Лабораторная работа №6

Информационная безопасность

Маметкадыров Ынтымак | НПМбд-02-20

Содержание

Список иллюстраций

1 Цель работы

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

2 Теоретическое введение

SELinux (Security-Enhanced Linux) обеспечивает усиление защиты путем внесения изменений как на уровне ядра, так и на уровне пространства пользователя, что превращает ее в действительно «непробиваемую» операционную систему. Впервые эта система появилась в четвертой версии CentOS, а в 5 и 6 версии реализация была существенно дополнена и улучшена.

SELinux имеет три основных режим работы:

- 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.

Политика SELinux определяет доступ пользователей к ролям, доступ ролей к доменам и доступ доменов к типам.

Контекст безопасности — все атрибуты SELinux — роли, типы и домены.

Более подробно см. в [1].

Apache — это свободное программное обеспечение, с помощью которого можно создать веб-сервер. Данный продукт возник как доработанная версия другого HTTP-клиента от национального центра суперкомпьютерных приложений (NCSA).

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.

Более подробно см. в [2].

3 Выполнение лабораторной работы

Вошли в систему под своей учетной записью и убедились, что SELinux работает в режиме enforcing политики targeted с помощью команд “getenforce” и “sestatus” (рис. 3.1).

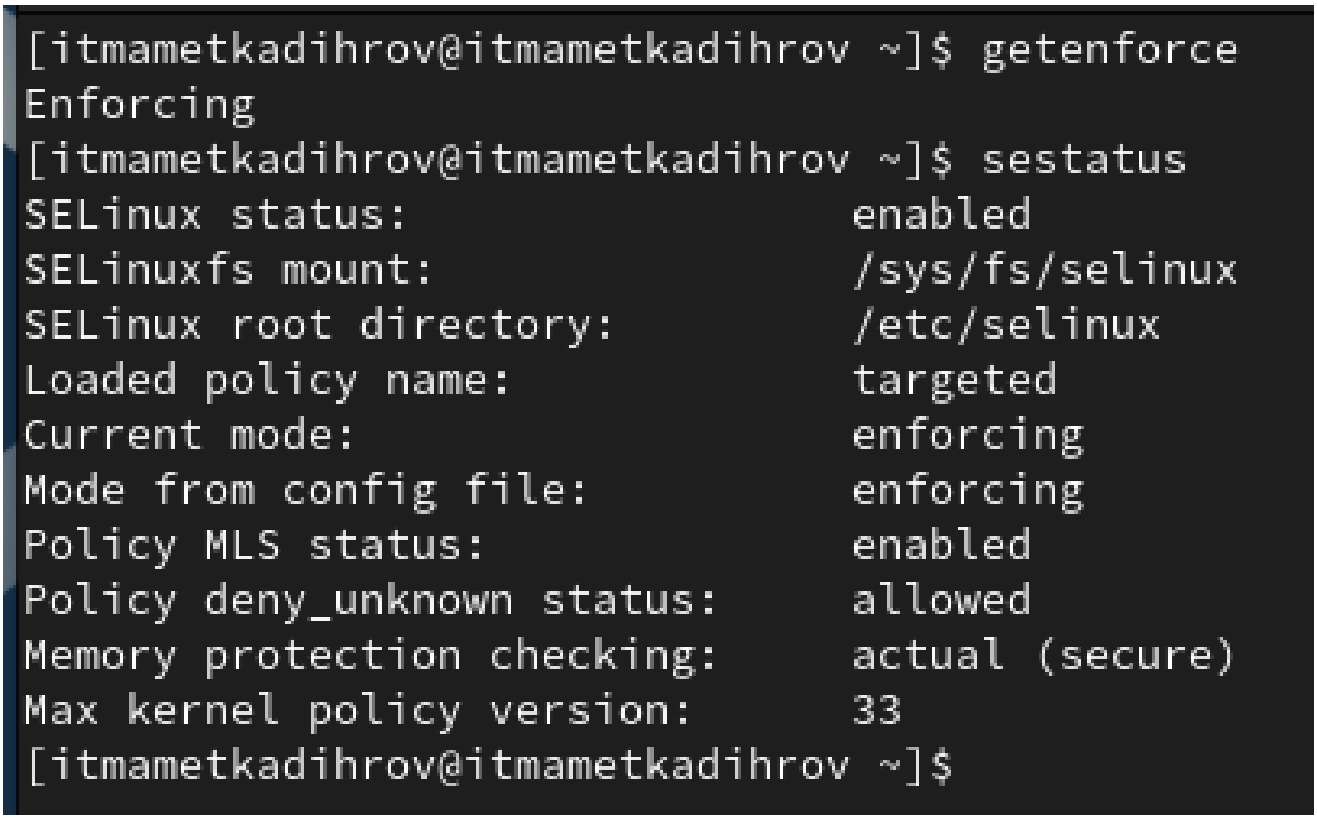

Рис. 3.1: Проверка режима enforcing политики targeted

Обратились с помощью браузера к веб-серверу, запущенному на нашем компьютере, и убедились, что последний работает с помощью команды “service httpd status” (рис. 3.2).

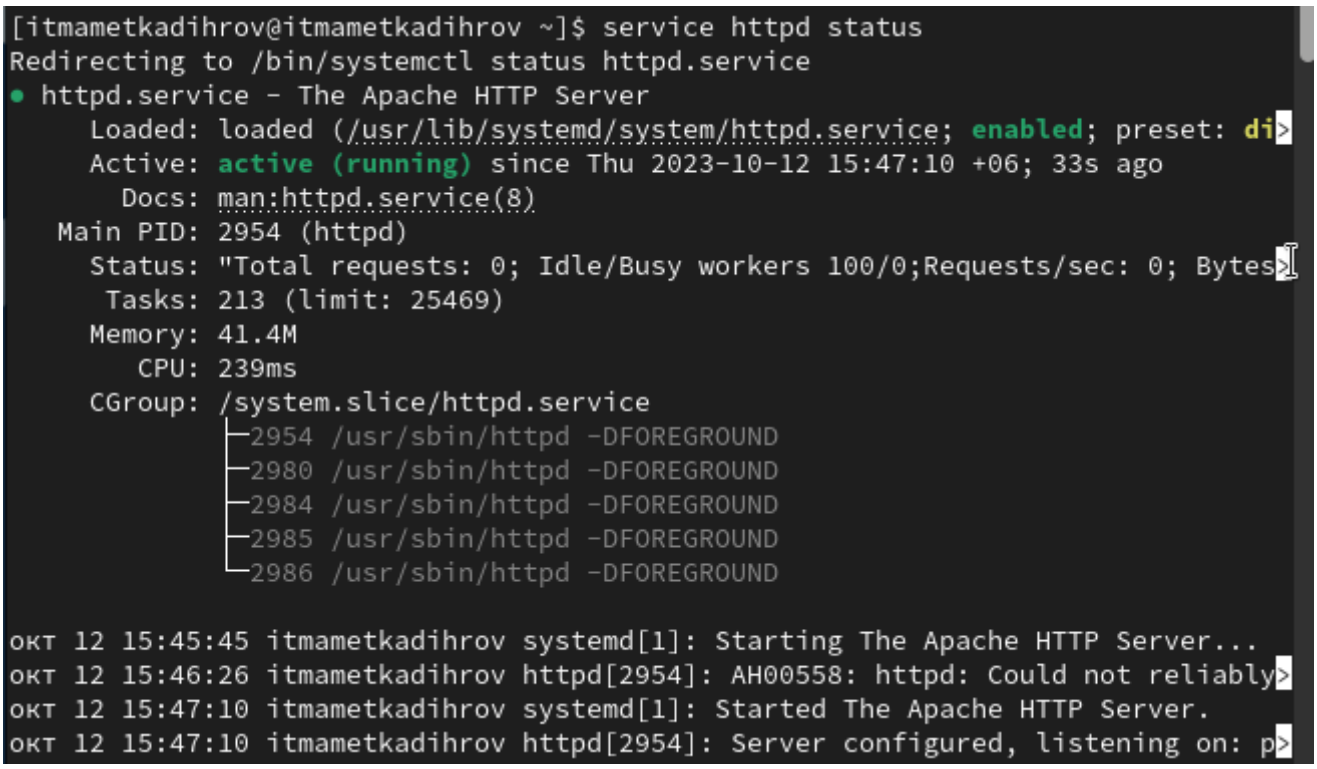

Рис. 3.2: Проверка работы веб-сервера

С помощью команды “ps auxZ | grep httpd” определили контекст безопасности веб-сервера Apache - httpd_t (рис. 3.3).

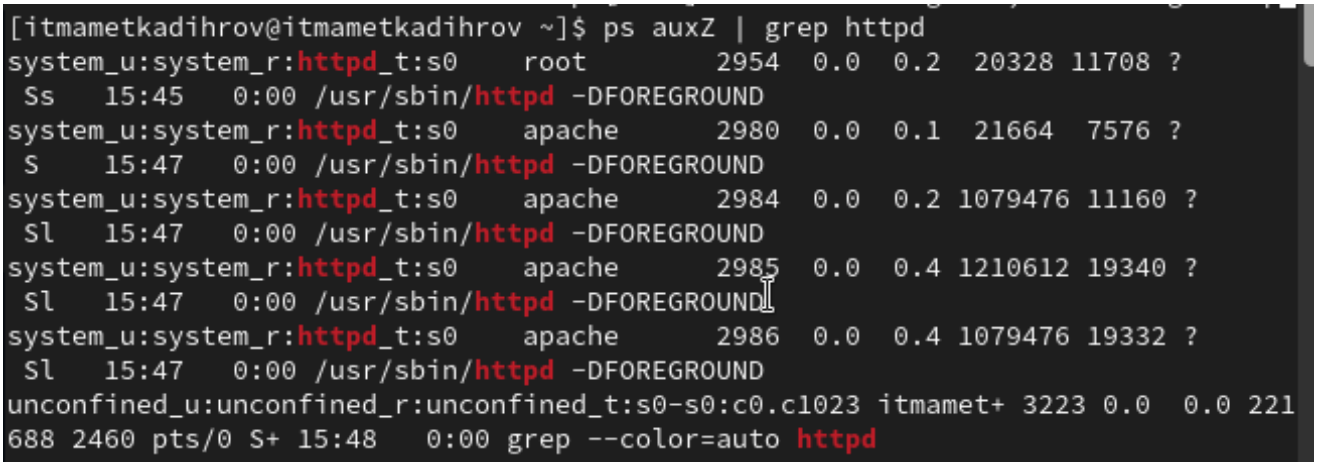

Рис. 3.3: Контекст безопасности веб-сервера Apache

Посмотрели текущее состояние переключателей SELinux для Apache с помощью команды “sestatus -bigrep httpd”, многие из переключателей находятся

в положении “off” (рис. 3.4).

```
[itmametkadihrov@itmametkadihrov ~]$ sestatus -bigrep httpd
sestatus: invalid option -- 'i'

Usage: sestatus [OPTION]

  -v  Verbose check of process and file contexts.
  -b  Display current state of booleans.

Without options, show SELinux status.
[itmametkadihrov@itmametkadihrov ~]$ 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
auditadm_exec_content                       on
authlogin_nsswitch_use_ldap                 off
authlogin_radius                            off
authlogin_yubikey                           off
```

Рис. 3.4: Текущее состояние переключателей SELinux

Посмотрели статистику по политике с помощью команды “seinfo”. Множество

пользователей - 8, ролей - 14, типов 4995 (рис. 3.5).

```
[itmametkadihrov@itmametkadihrov ~]$ seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:             33 (MLS enabled)
Target Policy:              selinux
Handle unknown classes:     allow
  Classes:             135    Permissions:          457
  Sensitivities:         1    Categories:          1024
  Types:              5100    Attributes:           258
  Users:                 8    Roles:                 14
  Booleans:            353    Cond. Expr.:          384
  Allow:             65008    Neverallow:             0
  Auditallow:          170    Dontaudit:           8572
  Type_trans:       265344    Type_change:           87
  Type_member:          35    Range_trans:         6164
  Role allow:           38    Role_trans:           420
  Constraints:          70    Validatetrans:          0
  MLS Constrain:        72    MLS Val. Tran:          0
  Permissives:           2    Polcap:                 6
  Defaults:              7    Typebounds:             0
  Allowxperm:            0    Neverallowxperm:        0
  Auditallowxperm:       0    Dontauditxperm:         0
  Ibendportcon:          0    Ibpkeycon:              0
  Initial SIDs:         27    Fs_use:                35
  Genfscon:            109    Portcon:              660
  Netifcon:              0    Nodecon:                0
[itmametkadihrov@itmametkadihrov ~]$
```

Рис. 3.5: Статистика по политике

С помощью команды “ls -lZ /var/www” посмотрели файлы и поддиректории, находящиеся в директории /var/www. Используя команду “ls -lZ /var/www/html”, определили, что в данной директории файлов нет. Только владелец/суперпользователь

может создавать файлы в директории /var/www/html (рис. 3.6).

```
[itmametkadihrov@itmametkadihrov ~]$ ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 мая 17 02:21 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0     6 мая 17 02:21 html
[itmametkadihrov@itmametkadihrov ~]$ ls -lZ /var/www/html
итого 0
[itmametkadihrov@itmametkadihrov ~]$
```

Рис. 3.6: Просмотр файлов и поддиректориий в директории /var/www

От имени суперпользователя создали html-файл /var/www/html/test.html. Контекст созданного файла - httpd_sys_content_t (рис. 3.7).

```
[itmametkadihrov@itmametkadihrov ~]$ su -
Пароль:
[root@itmametkadihrov ~]# touch /var/www/html/test.html
[root@itmametkadihrov ~]# nano /var/www/html/test.html
[root@itmametkadihrov ~]# cat /var/www/html/test.html
<html>
<body>test</body>
</html>
[root@itmametkadihrov ~]# su - itmametkadihrov
[itmametkadihrov@itmametkadihrov ~]$ ls -lZ /var/www/html
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 окт 13 18:11 test.html
[itmametkadihrov@itmametkadihrov ~]$
```

Рис. 3.7: Создание файла /var/www/html/test.html

Обратились к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html”. Файл был успешно отображен (рис. 3.8).

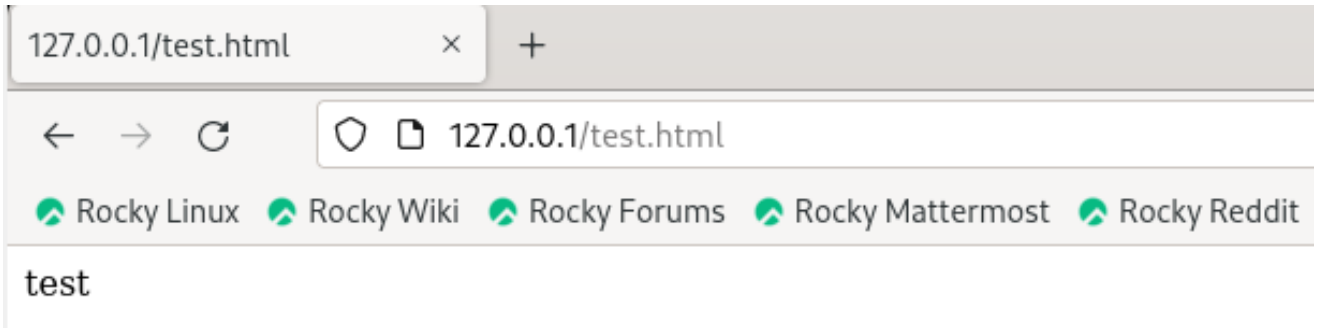

Рис. 3.8: Обращение к файлу через веб-сервер

Изучив справку man httpd_selinux, выяснили, что для httpd определены следующие контексты файлов: httpd_sys_content_t, httpd_sys_script_exec_t, httpd_sys_script_ro_t, httpd_sys_script_rw_t, httpd_sys_script_ra_t, httpd_unconfined_script_exec_t. Контекст моего файла - httpd_sys_content_t (в таком случае содержимое должно быть доступно для всех скриптов httpd и для самого демона). Изменили контекст файла на samba_share_t командой “sudo chcon -t samba_share_t /var/www/html/test.html” и проверили, что контекст поменялся (рис. 3.9).

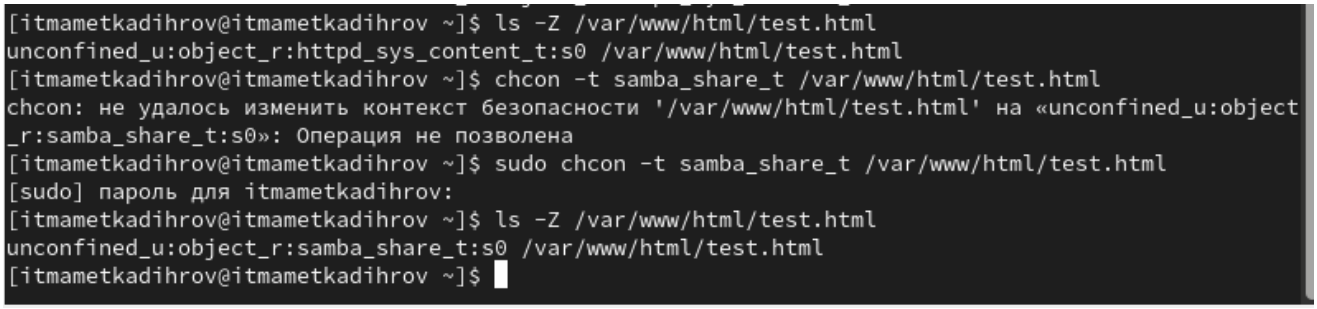

Рис. 3.9: Изменение контекста

Попробовали еще раз получить доступ к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html” и получили сообщение об ошибке (т.к. к установленному ранее контексту процесс httpd не имеет доступа) (рис. 3.10).

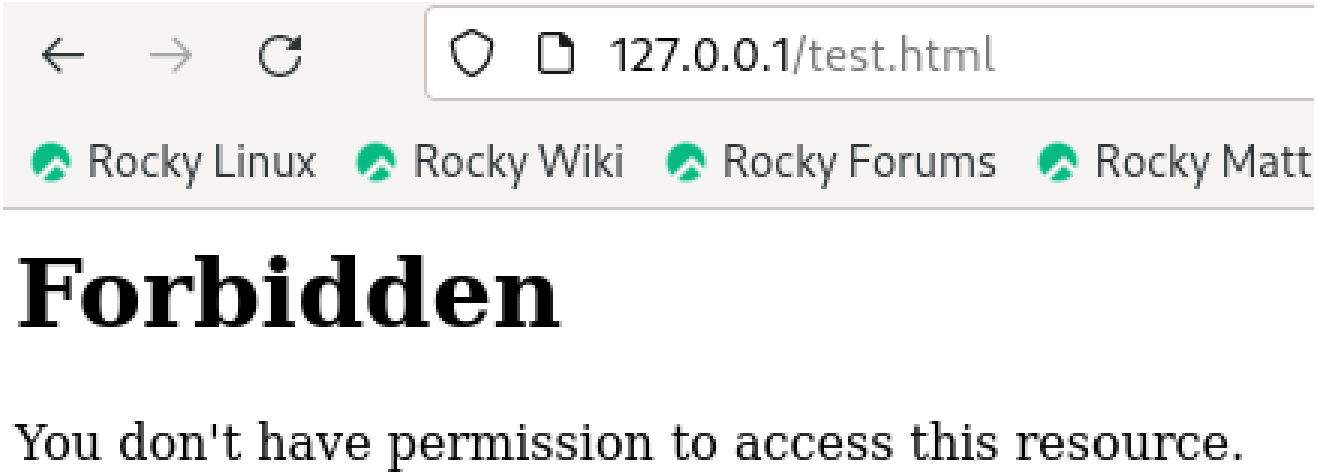

Рис. 3.10: Обращение к файлу через веб-сервер

Командой “ls -l /var/www/html/test.html” убедились, что читать данный файл

может любой пользователь. Просмотрели системный лог-файл веб-сервера Apache командой “sudo tail /var/log/messages”, отображающий ошибки (рис. 3.11).

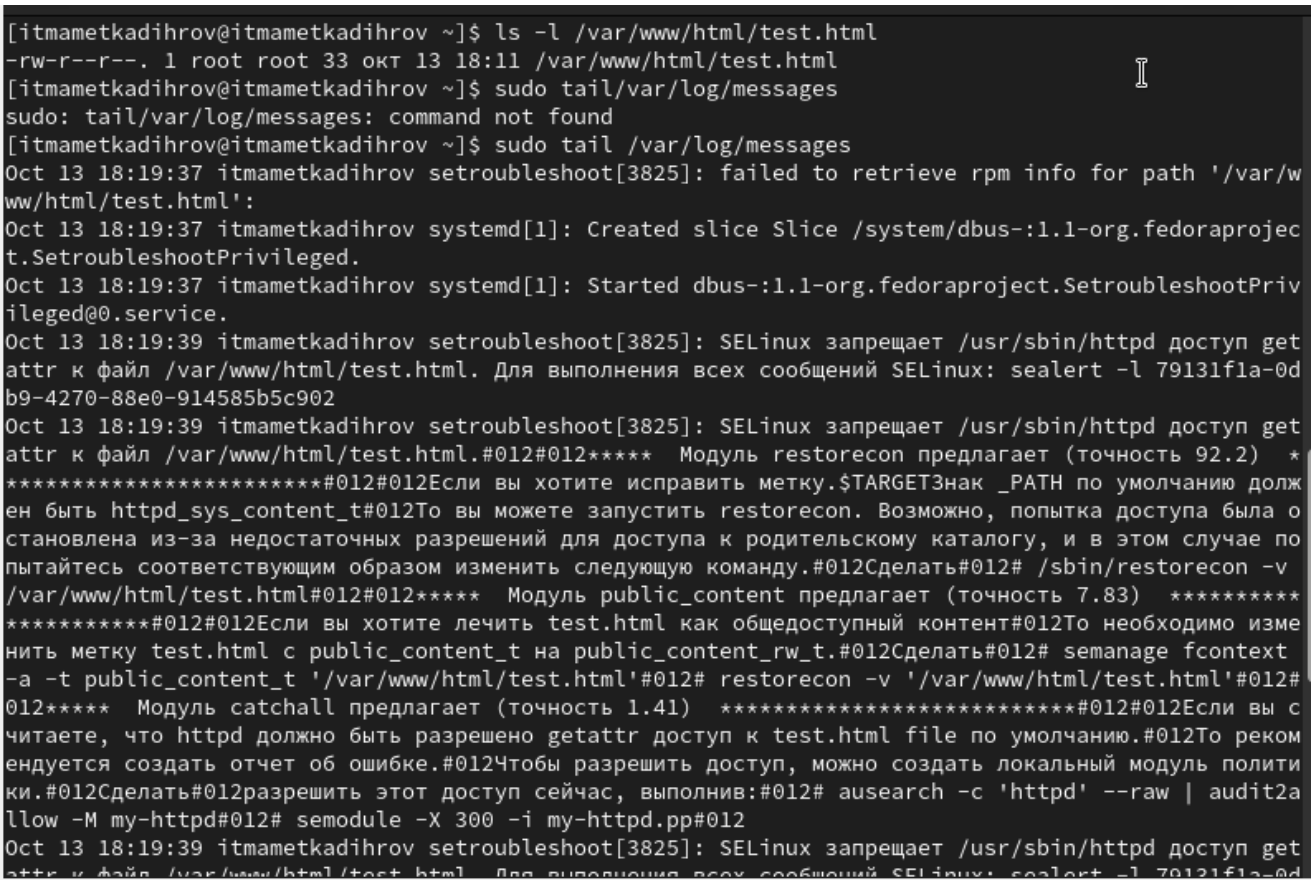

Рис. 3.11: Просмотр log-файла

В файле /etc/httpd/conf/httpd.conf заменили строчку “Listen 80” на “Listen 81”, чтобы установить веб-сервер Apache на прослушивание TCP-порта 81 (рис. 3.12).

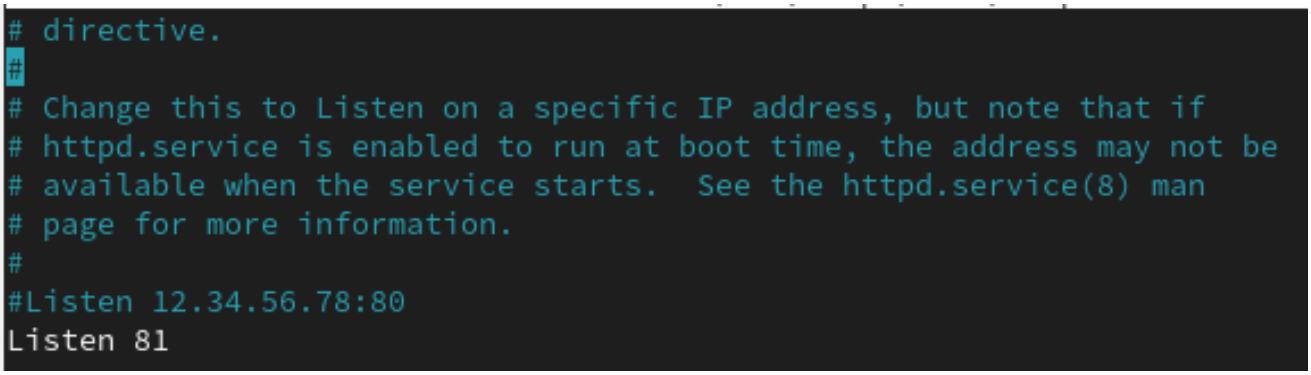

Рис. 3.12: Установка веб-сервера Apache на прослушивание TCP-порта 81

Перезапускаем веб-сервер Apache и анализирует лог-файлы командой “tail -nl /var/log/messages” (рис. 3.13).

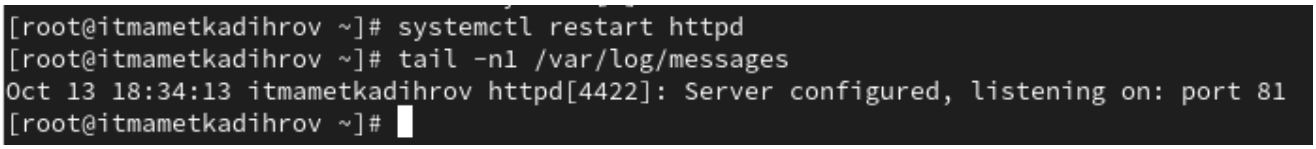

Рис. 3.13: Перезапуск веб-сервера и анализ лог-файлов

Выполнили команду “semanage port -a -t http_port_t -p tcp 81” и убедились, что порт TCP-81 установлен. Проверили список портов командой “semanage port -l | grep http_port_t”, убедились, что порт 81 есть в списке и запускаем веб-сервер Apache снова (рис. 3.14).

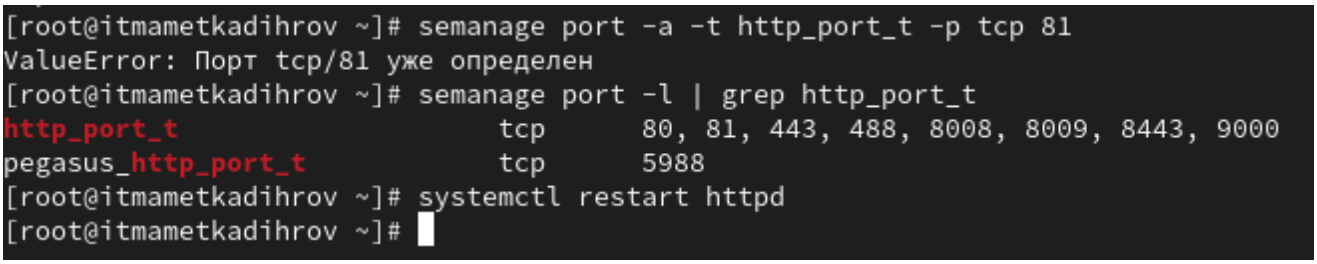

Рис. 3.14: Проверка установки порта 81

Вернули контекст “httpd_sys_content_t” файлу “/var/www/html/test.html” командой “chcon -t httpd_sys_content_t /var/www/html/test.html” (рис. 3.15) и после этого попробовали получить доступ к файлу через веб-сервер, введя адрес “http://127.0.0.1:81/test.html”, в результате чего увидели содежимое файла - слово “test” (рис. 3.15).

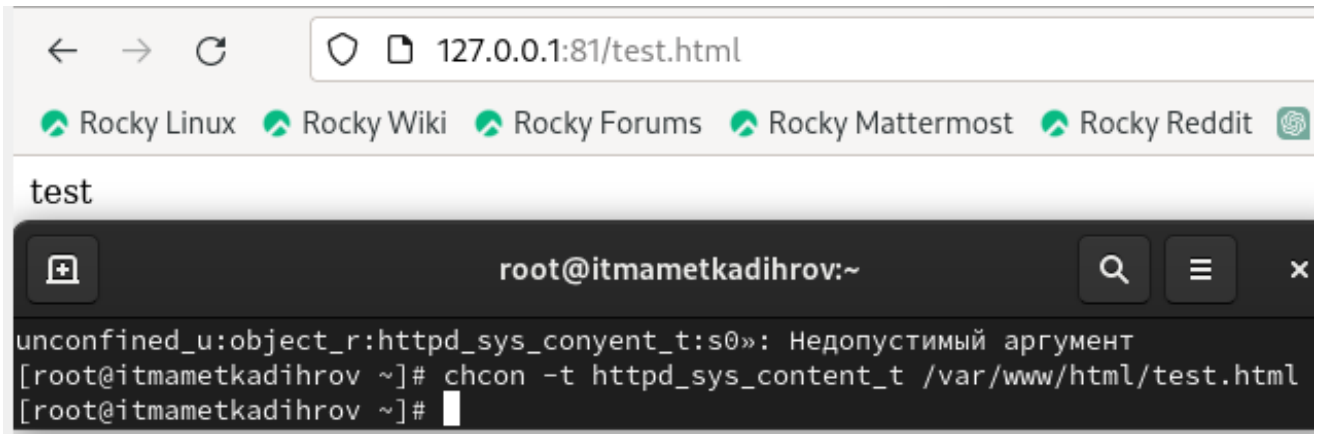

Рис. 3.15: Возвращение исходного контекста файлу и обращение к файлу через веб-сервер

Исправили обратно конфигурационный файл apache, вернув “Listen 80”. Попытались удалить привязку http_port к 81 порту командой “semanage port -d -t http_port_t -p tcp 81”, но этот порт определен на уровне политики, поэтому его нельзя удалить (рис. 3.16).

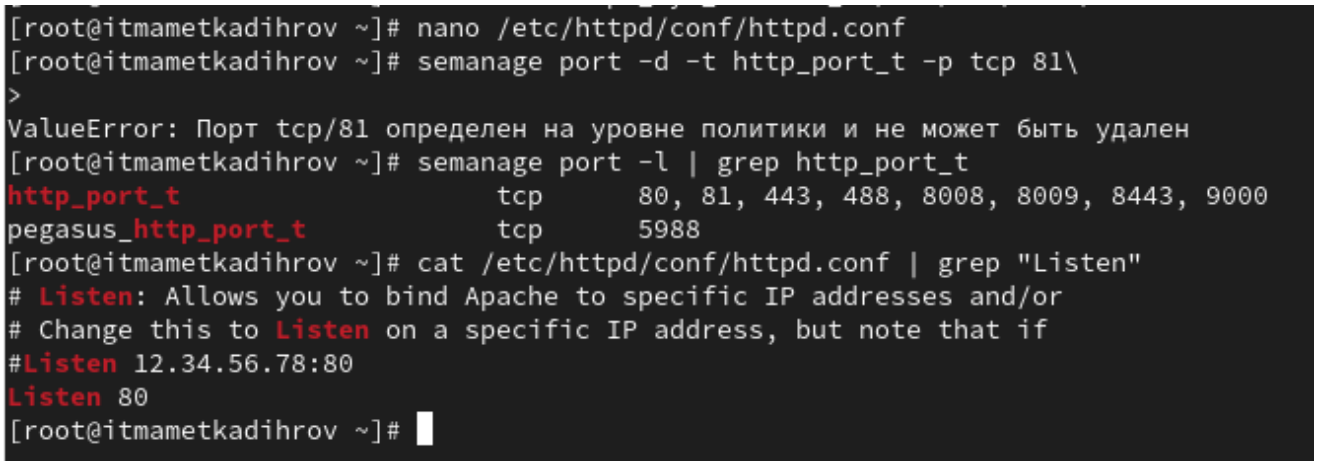

Рис. 3.16: Возвращение Listen 80 и попытка удалить порт 81

Удалили файл “/var/www/html/test.html” командой “rm /var/www/html/test.html” (рис. 3.17).

```
[root@itmametkadihrov ~]# rm /var/www/html/test.html
rm: удалить обычный файл '/var/www/html/test.html'? y
[root@itmametkadihrov ~]# ls /var/www/html
[root@itmametkadihrov ~]#
```

Рис. 3.17: Удаление файла test.html

4 Выводы

В ходе выполнения данной лабораторной работы мы развили навыки администрирования ОС Linux, получили первое практическое знакомство с технологией SELinux и проверили работу SELinux на практике совместно с веб-сервером Apache.

Список литературы

1. SELinux – описание и особенности работы с системой [Электронный ресурс]. URL: https://habr.com/ru/company/kingservers/blog/209644/.
2. Что такое Apache и зачем он нужен? [Электронный ресурс]. URL: https://2domains.ru/support/vps-i-servery/shto-takoye-apache.