

# **Лабораторная работа № 6**

**Мандатное разграничение прав в Linux**

Алибаева Данагуль НБибд-01-18

# 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>9</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>11</b>
<b>5</b>	<b>Выводы</b>	<b>22</b>
<b>6</b>	<b>Список литературы</b>	<b>23</b>

# List of Tables

# List of Figures

4.1	1.1. SELinux в режиме enforcing . . . . .	11
4.2	1.2. Политика targeted . . . . .	12
4.3	1.3. Обращение к веб-серверу . . . . .	12
4.4	1.4. Обращение к веб-серверу (2) . . . . .	12
4.5	1.5. Занесение контекста в отчет . . . . .	13
4.6	1.6. Просмотр текущего состояния переключателей . . . . .	13
4.7	1.7. Просмотр статистики . . . . .	14
4.8	1.8. Определение типа файлов и поддиректорий . . . . .	14
4.9	1.9. Определение типа файлов . . . . .	14
4.10	1.10. Создание html-файла . . . . .	15
4.11	1.11. Проверка контекста созданного файла . . . . .	15
4.12	1.12. Обращение к файлу через веб-сервер . . . . .	15
4.13	1.13. Справка selinux . . . . .	16
4.14	1.14. Определение контекстов файлов . . . . .	16
4.15	1.15. Изменение контекста файла . . . . .	16
4.16	1.16. Получение сообщения об ошибке . . . . .	17
4.17	1.17. Просмотр системного лог-файла . . . . .	17
4.18	1.18. Просмотр системного лог-файла (2) . . . . .	17
4.19	1.19. Замена строки Listen 80 . . . . .	18
4.20	1.20. Запуск веб-сервера Apache . . . . .	18
4.21	1.21. Анализ лог-файла messages . . . . .	18
4.22	1.22. Просмотр файлов error_log и access_log . . . . .	19
4.23	1.23. Просмотр файла audit.log . . . . .	19
4.24	1.24. Выполнение команды semanage . . . . .	19
4.25	1.25. Проверка списка портов . . . . .	20
4.26	1.26. Возвращение контекста . . . . .	20
4.27	1.27. Получение доступа к веб-серверу . . . . .	20
4.28	1.28. Исправление конфигурационного файла . . . . .	21
4.29	1.29. Удаление привязки к порту . . . . .	21
4.30	1.30. Удаление файла . . . . .	21

# 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 Задание

1. Войти в систему с полученными учётными данными и убедиться, что SELinux работает в режиме enforcing политики targeted.
2. Обратится с помощью браузера к веб-серверу, запущенному на компьютере, и убедиться, что последний работает. Если не работает, запустить его так же, но с параметром start.
3. Найти веб-сервер Apache в списке процессов, определить его контекст безопасности и занести эту информацию в отчёт.
4. Посмотреть текущее состояние переключателей SELinux для Apache. Обратить внимание, что многие из них находятся в положении «off».
5. Посмотреть статистику по политике с помощью команды seinfo, также определить множество пользователей, ролей, типов.
6. Определить тип файлов и поддиректорий, находящихся в директории /var/www.
7. Определить тип файлов, находящихся в директории /var/www/html.
8. Определить круг пользователей, которым разрешено создание файлов в директории /var/www/html.
9. Создать от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания: Test

10. Проверить контекст созданного файла. Занести в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории `/var/www/html`.
11. Обратиться к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Убедится, что файл был успешно отображён.
12. Изучить справку `man httpd_selinux` и выяснить, какие контексты файлов определены для `httpd`. Сопоставить их с типом файла `test.html`.
13. Изменить 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`. После этого проверить, что контекст поменялся.
14. Попробовать ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Должны получить сообщение об ошибке: `Forbidden You don't have permission to access /test.html on this server.`
15. Проанализировать ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю? Просмотреть `log`-файлы веб-сервера `Apache`. Также просмотреть системный `лог`-файл. Если в системе окажутся запущенными процессы `setroubleshootd` и `auditd`, то можно увидеть ошибки, аналогичные указанным выше, в файле `/var/log/audit/audit.log`.
16. Попробовать 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/httpd.conf` найти строчку `Listen 80` и заменить её на `Listen 81`.
17. Выполнить перезапуск веб-сервера `Apache`. Произошёл сбой? Пояснить почему?
18. Проанализировать `лог`-файлы. Просмотреть файлы `/var/log/http/error_log`, `/var/log/http/access_log` и `/var/log/audit/audit.log` и выяснить, в каких файлах

появились записи.

19. Проверить список портов. Убедиться, что порт 81 появился в списке.
20. Попробовать запустить веб-сервер Apache ещё раз.
21. Вернуть контекст `httpd_sys_content_t` к файлу `/var/www/html/test.html`. После этого попробовать получить доступ к файлу через веб-сервер. Должны увидеть содержимое файла — слово «test».
22. Исправить обратно конфигурационный файл `apache`, вернув `Listen 80`.
23. Удалить привязку `http_port_t` к 81 порту и проверить, что порт 81 удалён.
24. Удалить файл `/var/www/html/test.html`.



### 3 Теоретическое введение

В Linux дискреционные механизмы разграничения доступа (DAC, discretionary access control) являются основными и всегда активны. Их использование предполагает, что владельцы объектов правильно распоряжаются правами доступа к находящимся в их владении объектам. [1]

Например, пользовательские закрытые ключи, используемые службой W:[SSH], в каталоге ~/.ssh или ключи W:[GnuPG] в каталоге ~/.gnupg, и прочие секретные данные (подобные ключи доступа в банковские информационные системы) - должны быть недоступны никому, кроме их владельца. Запускаемые пользователем программы выполняются от лица запустившего их пользователя и имеют доступ к файлам согласно установленным режимам или спискам доступа. [1]

В примере из листинга ниже клиент ssh, браузер firefox и коммуникатор skype имеют абсолютно равные возможности по чтению и модификации пользовательского закрытого ключа ~/.ssh/id\_rsa, тогда как настоящим «владельцем» ключей является только ssh. [1]

Абсолютно естественно предполагать, что программы firefox и skype не имеют никаких намерений доступа к пользовательским ключам SSH. [1]

Можно даже доверять программе firefox, штатно установленной из доверенного источника (дистрибутива), где она была изготовлена из открытых исходных текстов, подлежащих верификации. Однако нет никаких оснований доверять закрытому skype, поставляемому в бинарном виде.[1]

Более того, предоставлять доступ программам firefox и skype к SSH-ключам пользователя нет никакой необходимости, во-первых, просто потому, что это

выходит за рамки набора минимально необходимых условий их целевого функционирования. [1]

Во-вторых, практически в любой программе есть ошибки, используя которые злоумышленник может осуществлять непреднамеренные действия в свою пользу. Таким воздействиям особенно подвержены программы, использующие сетевой обмен с недоверенной внешней средой — клиенты и серверы сетевых служб операционной системы. [1]

Тем временем, дискреционный подход и механизмы служат для разграничения доступа разных пользователей к файлам, но никак не предназначены для разграничения доступа программ одного и того же пользователя к разным файлам этого пользователя. [1]

Для разграничения доступа субъектов — программ к объектам — файлам дерева каталогов используют так называемый мандатный (от англ, mandatory — обязательный или принудительный) подход (MAC, mandatory access control), предполагающий следование обязательным правилам доступа к файлам, назначаемым администраторами системы. [1]

Правила доступа строятся на основе знания о внутреннем устройстве программ и представляют собой описание набора минимально необходимых условий их целевого функционирования. [1]

Таким образом, в мандатных правилах, ограничивающих доступ к SSH-ключам пользователя, только программе ssh должен быть разрешен доступ для непосредственного выполнения своих прямых функций, а программам firefox и skype в доступе к SSH-ключам должно быть отказано. [1]

## 4 Выполнение лабораторной работы

1.Лабораторная работа выполнялась дома со следующими характеристиками техники:

- Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz, 2304 МГц, ядер: 4, логических процессоров: 8
- ОС Майкрософт Windows 10 Pro
- VirtualBox верс. 6.1.26

2. Вошла в систему с полученными учётными данными и убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` (рис 1.1) и `sestatus` (рис 1.2).

```
[root@dalibaeva ~]# getenforce
Enforcing
[root@dalibaeva ~]# sestatus
```

Figure 4.1: 1.1. SELinux в режиме enforcing

```

Установлены зависимости:
  httpd-tools.i686 0:2.4.6-97.el7.centos.2      mailcap.noarch 0:2.1.41-2.el7

Выполнено!
[root@dalibaeva dalibaeva]# mc

[root@dalibaeva conf]# mcedit httpd.conf

[root@dalibaeva conf]# cd
[root@dalibaeva ~]# iptables -F
[root@dalibaeva ~]# iptables -P INPUT ACCEPT
[root@dalibaeva ~]# iptables -P OUTPUT ACCEPT
[root@dalibaeva ~]# getenforce
Enforcing
[root@dalibaeva ~]# 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
Max kernel policy version:    31
[root@dalibaeva ~]#

```

Figure 4.2: 1.2. Политика targeted

3. Обратилась с помощью браузера к веб-серверу, запущенному на компьютере, и убедилась, что последний работает: `service httpd status` или `/etc/rc.d/init.d/httpd status` (рис 1.3), (рис 1.4).

```

[root@dalibaeva ~]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@dalibaeva ~]# service httpd status

```

Figure 4.3: 1.3. Обращение к веб-серверу

```

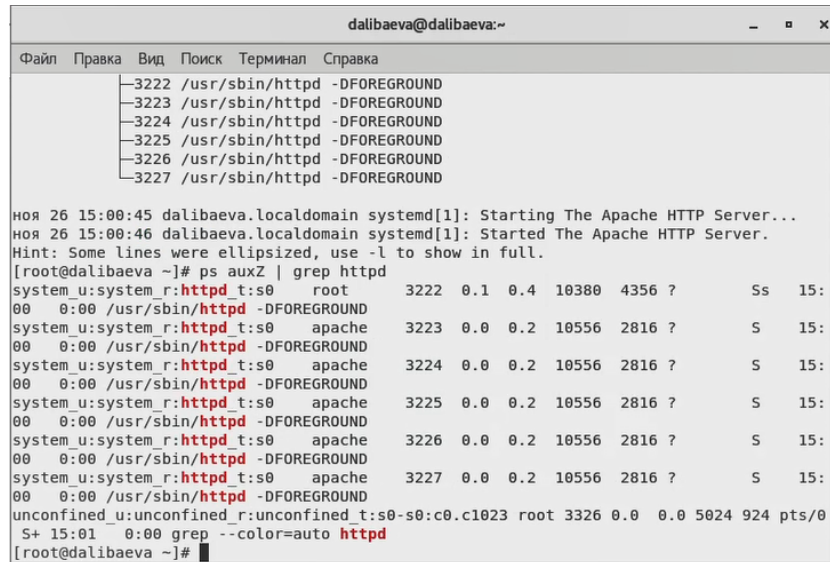
Policy deny_unknown status:    allowed
Max kernel policy version:    31
[root@dalibaeva ~]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@dalibaeva ~]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Пн 2021-11-26 15:00:46 MSK; 4s ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Main PID: 3222 (httpd)
    Status: "Processing requests..."
    CGroup: /system.slice/httpd.service
            └─3222 /usr/sbin/httpd -DFOREGROUND
              └─3223 /usr/sbin/httpd -DFOREGROUND
                └─3224 /usr/sbin/httpd -DFOREGROUND
                  └─3225 /usr/sbin/httpd -DFOREGROUND
                    └─3226 /usr/sbin/httpd -DFOREGROUND
                      └─3227 /usr/sbin/httpd -DFOREGROUND

ноя 26 15:00:45 dalibaeva.localdomain systemd[1]: Starting The Apache HTTP Server...
ноя 26 15:00:46 dalibaeva.localdomain systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[root@dalibaeva ~]#

```

Figure 4.4: 1.4. Обращение к веб-серверу (2)

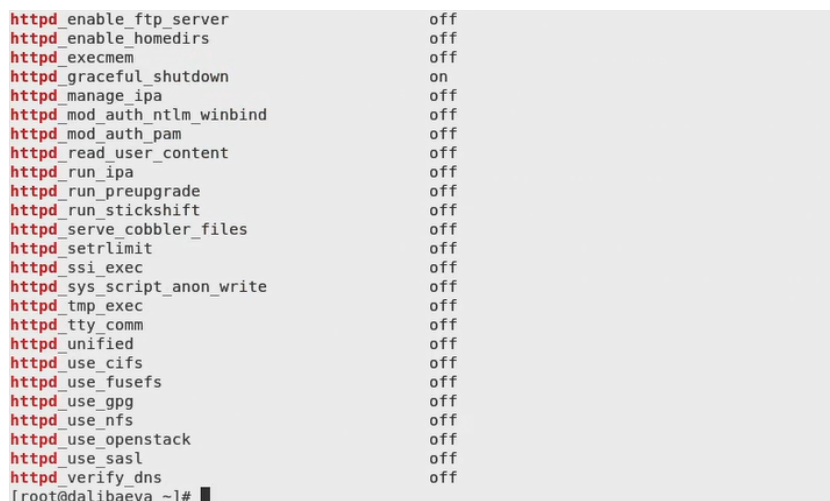
4. Нашла веб-сервер Apache в списке процессов, определила его контекст безопасности и занесла эту информацию в отчёт с помощью команды `ps auxZ | grep httpd` или `ps -eZ | grep httpd` (рис 1.5).



```
dalibaeva@dalibaeva:~  
Файл Правка Вид Поиск Терминал Справка  
--3222 /usr/sbin/httpd -DFOREGROUND  
--3223 /usr/sbin/httpd -DFOREGROUND  
--3224 /usr/sbin/httpd -DFOREGROUND  
--3225 /usr/sbin/httpd -DFOREGROUND  
--3226 /usr/sbin/httpd -DFOREGROUND  
--3227 /usr/sbin/httpd -DFOREGROUND  
ноя 26 15:00:45 dalibaeva.localdomain systemd[1]: Starting The Apache HTTP Server...  
ноя 26 15:00:46 dalibaeva.localdomain systemd[1]: Started The Apache HTTP Server.  
Hint: Some lines were ellipsized, use -l to show in full.  
[root@dalibaeva ~]# ps auxZ | grep httpd  
system_u:system_r:httpd_t:s0 root 3222 0.1 0.4 10380 4356 ? Ss 15:  
00 0:00 /usr/sbin/httpd -DFOREGROUND  
system_u:system_r:httpd_t:s0 apache 3223 0.0 0.2 10556 2816 ? S 15:  
00 0:00 /usr/sbin/httpd -DFOREGROUND  
system_u:system_r:httpd_t:s0 apache 3224 0.0 0.2 10556 2816 ? S 15:  
00 0:00 /usr/sbin/httpd -DFOREGROUND  
system_u:system_r:httpd_t:s0 apache 3225 0.0 0.2 10556 2816 ? S 15:  
00 0:00 /usr/sbin/httpd -DFOREGROUND  
system_u:system_r:httpd_t:s0 apache 3226 0.0 0.2 10556 2816 ? S 15:  
00 0:00 /usr/sbin/httpd -DFOREGROUND  
system_u:system_r:httpd_t:s0 apache 3227 0.0 0.2 10556 2816 ? S 15:  
00 0:00 /usr/sbin/httpd -DFOREGROUND  
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 3326 0.0 0.0 5024 924 pts/0  
S+ 15:01 0:00 grep --color=auto httpd  
[root@dalibaeva ~]#
```

Figure 4.5: 1.5. Занесение контекста в отчет

5. Посмотрела текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -bigrep httpd`. Обратила внимание, что многие из них находятся в положении «off» (рис 1.6).



```
httpd_enable_ftp_server off  
httpd_enable_homedirs off  
httpd_execmem off  
httpd_graceful_shutdown on  
httpd_manage_ipa off  
httpd_mod_auth_ntlm_winbind off  
httpd_mod_auth_pam off  
httpd_read_user_content off  
httpd_run_ipa off  
httpd_run_preupgrade off  
httpd_run_stickshift off  
httpd_serve_cobbler_files off  
httpd_setrlimit off  
httpd_ssi_exec off  
httpd_sys_script_anon_write off  
httpd_tmp_exec off  
httpd_tty_comm off  
httpd_unified off  
httpd_use_cifs off  
httpd_use_fusefs off  
httpd_use_gpg off  
httpd_use_nfs off  
httpd_use_openstack off  
httpd_use_sasl off  
httpd_verify_dns off  
[root@dalibaeva ~]#
```

Figure 4.6: 1.6. Просмотр текущего состояния переключателей

6. Посмотрела статистику по политике с помощью команды `seinfo`, также определила множество пользователей, ролей, типов. (рис 1.7).

```
[root@dalibaeva ~]# seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version & Type: v.31 (binary, mls)

Classes:          130   Permissions:      272
Sensitivities:    1     Categories:      1024
Types:            4793  Attributes:       253
Users:            8     Roles:           14
Booleans:         316   Cond. Expr.:     362
Allow:            107834 Neverallow:       0
Auditallow:       158   Dontaudit:       10022
Type_trans:       18153 Type_change:      74
Type_member:      35    Role_allow:      37
Role_trans:       414   Range_trans:     5899
Constraints:      143   Validatetrans:   0
Initial SIDs:     27    Fs_use:          32
Genfscon:         103   Portcon:         614
Netifcon:         0     Nodecon:         0
Permissives:      0     Polcap:          5

[root@dalibaeva ~]#
```

Figure 4.7: 1.7. Просмотр статистики

7. Определила тип файлов и поддиректорий, находящихся в директории `/var/www`, с помощью команды `ls -lZ /var/www` (рис 1.8).

```
[root@dalibaeva ~]# ls -lZ /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
[root@dalibaeva ~]#
```

Figure 4.8: 1.8. Определение типа файлов и поддиректорий

8. Определила тип файлов, находящихся в директории `/var/www/html`: `ls -lZ /var/www/html` (рис 1.9).

```
[root@dalibaeva ~]# ls -lZ /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
[root@dalibaeva ~]# ls -lZ /var/www/html
[root@dalibaeva ~]#
```

Figure 4.9: 1.9. Определение типа файлов

9. Создала от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) `html`-файл `/var/www/html/test.html` (рис 1.10)

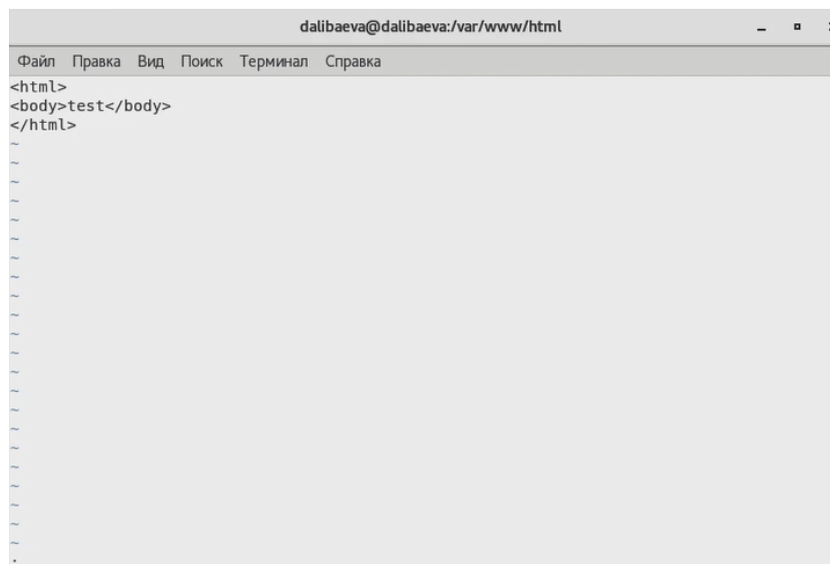


Figure 4.10: 1.10. Создание html-файла

10. Проверила контекст созданного файла. Занесла в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html. (рис 1.11).



Figure 4.11: 1.11. Проверка контекста созданного файла

11. Обратилась к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html>. Убедилась, что файл был успешно отображён. (рис 1.12).

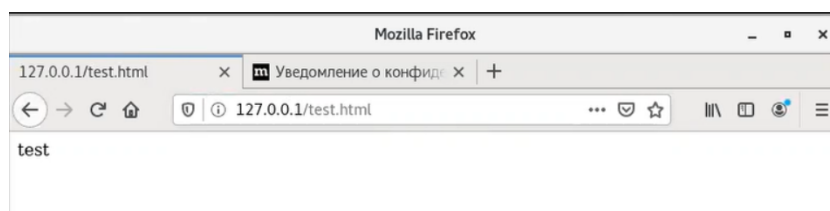


Figure 4.12: 1.12. Обращение к файлу через веб-сервер

12. Изучила справку `man httpd_selinux` и выяснила, какие контексты файлов определены для `httpd` с помощью команды `ls -Z ls -Z /var/www/html/test.html`. Сопоставила их с типом файла `test.html` (рис 1.13), (рис 1.14).

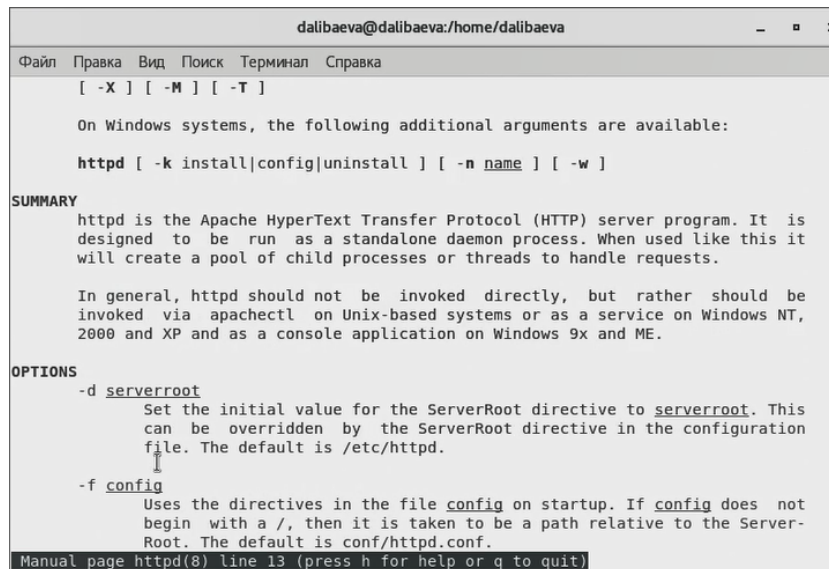


Figure 4.13: 1.13. Справка selinux

```
[root@dalibaeva dalibaeva]# man httpd_selinux
--Man-- след: selinux(8) [ просм (ввод) | пропуск (Ctrl-D) | выход (Ctrl-C) ]
^C
[root@dalibaeva dalibaeva]# ls -Z /var/www/html/test.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[root@dalibaeva dalibaeva]#
```

Figure 4.14: 1.14. Определение контекстов файлов

13. Изменила контекст файла `/var/www/html/test.html` с `httpd_sys_content_t` на `samba_share_t`: `chcon -t samba_share_t /var/www/html/test.html` `ls -Z /var/www/html/test.html` (рис 1.15).

```
[root@dalibaeva dalibaeva]# chcon -t samba_share_t /var/www/html/test.html
[root@dalibaeva dalibaeva]# ls -Z /var/www/html/test.html
-rw-r--r--. root root unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[root@dalibaeva dalibaeva]#
```

Figure 4.15: 1.15. Изменение контекста файла

14. Попробовала ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Получила сообщение об ошибке:



Forbidden You don't have permission to access /test.html on this server. (рис 1.16).

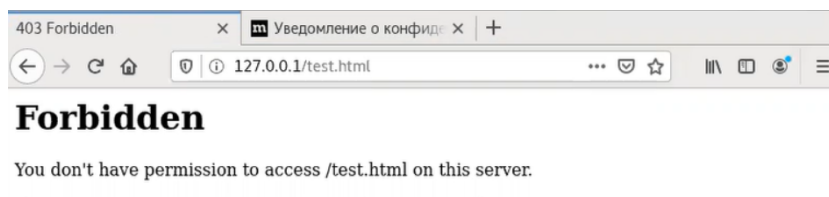


Figure 4.16: 1.16. Получение сообщения об ошибке

15. Проанализировала ситуацию. Просмотрела log-файлы веб-сервера Apache. Также просмотрела системный лог-файл: tail /var/log/messages. (рис 1.17), (рис 1.18).

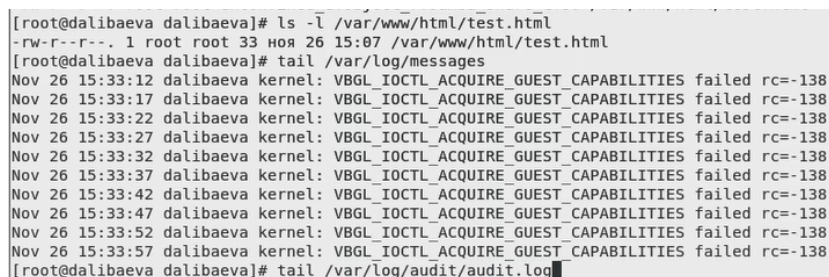


Figure 4.17: 1.17. Просмотр системного лог-файла

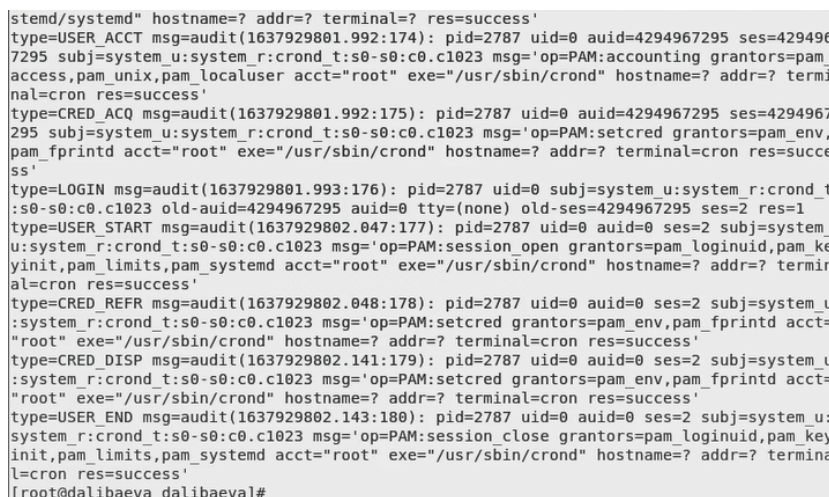


Figure 4.18: 1.18. Просмотр системного лог-файла (2)

16. Попробовала 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/httpd.conf нашла строчку Listen 80 и заменила её на Listen 81. (рис 1.19), (рис 1.20).

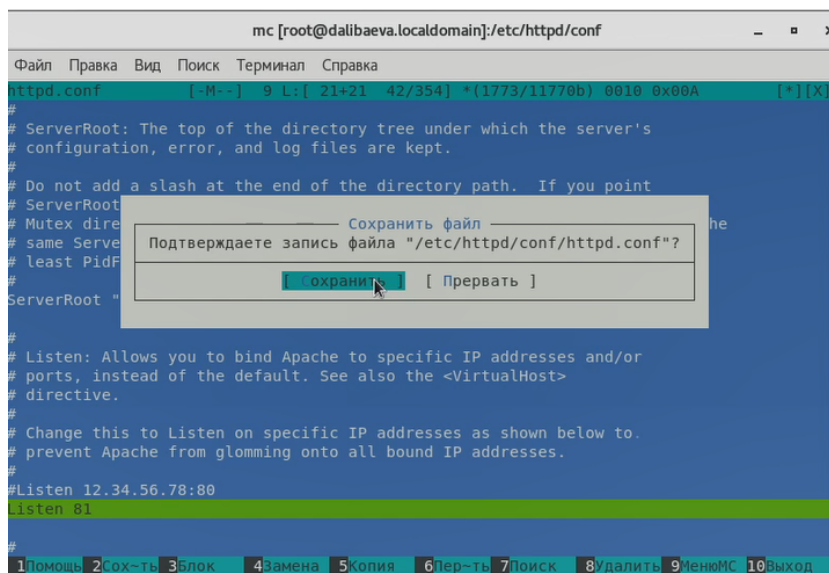


Figure 4.19: 1.19. Замена строчки Listen 80

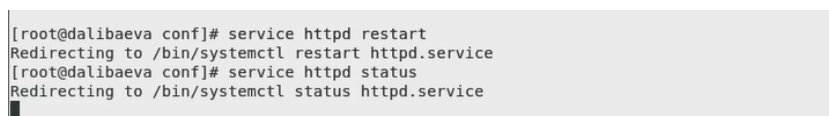


Figure 4.20: 1.20. Запуск веб-сервера Apache

17. Проанализировала лог-файлы: tail -n1 /var/log/messages (рис 1.21). Просмотрела файлы /var/log/http/error\_log, /var/log/http/access\_log (рис 1.22) и /var/log/audit/audit.log и выяснила, в каких файлах появились записи. (рис 1.23).



Figure 4.21: 1.21. Анализ лог-файла messages

```

Hint: Some lines were ellipsized, use -l to show in full.
[root@dalibaeva conf]# tail -n1 /var/log/messages
Nov 26 15:37:12 dalibaeva kernel: VBGL_IOCTL_ACQUIRE_GUEST_CAPABILITIES failed rc=-138
[root@dalibaeva conf]# cat /var/log/httpd/error_log
[Fri Nov 26 15:00:46.262125 2021] [core:notice] [pid 3222] SELinux policy enabled; http
d running as context system_u:system_r:httpd_t:s0
[Fri Nov 26 15:00:46.279018 2021] [suexec:notice] [pid 3222] AH01232: suEXEC mechanism
enabled (wrapper: /usr/sbin/suexec)
[Fri Nov 26 15:00:46.290898 2021] [lbmethod_heartbeat:notice] [pid 3222] AH02282: No sl
otmem from mod_heartbeat
[Fri Nov 26 15:00:46.292655 2021] [mpm_prefork:notice] [pid 3222] AH00163: Apache/2.4.6
(CentOS) configured -- resuming normal operations
[Fri Nov 26 15:00:46.292678 2021] [core:notice] [pid 3222] AH00094: Command line: '/usr
/sbin/httpd -D FOREGROUND'
[Fri Nov 26 15:36:34.823938 2021] [core:notice] [pid 3172] SELinux policy enabled; http
d running as context system_u:system_r:httpd_t:s0
[Fri Nov 26 15:36:34.840990 2021] [suexec:notice] [pid 3172] AH01232: suEXEC mechanism
enabled (wrapper: /usr/sbin/suexec)
[Fri Nov 26 15:36:34.850461 2021] [lbmethod_heartbeat:notice] [pid 3172] AH02282: No sl
otmem from mod_heartbeat
[Fri Nov 26 15:36:34.852039 2021] [mpm_prefork:notice] [pid 3172] AH00163: Apache/2.4.6
(CentOS) configured -- resuming normal operations
[Fri Nov 26 15:36:34.852063 2021] [core:notice] [pid 3172] AH00094: Command line: '/usr
/sbin/httpd -D FOREGROUND'
[root@dalibaeva conf]# cat /var/log/httpd/access_log
[root@dalibaeva conf]#

```

Figure 4.22: 1.22. Просмотр файлов error\_log и access\_log

```

access,pam_unix,pam_localuser acct="root" exe="/usr/sbin/crond" hostname=? addr=? termi
nal=cron res=success'
type=CRED_ACQ msg=audit(1637929801.992:175): pid=2787 uid=0 auid=4294967295 ses=4294967
295 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:setcred grantors=pam_env,
pam_fprintd acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=succe
ss'
type=LOGIN msg=audit(1637929801.993:176): pid=2787 uid=0 subj=system_u:system_r:crond_t
:s0-s0:c0.c1023 old-auid=4294967295 auid=0 tty=(none) old-ses=4294967295 ses=2 res=1
type=USER_START msg=audit(1637929802.047:177): pid=2787 uid=0 auid=0 ses=2 subj=system_
u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:session open grantors=pam_loginuid,pam_ke
yinit,pam_limits,pam_systemd acct="root" exe="/usr/sbin/crond" hostname=? addr=? termina
l=cron res=success'
type=CRED_REFF msg=audit(1637929802.048:178): pid=2787 uid=0 auid=0 ses=2 subj=system_u
:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:setcred grantors=pam_env,pam_fprintd acct=
"root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=CRED_DISP msg=audit(1637929802.141:179): pid=2787 uid=0 auid=0 ses=2 subj=system_u
:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:setcred grantors=pam_env,pam_fprintd acct=
"root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=USER_END msg=audit(1637929802.143:180): pid=2787 uid=0 auid=0 ses=2 subj=system_u:
system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:session close grantors=pam_loginuid,pam_key
init,pam_limits,pam_systemd acct="root" exe="/usr/sbin/crond" hostname=? addr=? termina
l=cron res=success'
type=SERVICE_START msg=audit(1637930194.852:181): pid=1 uid=0 auid=4294967295 ses=42949
67295 subj=system_u:system_r:init_t:s0 msg='unit=httpd comm="systemd" exe="/usr/lib/sys
temd/systemd" hostname=? addr=? terminal=? res=success'

```

Figure 4.23: 1.23. Просмотр файла audit.log

18. Выполнила команду `semanage port -a -t http_port_t -p tcp 81` (рис 1.24). После этого проверила список портов командой `semanage port -l | grep http_port_t`. Убедилась, что порт 81 появился в списке. (рис 1.25).

```

[root@dalibaeva conf]# semanage port -a -t http_port_t -p tcp 81
ValueError: Порт tcp/81 уже определен
[root@dalibaeva conf]# semanage port -l | grep http_port_

```

Figure 4.24: 1.24. Выполнение команды semanage

```

ss'
type=LOGIN msg=audit(1637929801.993:176): pid=2787 uid=0 subj=system_u:system_r:crond_t
:s0-s0:c0.c1023 old-auid=4294967295 auid=0 tty=(none) old-ses=4294967295 ses=2 res=1
type=USER_START msg=audit(1637929802.047:177): pid=2787 uid=0 auid=0 ses=2 subj=system_
u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:session_open grantors=pam_loginuid,pam_ke
yinit,pam_limits,pam_systemd acct="root" exe="/usr/sbin/crond" hostname=? addr=? termin
al=cron res=success'
type=CRED_REFR msg=audit(1637929802.048:178): pid=2787 uid=0 auid=0 ses=2 subj=system_u
:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:setcred grantors=pam_env,pam_fprintd acct=
"root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=CRED_DISP msg=audit(1637929802.141:179): pid=2787 uid=0 auid=0 ses=2 subj=system_u
:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:setcred grantors=pam_env,pam_fprintd acct=
"root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=USER_END msg=audit(1637929802.143:180): pid=2787 uid=0 auid=0 ses=2 subj=system_u:
system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:session_close grantors=pam_loginuid,pam_key
init,pam_limits,pam_systemd acct="root" exe="/usr/sbin/crond" hostname=? addr=? termina
l=cron res=success'
type=SERVICE_START msg=audit(1637930194.852:181): pid=1 uid=0 auid=4294967295 ses=42949
67295 subj=system_u:system_r:init_t:s0 msg='unit=httpd comm="systemd" exe="/usr/lib/sys
temd/systemd" hostname=? addr=? terminal=? res=success'
[root@dalibaeva conf]# semanage port -a -t http_port_t -p tcp 81
ValueError: Порт tcp/81 уже определен
[root@dalibaeva conf]# 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@dalibaeva conf]#

```

Figure 4.25: 1.25. Проверка списка портов

19. Вернула контекст `httpd_sys_content_t` к файлу `/var/www/html/test.html`: `chcon -t httpd_sys_content_t /var/www/html/test.html` (рис 1.26). После этого попробовала получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html` (рис 1.27).

```

[root@dalibaeva conf]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@dalibaeva conf]#

```

Figure 4.26: 1.26. Возвращение контекста

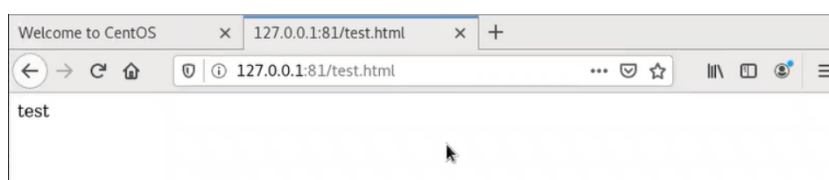
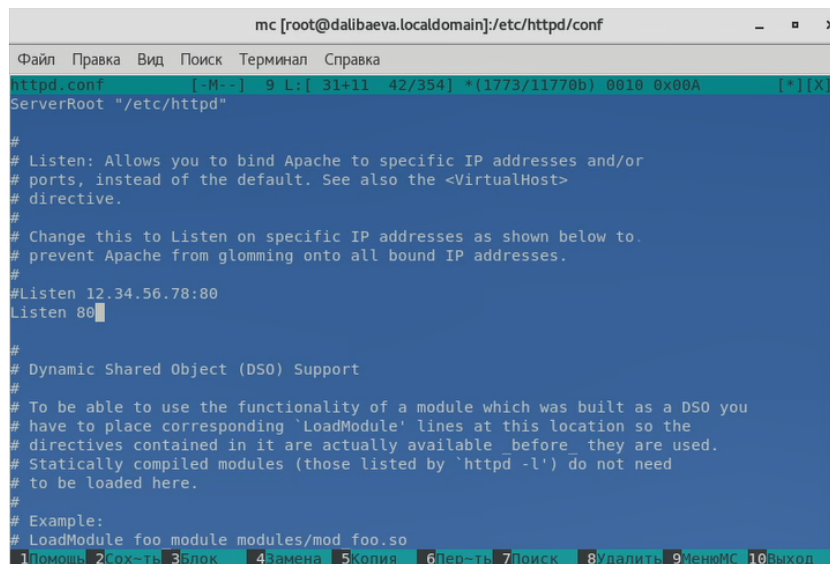


Figure 4.27: 1.27. Получение доступа к веб-серверу

20. Исправила обратно конфигурационный файл `apache`, вернув `Listen 80` (рис 1.28).



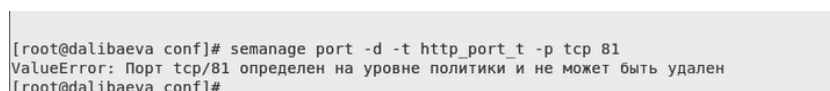
```
mc [root@dalibaeva.localdomain]:/etc/httpd/conf
Файл Правка Вид Поиск Терминал Справка
httpd.conf [-M--] 9 L:[ 31+11 42/354] *(1773/11770b) 0010 0x00A [*][X]
ServerRoot "/etc/httpd"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# 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 built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available before they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo module/modules/mod_foo.so
1Помощь 2Сок-ть 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Удалить 9МенюМС 10Выход
```

Figure 4.28: 1.28. Исправление конфигурационного файла

21. Удалила привязку `http_port_t` к 81 порту: `semanage port -d -t http_port_t -p tcp 81` и проверила, что порт 81 удалён. (рис 1.29).



```
[root@dalibaeva conf]# semanage port -d -t http_port_t -p tcp 81
ValueError: Порт tcp/81 определен на уровне политики и не может быть удален
[root@dalibaeva conf]#
```

Figure 4.29: 1.29. Удаление привязки к порту

22. Удалила файл `/var/www/html/test.html`: `rm /var/www/html/test.html` (рис 1.30).



```
[root@dalibaeva conf]# rm /var/www/html/test.html
rm: удалить обычный файл «/var/www/html/test.html»? y
[root@dalibaeva conf]#
```

Figure 4.30: 1.30. Удаление файла

## 5 Выводы

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

## 6 Список литературы

1. Мандатное разграничение прав в Linux// URL: <https://debianinstall.ru/mandatnoe-prinuditelnoe-razgranichenie-dostupa-linux/> (дата обращения: 26.11.2021).