

Лабораторная работа №6

Основы информационной безопасности

Сабралиева М. Н.

Российский университет дружбы народов, Москва, Россия

Информация

- Сабралиева Марворид Нуралиевна
- студентка НБИбд-01-22 кафедры прикладной информатики и теории вероятностей
- Российский университет дружбы народов

Элементы презентации

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

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

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`.

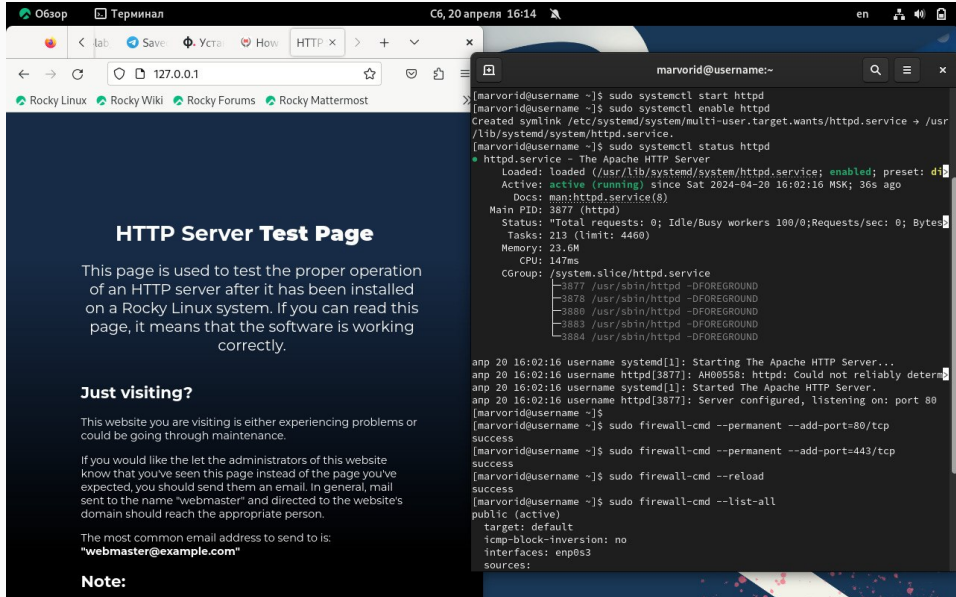
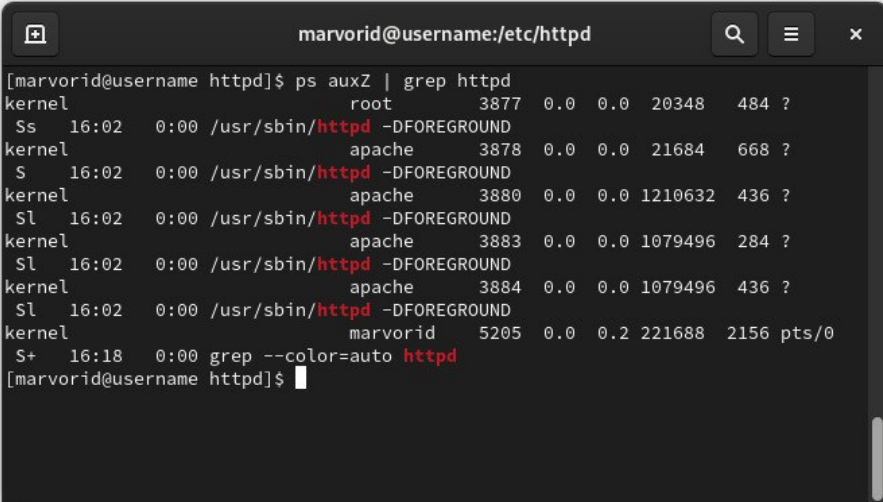


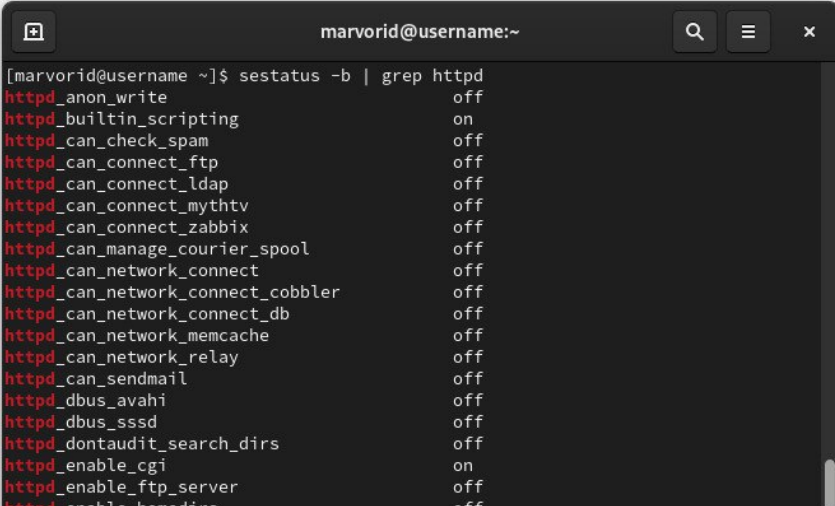
Рис. 1: запуск http

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`



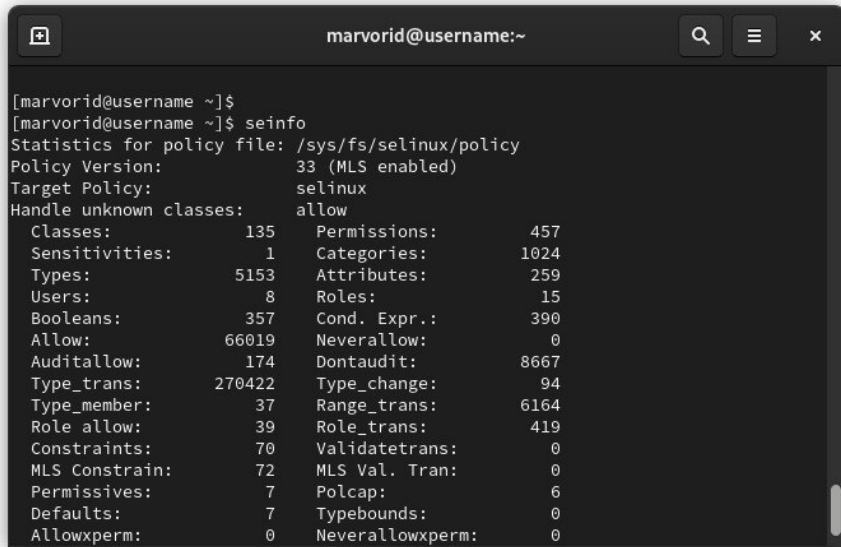
```
marvorid@username:/etc/httpd
[marvorid@username httpd]$ ps auxZ | grep httpd
kernel      root          3877  0.0  0.0  20348  484 ?
Ss   16:02    0:00 /usr/sbin/httpd -DFOREGROUND
kernel      apache        3878  0.0  0.0  21684  668 ?
S    16:02    0:00 /usr/sbin/httpd -DFOREGROUND
kernel      apache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

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

A terminal window titled 'marvorid@username:~' with search, menu, and close buttons. It displays the command '[marvorid@username ~]\$ sestatus -b | grep httpd' and its output, which lists various SELinux booleans for the httpd process and their current status (on or off).

```
[marvorid@username ~]$ sestatus -b | grep httpd
httpd_anon_write                off
httpd_builtin_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_sss                  off
httpd_dontaudit_search_dirs     off
httpd_enable_cgi                on
httpd_enable_ftp_server         off
httpd_enable_homedir            off
```

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



A terminal window titled 'marvorid@username:~' with search, menu, and close icons in the title bar. The terminal shows the execution of the 'seinfo' command, which displays statistics for the SELinux policy file located at '/sys/fs/selinux/policy'. The output includes the policy version (33, with MLS enabled), target policy (selinux), and a list of statistics for various SELinux components.

```
[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
```

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`

10. Проверьте контекст созданного вами файла.
11. Обратитесь к файлу через веб-сервер, введя в браузере адрес Файл был успешно отображён.

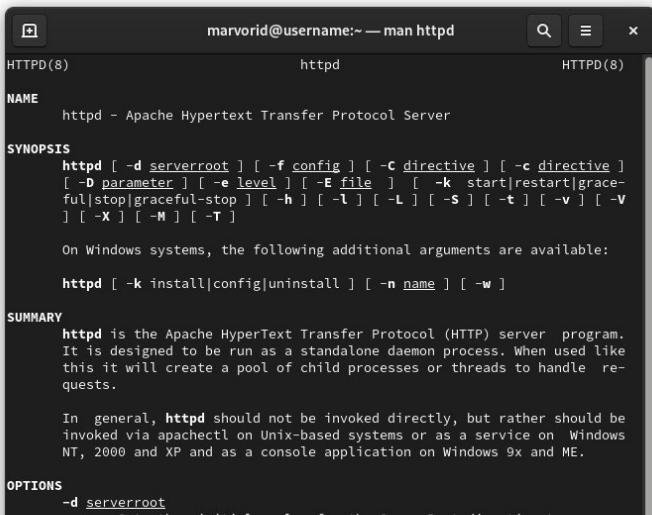
The screenshot shows a terminal window and a web browser. The terminal window, titled "marvolid@username: /var/www/html", displays the following commands and output:

```
[marvolid@username ~]$ ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 окт 28 1
2:35 cgi-bin
drwxr-xr-x. 3 root root system_u:object_r:git_content_t:s0 38 фев 15 0
7:39 git
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 окт 28 1
2:35 html
[marvolid@username ~]$ ls -lZ /var/www/html
итого 0
[marvolid@username ~]$ cd /var/www/html
[marvolid@username html]$ echo test >> test.html
bash: test.html: Отказано в доступе
[marvolid@username html]$ sudo -i
[sudo] пароль для marvolid:
[root@username ~]# cd /var/www/html/test.html
-bash: cd: /var/www/html/test.html: Нет такого файла или каталога
[root@username ~]# cd /var/www/html
[root@username html]# touch test.html
[root@username html]# mc
[root@username html]# exit
выход
[marvolid@username html]$ echo test >> test.html
bash: test.html: Отказано в доступе
[marvolid@username html]$ ls -lZ /var/www/html
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 32 апр 20 1
6:42 test.html
[marvolid@username html]$
```

The web browser, titled "Обзор Терминал", shows the address bar with the URL "127.0.0.1/test.html". The page content displays the word "test".

12. Изучите справку `man httpd_selinux` и выясните, какие контексты файлов определены для `httpd`. Сопоставьте их с типом файла `test.html`. 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
```



The screenshot shows a terminal window titled "marvori@username:~ — man httpd". The window displays the manual page for the `httpd` command. The page is divided into sections: NAME, SYNOPSIS, SUMMARY, and OPTIONS. The NAME section identifies `httpd` as the Apache Hypertext Transfer Protocol Server. The SYNOPSIS section lists various command-line options for `httpd`, including `-d` for server root, `-f` for config file, `-C` for directives, `-D` for parameters, `-e` for error level, `-E` for error file, `-k` for action (start, restart, graceful-stop, stop), `-h` for help, `-l` for log file, `-L` for log level, `-s` for server name, `-t` for test, `-v` for version, `-X` for X11, `-M` for modules, and `-T` for threads. The SUMMARY section describes `httpd` as the Apache HyperText Transfer Protocol (HTTP) server program, designed to be run as a standalone daemon process. The OPTIONS section lists the `-d` option for server root.

```
marvori@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|graceful|stop|graceful-stop]
 [-h] [-l] [-L] [-s] [-t] [-v] [-X] [-M] [-T]

 On Windows systems, the following additional arguments are available:

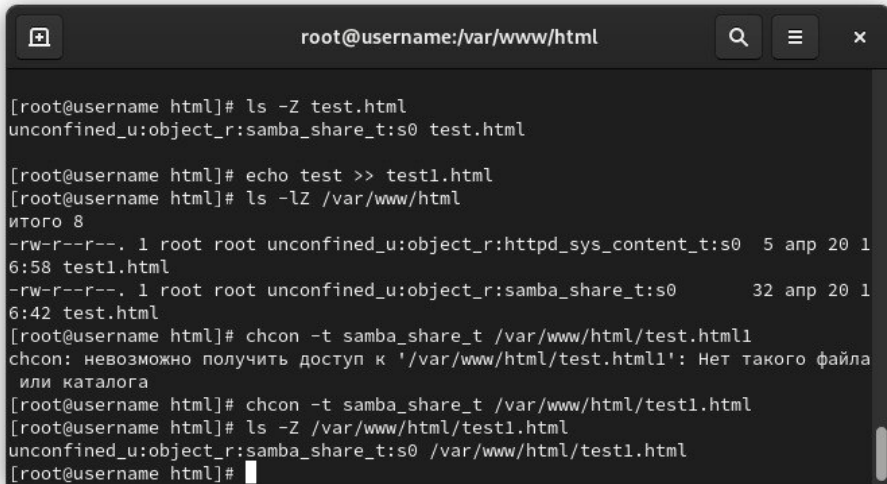
 httpd [-k install|config|uninstall] [-n name] [-w]

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
```



A terminal window with a dark background and light text. The title bar at the top shows 'root@username:/var/www/html' and standard window controls (minimize, maximize, close). The terminal content shows a series of commands and their outputs. The first command 'ls -Z test.html' shows the file's context as 'unconfined\_u:object\_r:samba\_share\_t:s0'. The second command 'echo test >> test1.html' creates a new file. The third command 'ls -lZ /var/www/html' shows a list of files with their permissions, owners, and contexts. The fourth command 'chcon -t samba\_share\_t /var/www/html/test.html1' fails with an error message. The fifth command 'chcon -t samba\_share\_t /var/www/html/test1.html' succeeds. The final command 'ls -Z /var/www/html/test1.html' shows the updated context for 'test1.html' as 'unconfined\_u:object\_r:samba\_share\_t:s0'.

```
root@username:/var/www/html

[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.html1
chcon: невозможно получить доступ к '/var/www/html/test.html1': 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. 7: Изменение контекста файла

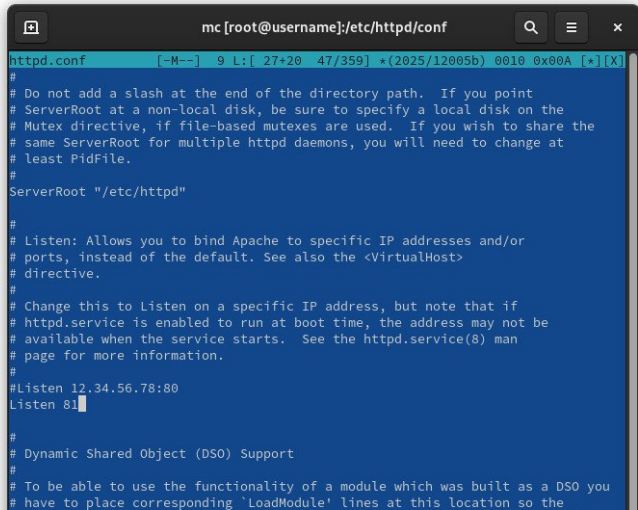
14. Попробуйте ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `.` . Вы должны получить сообщение об ошибке: `Forbidden You don't have permission to access /test.html on this server.`
15. Проанализируйте ситуацию. Почему файл не был отображён, если права доступа позволяют читать 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 исправить метку.$TARGETЗнак _PATH по умолчанию должен быть httpd_sys_content_t#012То вы можете запустить restorecon. Возможно, попытка доступа была остановлена из-за недостаточных разрешений для доступа к родительскому каталогу, и в этом случае попытайтесь соответствующим образом изменить следующую команду.#012Сделать#012#012# /sbin/restorecon -v /var/www/html/test1.html#012#012***** Модуль public_content предлагает (точность 7.83) *****#012#012Если вы хотите лечить test1.html как общедоступный контент#012То необходимо изменить метку test1.html с public_content_t на public_content_rw_t.#012Сделать#012#012# semanage fcontext -a -t public_content_t '/var/www/html/test1.html'#012#012# restorecon -v '/var/www/html/test1.html'#012#012***** Модуль catchall предлагает (точность 1.41) *****#012#012Если вы считаете, что httpd должно быть разрешено getattr доступ к test1.html file по умолчанию.#012То рекомендуется создать отчет об ошибке.#012Чтобы разрешить доступ, можно создать локальный модуль политики.#012Сделать#012разрешить этот доступ сейчас, выполнив:#012#012# ausearch -c 'httpd' --raw | audit2allow -M my-httpd#012#012# semodule -X 300 -i my-httpd.pp#012#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 как общедоступный контент#012То необходимо изменить метку test1.html с public_content_t на public_content_rw_t.#012Сделать#012#012# semanage fcontext -a -t public_content_t 'test1.html'#012#012# restorecon -v 'test1.html'#012#012***** Модуль catcha
```

Рис. 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.



```
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
```

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 Вы должны увидеть содержимое файла — слово «test».

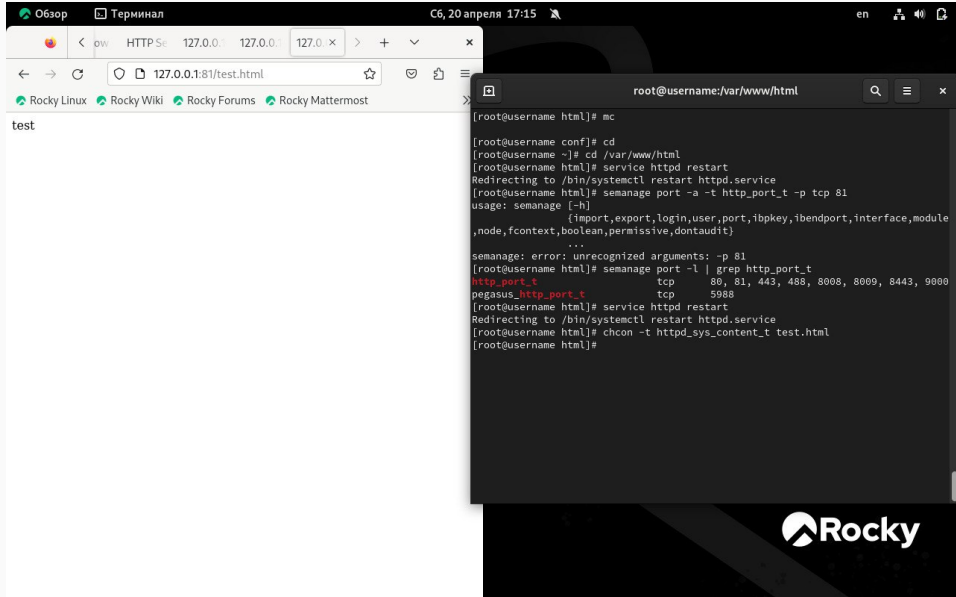
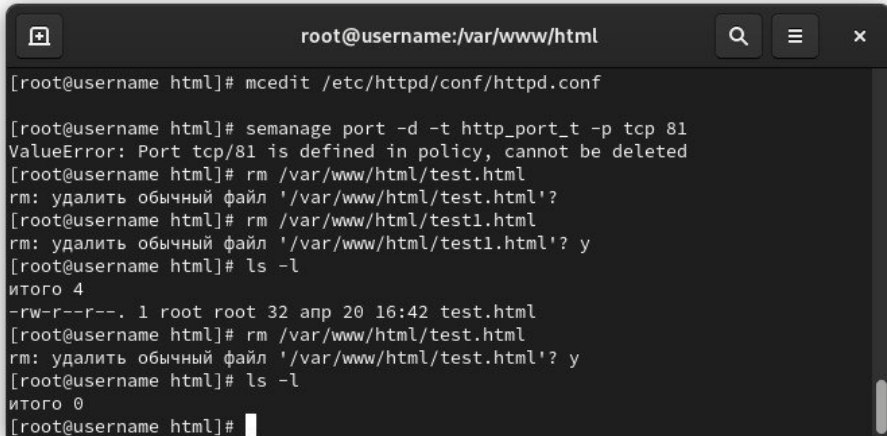


Рис. 10: доступ по http на 81 порт

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`



A terminal window with a dark background and light text. The title bar shows 'root@username:/var/www/html' and standard window controls. The terminal content shows a sequence of commands and their outputs: editing httpd.conf, deleting a port from semanage, removing test.html and test1.html files, and listing the directory contents before and after each deletion. The output shows the files being removed and the directory becoming empty.

```
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 апр 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. 11: Удаление файлов

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

- Запоминается последняя фраза. © Штирлиц

...