## Front matter

title: "Отчёта по лабораторной работе № 6" subtitle: "Информационная безопасность" author: "Надиа Эззакат"

## Generic otions

lang: ru-RU toc-title: "Содержание"

## Bibliography

bibliography: bib/cite.bib csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl

## Pdf output format

toc: true # Table of contents toc-depth: 2 lof: true # List of figures lot: true # List of tables fontsize: 12pt linestretch: 1.5 papersize: a4 documentclass: scrreprt

## l18n polyglossia

polyglossia-lang: name: russian options: - spelling=modern - babelshorthands=true polyglossia-otherlangs: name: english

## l18n babel

babel-lang: russian babel-otherlangs: english

## Fonts

mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase,Scale=0.9

## Biblatex

biblatex: true biblio-style: "gost-numeric" biblatexoptions:

- parentracker=true
- backend=biber
- hyperref=auto
- language=auto
- autolang=other*
- citestyle=gost-numeric

## Pandoc-crossref LaTeX customization

figureTitle: "Рис." tableTitle: "Таблица" listingTitle: "Листинг" lofTitle: "Список иллюстраций" lotTitle: "Список таблиц" lolTitle: "Листинги"

### Misc options

indent: true header-includes:

- \usepackage{indentfirst}
- \usepackage{float} # keep figures where there are in the text
- \floatplacement{figure}{H} # keep figures where there are in the text

---

# Цель работы

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

# Теорическое 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.

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.

# Выполнение лабораторной работы

Вошел в систему под своей учетной записью и убедился, что SELinux работает в режиме enforcing политики targeted с помощью команд “getenforce” и “sestatus”

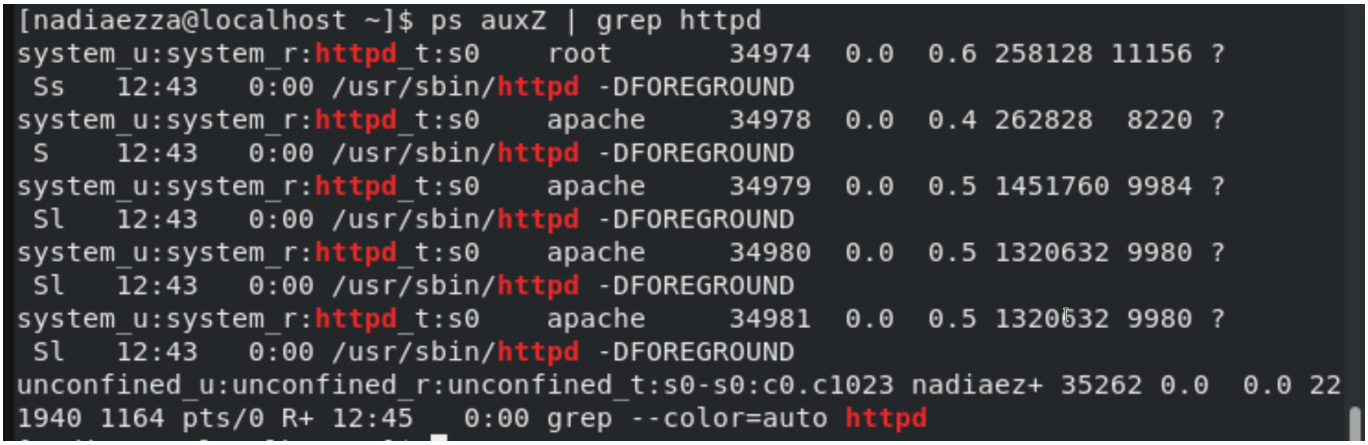
```
[nadiaezza@localhost ~]$ getenforce
Enforcing
[nadiaezza@localhost ~]$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
[nadiaezza@localhost ~]$
```

Обратилась с помощью браузера к веб-серверу, запущенному на моем компьютере, и убедилась, что последний работает с помощью команды “service httpd status”

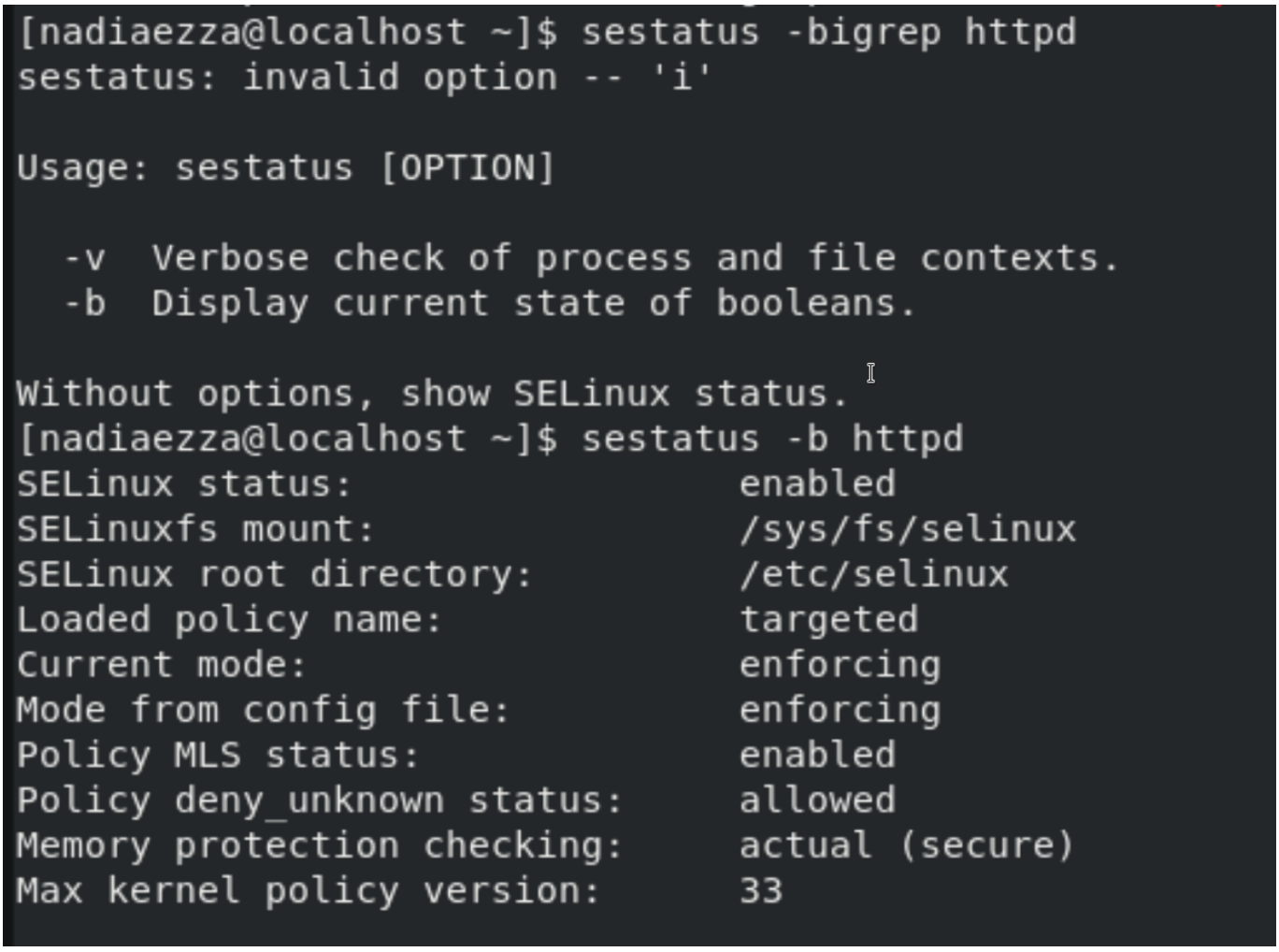
```
[nadiaezza@localhost ~]$ sudo systemctl start httpd
[nadiaezza@localhost ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor pres>
   Active: active (running) since Sat 2023-10-14 12:43:47 MSK; 21s ago
     Docs: man:httpd.service(8)
 Main PID: 34974 (httpd)
   Status: "Running, listening on: port 80"
    Tasks: 213 (limit: 11221)
   Memory: 17.6M
   CGroup: /system.slice/httpd.service
           ├─34974 /usr/sbin/httpd -DFOREGROUND
           ├─34978 /usr/sbin/httpd -DFOREGROUND
           ├─34979 /usr/sbin/httpd -DFOREGROUND
           ├─34980 /usr/sbin/httpd -DFOREGROUND
           └─34981 /usr/sbin/httpd -DFOREGROUND

Oct 14 12:43:47 localhost.localdomain systemd[1]: Starting The Apache HTTP Serv>
Oct 14 12:43:47 localhost.localdomain httpd[34974]: AH00558: httpd: Could not r>
Oct 14 12:43:47 localhost.localdomain systemd[1]: Started The Apache HTTP Serve>
Oct 14 12:43:47 localhost.localdomain httpd[34974]: Server configured, listenin>
```

С помощью команды “ps auxZ | grep httpd” определила контекст безопасности веб-сервера Apache - httpd_t




Посмотрела текущее состояние переключателей SELinux для Apache с помощью команды “sestatus -bigrep httpd”, многие из переключателей находятся в положении “off”



```
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
awstats_purge_apache_log_files              off
boinc_execmem                               on
cdrecord_read_content                       off
cluster_can_network_connect                 off
cluster_manage_all_files                    off
cluster_use_execmem                         off
cobbler_anon_write                          off
cobbler_can_network_connect                 off
cobbler_use_cifs                            off
cobbler_use_nfs                             off
```

Посмотрела статистику по политике с помощью команды “seinfo”. Множество пользователей - 8, ролей - 14, типов 4995

```
[nadiaezza@localhost ~]$ seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:             31 (MLS enabled)
Target Policy:              selinux
Handle unknown classes:     allow
  Classes:             132    Permissions:         464
  Sensitivities:         1    Categories:         1024
  Types:              5010    Attributes:          257
  Users:                 8    Roles:                14
  Booleans:            342    Cond. Expr.:         390
  Allow:            115052    Neverallow:            0
  Auditallow:          168    Dontaudit:         10439
  Type_trans:       257620    Type_change:          87
  Type_member:          35    Range_trans:        5989
  Role allow:           38    Role_trans:          422
  Constraints:          72    Validatetrans:         0
  MLS Constrain:        72    MLS Val. Tran:         0
  Permissives:           0    Polcap:                5
  Defaults:              7    Typebounds:            0
  Allowxperm:            0    Neverallowxperm:       0
  Auditallowxperm:       0    Dontauditxperm:        0
```

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

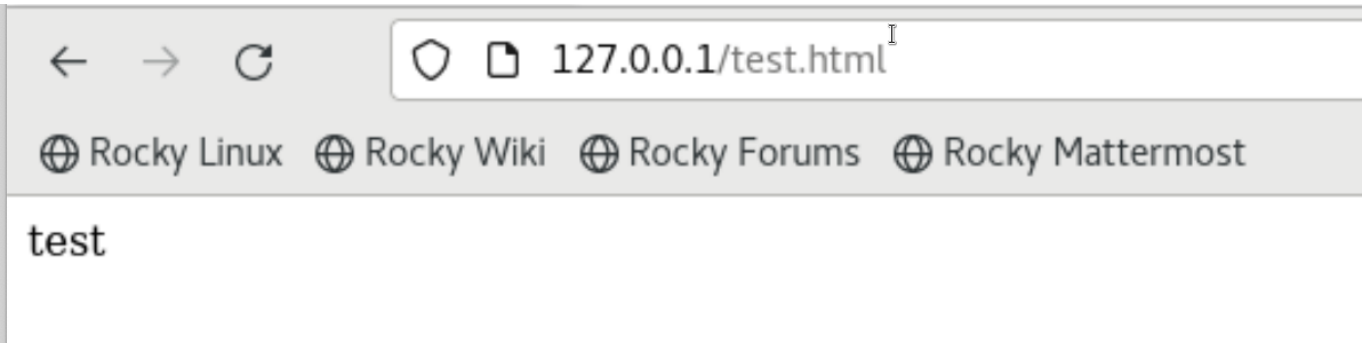
```
[nadiaezza@localhost ~]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Sep 23 02
:22 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0     6 Sep 23 02
:22 html
[nadiaezza@localhost ~]$
```

```
[nadiaezza@localhost ~]$ ls -lZ /var/www/html
total 0
[nadiaezza@localhost ~]$
```

От имени суперпользователя создала html-файл /var/www/html/test.html. Контекст созданного файла - httpd_sys_content_t

```
[nadiaezza@localhost ~]$ su
Password:
[root@localhost nadiaezza]# touch /var/www/html/test.html
[root@localhost nadiaezza]# vim /var/www/html/test.html
[root@localhost nadiaezza]# cat /var/www/html/test.html
<html>
        <body>test</body>
</html>
[root@localhost nadiaezza]#
```

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

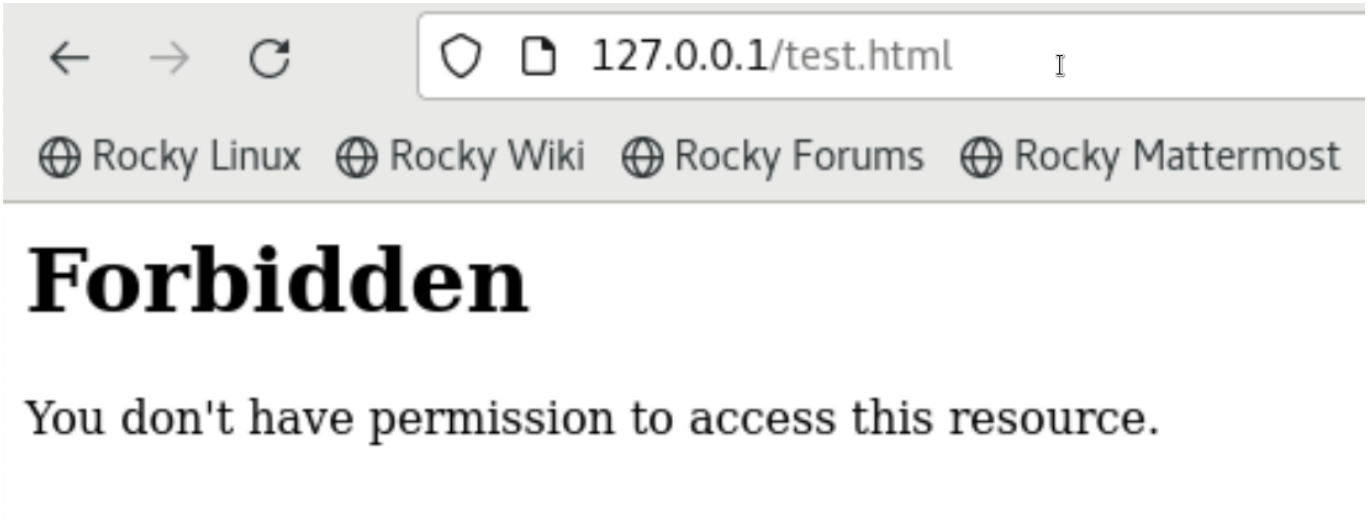



Изучив справку man httpd_selinux, выяснила, что для httpd определены следующие контексты файлов: httpd_sys_content_t, httpd_sys_script_exec_t, httpd_sys_script_ro_t, httpd_sys_script_rw_t, httpd_sys_script_ra_t, httpd_unconfined_script_exec_t.Контекст моего файла - httpd_sys_content_t (в таком случае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

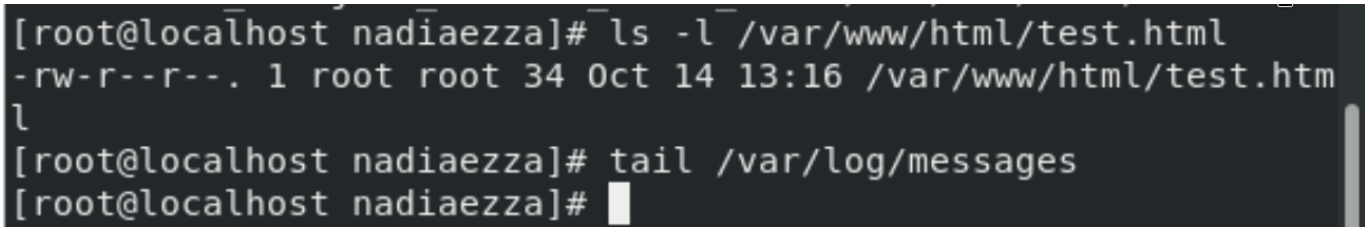
```
[root@localhost nadiaezza]# man httpd_selinux
No manual entry for httpd_selinux
[root@localhost nadiaezza]# exit
exit
[nadiaezza@localhost ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/tes
t.html
```

```
[nadiaezza@localhost ~]$ su
Password:
[root@localhost nadiaezza]# chcon -t samba_share_t /var/www/ht
ml/test.html
[root@localhost nadiaezza]# ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[root@localhost nadiaezza]#
```

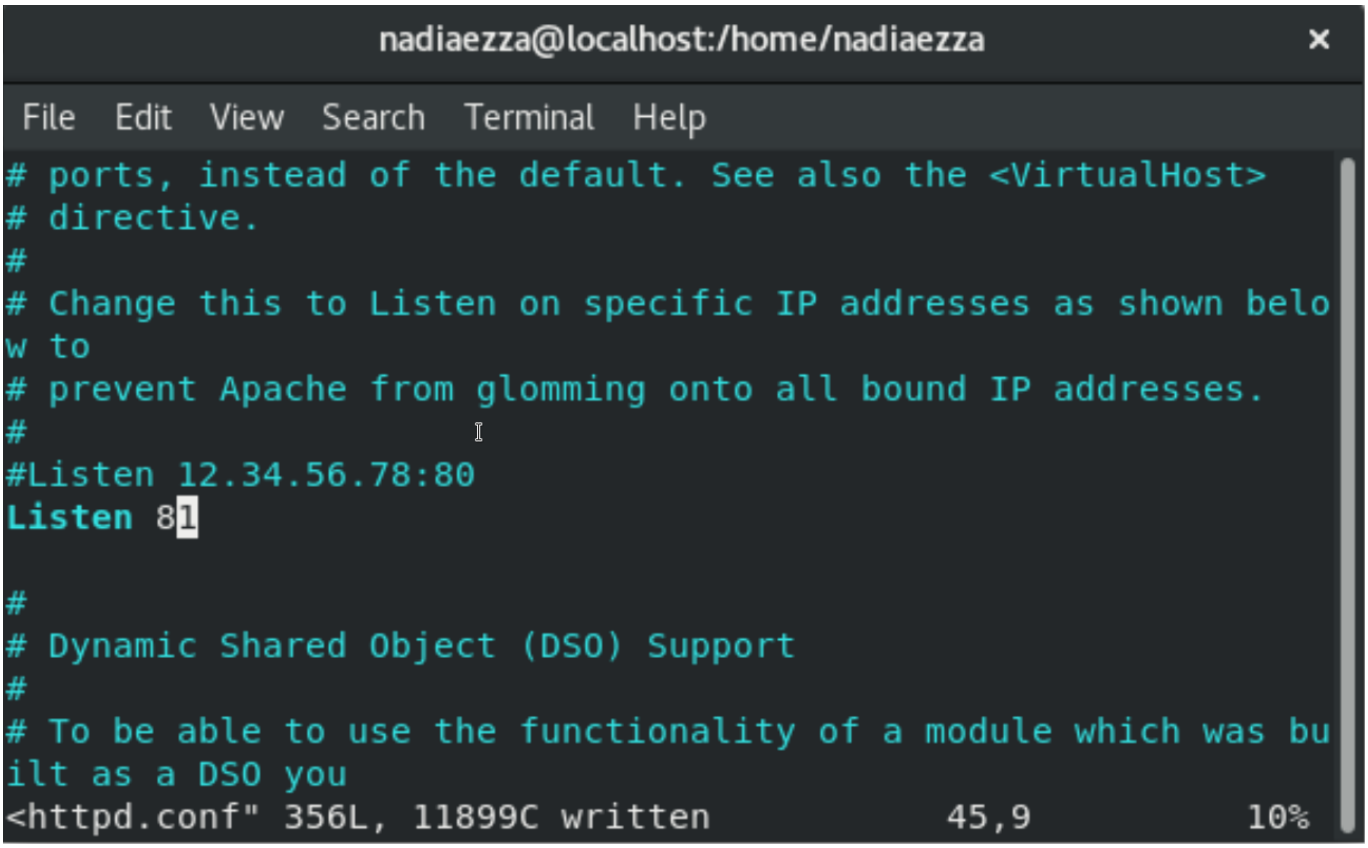
Попробовала еще раз получить доступ к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html” и получила сообщение об ошибке (т.к. к установленному ранее контексту процесс httpd не имеет доступа)




Командой “ls -l /var/www/html/test.html” убедился, что читать данный файл может любой пользователь. Просмотрела системный лог-файл веб-сервера Apache командой “sudo tail /var/log/messages”, отображающий ошибки

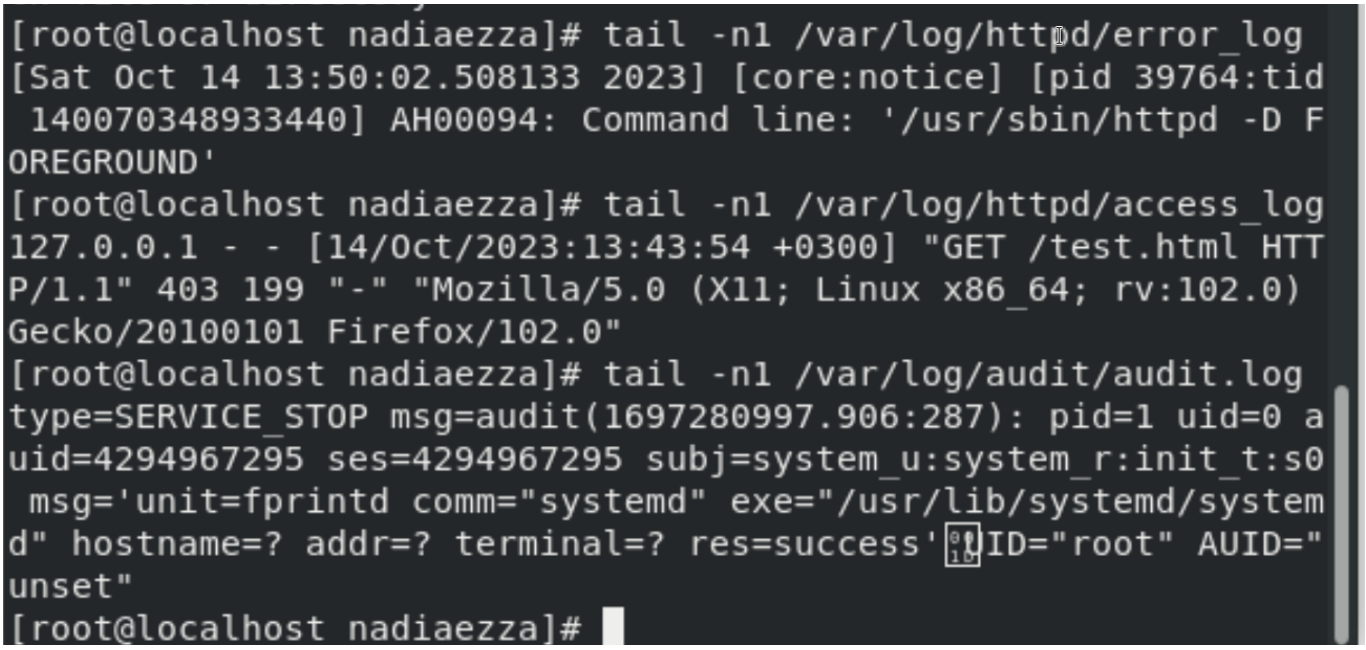



В файле /etc/httpd/conf/httpd.conf заменил строчку “Listen 80” на “Listen 81”, чтобы установить веб-сервер Apache на прослушивание TCP-порта 81




Перезапускаем веб-сервер Apache и анализирует лог-файлы командой “tail -nl /var/log/messages”

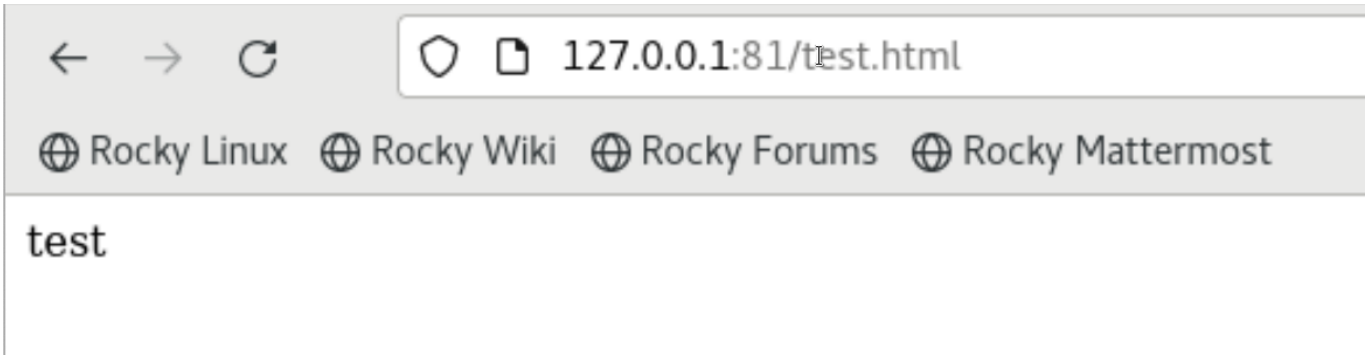
Просмотрела файлы “var/log/http/error_log”, “/var/log/http/access_log” и “/var/log/audit/audit.log” и выяснила, что запись появился в последнем файле




Выполнила команду “semanage port -a -t http_port_t -p tcp 81” и убедился, что порт TCP-81 установлен. Проверил список портов командой “semanage port -l | grep http_port_t”, убедился, что порт 81 есть в списке и запускаем веб-сервер Apache снова

```
[root@localhost nadiaezza]# semanage port -a -t http_port_t -p tcp 81
ValueError: Port tcp/81 already defined
[root@localhost nadiaezza]# semanage port -l | grep http_port_t
semanage port: error: one of the arguments -a/--add -d/--delete -m/--mod
ify -l/--list -E/--extract -D/--deleteall is required
[root@localhost nadiaezza]# semanage port -l | grep http_port_t
http_port_t                    tcp      80, 81, 443, 488, 8008, 8009, 84
43, 9000
pegasus_http_port_t            tcp      5988
[root@localhost nadiaezza]#
```

Вернула контекст “httpd_sys_content_t” файлу “/var/www/html/test.html” командой “chcon -t httpd_sys_content_t /var/www/html/test.html” (рис. 3.16) и после этого попробовал получить доступ к файлу через веб-сервер, введя адрес “http://127.0.0.1:81/test.html”, в результате чего увидел содежимое файла - слово “test”




Исправила обратно конфигурационный файл apache, вернув “Listen 80”. Попытался 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. Удалил файл “/var/www/html/test.html” командой “rm /var/www/html/test.html”

```
File  Edit  View  Search  Terminal  Help
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown bel
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was b
# have to place corresponding `LoadModule' lines at this loca
# directives contained in it are actually available _before_
"/etc/httpd/conf/httpd.conf" 356L, 11899C written
```

# Выводы

В ходе выполнения данной лабораторной работы я развил навыки администрирования ОС Linux, получил первое практическое знакомство с технологией SELinux и проверил работу SELinux на практике совместно с веб-сервером Apache.

# Список литературы

1. Кулябов Д. С. *Лабораторная работа №5**: 006-lab_selinux.pdf*

2. Использование SETUID, SETGID и Sticky bit для расширенной настройки прав доступа в операционных системах Linux [Электронный ресурс]. 2023.URL: https://ruvds.com/ru/helpcenter/suid-sgid-sticky-bit-linux/ (дата обращения: 05.10.2023)