

# **Лабораторная работа №6**

**Дисциплина: Основы информационной безопасности**

Феоктистов Владислав Сергеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
3.1	Технология SELinux . . . . .	7
3.2	Apache . . . . .	9
3.3	Таблицы . . . . .	10
<b>4</b>	<b>Подготовка лабораторного стенда</b>	<b>13</b>
<b>5</b>	<b>Выполнение лабораторной работы</b>	<b>15</b>
<b>6</b>	<b>Выводы</b>	<b>27</b>
	<b>Список литературы</b>	<b>28</b>

# Список иллюстраций

4.1	Подготовка лабораторного стенда . . . . .	14
4.2	Подготовка лабораторного стенда . . . . .	14
4.3	Подготовка лабораторного стенда . . . . .	14
5.1	Просмотр режима работы и политики SELinux . . . . .	15
5.2	Запуск и проверка работы сервера Apache . . . . .	16
5.3	Проверка работы веб-сервиса . . . . .	16
5.4	Контекст безопасности процесса веб-сервера и состояние переключателей SELinux . . . . .	17
5.5	Статистика по политике . . . . .	17
5.6	Типы поддиректорий и определение пользователя с правами создания файла . . . . .	18
5.7	Таблица с минимальными правами для совершения операций из 2ой лаб. работы . . . . .	18
5.8	Создание html-файла . . . . .	19
5.9	Контекст по умолчанию . . . . .	19
5.10	Обращение к файлу через веб-сервер . . . . .	19
5.11	Справка httpd_selinux . . . . .	20
5.12	Справка httpd_selinux . . . . .	20
5.13	Изменение контекста файла . . . . .	21
5.14	Обращение к файлу через веб-сервер при измененном контексте . . . . .	21
5.15	Log-файл error_log . . . . .	22
5.16	Log-файл messages . . . . .	22
5.17	Установка TCP-порта 81 . . . . .	23
5.18	Запуск веб-сервера под старым адресом, но с новым портом . . . . .	23
5.19	Анализ лог-файлов . . . . .	24
5.20	Добавление TCP-порта . . . . .	24
5.21	Запуск веб-сервера под портом 81 . . . . .	24
5.22	Возвращение контекста . . . . .	25
5.23	Запуск веб-сервера под портом 81 с новым контекстом . . . . .	25
5.24	Завершение лабораторной работы . . . . .	26

## Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . .	11
3.2	Описание некоторых используемых в работе команд . . . . .	11

# 1 Цель работы

Целью данной работы является:

- развитие навыка администрирования ОС Linux;
- получение первого практического знакомства с технологией SELinux;
- проверка работы SELinux на практике совместно с веб-сервером Apache.

## 2 Задание

- Проверить режим работы, политику и состояние переключателей SELinux, проверить работу веб-сервера;
- Найти процесс веб-сервера, определить его контекст;
- Посмотреть статистику политики, определить типы поддиректорий и файлов, определить круг пользователей с правами создания файлов в указанной поддиректории;
- Создать html-файл, определить его контекст по умолчанию, подключиться к веб-серверу в браузере;
- Изучить справку `httpd_selinux`;
- Попробовать подключиться к веб-серверу в браузере с другим контекстом html-файла;
- Попробовать подключиться к веб-серверу в браузере с другим контекстом html-файла и другим TCP-портом;
- Попробовать подключиться к веб-серверу по новому адресу в браузере с другим контекстом html-файла и другим TCP-портом;
- Вернуть старые настройки и проанализировать все необходимые Apache лог-файлы.

## 3 Теоретическое введение

### 3.1 Технология SELinux

**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.

Основные термины, используемые в SELinux:

- Домен — список действий, которые может выполнять процесс. Обычно в качестве домена определяется минимально-возможный набор действий, при помощи которых процесс способен функционировать. Таким образом, если процесс дискредитирован, злоумышленнику не удастся нанести большого вреда.
- Роль — список доменов, которые могут быть применены. Если какого-то домена нет в списке доменов какой-то роли, то действия из этого домена не могут быть применены.
- Тип — набор действий, которые допустимы по отношению к объекту. Тип отличается от домена тем, что он может применяться к пайпам, каталогам и файлам, в то время как домен применяется к процессам.
- Контекст безопасности — все атрибуты SELinux — роли, типы и домены.

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.

Вы можете посмотреть текущий режим и другие настройки SELinux (а в случае необходимости и изменить его) при помощи специального GUI-инструмента, доступного в меню «Администрирование» (system-config-selinux). Если же вы привыкли работать в консоли, то можете посмотреть текущий статус командой `sestatus`.

Также вы можете узнать статус SELinux при помощи команды `getenforce`.

Команда «`setenforce`» позволяет быстро переключаться между режимами Enforcing и Permissive, изменения вступают в силу без перезагрузки. Но если вы включаете или отключаете SELinux, требуется перезагрузка, ведь нужно заново устанавливать метки безопасности в файловой системе.

Для того, чтобы выбрать режим по-умолчанию, который будет применяться при каждой загрузке системы, задайте значение строки 'SELINUX=' в файле `/etc/selinux/config`, задав один из режимов — 'enforcing', 'permissive' или 'disabled'. Например: 'SELINUX=permissive'.

Более подробно на сайте [1].



## 3.2 Apache

**Apache** — это свободное программное обеспечение, с помощью которого можно создать веб-сервер. Несмотря на то, что Apache чаще всего называют сервером (более того, его официальное название — Apache HTTP Server) — это всё-таки программа, которую устанавливают на сервер, чтобы добиться определённых результатов. Русскоязычная аудитория нередко называет серверы с такими программами коротко — Апач.

Для чего нужен Apache сервер:

- чтобы открывать динамические PHP-страницы,
- для распределения поступающей на сервер нагрузки,
- для обеспечения отказоустойчивости сервера,
- чтобы потренироваться в настройке сервера и запуске PHP-скриптов.

Apache является кроссплатформенным ПО и поддерживает такие операционные системы, как Linux, BSD, MacOS, Microsoft, BeOS и другие.

Основные компоненты архитектуры сервера Апач — динамические модули, ядро и конфигурационные файлы.

Основные задачи ядра веб-сервера — модерация работы конфигурационных файлов, а также исполнение HTTP и HTTPS протоколов. Однако в чистом виде ядро имеет весьма ограниченный функционал и не справляется с такими задачами. Как можно расширить функционал веб-сервера? Для этого ядро должно работать сообща с системой модулей.

Модули — это по сути файлы, которые помогают расширять возможности той или иной системы. Базовая часть модулей для Апач устанавливается по умолчанию, а дополнительные модули нужно подключать самостоятельно. При этом каждый модуль отвечает за отдельный компонент работы с запросом. Например, аутентификацию или кэширование. Для оптимизации ядра существует свыше 500 различных модулей — под любую задачу или проект.

Конфигурационный файл — это файл, который хранит настройки операционной системы и приложений, а также позволяет вносить в них изменения. Конфигурация сервера Apache основана на текстовых конфигурационных файлах. Эти файлы отвечают за каждый из трёх уровней:

- Файл уровня конфигурации сервера — `httpd.conf`. Он содержит директивы, которые управляют работой веб-сервера. В каждой операционной системе `httpd.conf` по-разному расположен. Чтобы узнать его местоположение, достаточно ввести в терминале команду: `httpd -V`
- Файл (или файлы) уровня конфигурации каталога — `.htaccess`. Файл `.htaccess` отвечает за настройки веб-сервера только в том каталоге, в котором он размещен, а также в его дочерних каталогах. То есть вносимые в `.htaccess` изменения не затрагивают глобальные настройки. Также настройки `.htaccess` имеют приоритет перед настройками `httpd.conf`.
- Файл уровня виртуального хоста — `extra/httpd-vhosts.conf`. Такие хосты нужны пользователям, которые запускают несколько сайтов на одном виртуальном сервере. На один сервер можно добавить неограниченное количество виртуальных хостов.

Как правило, основные конфигурационные файлы располагаются в папке `conf`, а дополнительные в папке `extra`. Изменения в эти файлы можно вносить как через командную строку, так и путем редактирования самого файла.

Более подробно на сайте [2].

### 3.3 Таблицы

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Таблица 3.2: Описание некоторых используемых в работе команд

Команда	Описание команды
getenforce	Получение статуса SELinux: enforcing, permissive, disabled
sestatus	Текущий режим и другие настройки SELinux
service	Выводит список всех сервисов при использовании опции –status-all и выводит статус конкретного сервиса при указании его названия
cat	Вывод содержимого указанного файла.
ls	Выводит содержимое каталога. Опция -l выводит дополнительную информацию, -a отображает скрытые файлы, в названии которых в самом начале стоит символ '.', -Z - выводит контекст файла в SELinux

Команда	
Команда	Описание команды
touch	Создает текстовый файл по указанному пути и с указанным именем внутри пути.
rm	Удаляет файл(ы) (каталог(и) при указании опции -r) по указанному(ым) пути(ям).
cd	Перемещение по файловой системе.
grep	Дает возможность вести поиск строк. Также можно передать вывод любой команды в grep, что сильно упрощает работу во время поиска
nano	Запуск в терминале текстовый редактор
ps	Выводит список запущенных процессов с их идентификаторами
chcon	Помогает изменить контекст SELinux
tail	Просмотр последних строк файла
semanage	Инструмент, используемый для настройки определенных элементов политики SELinux без изменения или перекомпиляции источников политики
systemctl	Позволяет управлять основными процессами Linux
matchpathcon	Выводит контекст безопасности по умолчанию для указанного файла

Более подробно об Unix см. в [3–8].

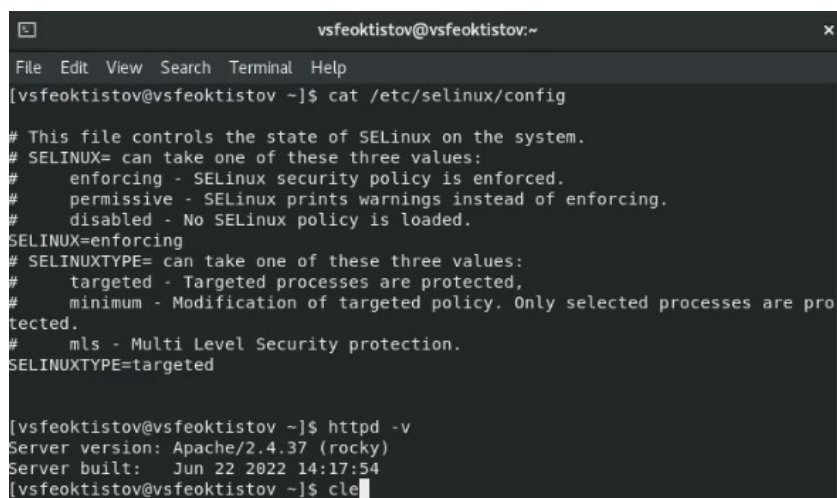
## 4 Подготовка лабораторного стенда

Для проведения указанной лабораторной работы на одно рабочее место требуется компьютер с установленной операционной системой Linux, поддерживающей технологию SELinux (Rocky Linux поддерживает технологию SELinux по умолчанию с включённой политикой SELinux targeted и режимом enforcing). При этом следует убедиться, что политика и режим включены, особенно когда работа будет проводиться повторно и велика вероятность изменений при предыдущем использовании системы.

Для выполнения заданий требуется наличие учётной записи администратора (root) и учётной записи обычного пользователя. Постоянно работать от учётной записи root неправильно с точки зрения безопасности (несмотря на это все команды выполнялись от имени root-пользователя, чтобы не тратить время).

Перед выполнением работы можно посмотреть содержимое конфигурационного файла `/etc/selinux/config` (рис. 4.1). В нем можно поменять режимы SELinux по умолчанию.

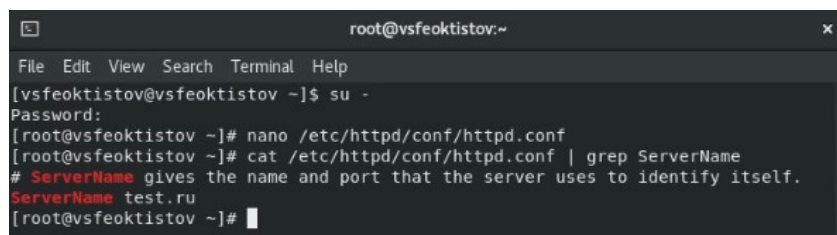
Необходимо, чтобы был установлен веб-сервер Apache. При установке системы в конфигурации «рабочая станция» указанный пакет не ставится. Проверить наличие Apache в Rocky Linux можно с помощью команды `httpd -v`. Если Apache установлен, то должна вывести его версия, иначе нужно будет установить его [**cmd:** `sudo yum install httpd`] (рис. 4.1).



```
vsfeoktistov@vsfeoktistov:~  
File Edit View Search Terminal Help  
[vsfeoktistov@vsfeoktistov ~]$ cat /etc/selinux/config  
# This file controls the state of SELinux on the system.  
# SELINUX= can take one of these three values:  
#   enforcing - SELinux security policy is enforced.  
#   permissive - SELinux prints warnings instead of enforcing.  
#   disabled - No SELinux policy is loaded.  
SELINUX=enforcing  
# SELINUXTYPE= can take one of these three values:  
#   targeted - Targeted processes are protected,  
#   minimum - Modification of targeted policy. Only selected processes are protected.  
#   mls - Multi Level Security protection.  
SELINUXTYPE=targeted  
  
[vsfeoktistov@vsfeoktistov ~]$ httpd -v  
Server version: Apache/2.4.37 (rocky)  
Server built:   Jun 22 2022 14:17:54  
[vsfeoktistov@vsfeoktistov ~]$ cle
```

Рис. 4.1: Подготовка лабораторного стенда

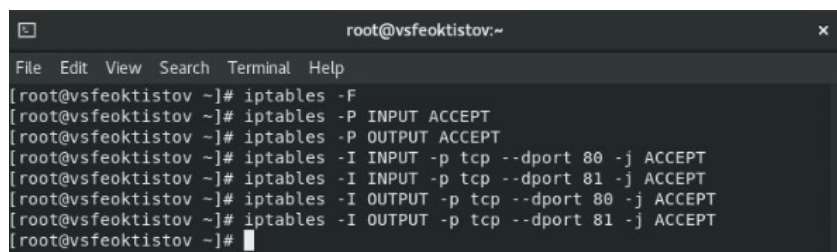
В конфигурационном файле `/etc/httpd/conf/httpd.conf` необходимо задать параметр `ServerName test.ru`, чтобы при запуске веб-сервера не выдавались лишние сообщения об ошибках, не относящихся к лабораторной работе (рис. 4.2).



```
root@vsfeoktistov:~  
File Edit View Search Terminal Help  
[vsfeoktistov@vsfeoktistov ~]$ su -  
Password:  
[root@vsfeoktistov ~]# nano /etc/httpd/conf/httpd.conf  
[root@vsfeoktistov ~]# cat /etc/httpd/conf/httpd.conf | grep ServerName  
# ServerName gives the name and port that the server uses to identify itself.  
ServerName test.ru  
[root@vsfeoktistov ~]#
```

Рис. 4.2: Подготовка лабораторного стенда

Также необходимо проследить, чтобы пакетный фильтр был отключён или в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола tcp (рис. 4.3).



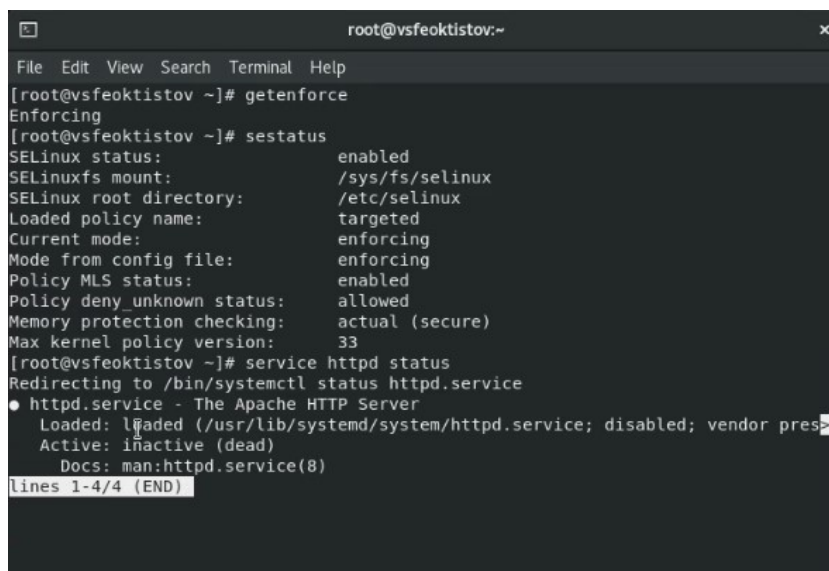
```
root@vsfeoktistov:~  
File Edit View Search Terminal Help  
[root@vsfeoktistov ~]# iptables -F  
[root@vsfeoktistov ~]# iptables -P INPUT ACCEPT  
[root@vsfeoktistov ~]# iptables -P OUTPUT ACCEPT  
[root@vsfeoktistov ~]# iptables -I INPUT -p tcp --dport 80 -j ACCEPT  
[root@vsfeoktistov ~]# iptables -I INPUT -p tcp --dport 81 -j ACCEPT  
[root@vsfeoktistov ~]# iptables -I OUTPUT -p tcp --dport 80 -j ACCEPT  
[root@vsfeoktistov ~]# iptables -I OUTPUT -p tcp --dport 81 -j ACCEPT  
[root@vsfeoktistov ~]#
```

Рис. 4.3: Подготовка лабораторного стенда

## 5 Выполнение лабораторной работы

В первую очередь, убедимся, что SELinux работает в режиме *enforcing* политики *targeted* с помощью команд *getenforce* и *sestatus*. Как видно из рисунка 5.1, команда *getenforce* вывела сообщение *Enforcing*, а команда *sestatus* в параметре *Loaded policy name* - *targeted*, что и было необходимо.

После этого стоит убедиться, что сервер работает [**cmd:** *service httpd status*] и обратиться к веб-сервису через браузер.

A terminal window titled 'root@vsfeoktistov:~' showing the execution of several commands. The first command is 'getenforce', which outputs 'Enforcing'. The second command is 'sestatus', which outputs a detailed status report for SELinux, including 'enabled' status, 'targeted' policy name, and 'enforcing' current mode. The third command is 'service httpd status', which is redirected to 'systemctl status httpd.service'. The output shows that the 'httpd.service' is 'loaded' but 'inactive (dead)'. The terminal text is as follows:

```
root@vsfeoktistov:~  
File Edit View Search Terminal Help  
[root@vsfeoktistov ~]# getenforce  
Enforcing  
[root@vsfeoktistov ~]# 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  
Memory protection checking: actual (secure)  
Max kernel policy version: 33  
[root@vsfeoktistov ~]# 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 pres  
   Active: inactive (dead)  
   Docs: man:httpd.service(8)  
lines 1-4/4 (END)
```

Рис. 5.1: Просмотр режима работы и политики SELinux

Если у Вас, как и на рисунке 5.1, значение *Active* - это *inactive (dead)*, т.е. режим работы Apache - выключен (неактивен), то его нужно будет запустить с помощью команды *systemctl start httpd* и снова проверить статус сервера [**cmd:** *service httpd status*], теперь он должен быть со значением *active (running)* (включен) (рис. 5.2).

```
root@vsfeoktistov:~  
File Edit View Search Terminal Help  
[root@vsfeoktistov ~]# systemctl start httpd  
[root@vsfeoktistov ~]# 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: enabled)  
   Active: active (running) since Thu 2022-10-13 18:46:43 MSK; 3s ago  
     Docs: man:httpd.service(8)  
  Main PID: 9732 (httpd)  
    Status: "Started, listening on: port 80"  
    Tasks: 213 (limit: 37644)  
   Memory: 45.0M  
    CGroup: /system.slice/httpd.service  
           └─9732 /usr/sbin/httpd -DFOREGROUND  
             └─9733 /usr/sbin/httpd -DFOREGROUND  
               └─9734 /usr/sbin/httpd -DFOREGROUND  
                 └─9735 /usr/sbin/httpd -DFOREGROUND  
                   └─9736 /usr/sbin/httpd -DFOREGROUND  
  
Oct 13 18:46:43 vsfeoktistov.localdomain systemd[1]: Starting The Apache HTTP S>  
Oct 13 18:46:43 vsfeoktistov.localdomain systemd[1]: Started The Apache HTTP Se>  
Oct 13 18:46:43 vsfeoktistov.localdomain httpd[9732]: Server configured, listen>  
lines 1-18/18 (END)
```

Рис. 5.2: Запуск и проверка работы сервера Apache

После этого можно будет обратиться к веб-сервису через любой браузер. Для этого нужно будет ввести в адресной строке браузера локальный адрес: *http://127.0.0.1* (рис. 5.3). В результате откроется сайт со справкой использования Apache сервера.

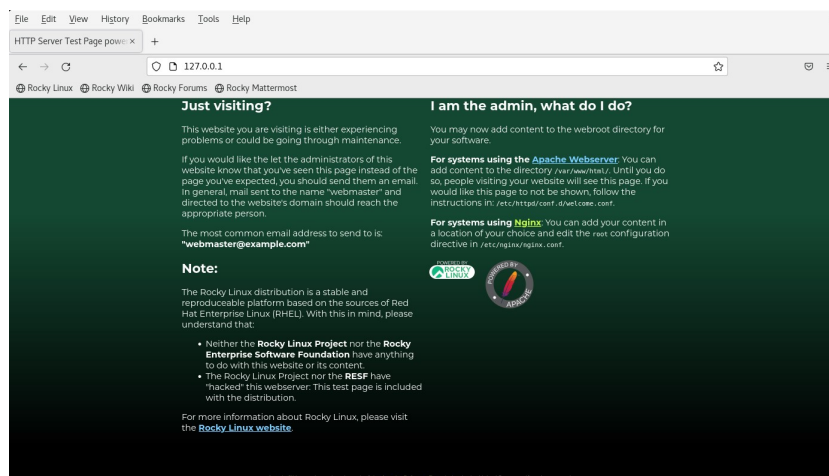
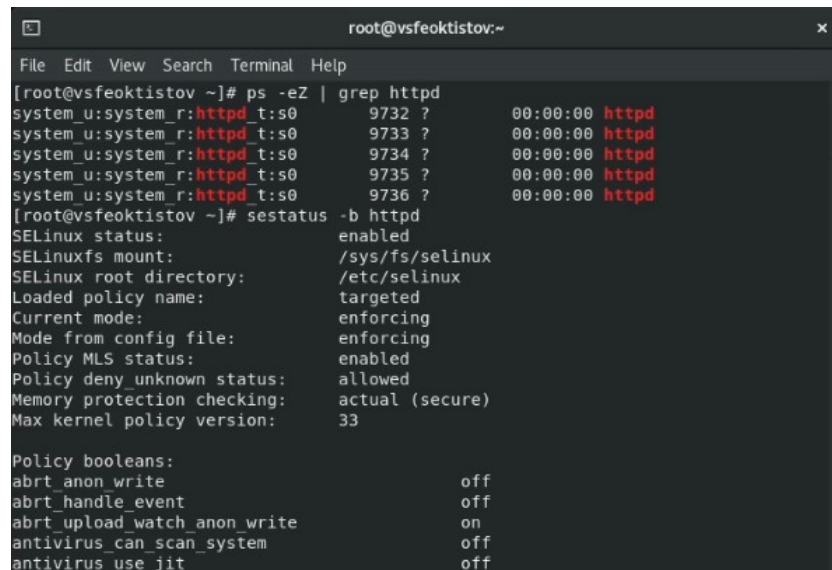


Рис. 5.3: Проверка работы веб-сервиса

Далее найдем веб-сервер Apache в списке процессов, определим его контекст безопасности [cmd: `ps -eZ | grep httpd`] и посмотрим текущее состояние переключателей SELinux [cmd: `sestatus -b httpd`]. Как видно из рисунка 5.4, контекст



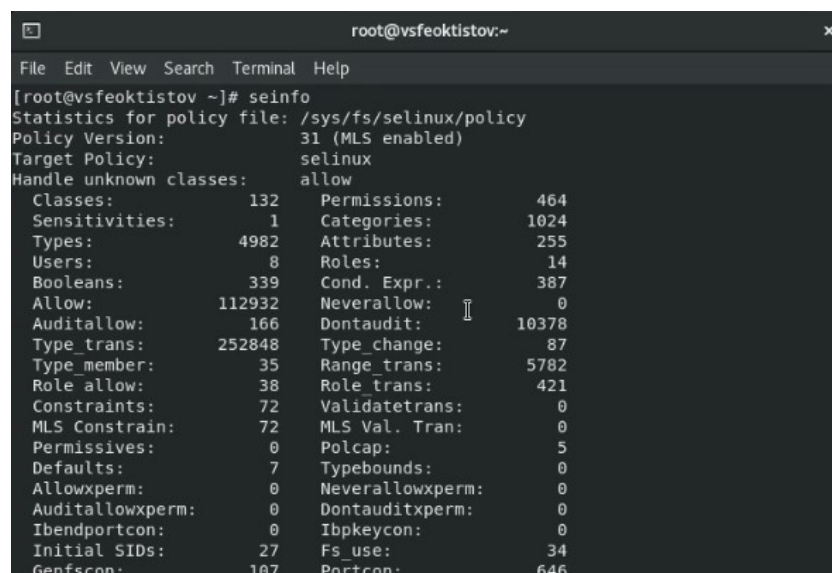
безопасности веб-сервера - `system_u:system_r:httpd_t:s0`, а многие переключатели находятся в положении “off”.



```
root@vsfeoktistov:~  
File Edit View Search Terminal Help  
[root@vsfeoktistov ~]# ps -eZ | grep httpd  
system_u:system_r:httpd_t:s0      9732 ?        00:00:00 httpd  
system_u:system_r:httpd_t:s0      9733 ?        00:00:00 httpd  
system_u:system_r:httpd_t:s0      9734 ?        00:00:00 httpd  
system_u:system_r:httpd_t:s0      9735 ?        00:00:00 httpd  
system_u:system_r:httpd_t:s0      9736 ?        00:00:00 httpd  
[root@vsfeoktistov ~]# sestatus -b httpd  
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  
Memory protection checking:     actual (secure)  
Max kernel policy version:      33  
  
Policy booleans:  
abrt_anon_write                 off  
abrt_handle_event               off  
abrt_upload_watch_anon_write    on  
antivirus_can_scan_system       off  
antivirus_use_jit               off
```

Рис. 5.4: Контекст безопасности процесса веб-сервера и состояние переключателей SELinux

С помощью команды `seinfo` определим, что в политике SELinux всего 8 пользователя, 14 ролей и 4982 типа (рис. 5.5).

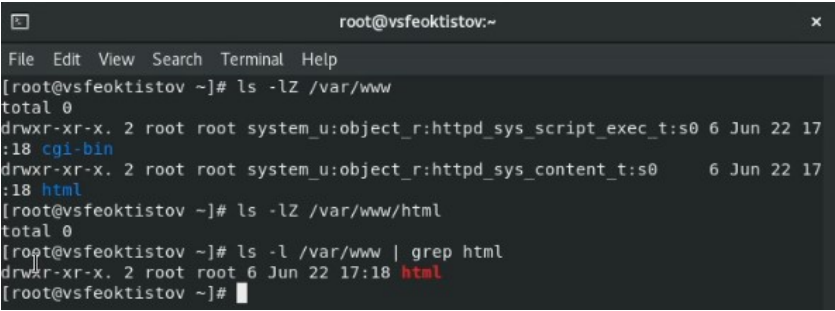


```
root@vsfeoktistov:~  
File Edit View Search Terminal Help  
[root@vsfeoktistov ~]# seinfo  
Statistics for policy file: /sys/fs/selinux/policy  
Policy Version:                 31 (MLS enabled)  
Target Policy:                  selinux  
Handle unknown classes:        allow  
Classes:                        132  
Sensitivities:                  1  
Types:                          4982  
Users:                          8  
Booleans:                       339  
Allow:                          112932  
Auditallow:                     166  
Type_trans:                     252848  
Type_member:                    35  
Role_allow:                     38  
Constraints:                    72  
MLS Constrain:                  72  
Permissives:                    0  
Defaults:                       7  
Allowxperm:                     0  
Auditallowxperm:               0  
Ibendportcon:                  0  
Initial SIDs:                   27  
Genfscon:                      107  
Permissions:                   464  
Categories:                   1024  
Attributes:                    255  
Roles:                         14  
Cond. Expr.:                   387  
Neverallow:                     0  
Dontaudit:                     10378  
Type_change:                    87  
Range_trans:                   5782  
Role_trans:                     421  
Validatetrans:                  0  
MLS Val. Tran:                 0  
Polcap:                         5  
Typebounds:                     0  
Neverallowxperm:               0  
Dontauditxperm:                0  
Ibpkeycon:                     0  
Fs_use:                        34  
Portcon:                       646
```

Рис. 5.5: Статистика по политике

С помощью команды `ls -lZ /var/www` узнаем, что тип директории `/var/www/cgi-`

*bin* - *httpd\_sys\_script\_exec\_t*, а директории *var/www/html* - *httpd\_sts\_content\_t*. Директория *var/www/html* не содержит никаких файлов и каталогов. Кроме того, видно, что только *root*-пользователь может создавать файлы в директории *var/www/html*, поскольку только не есть для этого минимальный набор прав: право на запись и исполнения для каталога (рис. 5.6). Минимальный набор прав мы определили во 2ой лабораторной работе (рис. 5.7).



```
root@vsfeoktistov:~  
File Edit View Search Terminal Help  
[root@vsfeoktistov ~]# ls -lZ /var/www  
total 0  
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Jun 22 17  
:18 cgi-bin  
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 Jun 22 17  
:18 html  
[root@vsfeoktistov ~]# ls -lZ /var/www/html  
total 0  
[root@vsfeoktistov ~]# ls -l /var/www | grep html  
drwxr-xr-x. 2 root root 6 Jun 22 17:18 html  
[root@vsfeoktistov ~]#
```

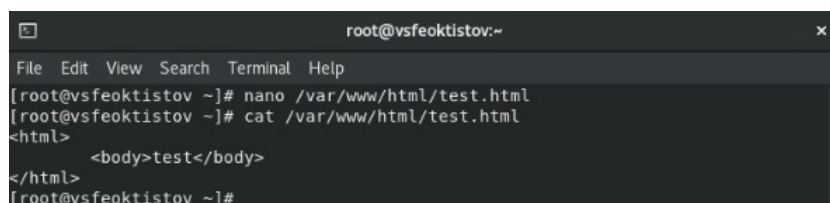
Рис. 5.6: Типы поддиректорий и определение пользователя с правами создания файла

Таблица 4.2: Минимальные права для совершения операций

Операция	Минимальные права на директорию	Минимальные права на файл
Создание файла	d-wx (300)	— (000)

Рис. 5.7: Таблица с минимальными правами для совершения операций из 2ой лаб. работы

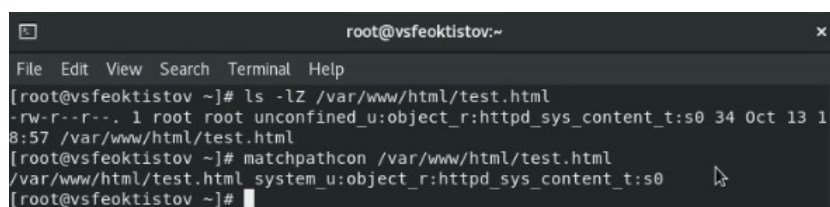
От имени *root*-пользователя (т.к. в дистрибутиве после установки только ему разрашена запись в директории */var/www/html*) создадим *html*-файл */var/www/html/test.html* (рис. 5.8). Можно либо сначала создать файл с помощью команды *touch*, а потом уже его отредактировать, либо можно сразу запустить *папо* редактор, по указанному пути. Если по указанному пути не будет текстового файла, то он автоматически создастся редактором.



```
root@vsfeoktistov:~  
File Edit View Search Terminal Help  
[root@vsfeoktistov ~]# nano /var/www/html/test.html  
[root@vsfeoktistov ~]# cat /var/www/html/test.html  
<html>  
  <body>test</body>  
</html>  
[root@vsfeoktistov ~]#
```

Рис. 5.8: Создание html-файла

Проверим контекст созданного файла [**cmd**: `matchpathcon /var/www/html/test.html`]. Как видно из рисунка 5.9, `system_u:object_r:httpd_sys_content_t:s0` - это присваиваемый по умолчанию контекст у вновь созданных файлов в директории `/var/www/html` и он полностью совпадает с контекстом безопасности каталога `/var/www/html` (рис. 5.6).



```
root@vsfeoktistov:~  
File Edit View Search Terminal Help  
[root@vsfeoktistov ~]# ls -lZ /var/www/html/test.html  
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 34 Oct 13 18:57 /var/www/html/test.html  
[root@vsfeoktistov ~]# matchpathcon /var/www/html/test.html  
/var/www/html/test.html system_u:object_r:httpd_sys_content_t:s0  
[root@vsfeoktistov ~]#
```

Рис. 5.9: Контекст по умолчанию

Обратимся к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`, и убедимся, что файл был успешно отображен (рис. 5.10).

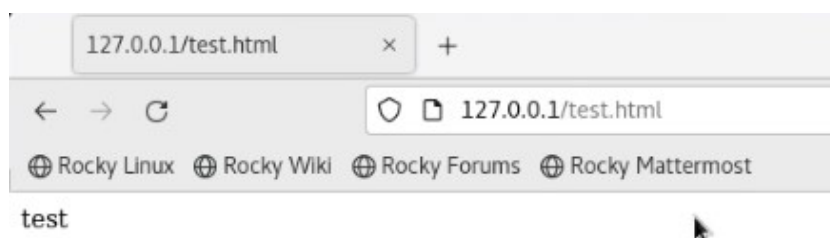


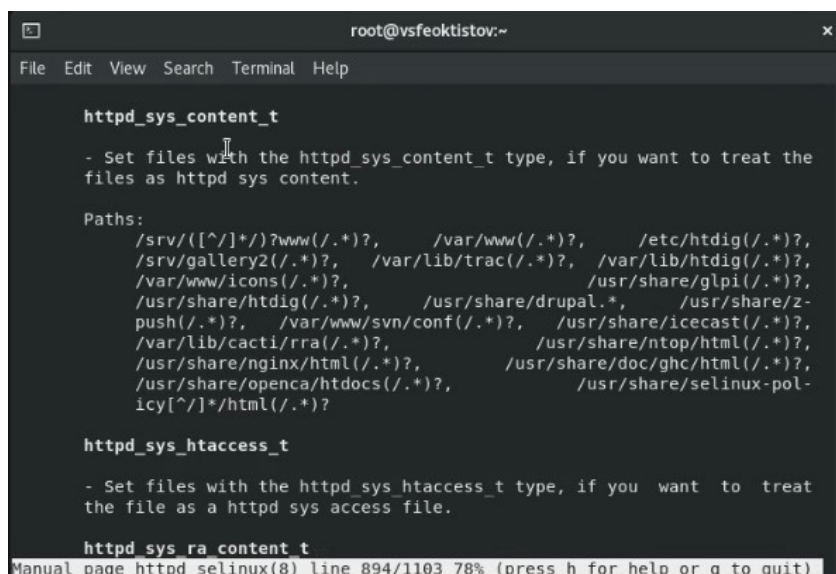
Рис. 5.10: Обращение к файлу через веб-сервер

Далее из справки [**cmd**: `man httpd_selinux`] узнаем, что для `httpd` определены следующие контексты файлов: `httpd_sys_content_t`, `httpd_sys_script_exec_t`, `httpd_script_ro_t`, `httpd_sys_script_rw_t`, `httpd_sys_script_ra_t`, `httpd_unconfined_script_exec_t`,

а также узнаем, что у файла *test.html* контекст файла есть в этом списке - *httpd\_sys\_content\_t* [cmd: *ls -Z /var/www/html/test.html*] (рис. 5.11 и рис. 5.12).

```
httpd_t, httpd_helper_t, httpd_php_t, httpd_rotate_logs_t, httpd_suexec_t,
httpd_sys_script_t, httpd_user_script_t, httpd_passwd_t, httpd_unconfined_scrip
t_t
```

Рис. 5.11: Справка *httpd\_selinux*



```
root@vsfeoktistov:~
File Edit View Search Terminal Help

httpd_sys_content_t

- Set files with the httpd_sys_content_t type, if you want to treat the
files as httpd sys content.

Paths:
/srv/([^\s]*)?www(/.*)?, /var/www(/.*)?, /etc/htdig(/.*)?,
/srv/gallery2(/.*)?, /var/lib/trac(/.*)?, /var/lib/htdig(/.*)?,
/var/www/icons(/.*)?, /usr/share/glpi(/.*)?,
/usr/share/htdig(/.*)?, /usr/share/drupal.*, /usr/share/z-
push(/.*)?, /var/www/svn/conf(/.*)?, /usr/share/icecast(/.*)?,
/var/lib/cacti/rra(/.*)?, /usr/share/ntop/html(/.*)?,
/usr/share/nginx/html(/.*)?, /usr/share/doc/ghc/html(/.*)?,
/usr/share/openca/htdocs(/.*)?, /usr/share/selinux-pol-
icy[^\s]*/html(/.*)?

httpd_sys_htaccess_t

- Set files with the httpd_sys_htaccess_t type, if you want to treat
the file as a httpd sys access file.

httpd_sys_ra_content_t

Manual page httpd_selinux(8) line 894/1103 78% (press h for help or q to quit)
```

Рис. 5.12: Справка *httpd\_selinux*

Рассмотрим полученный контекст детально. Так как по умолчанию пользователи Rocky Linux являются свободными от типа (*unconfined* в переводе с англ. означает свободный), созданному нами файлу *test.html* был сопоставлен SELinux, пользователь *unconfined\_u*.

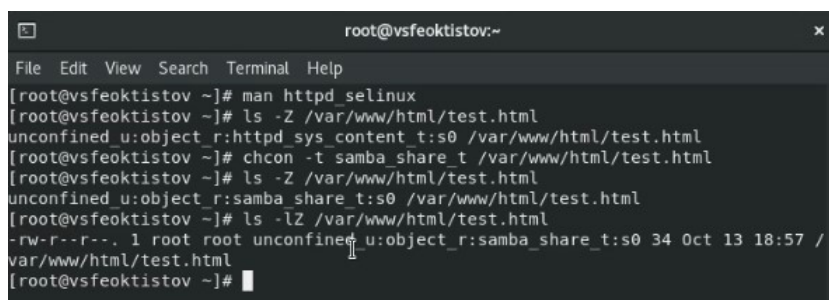
Это первая часть контекста.

Далее политика ролевого разделения доступа RBAC используется 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. (В директории */proc* 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 рассмат-

ривать не будем, как и предназначение :s0).

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.

С помощью команды `chcon -t samba_share_t /var/www/html/test.html` изменим контекст файла `/var/www/html/test.html` с `httpd_sys_content_t` на `samba_share_t`, к которому процесс `httpd` не должен иметь доступа. Проверить, что контекст поменялся, можно с помощью команды `ls -Z /var/www/html/test.html` (рис. 5.13).



```
root@vsfeoktistov:~  
File Edit View Search Terminal Help  
[root@vsfeoktistov ~]# man httpd_selinux  
[root@vsfeoktistov ~]# ls -Z /var/www/html/test.html  
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html  
[root@vsfeoktistov ~]# chcon -t samba_share_t /var/www/html/test.html  
[root@vsfeoktistov ~]# ls -Z /var/www/html/test.html  
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html  
[root@vsfeoktistov ~]# ls -lZ /var/www/html/test.html  
-rw-r--r--. 1 root root unconfined_u:object_r:samba_share_t:s0 34 Oct 13 18:57 /  
var/www/html/test.html  
[root@vsfeoktistov ~]#
```

Рис. 5.13: Изменение контекста файла

Попробуем еще раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. В итоге получаем следующее сообщение об ошибке: *Forbidden You don't have permission to access this resource* (рис. 5.14). Это произошло, поскольку для файла `test.html` мы установили контекст файла, к которому процесс `httpd` не имеет доступа.

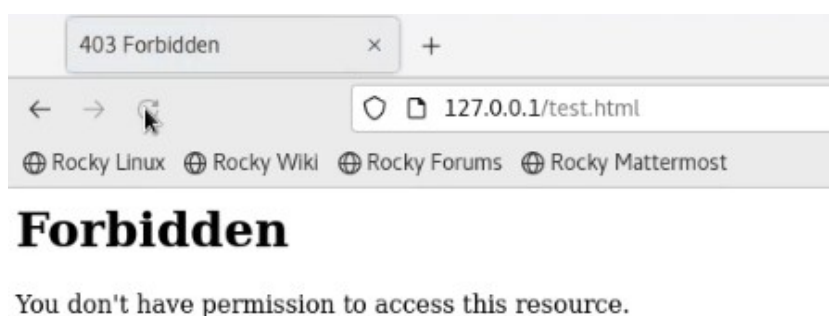


Рис. 5.14: Обращение к файлу через веб-сервер при измененном контексте

Посмотрим log-файлы веб-сервера Apache (рис. 5.15 и 5.16).

```
[root@vsfeoktistov ~]# ls -l /var/log/httpd
total 8
-rw-r--r--. 1 root root 633 Oct 13 19:28 access_log
-rw-r--r--. 1 root root 1121 Oct 13 19:28 error_log
[root@vsfeoktistov ~]# tail /var/log/httpd/error_log
[Thu Oct 13 18:46:43.649357 2022] [core:notice] [pid 9732:tid 139781220501824] S
ELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Thu Oct 13 18:46:43.651219 2022] [suexec:notice] [pid 9732:tid 139781220501824]
AH01232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Thu Oct 13 18:46:43.703605 2022] [lbmethod heartbeat:notice] [pid 9732:tid 1397
81220501824] AH02282: No slotmem from mod_heartbeat
[Thu Oct 13 18:46:43.704964 2022] [http2:warn] [pid 9732:tid 139781220501824] AH
02951: mod_ssl does not seem to be enabled
[Thu Oct 13 18:46:43.715535 2022] [mpm event:notice] [pid 9732:tid 1397812205018
24] AH00489: Apache/2.4.37 (rocky) configured -- resuming normal operations
[Thu Oct 13 18:46:43.715611 2022] [core:notice] [pid 9732:tid 139781220501824] A
H00094: Command line: '/usr/sbin/httpd -D FOREGROUND'
[Thu Oct 13 19:28:49.598194 2022] [core:error] [pid 9734:tid 139780197127936] (1
3)Permission denied: [client 127.0.0.1:56992] AH00035: access to /test.html deni
ed (filesystem path '/var/www/html/test.html') because search permissions are mi
ssing on a component of the path
[root@vsfeoktistov ~]#
```

Рис. 5.15: Log-файл error\_log

```
root@vsfeoktistov:~
File Edit View Search Terminal Help
Oct 13 19:29:00 vsfeoktistov setroubleshoot[15624]: failed to retrieve rpm info
for /var/www/html/test.html
Oct 13 19:29:00 vsfeoktistov setroubleshoot[15624]: SELinux is preventing /usr/s
bin/httpd from getattr access on the file /var/www/html/test.html. For complete
SELinux messages run: sealert -l 16474a63-8dce-45a6-a438-72clee5b9e
Oct 13 19:29:00 vsfeoktistov setroubleshoot[15624]: SELinux is preventing /usr/s
bin/httpd from getattr access on the file /var/www/html/test.html.#012#012****
Plugin restorecon (92.2 confidence) suggests *****#012#012
If you want to fix the label. #012/var/www/html/test.html default label should b
e httpd_sys_content_t.#012Then you can run restorecon. The access attempt may ha
ve been stopped due to insufficient permissions to access a parent directory in
which case try to change the following command accordingly.#012Do#012# /sbin/res
torecon -v /var/www/html/test.html#012#012***** Plugin public_content (7.83 con
fidence) suggests *****#012#012If you want to treat test.html a
s public content#012Then you need to change the label on test.html to public co
ntent_t or public_content_rw_t.#012Do#012# semanage fcontext -a -t public_content
_t '/var/www/html/test.html'#012# restorecon -v '/var/www/html/test.html'#012#01
2***** Plugin catchall (1.41 confidence) suggests *****#012#012
If you believe that httpd should be allowed getattr access on the test.ht
ml file by default.#012Then you should report this as a bug.#012You can generate
a local policy module to allow this access.#012Do#012allow this access for now
by executing:#012# ausearch -c 'httpd' --raw | audit2allow -M my-httpd#012# semo
dule -X 300 -i my-httpd.pp#012
[root@vsfeoktistov ~]#
```

Рис. 5.16: Log-файл messages

Из логов видно, что для файла *test.html* нужно установить контекст *httpd\_sys\_content\_t*.

Попробуем запустить веб-сервер Apache на прослушивание TCP-порта 81, а не 80, как рекомендует IANA и прописано в */etc/services*. Для этого в файле */etc/httpd/conf/httpd.conf* найдем строчку *Listen 80* и заменим её на *Listen 81* (поиск лучше производить через зажатие клавиш *Ctrl + w*), а после перезапустим Apache [**cmd: *systemctl restart httpd***] (рис. 5.17).



```
root@vsfeoktistov:~  
File Edit View Search Terminal Help  
[root@vsfeoktistov ~]# nano /etc/httpd/conf/httpd.conf  
[root@vsfeoktistov ~]# cat /etc/httpd/conf/httpd.conf | grep Listen  
# Listen: Allows you to bind Apache to specific IP addresses and/or  
# Change this to Listen on specific IP addresses as shown below to  
#Listen 12.34.56.78:80  
Listen 81  
[root@vsfeoktistov ~]# systemctl restart httpd  
[root@vsfeoktistov ~]#
```

Рис. 5.17: Установка TCP-порта 81

При попытке запустить веб-сервер под старым адресом, получим ошибку: *Unable to connect* (рис. 5.18). Если мы пишем адрес *http://127.0.0.1/test.html*, то по умолчанию подключение идет по 80 порту, т.е. этот адрес эквивалентен адресу *http://127.0.0.1:80/test.html*. После смены TCP-порта с 80 на 81, подключаться к веб-серверу нужно по адресу *http://127.0.0.1:81/test.html*, явно указывая порт.

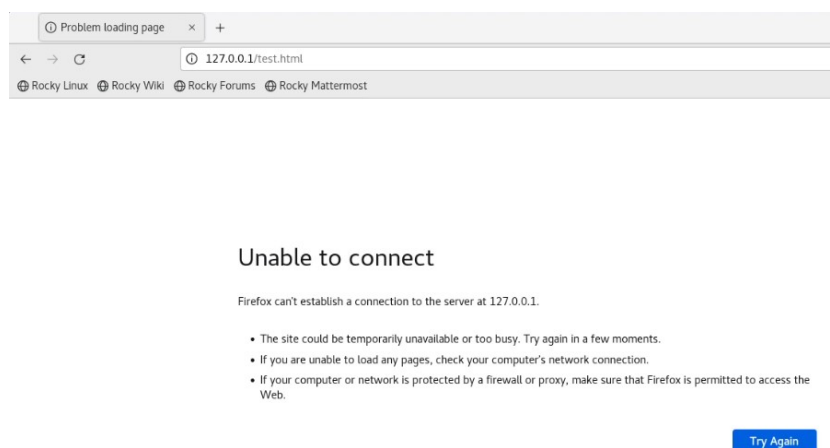


Рис. 5.18: Запуск веб-сервера под старым адресом, но с новым портом

Проанализируем лог-файлы */var/log/messages*, */var/log/http/error\_log*, */var/log/http/access\_log* и */var/log/audit/audit.log*. Как видно из рисунка 5.19, в файлах *messages* и *error\_log* появились новые записи. Для просмотра удобно использовать команду *tail -n 1* для просмотра последней строки.

```
[root@vsfeoktistov ~]# tail -n1 /var/log/messages
Oct 13 19:40:47 vsfeoktistov org.gnome.Shell.desktop[2020]: libinput error: even
t2 - AT Translated Set 2 keyboard: client bug: event processing lagging behind
by 28ms, your system is too slow
[root@vsfeoktistov ~]# tail -n1 /var/log/httpd/error_log
[Thu Oct 13 19:39:40.314385 2022] [core:notice] [pid 15788:tid 139820574599488]
AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'
[root@vsfeoktistov ~]# tail -n1 /var/log/httpd/access_log
127.0.0.1 - - [13/Oct/2022:19:28:49 +0300] "GET /test.html HTTP/1.1" 403 199 "-"
"Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0"
[root@vsfeoktistov ~]# tail -n1 /var/log/audit/audit.log
type=SERVICE_START msg=audit(1665679180.267:286): pid=1 uid=0 auid=4294967295 se
s=4294967295 subj=system_u:system_r:init t:s0 msg='unit=httpd comm="systemd" exe
="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success' ID="root"
AUID="unset"
[root@vsfeoktistov ~]#
```

Рис. 5.19: Анализ лог-файлов

Однако, перед этим рекомендуется добавить этот порт в список портов с помощью команды `semanage port -a -t http_port_t -p tcp 81`. Проверить наличие порта в списке можно командой `semanage port -l | grep http_port_t`. По умолчанию порт 81 уже добавлен в список ТСП-портов Apache (рис. 5.20).

```
root@vsfeoktistov:~
File Edit View Search Terminal Help
[root@vsfeoktistov ~]# semanage port -a -t http_port_t -p tcp 81
ValueError: Port tcp/81 already defined
[root@vsfeoktistov ~]# semanage port -l | grep http_port_t
http_port_t      tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t  tcp      5988
[root@vsfeoktistov ~]# systemctl restart httpd
```

Рис. 5.20: Добавление ТСП-порта

Теперь же, если зайти на веб-сервер по адресу `http://127.0.0.1:81/test.html`, то мы снова сможем подключиться к серверу и получить ошибку *Forbidden You don't have permission to access this resource*, т.к. контекст файла `test.html` мы еще пока не изменили на нужный (рис. 5.21).

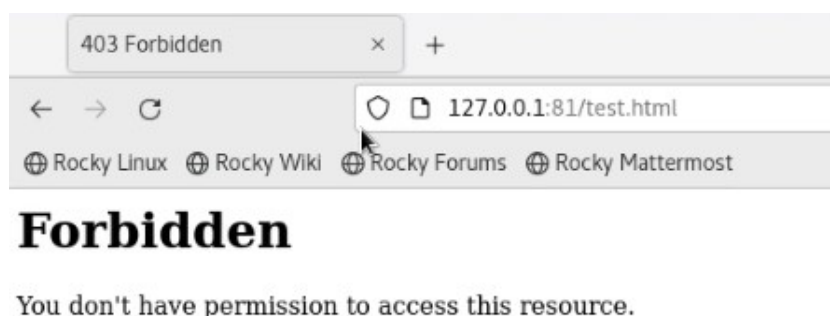
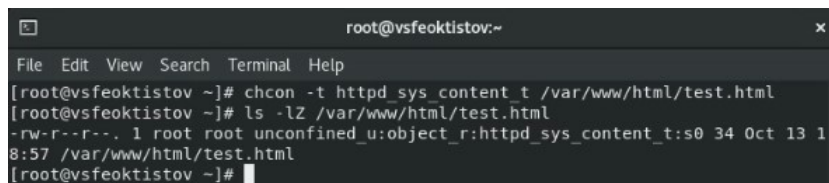


Рис. 5.21: Запуск веб-сервера под портом 81



Вернем контекст `httpd_sys_content_t` к файлу `/var/www/html/test.html` командой `chcon -t httpd_sys_content_t /var/www/html/test.html` (рис. 5.22) и попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html` (рис. 5.23).



```
root@vsfeoktistov:~  
File Edit View Search Terminal Help  
[root@vsfeoktistov ~]# chcon -t httpd_sys_content_t /var/www/html/test.html  
[root@vsfeoktistov ~]# ls -lZ /var/www/html/test.html  
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 34 Oct 13 1  
8:57 /var/www/html/test.html  
[root@vsfeoktistov ~]#
```

Рис. 5.22: Возвращение контекста

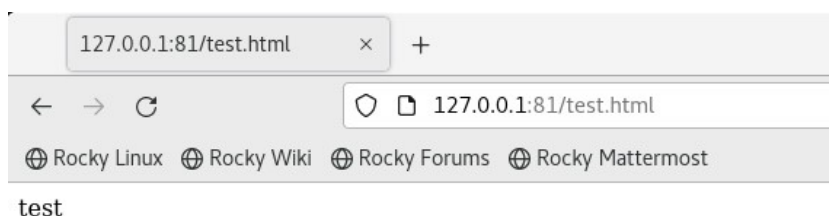


Рис. 5.23: Запуск веб-сервера под портом 81 с новым контекстом

Как можно заметить, теперь наконец-то файл `test.html` нормально отображается на веб-сервере под 81 TCP-портом.

Перед завершением работы, исправим конфигурационный файл Apache, вернув `Listen 80`, удалим привязку `http_port` к 81 порту (81 порт из списка нельзя удалить, поскольку по политике он установлен там по умолчанию) и удалим файл `/var/www/html/test.html` (рис. 5.24).

```
root@vsfeoktistov:~  
File Edit View Search Terminal Help  
[root@vsfeoktistov ~]# nano /etc/httpd/conf/httpd.conf  
[root@vsfeoktistov ~]# cat /etc/httpd/conf/httpd.conf | grep Listen  
# Listen: Allows you to bind Apache to specific IP addresses and/or  
# Change this to Listen on specific IP addresses as shown below to  
#Listen 12.34.56.78:80  
Listen 80  
[root@vsfeoktistov ~]# semanage port -d -t http_port_t -p tcp 81  
ValueError: Port tcp/81 is defined in policy, cannot be deleted  
[root@vsfeoktistov ~]# semanage port -l | grep http_port_t  
http_port_t tcp 80, 81, 443, 488, 8008, 8009, 8443, 9000  
pegasus_http_port_t tcp 5988  
[root@vsfeoktistov ~]# rm -v /var/www/html/test.html  
rm: remove regular file '/var/www/html/test.html'? y  
removed '/var/www/html/test.html'  
[root@vsfeoktistov ~]#
```

Рис. 5.24: Завершение лабораторной работы

## 6 Выводы

В процессе выполнения лабораторной работы:

- развил навыки администрирования ОС Linux;
- получил первое практическое знакомство с технологией SELinux;
- проверил работу SELinux на практике совместно с веб-сервером Apache.

## Список литературы

1. SELinux – описание и особенности работы с системой. Часть 1 [Электронный ресурс]. habr, 2014. URL: <https://habr.com/ru/company/kingservers/blog/209644/>.
2. Что такое Apache [Электронный ресурс]. 2domains. URL: <https://2domains.ru/support/vps-i-servery/shto-takoye-apache>.
3. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: <https://www.gnu.org/software/bash/manual/>.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
5. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
6. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
7. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
8. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.