

Отчет по лабораторной работе №6

Основы информационной безопасности

Закиров Нурислам, НКАбд-01-23

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Выводы	19
	Список литературы	20

Список иллюстраций

3.1	проверка режима работы SELinux	8
3.2	Проверка работы Apache	9
3.3	Контекст безопасности Apache	9
3.4	Состояние переключателей SELinux	10
3.5	Статистика по политике	10
3.6	Типы поддиректорий	11
3.7	Типы файлов	11
3.8	Создание файла	11
3.9	Контекст файла	12
3.10	Отображение файла	12
3.11	Изучение справки по команде	13
3.12	Изменение контекста	13
3.13	Отображение файла	13
3.14	Попытка прочесть лог-файл	14
3.15	Изменение файла	14
3.16	Изменение порта	15
3.17	Попытка прослушивания другого порта	16
3.18	Проверка лог-файлов	16
3.19	Проверка лог-файлов	17
3.20	Проверка портов	17
3.21	Перезапуск сервера	17
3.22	Проверка сервера	18
3.23	Проверка порта 81	18
3.24	Удаление файла	18

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 Теоретическое введение

1. **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.

2. **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).

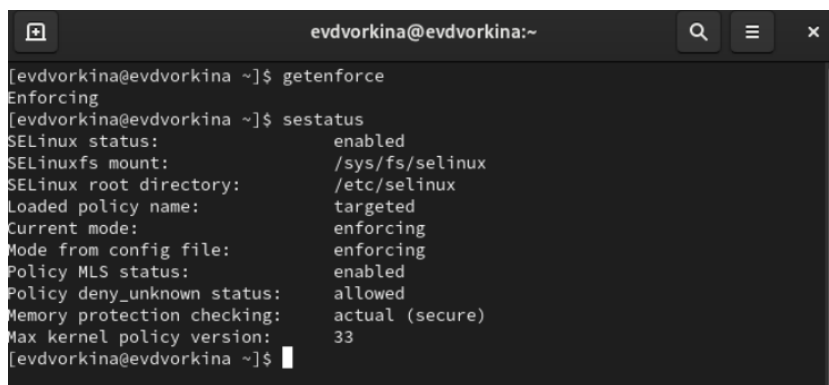
Для чего нужен Apache сервер:

- 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

3 Выполнение лабораторной работы

Вошёл в систему под своей учетной записью. Убедился, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus` (рис. 3.1).

A screenshot of a terminal window titled "evdvorkina@evdvorkina:~". The terminal shows the output of two commands. First, the user enters `getenforce`, and the output is `Enforcing`. Then, the user enters `sestatus`, and the output shows various SELinux configuration details.

```
[evdvorkina@evdvorkina ~]$ getenforce
Enforcing
[evdvorkina@evdvorkina ~]$ 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
[evdvorkina@evdvorkina ~]$
```

Рис. 3.1: 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. 3.2).


```
[evdvorkina@evdvorkina ~]$ sudo systemctl start httpd
[evdvorkina@evdvorkina ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[evdvorkina@evdvorkina ~]$ 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; preset: disabled)
   Active: active (running) since Sat 2024-04-20 04:52:10 MSK; 31s ago
     Docs: man:httpd.service(8)
   Main PID: 30093 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served: 0"
      Tasks: 213 (limit: 10899)
    Memory: 37.9M
       CPU: 301ms
    CGroup: /system.slice/httpd.service
            └─30093 /usr/sbin/httpd -DFOREGROUND
              └─30133 /usr/sbin/httpd -DFOREGROUND
                └─30134 /usr/sbin/httpd -DFOREGROUND
                  └─30135 /usr/sbin/httpd -DFOREGROUND
                    └─30136 /usr/sbin/httpd -DFOREGROUND

anp 20 04:52:10 evdvorkina systemd[1]: Starting The Apache HTTP Server...
anp 20 04:52:10 evdvorkina httpd[30093]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 instead; this can be fixed by editing the 'ServerName' directive in the configuration file at /etc/httpd/conf/httpd.conf.
```

Рис. 3.2: Проверка работы Apache

С помощью команды `ps auxZ | grep httpd` нашёл веб-сервер Apache в списке процессов. Его контекст безопасности - `httpd_t` (рис. 3.3).

```
[evdvorkina@evdvorkina ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 30093 0.1 0.6 20340 11624 ? Ss 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 30133 0.0 0.4 21676 7436 ? S 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 30134 0.0 1.0 2193664 19320 ? Sl 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 30135 0.0 0.8 2062528 15228 ? Sl 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 30136 0.0 0.8 2062528 15228 ? Sl 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 evdvork+ 42224 0.0 0.1 221688 2388 pts/0 S+ 04:53 0:00 grep --color=auto httpd
[evdvorkina@evdvorkina ~]$
```

Рис. 3.3: Контекст безопасности Apache

Просмотрел текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -bigrep httpd` (рис. 3.4).

```
[evdvorkina@evdvorkina ~]$ 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
cdrecord_read_content           off
cluster_can_network_connect    off
...
```

Рис. 3.4: Состояние переключателей SELinux

Просмотрел статистику по политике с помощью команды `seinfo`. Множество пользователей - 8, ролей - 39, типов - 5135. (рис. 3.5).

```
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:            33 (MLS enabled)
Target Policy:             selinux
Handle unknown classes:    allow
Classes:                   135
Sensitivities:             1
Types:                     5135
Users:                     8
Booleans:                  357
Allow:                     65409
Auditallow:                172
Type_trans:                267813
Type_member:               37
Role allow:                39
Constraints:               70
MLS Constrain:             72
Permissives:               2
Defaults:                  7
Allowxperm:                0
Auditallowxperm:          0
Ibendportcon:             0
Initial SIDs:              27
Genfscon:                  109
Netifcon:                  0
Permissions:               457
Categories:               1024
Attributes:                259
Roles:                     15
Cond. Expr.:               390
Neverallow:                0
Dontaudit:                 8647
Type_change:               94
Range_trans:               6164
Role_trans:                419
Validatetrans:             0
MLS Val. Tran:             0
Polcap:                    6
Typebounds:                0
Neverallowxperm:          0
Dontauditxperm:           0
Ibpkeycon:                 0
Fs_use:                    35
Portcon:                   665
Nodecon:                   0
```

Рис. 3.5: Статистика по политике

Типы поддиректорий, находящихся в директории `/var/www`, с помощью команды `ls -lZ /var/www` следующие: владелец - root, права на изменения только у владельца. Файлов в директории нет (рис. 3.6).

```
[evdvorkina@evdvorkina ~]$ 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 12:35 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 окт 28 12:35 html
```

Рис. 3.6: Типы поддиректорий

В директории `/var/www/html` нет файлов. (рис. 3.7).

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

Рис. 3.7: Типы файлов

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

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

(рис. 3.8).

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

Рис. 3.8: Создание файла

Проверяю контекст созданного файла. По умолчанию это `httpd_sys_content_t` (рис. 3.9).

```

[evdvorkina@evdvorkina ~]$ ls -lZ /var/www/html/
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 anp 20 05:01 test.html
[evdvorkina@evdvorkina ~]$

```

Рис. 3.9: Контекст файла

Обращаюсь к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Файл был успешно отображён (рис. 3.10).

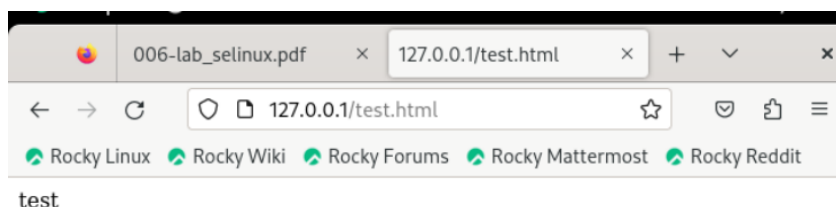
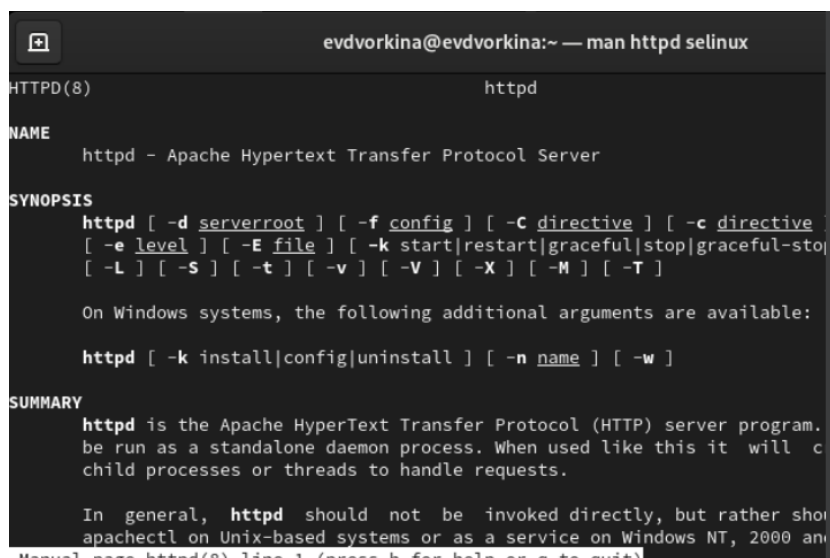


Рис. 3.10: Отображение файла

Изучил 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. 3.11).



```
evdvorkina@evdvorkina:~ — man httpd selinux
HTTPD(8)                                httpd

NAME
    httpd - Apache Hypertext Transfer Protocol Server

SYNOPSIS
    httpd [ -d serverroot ] [ -f config ] [ -C directive ] [ -c directive ]
    [ -e level ] [ -E file ] [ -k start|restart|graceful|stop|graceful-stop ]
    [ -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 ]

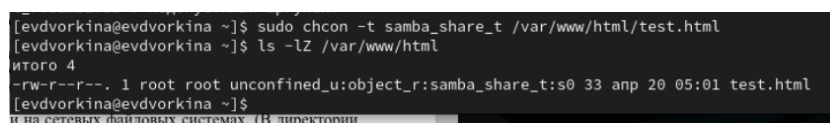
SUMMARY
    httpd is the Apache HyperText Transfer Protocol (HTTP) server program.
    It can be run as a standalone daemon process. When used like this it will
    create child processes or threads to handle requests.

    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 NT, 2000 and
    later.

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

Рис. 3.11: 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. 3.12).



```
[evdvorkina@evdvorkina ~]$ sudo chcon -t samba_share_t /var/www/html/test.html
[evdvorkina@evdvorkina ~]$ ls -lZ /var/www/html
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:samba_share_t:s0 33 anp 20 05:01 test.html
[evdvorkina@evdvorkina ~]$
```

Рис. 3.12: Изменение контекста

При попытке отображения файла в браузере получаем сообщение об ошибке (рис. 3.13).

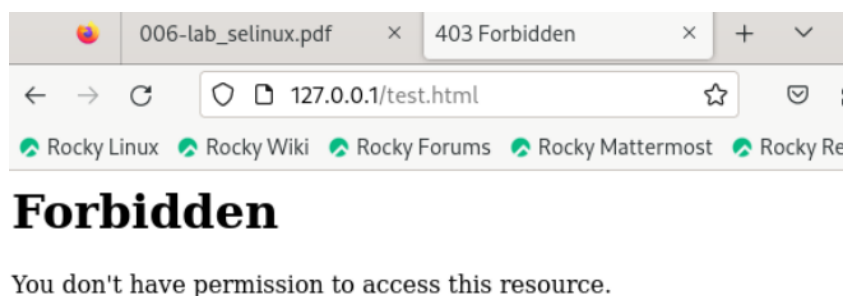


Рис. 3.13: Отображение файла

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`. 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`. (рис. 3.14).

```
[evdvorkina@evdvorkina ~]$ ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 33 апр 20 05:01 /var/www/html/test.html
[evdvorkina@evdvorkina ~]$ tail /var/log/audit/audit.log
tail: невозможно открыть '/var/log/audit/audit.log' для чтения: Отказано в доступе
[evdvorkina@evdvorkina ~]$ sudo tail /var/log/audit/audit.log
type=SYSCALL msg=audit(1713578987.972:279): arch=c000003e syscall=262 success=no exit=-13 a0=ffffff9c a1=7f97cc004c10 a2=7f97c2ffc8b0 a3=100 items=0 p
pid=30093 ppid=30130 uid=0 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="httpd" exe="/usr/s
bin/httpd" subj=system_u:system_r:httpd_t:s0 key=(null)ARCH=x86_64 SYSCALL=newfstatat AUDID="unset" UID="apache" GID="apache" EUID="apach
e" FSUID="apache" EGD="apache" SGID="apache" FSGID="apache"
type=PROC TITLE msg=audit(1713578987.972:279): proc_title=2f57372f7326208e3f6874747064602d446df324547224f504e44
type=SERVICE_START msg=audit(1713578989.341:280): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg="unit=setroubleshoot
d comm='systemd' exe='/usr/lib/systemd/systemd' hostname=? addr=? terminal=? res=success"UID="root" AUDID="unset"
type=SERVICE_START msg=audit(1713578990.334:281): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg="unit=dbus-1.11-org.
fedora.project.SetroubleshootPrivileged1 comm='systemd' exe='/usr/lib/systemd/systemd' hostname=? addr=? terminal=? res=success"UID="root" AUDID="unset"
type=SERVICE_STOP msg=audit(1713579003.845:282): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg="unit=dbus-1.11-org.f
edora.project.SetroubleshootPrivileged1 comm='systemd' exe='/usr/lib/systemd/systemd' hostname=? addr=? terminal=? res=success"UID="root" AUDID="unset"
type=SERVICE_STOP msg=audit(1713579003.985:283): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg="unit=setroubleshoot
d comm='systemd' exe='/usr/lib/systemd/systemd' hostname=? addr=? terminal=? res=success"UID="root" AUDID="unset"
type=USER_ACCT msg=audit(1713579112.875:284): pid=43179 uid=1000 auid=1000 ses=3 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg="op=PA
rtAccounting grants=pam_unix,pam_localuser acct='evdvorkina' exe='/usr/bin/sudo' hostname=? addr=? terminal=/dev/pts/0 res=success"UID="evdvorkina"
AUDID="evdvorkina"
type=USER_CMD msg=audit(1713579112.876:285): pid=43179 uid=1000 auid=1000 ses=3 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg="cmd='/'
home/evdvorkina" cmd='7461696c202f7061727270636f672f61756469742f61756469742e636f67 exe='/usr/bin/sudo' terminal=pts/0 res=success"UID="evdvorkina" AUDID="
```

Рис. 3.14: 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. (рис. 3.15).

```
evdvorkina@evdvorkina ~]$ sudo nano /etc/httpd/conf/httpd.conf
evdvorkina@evdvorkina ~]$
```

Рис. 3.15: Изменение файла

Нахожу строчку `Listen 80` и заменяю её на `Listen 81`. (рис. 3.16).

```
# 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
# to be loaded here.
```

^G Справка	^O Записать	^W Поиск	^K Вырезать	^T Выполнить
^X Выход	^R ЧитФайл	^N Замена	^U Вставить	^J Выводить

Рис. 3.16: Изменение порта

Выполняю перезапуск веб-сервера Apache. Произошёл сбой, потому что порт 80 для локальной сети, а 81 нет (рис. 3.17).

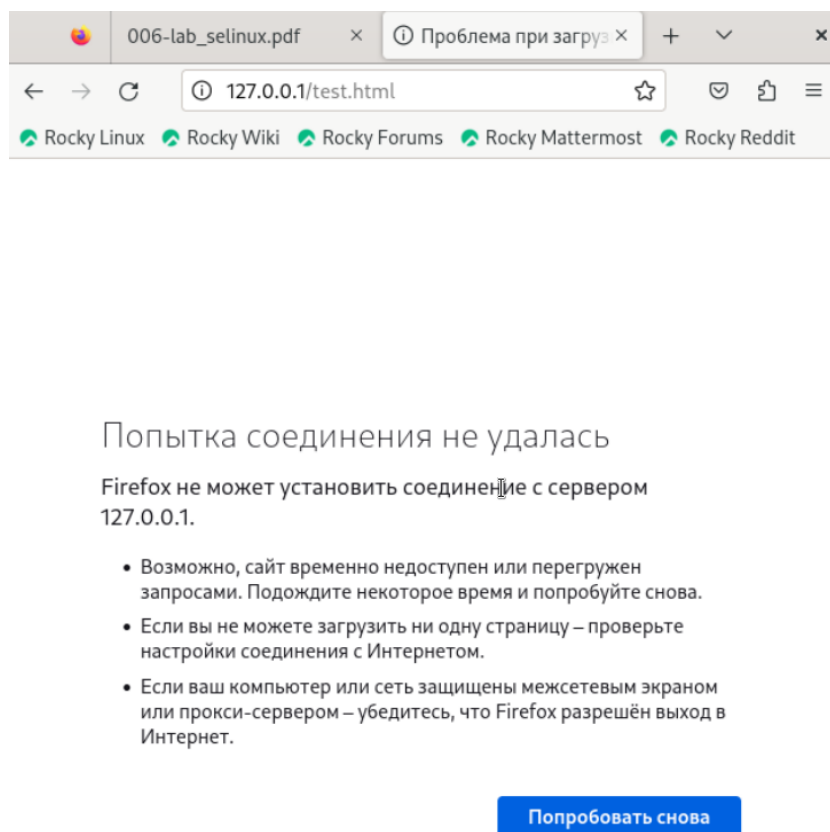


Рис. 3.17: Попытка прослушивания другого порта

Проанализируйте лог-файлы: `tail -nl /var/log/messages` (рис. 3.18).

```
[evdvorkina@evdvorkina ~]$ sudo tail -nl /var/log/messages
Apr 20 05:16:31 evdvorkina systemd[1]: Started The Apache HTTP Server.
[evdvorkina@evdvorkina ~]$
```

Рис. 3.18: 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 файле `error_log` (рис. 3.19).


```

[evdvorkina@evdvorkina ~]$ sudo cat /var/log/httpd/error_log
[Sat Apr 20 04:52:10.304359 2024] [core:notice] [pid 30093:tid 30093] SELinux
policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Sat Apr 20 04:52:10.307330 2024] [suexec:notice] [pid 30093:tid 30093] AH0123
2: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
AH00558: httpd: Could not reliably determine the server's fully qualified doma
in name, using fe80::a00:27ff:fe98:bdea%enp0s3. Set the 'ServerName' directive
globally to suppress this message
[Sat Apr 20 04:52:10.371973 2024] [lbmethod_heartbeat:notice] [pid 30093:tid 3
0093] AH02282: No slotmem from mod_heartbeat
[Sat Apr 20 04:52:10.389422 2024] [mpm_event:notice] [pid 30093:tid 30093] AH0
0489: Apache/2.4.57 (Rocky Linux) configured -- resuming normal operations
[Sat Apr 20 04:52:10.389524 2024] [core:notice] [pid 30093:tid 30093] AH00094:
Command line: '/usr/sbin/httpd -D FOREGROUND'
[Sat Apr 20 05:09:47.974451 2024] [core:error] [pid 30136:tid 30312] (13)Permi
ssion denied: [client 127.0.0.1:44098] AH00035: access to /test.html denied (f
ilesystem path '/var/www/html/test.html') because search permissions are missi
ng on a component of the path
[Sat Apr 20 05:15:41.743945 2024] [core:error] [pid 30134:tid 30322] (13)Permi
ssion denied: [client 127.0.0.1:58006] AH00035: access to /test.html denied (f
ilesystem path '/var/www/html/test.html') because search permissions are missi
ng on a component of the path
[Sat Apr 20 05:16:30.614988 2024] [mpm_event:notice] [pid 30093:tid 30093] AH0
0492: caught SIGWINCH, shutting down gracefully
[Sat Apr 20 05:16:31.953947 2024] [core:notice] [pid 43296:tid 43296] SELinux

```

Рис. 3.19: Проверка лог-файлов

Выполняю команду `semanage port -a -t http_port_t -p tcp 81` После этого проверяю список портов командой `semanage port -l | grep http_port_t` Порт 81 появился в списке (рис. 3.20).

```

[evdvorkina@evdvorkina ~]$ sudo semanage port -a -t http_port_t -p tcp 81
ValueError: Port tcp/81 already defined
[evdvorkina@evdvorkina ~]$ semanage port -l | grep http_port_t
ValueError: Политика SELinux не задана, или нет доступа к хранилищу.
[evdvorkina@evdvorkina ~]$ sudo semanage port -l | grep http_port_t
http_port_t                tcp      80, 81, 443, 488, 8008, 8009, 8443, 90
00
pegasus_http_port_t        tcp      5988
[evdvorkina@evdvorkina ~]$

```

Рис. 3.20: Проверка портов

Перезапускаю сервер Apache (рис. 3.21).

```

command not found
[evdvorkina ~]$ sudo systemctl restart httpd
[evdvorkina ~]$ sudo chcon -t httpd_sys_content_t /var/www/html/test.html
[evdvorkina ~]$ sudo systemctl restart httpd
[evdvorkina ~]$

```

Рис. 3.21: Перезапуск сервера

Теперь он работает, ведь мы внесли порт 81 в список портов `httpd_port_t` (рис. 3.22).

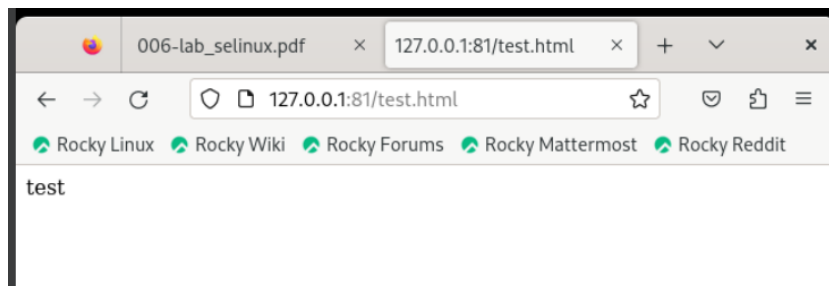


Рис. 3.22: Проверка сервера

Возвращаю в файле `/etc/httpd/httpd.conf` порт 80, вместо 81. Проверяю, что порт 81 удален, это правда. (рис. 3.23).

```
[evdvorkina@evdvorkina ~]$ sudo nano /etc/httpd/conf/httpd.conf
[evdvorkina@evdvorkina ~]$ semanage port -d -t http_port_t -p tcp 81
ValueError: Политика SELinux не задана, или нет доступа к хранилищу.
[evdvorkina@evdvorkina ~]$ sudo semanage port -d -t http_port_t -p tcp 81
ValueError: Port tcp/81 is defined in policy, cannot be deleted
[evdvorkina@evdvorkina ~]$
```

Рис. 3.23: Проверка порта 81

Далее удаляю файл `test.html`, проверяю, что он удален(рис. 3.24).

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

Рис. 3.24: Удаление файла

4 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