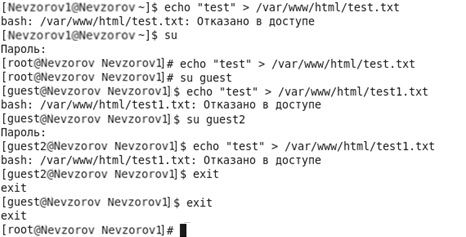
1. Определим тип файлов и поддиректорий, находящихся в директории */var/www* с помощью команды *ls -lZ /var/www*



1. Определим тип файлов, находящихся в директории */var/www/html* с помощью команды *ls –lZ /var/www/html*

https://sun9-13.userapi.com/impg/BrJxGg654hb5XJZ3QcQUg09QnhDoZHrogZwkVQ/l90_liTKiDY.jpg?size=463x34&quality=96&sign=77fb407b9a4ef24bf2c17c4013734aec&type=album

1. Определим круг пользователей, которым разрешено создание файлов в директории */var/www/html*.



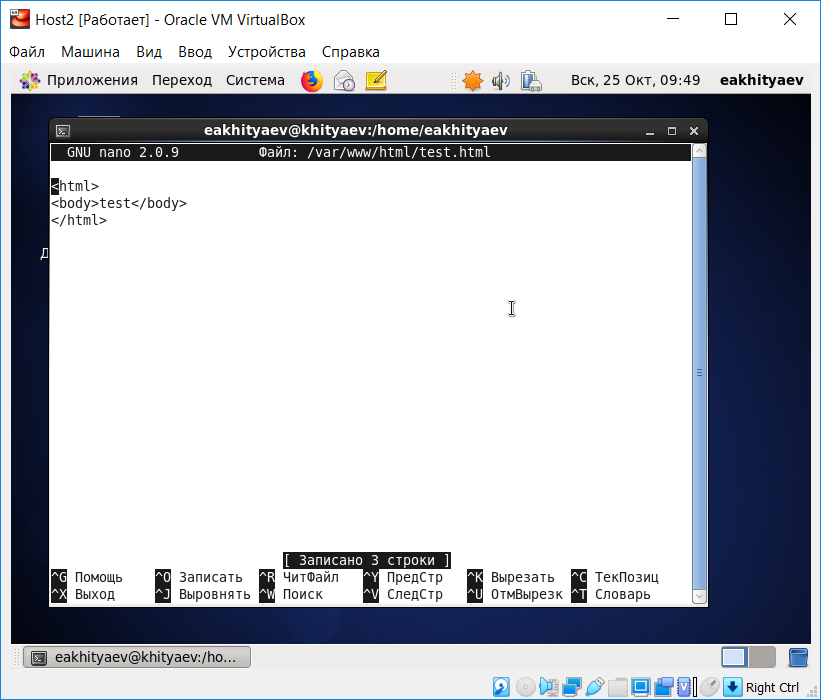
Видно, что только суперпользователь может создать файл в данной директории.

1. В следствие этого создадим от имени суперпользователя html-файл */var/www/html/test.html* следующего содержания:

**<html>**

**<body>**test**</body>**

**</html>**



1. Проверим контекст созданного файла.

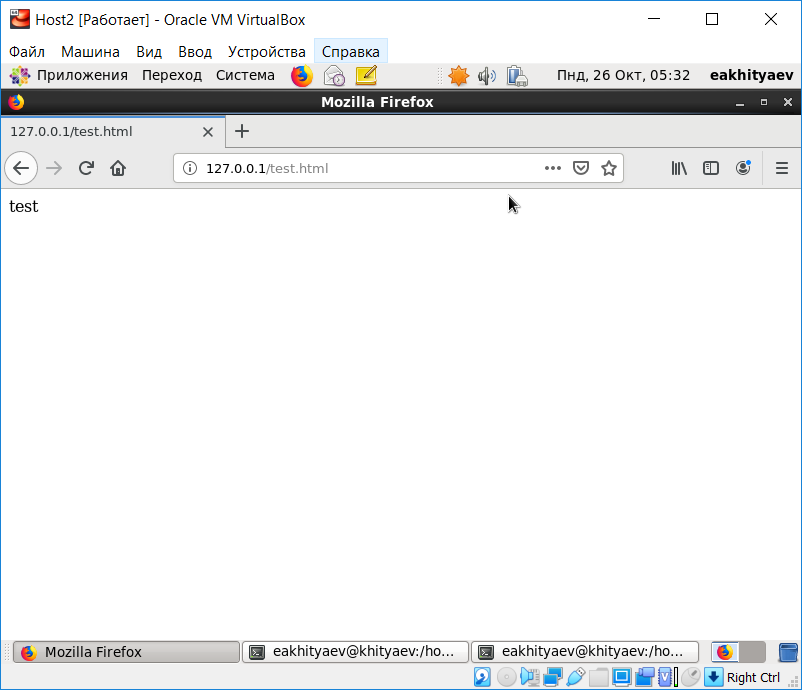


Контекст, присваиваемый по умолчанию вновь созданным файлам в директории */var/www/html*: unconfined\_u:object\_r:httpd\_sys\_content\_t

1. Обратимся к файлу через веб-сервер, введя в браузере firefox адрес

[*http://127.0.0.1/test.html*](http://127.0.0.1/test.html)

Убедимся, что файл был успешно отображен.



1. Изучим справку *man httpd\_selinux* и выясним, какие контексты файлов определены для *httpd* и сопоставим их с типом *файла test.html*. Проверим контекст файла командой *ls –Z /var/www/html/test.htm*

**

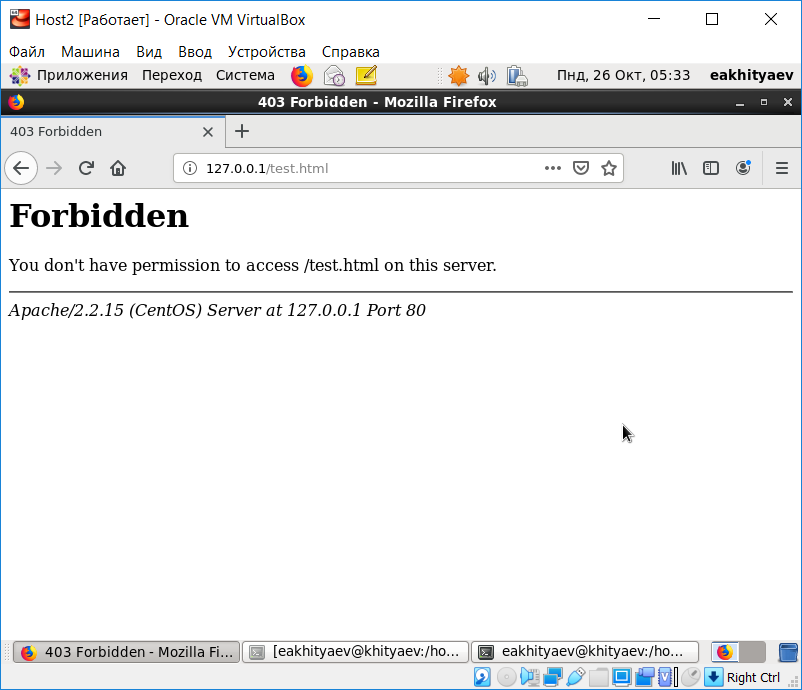
Т.к. по умолчанию пользователи CentOS являются свободными (unconfined) от типа, созданному нами файлу *test.html* был сопоставлен SELinux, пользователь unconfined\_u. Это первая часть контекста. Далее политика ролевого разделения доступа RBAC используется процессами, но не файлами, поэтому роли не имеют никакого значения для файлов. Роль object\_r используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах. Тип httpd\_sys\_content\_t позволяет процессу httpd получить доступ к файлу. Благодаря наличию последнего типа мы получили доступ к файлу при обращении к нему через браузер.

1. Изменим контекст файла */var/www/html/test.html* с httpd\_sys\_content\_t на другой, к которому процесс httpd не должен иметь доступа, в нашем случае, на samba\_share\_t:

*chcon –t samba\_share\_t /var/www/html/test.html*

*ls –Z /var/www/html/test.html*

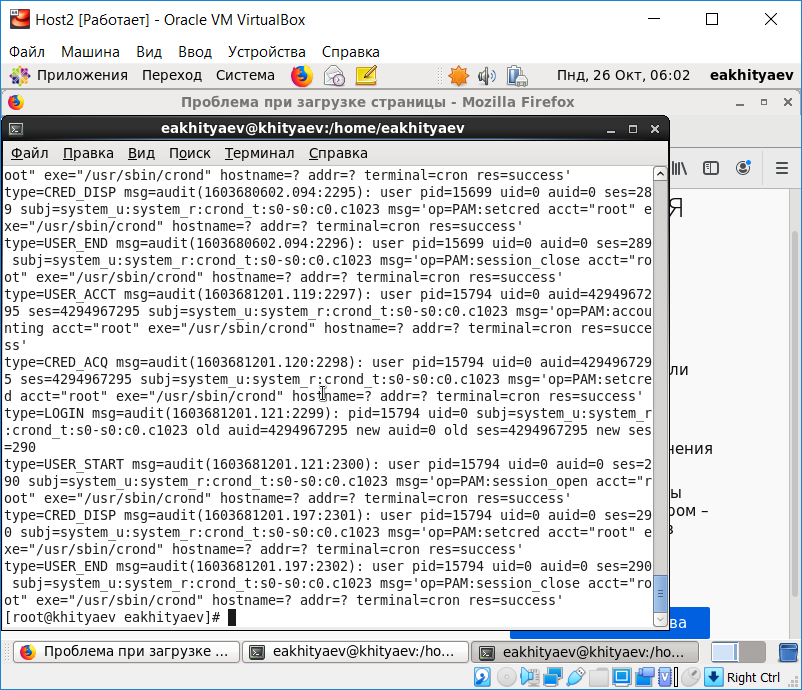
1. Попробуем еще раз получить доступ к файлу через веб-сервер, введя в браузере firefox адрес [*http://127.0.0.1/test.html*](http://127.0.0.1/test.html)



Мы получили сообщение об ошибке.

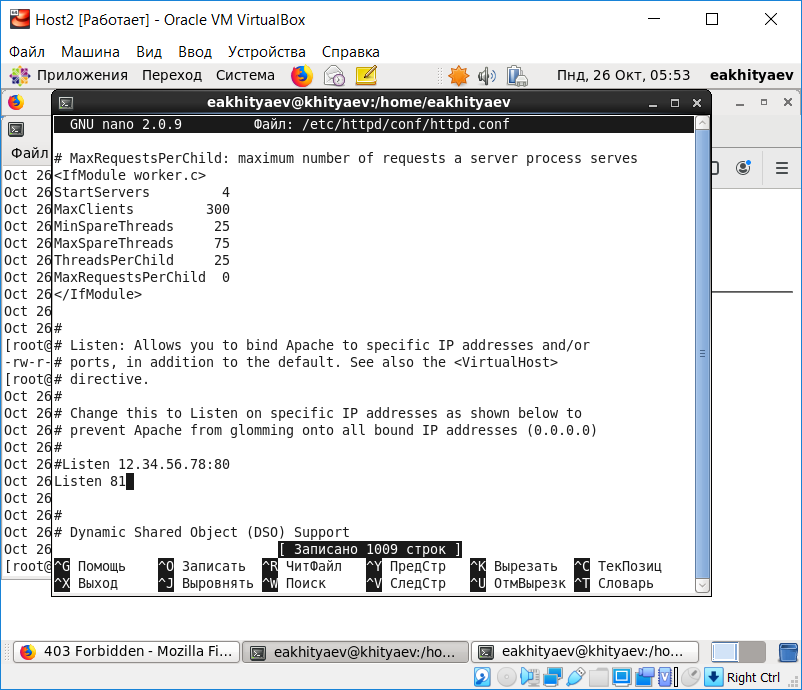
1. Проанализируем ситуацию, просмотрев log-файлы веб-сервера Apache, системный log-файл и audit.log при условии уже запущенных процессов setroubleshootd и audtd.



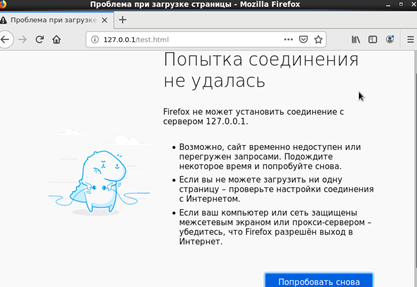


Исходя из log-файлов, мы можем заметить, что проблема в измененном контексте на шаге 13, т.к. процесс httpd не имеет доступа на samba\_share\_t. В системе оказались запущены процессы setroubleshootd и audtd, поэтому ошибки, связанные с измененным контекстом, также есть в файле */var/log/audit/audit.log*.

1. Попробуем запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в */etc/services*), заменив в файле */etc/httpd/conf/httpd.conf* строчку *Listen 80* на *Listen 81.*



1. Перезапустим веб-сервер Apache и попробуем обратиться к файлу через веб-сервер, введя в браузере firefox адрес [*http://127.0.0.1/test.html*](http://127.0.0.1/test.html)

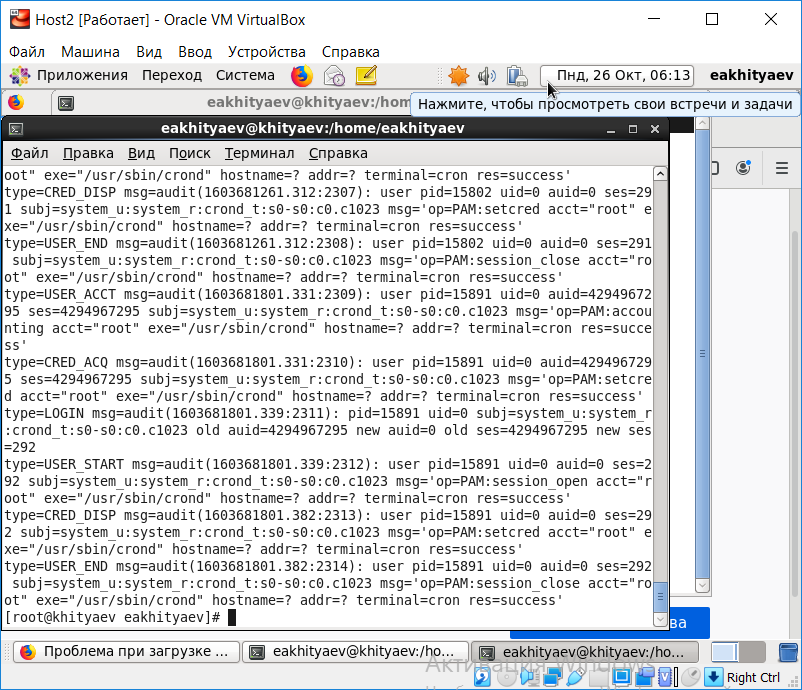


Из того, что при запуске файла через браузер появилась ошибка, можно сделать предположение, что в списках портов, работающих с веб-сервером Apache, отсутствует порт 81.

1. Подтвердим свои догадки, проанализировав log-файлы: *tail –n1 /var/log/messages* и просмотрев файлы */var/log/http/error\_log,* */var/log/http/access\_log* и */var/log/audit/audit.log*



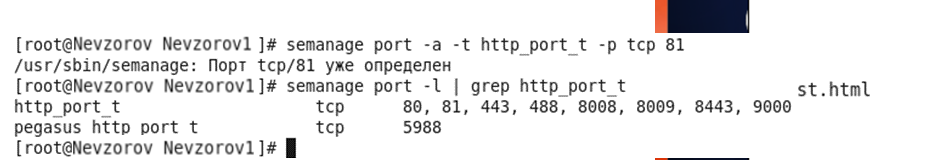
~~~~



Во всех log-файлах появились записи, кроме /var/log/messages.

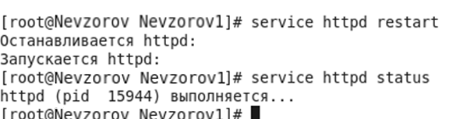
1. Выполним команду *semanage port –a –t http\_port\_t –p tcp 81*

После этого проверим список портов командой *semanage port –l | grep http\_port\_t*

**

Убедились, что порт 81 присутствует в списке.

1. Попробуем теперь запустить веб-сервер Apache еще раз.

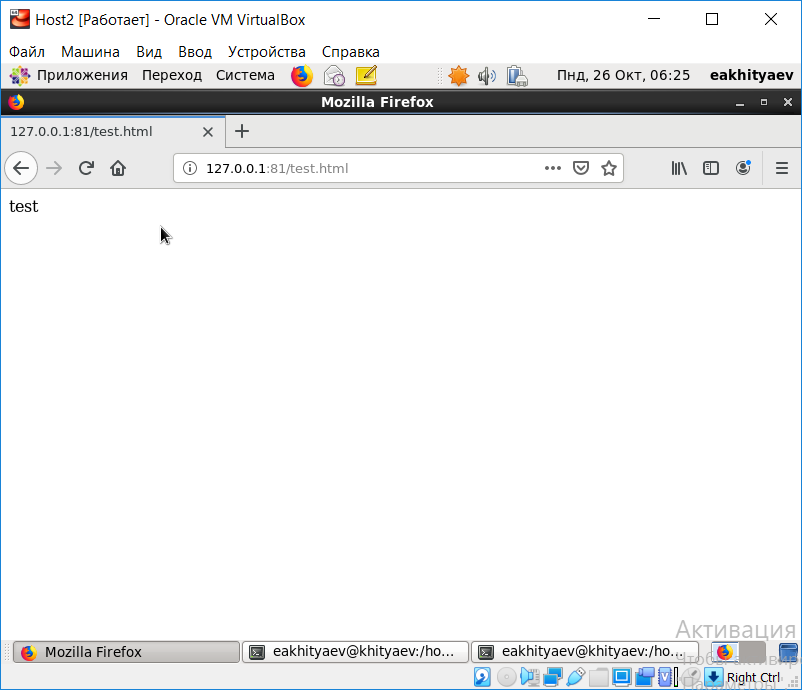


1. Вернем контекст *httpd\_sys\_content\_t* к файлу */var/www/html/test.html*:

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

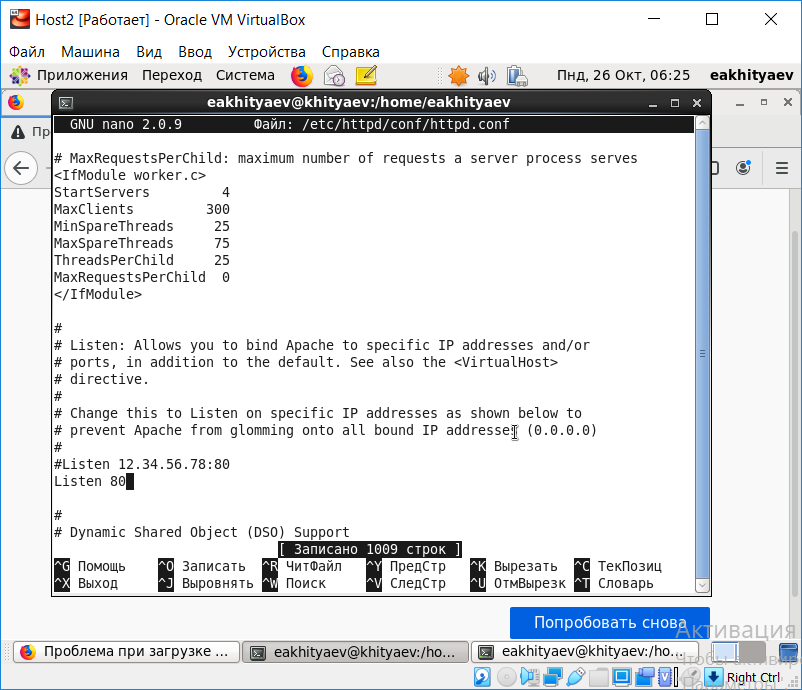


После этого вновь попробуем получить доступ к файлу через веб-сервер, введя в браузере firefox адрес [*http://127.0.0.1:81/test.html*](http://127.0.0.1:81/test.html)



Увидели слово содержимое файла - слово «test».

1. Исправим обратно конфигурационный файл apache, вернув Listen 80.



1. Удалим привязку http\_port\_t к 81 порту: *semanage port –d –t http\_port\_t –p tcp 81.* Данную команду выполнить невозможно на моей версии CentOS, поэтому получаем ошибку.



1. Удалим файл */var/www/html/test.html*: *rm /var/www/html/test.html*



**Вывод**

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