

# **Отчет по лабораторной работе №6**

**Основы информационной безопасности**

Машков Илья Евгеньевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>16</b>
	<b>Список литературы</b>	<b>17</b>

## Список иллюстраций

2.1	проверка режима работы SELinux . . . . .	6
2.2	Проверка работы Apache . . . . .	7
2.3	Контекст безопасности Apache . . . . .	7
2.4	Состояние переключателей SELinux . . . . .	8
2.5	Статистика по политике . . . . .	8
2.6	Типы поддиректорий . . . . .	9
2.7	Типы файлов . . . . .	9
2.8	Создание файла . . . . .	9
2.9	Контекст файла . . . . .	9
2.10	Отображение файла . . . . .	10
2.11	Изучение справки по команде . . . . .	11
2.12	Изменение контекста . . . . .	11
2.13	Отображение файла . . . . .	12
2.14	Попытка прочесть лог-файл . . . . .	12
2.15	Изменение порта . . . . .	13
2.16	Попытка прослушивания другого порта . . . . .	13
2.17	Проверка лог-файлов . . . . .	13
2.18	Проверка лог-файлов . . . . .	14
2.19	Проверка портов . . . . .	14
2.20	Перезапуск сервера . . . . .	14
2.21	Проверка сервера . . . . .	15
2.22	Проверка порта 81 . . . . .	15
2.23	Удаление файла . . . . .	15

## **Список таблиц**

# 1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux<sup>1</sup>. Проверить работу SELinx на практике совместно с веб-сервером Apache. [course?]

## 2 Выполнение лабораторной работы

Вошла в систему под своей учетной записью. Убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus` (рис. 2.1).

```
[iemashkov@localhost ~]$ getenforce
Enforcing
[iemashkov@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
[iemashkov@localhost ~]$
```

Рис. 2.1: проверка режима работы SELinux

Запускаю сервер `apache`, далее обращаюсь с помощью браузера к веб-серверу, запущенному на компьютере, он работает, что видно из вывода команды `service httpd status` (рис. 2.2).

```

[iemashkov@localhost ~]$ sudo systemctl start httpd
[iemashkov@localhost ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr
/lib/systemd/system/httpd.service.
[iemashkov@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; enabled; vendor prese
   Active: active (running) since Sat 2025-05-03 17:33:44 MSK; 52s ago
     Docs: man:httpd.service(8)
   Main PID: 129016 (httpd)
    Status: "Running, listening on: port 80"
     Tasks: 213 (limit: 37628)
    Memory: 41.9M
    CGroup: /system.slice/httpd.service
            └─129016 /usr/sbin/httpd -DFOREGROUND
              └─129023 /usr/sbin/httpd -DFOREGROUND
                └─129024 /usr/sbin/httpd -DFOREGROUND
                  └─129025 /usr/sbin/httpd -DFOREGROUND
                    └─129026 /usr/sbin/httpd -DFOREGROUND

```

Рис. 2.2: Проверка работы Apache

С помощью команды `ps auxZ | grep httpd` нашла веб-сервер Apache в списке процессов. Его контекст безопасности - `httpd_t` (рис. 2.3).

```

[iemashkov@localhost ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 129016 0.0 0.1 258208 10996 ?
Ss 17:33 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 129023 0.0 0.1 262912 8464 ?
S 17:33 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 129024 0.0 0.3 2697028 18260 ?
Sl 17:33 0:01 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 129025 0.1 0.3 2500364 18344 ?
Sl 17:33 0:01 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 129026 0.1 0.3 2500364 18332 ?
Sl 17:33 0:01 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 iemashk+ 129600 0.0 0.0 2
22012 1096 pts/0 S+ 17:54 0:00 grep --color=auto httpd
[iemashkov@localhost ~]$

```

Рис. 2.3: Контекст безопасности Apache

Просмотрела текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -b httpd` (рис. 2.4).

```
[iemashkov@localhost ~]$ 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
awstats_purge_apache_log_files off
boinc_execmem                 on
```

Рис. 2.4: Состояние переключателей SELinux

Просмотрела статистику по политике с помощью команды `seinfo`. Множество пользователей - 8, ролей - 15, типов - 5015. (рис. 2.5).

```
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:           31 (MLS enabled)
Target Policy:            selinux
Handle unknown classes:   allow
Classes:                  132
Sensitivities:            1
Types:                    5015
Users:                    8
Booleans:                 349
Allow:                    116272
Auditallow:               172
Type_trans:               262670
Type_member:              37
Role_allow:               40
Constraints:              72
MLS Constrains:           72
Permissives:              0
Defaults:                 7
Allowxperm:               0
Auditallowxperm:          0
Ibendportcon:             0
Initial SIDs:             27
Genfscon:                 107
Permissions:              464
Categories:               1024
Attributes:               258
Roles:                    15
Cond. Expr.:              399
Neverallow:               0
Dontaudit:                10529
Type_change:              94
Range_trans:              5989
Role_trans:               421
Validatetrans:            0
MLS Val. Tran:            0
Polcap:                   5
Typebounds:               0
Neverallowxperm:          0
Dontauditxperm:           0
Ibpkeycon:                0
Fs_use:                   34
Portcon:                  649
```

Рис. 2.5: Статистика по политике

Типы поддиректорий, находящихся в директории `/var/www`, с помощью коман-



ды `ls -lZ /var/www` следующие: владелец - root, права на изменения только у владельца. Файлов в директории нет (рис. 2.6).

```
[iemashkov@localhost ~]$ ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 фев 19 23
:08 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 фев 19 23
:08 html
```

Рис. 2.6: Типы поддиректорий

В директории `/var/www/html` нет файлов. (рис. 2.7).

```
[iemashkov@localhost ~]$ ls -lZ /var/www/html
итого 0
```

Рис. 2.7: Типы файлов

Создать файл может только суперпользователь, поэтому от его имени создаем файл `touch.html` со следующим содержанием:

```
html
<html>
<body>test</body>
</html>
```

(рис. 2.8).

```
[iemashkov@localhost ~]$ sudo touch /var/www/html/test.html
[sudo] пароль для iemashkov:
[iemashkov@localhost ~]$ sudo nano /var/www/html/test.html
[iemashkov@localhost ~]$ sudo cat /var/www/html/test.html
<html>
<body>test</body>
</html>
```

Рис. 2.8: Создание файла

Проверяю контекст созданного файла. По умолчанию это `httpd_sys_content_t` (рис. 2.9).

```
[iemashkov@localhost ~]$ ls -lZ /var/www/html
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 мая 3 1
7:59 test.html
```

Рис. 2.9: Контекст файла

Обращаюсь к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Файл был успешно отображён (рис. 2.10).

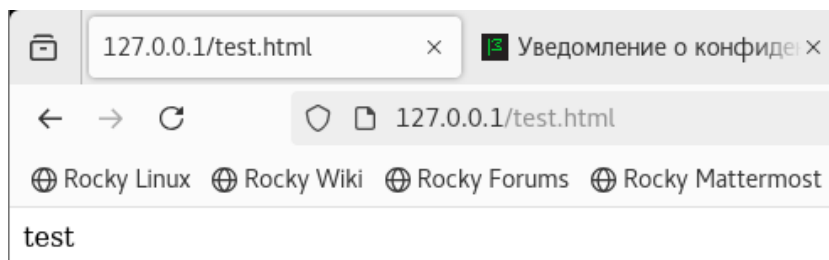
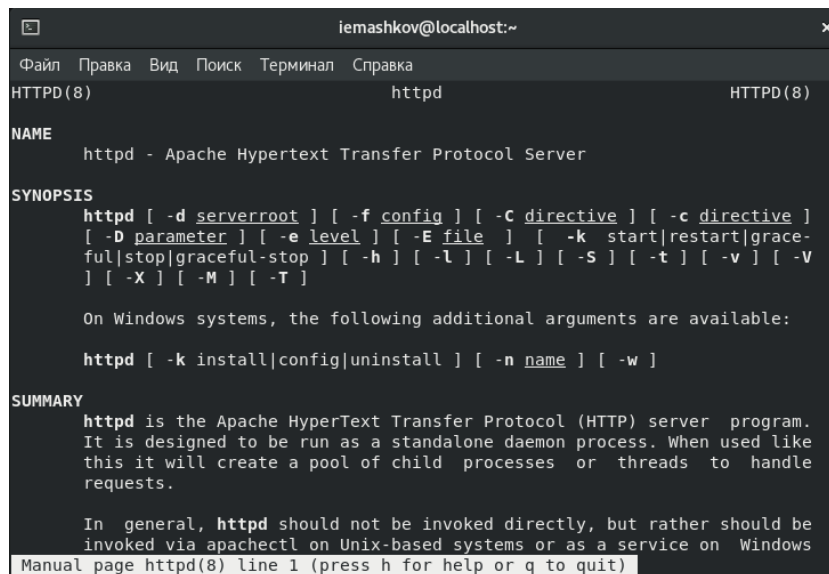


Рис. 2.10: Отображение файла

Изучила справку `man httpd_selinux`. Рассмотрим полученный контекст детально. 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 `/ргос` 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. (рис. 2.11).



```
iemashkov@localhost:~
Файл Правка Вид Поиск Терминал Справка
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
    requests.

    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
    Manual page httpd(8) line 1 (press h for help or q to quit)
```

Рис. 2.11: Изучение справки по команде

Изменяю 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` Контекст действительно поменялся (рис. 2.12).



```
[iemashkov@localhost ~]$ sudo chcon -t samba_share_t /var/www/html/test.html
[sudo] пароль для iemashkov:
[iemashkov@localhost ~]$ ls -lZ /var/www/html
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:samba_share_t:s0 33 мая  3 17:59 t
est.html
[iemashkov@localhost ~]$
```

Рис. 2.12: Изменение контекста

При попытке отображения файла в браузере получаем сообщение об ошибке (рис. 2.13).

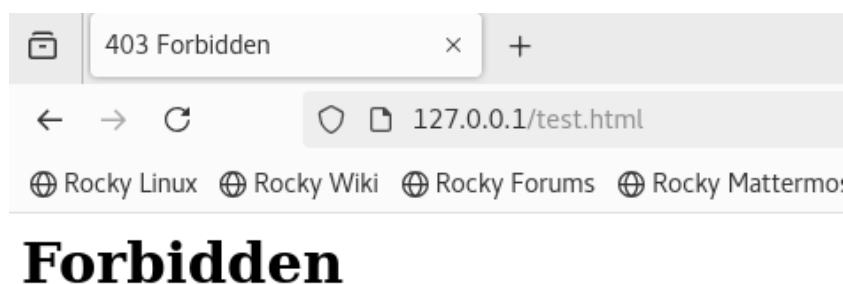


Рис. 2.13: Отображение файла

Файл не был отображён, хотя права доступа позволяют читать этот файл любому пользователю, потому что установлен контекст, к которому процесс httpd не должен иметь доступа.

Просматриваю log-файлы веб-сервера Apache и системный лог-файл: `tail /var/log/messages`. (рис. 2.14).

```
[iemashkov@localhost ~]$ ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 33 мая  3 17:59 /var/www/html/test.html
[iemashkov@localhost ~]$ sudo tail /var/log/messages
May  3 18:06:12 localhost systemd[1]: timedatex.service: Succeeded.
May  3 18:06:13 localhost dbus-daemon[826]: [system] Successfully activated service 'org.fedoraproject.Setroubleshootd'
May  3 18:06:13 localhost systemd[1]: Started SETroubleshoot daemon for processing new SELinux denial logs.
May  3 18:06:14 localhost setroubleshoot[131479]: failed to retrieve rpm info for /var/www/html/test.html
May  3 18:06:14 localhost dbus-daemon[826]: [system] Activating service name='org.fedoraproject.SetroubleshootPrivileged' requested by ':1.1093' (uid=984 pid=131479 comm="/usr/libexec/platform-python -Es /usr/sbin/setroub label="system u:s
```

Рис. 2.14: Попытка прочесть лог-файл

Чтобы запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в `/etc/services`) открываю файл `/etc/httpd/httpd.conf` для изменения.

Нахожу строку `Listen 80` и заменяю её на `Listen 81`. (рис. 2.15).

```
# 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 specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 81
#
# Dynamic Shared Object (DSO) Support
#
```

Рис. 2.15: Изменение порта

Выполняю перезапуск веб-сервера Apache. Произошёл сбой, потому что порт 80 для локальной сети, а 81 нет (рис. 2.16).

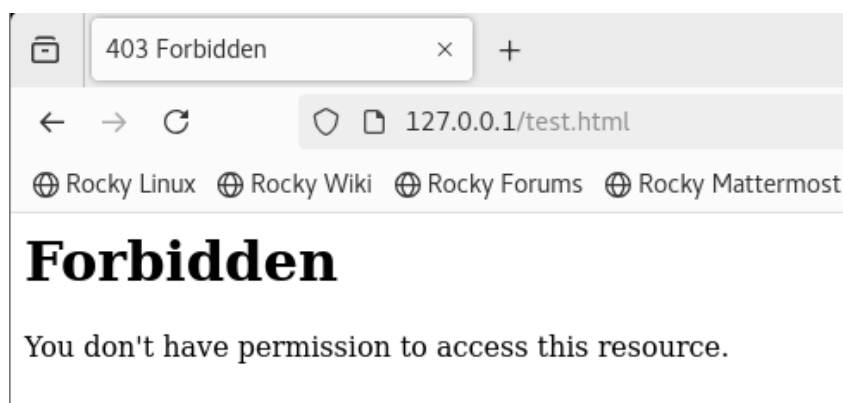


Рис. 2.16: Попытка прослушивания другого порта

Проанализируйте лог-файлы: `tail -nl /var/log/messages` (рис. 2.17).

```
[iemashkov@localhost ~]$ sudo tail -nl /var/log/messages
May  3 18:12:19 localhost org.gnome.Shell.desktop[2063]: libinput error: event2
- AT Translated Set 2 keyboard: client bug: event processing lagging behind by
12ms, your system is too slow
```

Рис. 2.17: Проверка лог-файлов

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. Запись появилась в файлу `error_log` (рис. 2.18).

```
[iemandkov@localhost ~]$ sudo cat /var/log/httpd/error_log
[Sat May 03 17:33:44.062054 2025] [core:notice] [pid 129016:tid 140193110817088]
SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Sat May 03 17:33:44.065107 2025] [suexec:notice] [pid 129016:tid 14019311081708
8] AH01232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
AH00558: httpd: Could not reliably determine the server's fully qualified domain
name, using localhost.localdomain. Set the 'ServerName' directive globally to s
uppress this message
[Sat May 03 17:33:44.081069 2025] [lbmethod heartbeat:notice] [pid 129016:tid 14
0193110817088] AH02282: No slotmem from mod_heartmonitor
[Sat May 03 17:33:44.082015 2025] [http2:warn] [pid 129016:tid 140193110817088]
AH02951: mod_ssl does not seem to be enabled
[Sat May 03 17:33:44.085125 2025] [mpm event:notice] [pid 129016:tid 14019311081
7088] AH00489: Apache/2.4.37 (Rocky Linux) configured -- resuming normal operati
ons
[Sat May 03 17:33:44.085152 2025] [core:notice] [pid 129016:tid 140193110817088]
AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'
[Sat May 03 18:02:18.518130 2025] [autoindex:error] [pid 129025:tid 140192514164
480] [client 127.0.0.1:37444] AH01276: Cannot serve directory /var/www/html/: No
matching DirectoryIndex (index.html) found, and server-generated directory inde
x forbidden by Options directive
[Sat May 03 18:05:46.981411 2025] [core:error] [pid 129025:tid 140192396588800]
(13)Permission denied: [client 127.0.0.1:46728] AH00035: access to /test.html de
```

Рис. 2.18: Проверка лог-файлов

Выполняю команду `semanage port -a -t http_port_t -p tcp 81` После это-  
го проверяю список портов командой `semanage port -l | grep http_port_t`  
Порт 81 появился в списке (рис. 2.19).

```
[iemandkov@localhost ~]$ semanage port -a -t http_port_t -p tcp 81
usage: semanage [-h]
               {import,export,login,user,port,ibpkey,ibendport,interface,module
,node,fcontext,boolean,permissive,dontaudit}
...
semanage: error: unrecognized arguments: -p 81
[iemandkov@localhost ~]$ semanage port -l | grep http_port_t
ValueError: Политика SELinux не задана, или нет доступа к Хранилищу.
[iemandkov@localhost ~]$ sudo 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
[iemandkov@localhost ~]$
```

Рис. 2.19: Проверка портов

Перезапускаю сервер Apache (рис. 2.20).

```
[iemandkov@localhost ~]$ sudo systemctl restart httpd
[iemandkov@localhost ~]$ sudo chcon -t httpd_sys_content_t /var/www/html/test.html
[iemandkov@localhost ~]$ sudo systemctl restart httpd
[iemandkov@localhost ~]$
```

Рис. 2.20: Перезапуск сервера

Теперь он работает, ведь мы внесли порт 81 в список портов `httpd_port_t`  
(рис. 2.21).

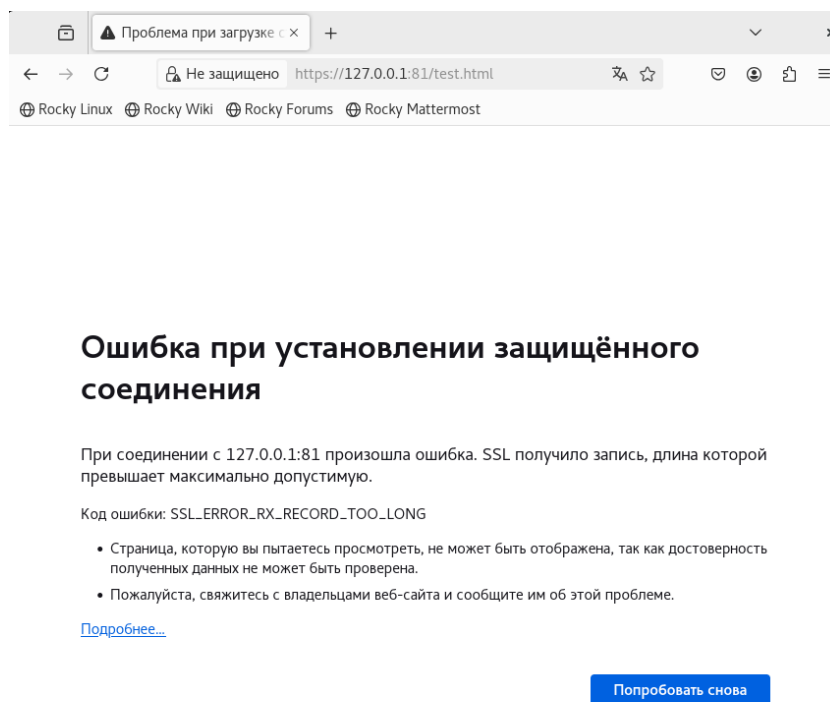


Рис. 2.21: Проверка сервера

Возвращаю в файле `/etc/httpd/httpd.conf` порт 80, вместо 81. Проверяю, что порт 81 удален, это правда. (рис. 2.22).

```
[iemashkov@localhost ~]$ sudo nano /etc/httpd/conf/httpd.conf
[iemashkov@localhost ~]$ sudo semanage port -d -t http_port_t -p tcp 81
ValueError: Порт tcp/81 определен на уровне политики и не может быть удален
```

Рис. 2.22: Проверка порта 81

Далее удаляю файл `test.html`, проверяю, что он удален(рис. 2.23).

```
[iemashkov@localhost html]$ ls -lZ /var/www/html
итого 0
[iemashkov@localhost html]$
```

Рис. 2.23: Удаление файла

## 3 Выводы

В ходе выполнения данной лабораторной работы были развиты навыки администрирования ОС Linux, получено первое практическое знакомство с технологией SELinux и проверена работа SELinux на практике совместно с веб-сервером Apache.



# Список литературы

ОИБ