

Лабораторная работа №6

Основы информационной безопасности

Сабралиева Марворид Нуралиевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Подготовка	6
2.2	Изучение механики SetUID	6
3	Выводы	18
	Список литературы	19

Список иллюстраций

2.1	запуск http	7
2.2	контекст безопасности http	8
2.3	переключатели SELinux для http	9
2.4	статистика по политике	10
2.5	тип файлов и поддиректорий	11
2.6	man httpd_selinux	12
2.7	Изменение контекста файла	13
2.8	лог ошибок	14
2.9	переключение порта	15
2.10	доступ по http на 81 порт	16
2.11	Удаление файлов	17

Список таблиц

1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux¹. Проверить работу SELinx на практике совместно с веб-сервером Apache.

2 Выполнение лабораторной работы

2.1 Подготовка

1. Установили httpd
2. Задали имя сервера
3. Открыли порты для работы с протоколом http

2.2 Изучение механики SetUID

1. Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`.
2. Обратимся с помощью браузера к веб-серверу, запущенному на вашем компьютере, и убедимся, что последний работает: `service httpd status` или `/etc/rc.d/init.d/httpd status` Если не работает, запустить его можно так же, но с параметром `start`.

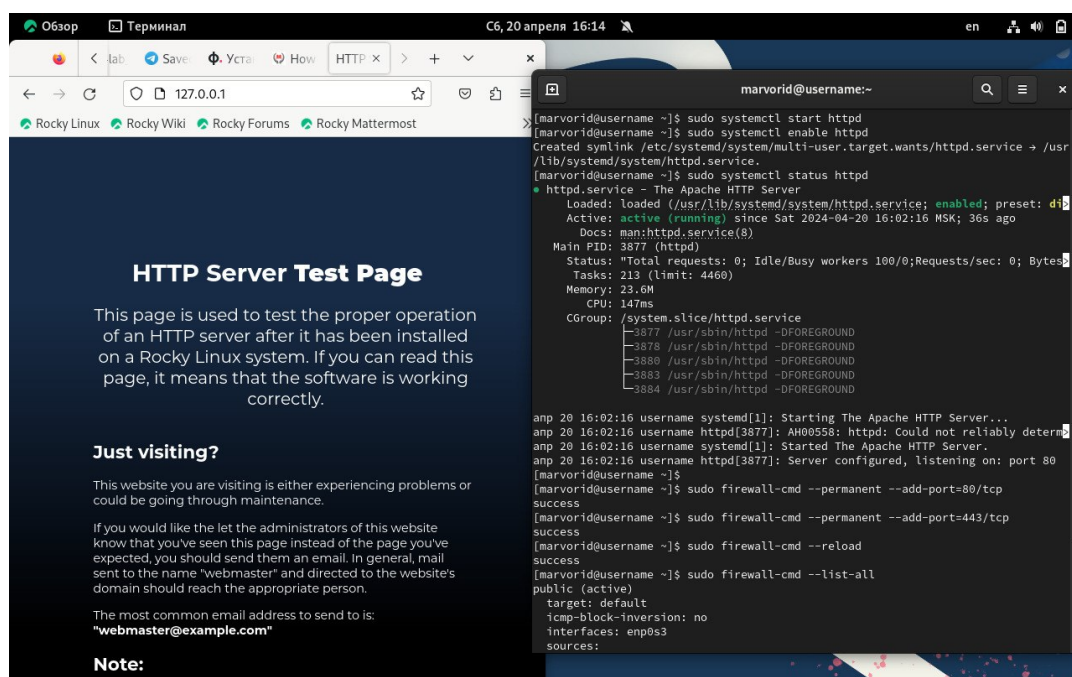
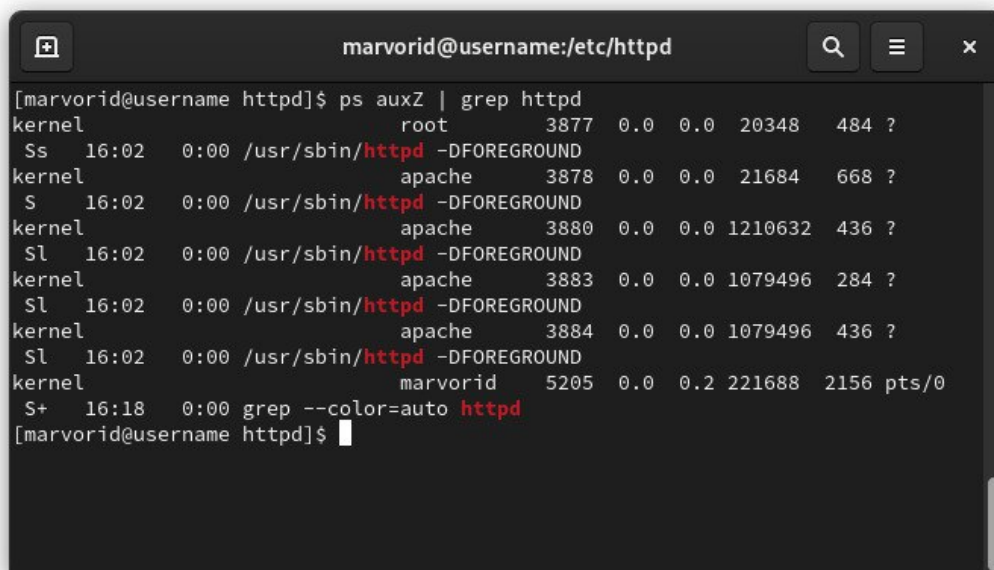


Рис. 2.1: запуск http

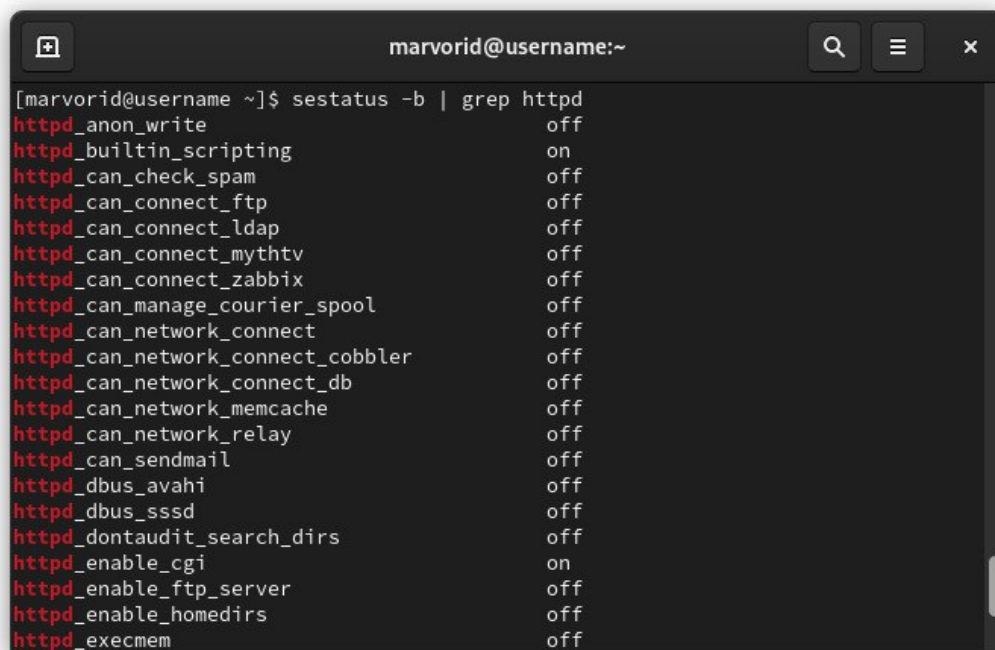
3. Найдем веб-сервер Apache в списке процессов, определите его контекст безопасности. Например, можно использовать команду `ps auxZ | grep httpd` или `ps -eZ | grep httpd` (рис. 2.2).



```
marvoriid@username:/etc/httpd
[marvoriid@username httpd]$ ps auxZ | grep httpd
kernel      root          3877   0.0  0.0  20348   484 ?
Ss   16:02    0:00 /usr/sbin/httpd -DFOREGROUND
kernel     apache       3878   0.0  0.0   21684   668 ?
S    16:02    0:00 /usr/sbin/httpd -DFOREGROUND
kernel     apache       3880   0.0  0.0  1210632   436 ?
Sl   16:02    0:00 /usr/sbin/httpd -DFOREGROUND
kernel     apache       3883   0.0  0.0  1079496   284 ?
Sl   16:02    0:00 /usr/sbin/httpd -DFOREGROUND
kernel     apache       3884   0.0  0.0  1079496   436 ?
Sl   16:02    0:00 /usr/sbin/httpd -DFOREGROUND
kernel     marvoriid    5205   0.0  0.2   221688   2156 pts/0
S+   16:18    0:00 grep --color=auto httpd
[marvoriid@username httpd]$
```

Рис. 2.2: контекст безопасности http

4. Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -bigrep httpd`. Обратим внимание, что многие из них находятся в положении «off». (рис. 2.3).

A terminal window with a dark background. The title bar shows 'marvorid@username:~'. The command '[marvorid@username ~]\$ sestatus -b | grep httpd' has been executed. The output is a list of SELinux booleans for the httpd process, each followed by its status (on or off).

```
[marvorid@username ~]$ sestatus -b | grep httpd
httpd_anon_write off
httpd_builtins_scripting on
httpd_can_check_spam off
httpd_can_connect_ftp off
httpd_can_connect_ldap off
httpd_can_connect_mythtv off
httpd_can_connect_zabbix off
httpd_can_manage_courier_spool off
httpd_can_network_connect off
httpd_can_network_connect_cobbler off
httpd_can_network_connect_db off
httpd_can_network_memcache off
httpd_can_network_relay off
httpd_can_sendmail off
httpd_dbus_avahi off
httpd_dbus_sssd off
httpd_dontaudit_search_dirs off
httpd_enable_cgi on
httpd_enable_ftp_server off
httpd_enable_homedirs off
httpd_execmem off
```

Рис. 2.3: переключатели SELinux для http

5. Посмотрим статистику по политике с помощью команды `seinfo`, также определим множество пользователей, ролей, типов. (рис. 2.4).

```
[marvorid@username ~]$  
[marvorid@username ~]$ 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:                   5153     Attributes:              259  
Users:                   8        Roles:                  15  
Booleans:                357     Cond. Expr.:            390  
Allow:                   66019    Neverallow:              0  
Auditallow:              174     Dontaudit:              8667  
Type_trans:              270422   Type_change:             94  
Type_member:              37      Range_trans:            6164  
Role allow:              39       Role_trans:             419  
Constraints:             70      Validatetrans:           0  
MLS Constrain:           72      MLS Val. Tran:           0  
Permissives:             7        Polcap:                 6  
Defaults:                7       Typebounds:             0  
Allowxperm:              0       Neverallowxperm:        0
```

Рис. 2.4: статистика по политике

6. Определим тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды `ls -lZ /var/www`
7. Определим тип файлов, находящихся в директории /var/www/html: `ls -lZ /var/www/html`
8. Определим круг пользователей, которым разрешено создание файлов в директории /var/www/html.
9. Создадим 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. Проверьте контекст созданного вами файла.

11. Обратитесь к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Файл был успешно отображён.

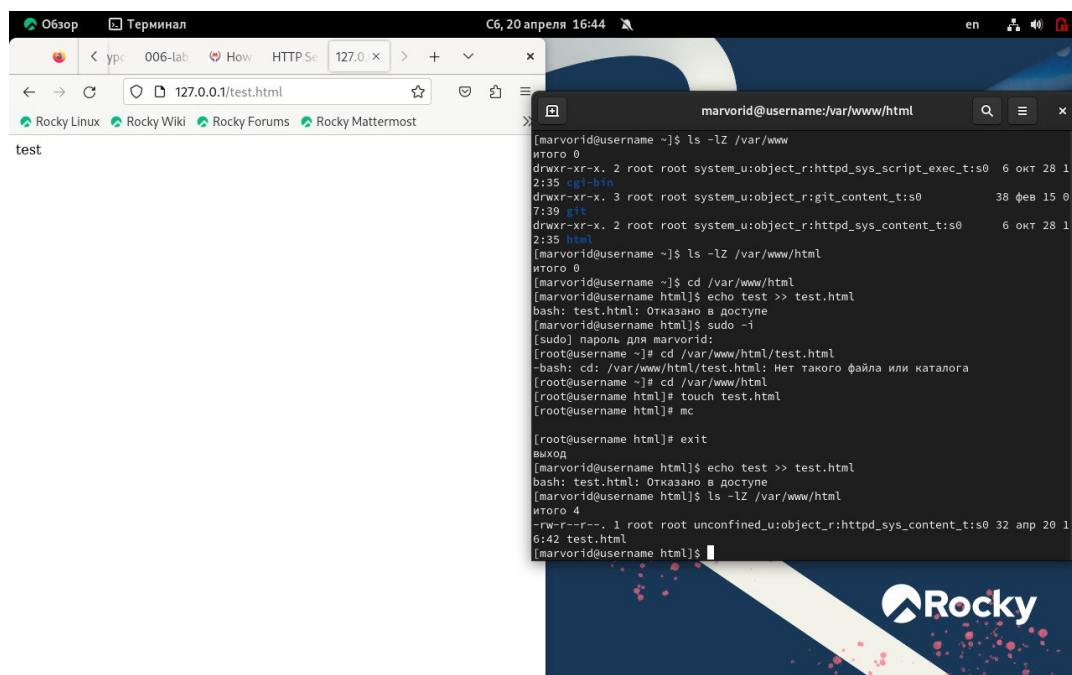
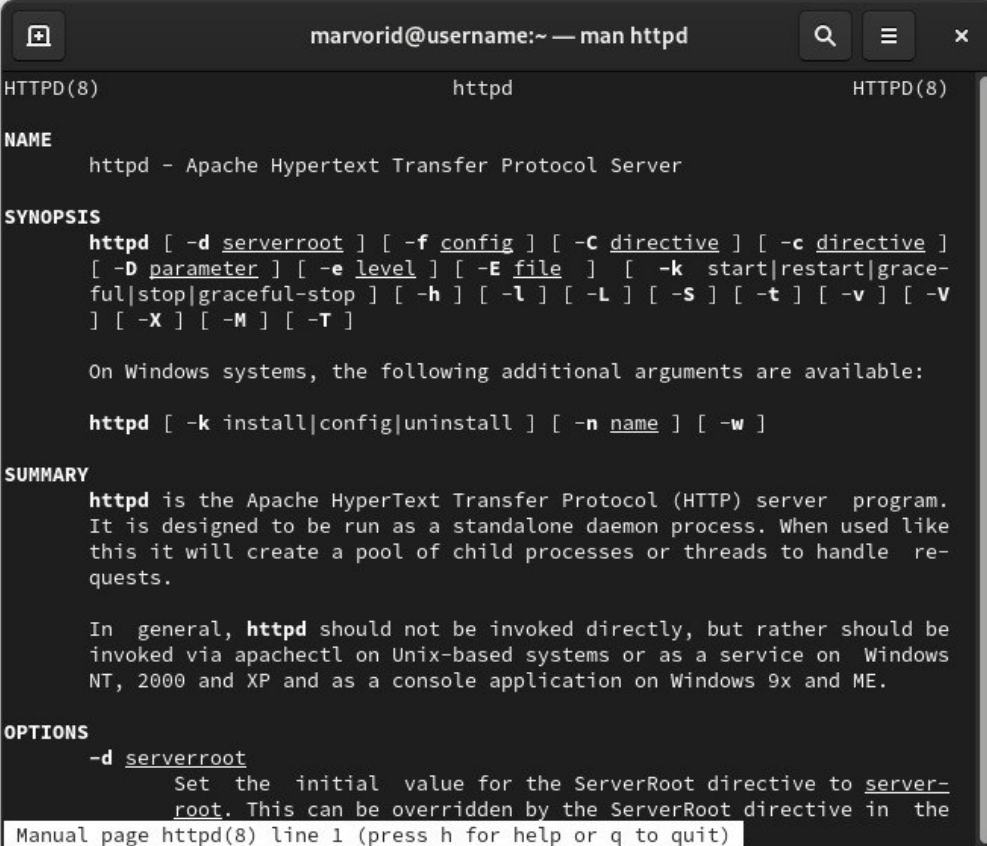


Рис. 2.5: тип файлов и поддиректорий

12. Изучите справку `man httpd_selinux` и выясните, какие контексты файлов определены для `httpd`. Сопоставьте их с типом файла `test.html`. (рис. 2.6). Проверить контекст файла можно командой `ls -Z`. Рассмотрим полученный контекст детально. Обратите внимание, что так как по умолчанию пользователи CentOS 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 `/usr` 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

рассматривать 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.

13. Измените 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`: `chcon -t samba_share_t /var/www/html/test.html ls -Z /var/www/html/test.html` (рис. 2.7).



```
marvolid@username:~ — man httpd
HTTPD(8)                                httpd                                HTTPD(8)

NAME
    httpd - Apache Hypertext Transfer Protocol Server

SYNOPSIS
    httpd [ -d serverroot ] [ -f config ] [ -C directive ] [ -c directive ]
    [ -D parameter ] [ -e level ] [ -E file ] [ -k start|restart|grace-
    ful|stop|graceful-stop ] [ -h ] [ -l ] [ -L ] [ -S ] [ -t ] [ -v ] [ -V
    ] [ -X ] [ -M ] [ -T ]

    On Windows systems, the following additional arguments are available:

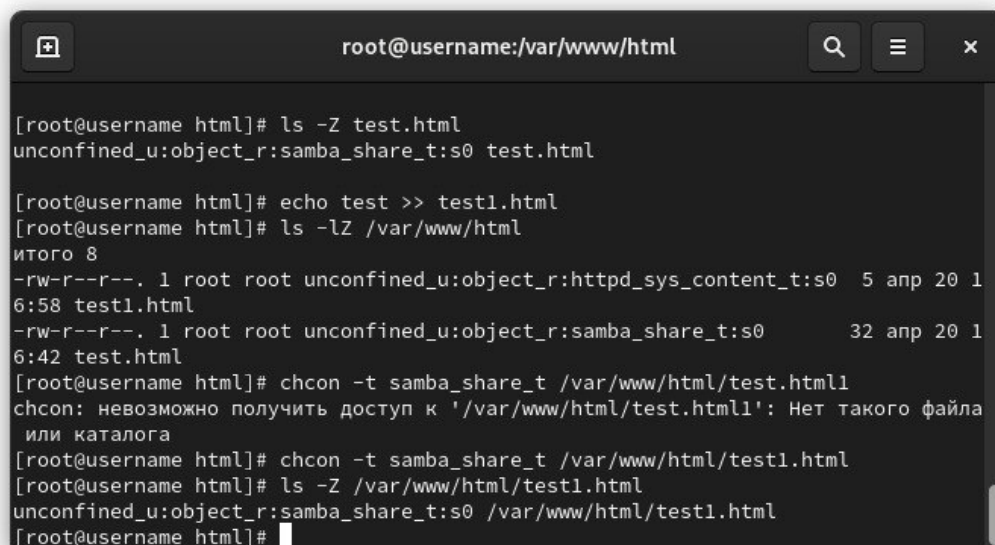
    httpd [ -k install|config|uninstall ] [ -n name ] [ -w ]

SUMMARY
    httpd is the Apache HyperText Transfer Protocol (HTTP) server program.
    It is designed to be run as a standalone daemon process. When used like
    this it will create a pool of child processes or threads to handle re-
    quests.

    In general, httpd should not be invoked directly, but rather should be
    invoked via apachectl on Unix-based systems or as a service on Windows
    NT, 2000 and XP and as a console application on Windows 9x and ME.

OPTIONS
    -d serverroot
        Set the initial value for the ServerRoot directive to server-
        root. This can be overridden by the ServerRoot directive in the
        Manual page httpd(8) line 1 (press h for help or q to quit)
```

Рис. 2.6: `man httpd_selinux`

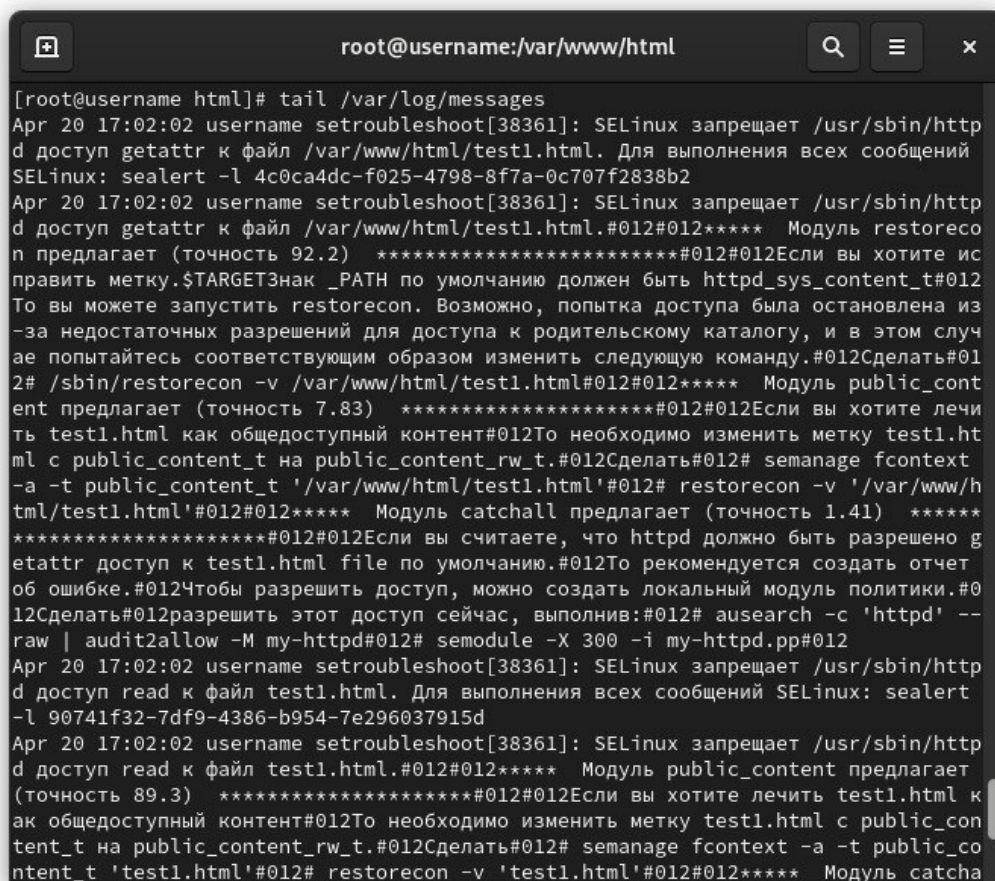
A terminal window titled 'root@username:/var/www/html' with search, menu, and close icons. The terminal shows the following commands and output:

```
[root@username html]# ls -Z test.html
unconfined_u:object_r:samba_share_t:s0 test.html

[root@username html]# echo test >> test1.html
[root@username html]# ls -lZ /var/www/html
итого 8
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0  5 апр 20 1
6:58 test1.html
-rw-r--r--. 1 root root unconfined_u:object_r:samba_share_t:s0      32 апр 20 1
6:42 test.html
[root@username html]# chcon -t samba_share_t /var/www/html/test.html
chcon: невозможно получить доступ к '/var/www/html/test.html': Нет такого файла
или каталога
[root@username html]# chcon -t samba_share_t /var/www/html/test1.html
[root@username html]# ls -Z /var/www/html/test1.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test1.html
[root@username html]#
```

Рис. 2.7: Изменение контекста файла

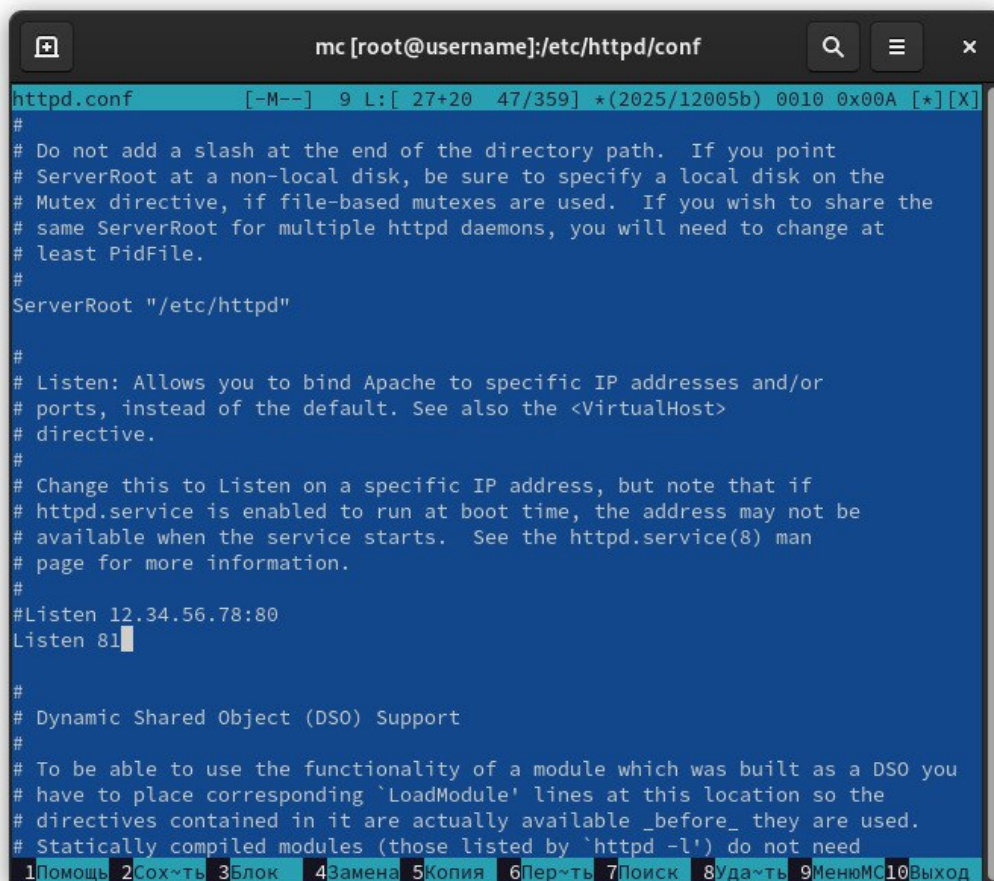
14. Попробуйте ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Вы должны получить сообщение об ошибке: `Forbidden You don't have permission to access /test.html on this server.`
15. Проанализируйте ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю? `ls -l /var/www/html/test.html` Просмотрите log-файлы веб-сервера Apache. Также просмотрите системный лог-файл: `tail /var/log/messages` Если в системе окажутся запущенными процессы `setroubleshootd` и `audtd`, то вы также сможете увидеть ошибки, аналогичные указанным выше, в файле `/var/log/audit/audit.log`.



```
root@username:/var/www/html
[root@username html]# tail /var/log/messages
Apr 20 17:02:02 username setroubleshoot[38361]: SELinux запрещает /usr/sbin/httpd
доступ getattr к файл /var/www/html/test1.html. Для выполнения всех сообщений
SELinux: sealert -l 4c0ca4dc-f025-4798-8f7a-0c707f2838b2
Apr 20 17:02:02 username setroubleshoot[38361]: SELinux запрещает /usr/sbin/httpd
доступ getattr к файл /var/www/html/test1.html.#012#012***** Модуль restorecon
предлагает (точность 92.2) *****#012#012Если вы хотите ис
править метку.$TARGETЗнак _PATH по умолчанию должен быть httpd_sys_content_t#012
То вы можете запустить restorecon. Возможно, попытка доступа была остановлена из
-за недостаточных разрешений для доступа к родительскому каталогу, и в этом случ
ае попытайтесь соответствующим образом изменить следующую команду.#012Сделать#01
2# /sbin/restorecon -v /var/www/html/test1.html#012#012***** Модуль public_cont
ent предлагает (точность 7.83) *****#012#012Если вы хотите лечи
ть test1.html как общедоступный контент#012То необходимо изменить метку test1.ht
ml с public_content_t на public_content_rw_t.#012Сделать#012# semanage fcontext
-a -t public_content_t '/var/www/html/test1.html'#012# restorecon -v '/var/www/h
tml/test1.html'#012#012***** Модуль catchall предлагает (точность 1.41) *****
*****#012#012Если вы считаете, что httpd должно быть разрешено g
etattr доступ к test1.html file по умолчанию.#012То рекомендуется создать отчет
об ошибке.#012Чтобы разрешить доступ, можно создать локальный модуль политики.#0
12Сделать#012разрешить этот доступ сейчас, выполнив:#012# ausearch -c 'httpd' --
raw | audit2allow -M my-httpd#012# semodule -X 300 -i my-httpd.pp#012
Apr 20 17:02:02 username setroubleshoot[38361]: SELinux запрещает /usr/sbin/httpd
доступ read к файл test1.html. Для выполнения всех сообщений SELinux: sealert
-l 90741f32-7df9-4386-b954-7e296037915d
Apr 20 17:02:02 username setroubleshoot[38361]: SELinux запрещает /usr/sbin/httpd
доступ read к файл test1.html.#012#012***** Модуль public_content предлагает
(точность 89.3) *****#012#012Если вы хотите лечить test1.html к
ак общедоступный контент#012То необходимо изменить метку test1.html с public_con
tent_t на public_content_rw_t.#012Сделать#012# semanage fcontext -a -t public_co
ntent_t 'test1.html'#012# restorecon -v 'test1.html'#012#012***** Модуль catcha
```

Рис. 2.8: лог ошибок

16. Попробуйте 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 найдите строчку Listen 80 и замените её на Listen 81. (рис. 2.9).



```
mc [root@username]:/etc/httpd/conf
httpd.conf [-M--] 9 L: [ 27+20 47/359] *(2025/12005b) 0010 0x00A [*] [X]
#
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used. If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
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 a specific IP address, but note that if
# httpd.service is enabled to run at boot time, the address may not be
# available when the service starts. See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
Listen 81
#
# 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
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Вставить 7Поиск 8Удалить 9Меню 10Выход
```

Рис. 2.9: переключение порта

17. Выполним перезапуск веб-сервера Apache. Произошёл сбой? Сбой не происходит, порт 81 уже вписан в разрешенные
18. Проанализируйте лог-файлы: `tail -nl /var/log/messages` Просмотрите файлы `/var/log/http/error_log`, `/var/log/http/access_log` и `/var/log/audit/audit.log` и выясните, в каких файлах появились записи.
19. Выполним команду `semanage port -a -t http_port_t -p tcp 81` После этого проверьте список портов командой `semanage port -l | grep http_port_t` Убедимся, что порт 81 появился в списке.
20. Попробуем запустить веб-сервер Apache ещё раз.

21. Вернем контекст `httpd_sys_content_t` к файлу `/var/www/html/ test.html`:
`chcon -t httpd_sys_content_t /var/www/html/test.html` После этого попробуйте получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html`. Вы должны увидеть содержимое файла — слово «test».

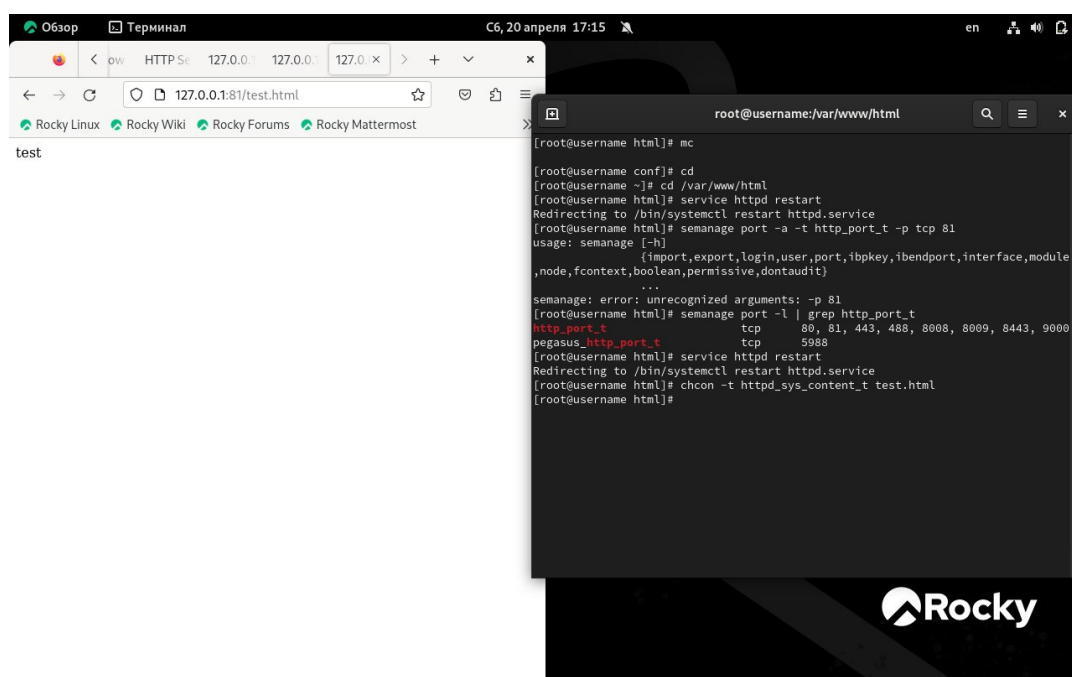
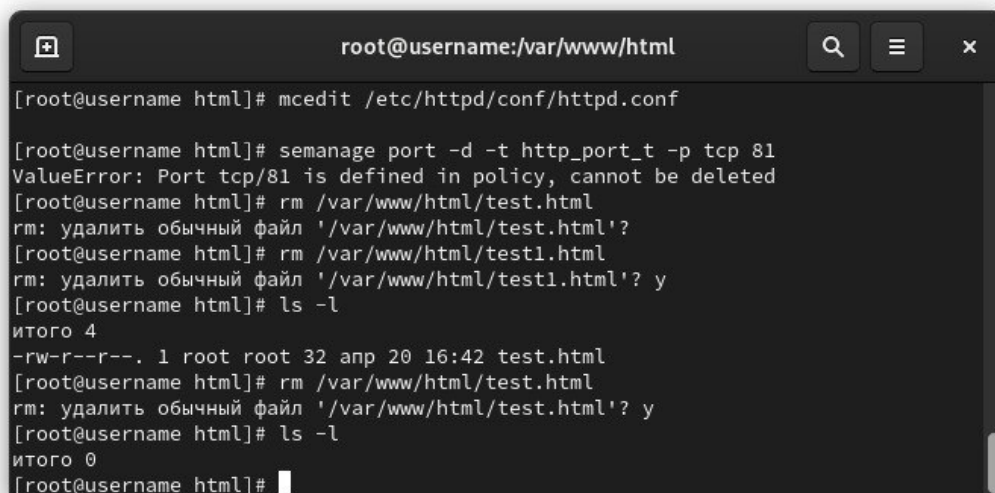


Рис. 2.10: доступ по http на 81 порт

22. Исправьте обратно конфигурационный файл `apache`, вернув `Listen 80`.
23. Удалите привязку `http_port_t` к 81 порту: `semanage port -d -t http_port_t -p tcp 81` и проверьте, что порт 81 удалён.
24. Удалите файл `/var/www/html/test.html`: `rm /var/www/html/test.html`



```
root@username:/var/www/html
[root@username html]# mcedit /etc/httpd/conf/httpd.conf
[root@username html]# semanage port -d -t http_port_t -p tcp 81
ValueError: Port tcp/81 is defined in policy, cannot be deleted
[root@username html]# rm /var/www/html/test.html
rm: удалить обычный файл '/var/www/html/test.html'?
[root@username html]# rm /var/www/html/test1.html
rm: удалить обычный файл '/var/www/html/test1.html'? y
[root@username html]# ls -l
итого 4
-rw-r--r--. 1 root root 32 anp 20 16:42 test.html
[root@username html]# rm /var/www/html/test.html
rm: удалить обычный файл '/var/www/html/test.html'? y
[root@username html]# ls -l
итого 0
[root@username html]#
```

Рис. 2.11: Удаление файлов

3 Выводы

В процессе выполнения лабораторной работы мною были получены базовые навыки работы с технологией SELinux

Список литературы