

Лабораторная работа №6

Дисциплина: основы информационной безопасности

Астраханцева А. А.

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Выводы	17
5	Список литературы. Библиография	18

Список иллюстраций

3.1	Проверка работы SELinux	7
3.2	Проверка работы веб-сервера	8
3.3	Контекст безопасности веб-сервера Apache	8
3.4	Текущее состояние переключателей SELinux для Apache	9
3.5	Статистика по политике	10
3.6	Типы файлов в директориях /var/www и /var/www/html	10
3.7	Создание html-файла	11
3.8	Контекст файла /var/www/html	11
3.9	Файл /var/www/html	11
3.10	Изменение контекста файла /var/www/html/test.html	11
3.11	Сообщение об ошибке	12
3.12	Просмотр log-файлов	12
3.13	Просмотр log-файлов	13
3.14	Просмотр файлов /var/log/messages, /var/log/http/error_log	14
3.15	Добавление порта 81	14
3.16	Запуск веб-сервера Apache	15
3.17	Исправление конфигурационного файла apache	15
3.18	Удаление привязки к порту 81 и файла /var/www/html/test.html:	16


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

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 [1].

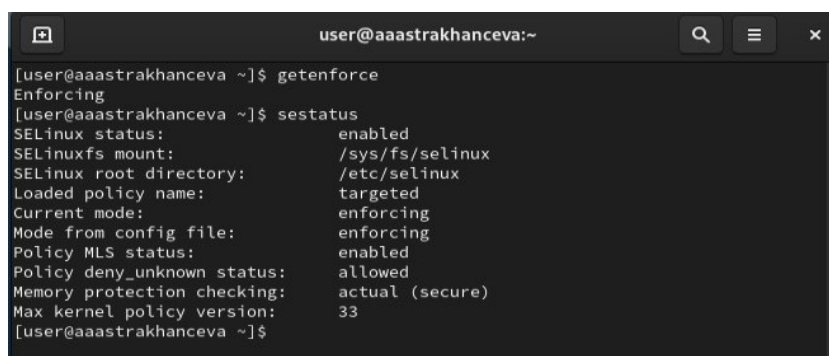
Apache  это программное обеспечение с открытым исходным кодом, которое позволяет создавать веб-сервер.

Веб-сервер — это программа, без которой не может работать сайт в интернете. Сайт — это набор файлов, например, HTML, CSS и JS. В каждом файле находится информация о картинках, тексте, кнопках, шрифтах и других элементах внешнего вида сайта. Все эти файлы находятся на физическом сервере, который имеет или арендует владелец сайта. Чтобы показать пользователю сайт, браузер должен связаться с сервером и получить эти файлы. Для связи с сервером браузер просит помощи у веб-сервера. Веб-сервер получает запрос от браузера, ищет ресурсы сайта и пересылает их ему. Только после этого браузер показывает контент пользователю.

Apache состоит из ядра и модулей. Ядро выполняет основные функции: обработка конфигурационных файлов, работа с протоколом HTTP, система загрузки модулей. Оно может работать самостоятельно без модулей, но тогда функционал программы будет крайне ограничен. Ядро создала команда компании Apache Software Foundation без вмешательства сторонних разработчиков. [2].

3 Выполнение лабораторной работы

Войдите в систему с полученными учётными данными и убедитесь, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`. (рис. 3.1)

A terminal window titled 'user@aaastrakhanceva:~' with search, menu, and close buttons. It shows the execution of 'getenforce' and 'sestatus' commands. The output of 'getenforce' is 'Enforcing'. The output of 'sestatus' shows SELinux is enabled, mounted at /sys/fs/selinux, with root directory /etc/selinux, loaded policy name 'targeted', and current mode 'enforcing'. Other details include 'Mode from config file: enforcing', 'Policy MLS status: enabled', 'Policy deny_unknown status: allowed', 'Memory protection checking: actual (secure)', and 'Max kernel policy version: 33'.

```
[user@aaastrakhanceva ~]$ getenforce
Enforcing
[user@aaastrakhanceva ~]$ 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
[user@aaastrakhanceva ~]$
```

Рис. 3.1: Проверка работы SELinux

Обратитесь с помощью браузера к веб-серверу, запущенному на вашем компьютере, и убедитесь, что последний работает: `service httpd status`: (рис. 3.2)

```

[root@aaastrakhanceva user]# sudo systemctl start httpd
[root@aaastrakhanceva user]# exit
exit
[user@aaastrakhanceva ~]$ 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; preset: 0)
   Active: active (running) since Sat 2024-04-27 12:31:26 MSK; 23s ago
     Docs: man:httpd.service(8)
  Main PID: 42807 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes"
       Tasks: 213 (limit: 10900)
      Memory: 23.7M
         CPU: 103ms
    CGroup: /system.slice/httpd.service
            └─42807 /usr/sbin/httpd -DFOREGROUND
              └─42808 /usr/sbin/httpd -DFOREGROUND
                └─42811 /usr/sbin/httpd -DFOREGROUND
                  └─42813 /usr/sbin/httpd -DFOREGROUND
                    └─42815 /usr/sbin/httpd -DFOREGROUND
[user@aaastrakhanceva ~]$
[user@aaastrakhanceva ~]$ sudo yum install httpd
[user@aaastrakhanceva ~]$ sudo systemctl enable httpd
[sudo] password for user:
user is not in the sudoers file. This incident will be reported.
[user@aaastrakhanceva ~]$ su
Password:
[root@aaastrakhanceva user]# sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@aaastrakhanceva user]# exit
exit
[user@aaastrakhanceva ~]$ 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: 0)
   Active: active (running) since Sat 2024-04-27 12:31:26 MSK; 2min 0s ago
     Docs: man:httpd.service(8)
  Main PID: 42807 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes"
       Tasks: 213 (limit: 10900)
      Memory: 23.7M
         CPU: 145ms
    CGroup: /system.slice/httpd.service
            └─42807 /usr/sbin/httpd -DFOREGROUND
              └─42808 /usr/sbin/httpd -DFOREGROUND
                └─42811 /usr/sbin/httpd -DFOREGROUND
                  └─42813 /usr/sbin/httpd -DFOREGROUND
                    └─42815 /usr/sbin/httpd -DFOREGROUND
lines 1-15/15 (END)...skipping...
● httpd.service - The Apache HTTP Server

```

Рис. 3.2: Проверка работы веб-сервера

Найдите веб-сервер Apache в списке процессов, определите его контекст безопасности и занесите эту информацию в отчёт (рис. 3.3).

```

[user@aaastrakhanceva ~]$ ps aux | grep httpd
system_u:system_r:httpd_t:s0 root 42807 0.0 0.6 20340 11612 ?
Ss 12:31 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 42808 0.0 0.4 21676 7460 ?
S 12:31 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 42811 0.0 0.6 1079488 11032 ?
Sl 12:31 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 42813 0.0 0.7 1210624 13080 ?
Sl 12:31 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 42815 0.0 0.6 1079488 11032 ?
Sl 12:31 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 user 43148 0.0 0.1 2216
64 2264 pts/0 S+ 12:35 0:00 grep --color=auto httpd
[user@aaastrakhanceva ~]$

```

Рис. 3.3: Контекст безопасности веб-сервера Apache

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


```

[root@aaastrakhanceva user]# 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
cluster_manage_all_files       off
cluster_use_execmem            off
cobbler_anon_write             off
cobbler_can_network_connect    off
cobbler_use_cifs               off
cobbler_use_nfs                off
collectd_tcp_network_connect   off
colord_use_nfs                 off
condor_tcp_network_connect     off
conman_can_network             off

```

Рис. 3.4: Текущее состояние переключателей SELinux для Apache

Посмотрите статистику по политике с помощью команды `seinfo`, также определите множество пользователей, ролей, типов (рис. 3.5).

```

[root@aaastrakhanceva user]# 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:                   5135     Attributes:           259
Users:                   8         Roles:               15
Booleans:                357      Cond. Expr.:         390
Allow:                   65409     Neverallow:          0
Auditallow:              172      Dontaudit:           8647
Type_trans:              267813   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:             2        Polcap:              6
Defaults:                7        Typebounds:          0
Allowxperm:              0        Neverallowxperm:     0
Auditallowxperm:         0        Dontauditxperm:     0
Ibendportcon:            0        Ibpkeycon:           0
Initial SIDs:            27       Fs_use:              35
Genfscon:                109      Portcon:             665
Netifcon:                0        Nodecon:             0

```

Рис. 3.5: Статистика по политике

Определите тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды `ls -lZ /var/www` (2 директории). Определите тип файлов, находящихся в директории /var/www/html: `ls -lZ /var/www/html` (пустая). (рис. 3.6).

```

[root@aaastrakhanceva user]# ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Oct 28
12:35 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 Oct 28
12:35 html
[root@aaastrakhanceva user]# ls -lZ /var/www/html
total 0
[root@aaastrakhanceva user]# ls -lZ /var/www/html
total 0
[root@aaastrakhanceva user]#

```

Рис. 3.6: Типы файлов в директориях /var/www и /var/www/html

Создайте от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания (рис. 3.7):

test

.

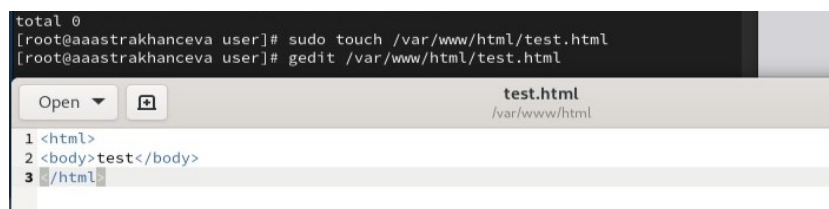


Рис. 3.7: Создание html-файла

Проверьте контекст созданного вами файла. Занесите в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории `/var/www/html`. (рис. 3.8).

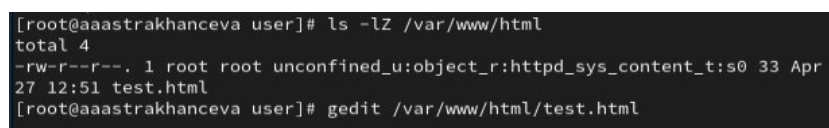


Рис. 3.8: Контекст файла `/var/www/html`

Обратитесь к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Убедитесь, что файл был успешно отображён. (рис. 3.9).

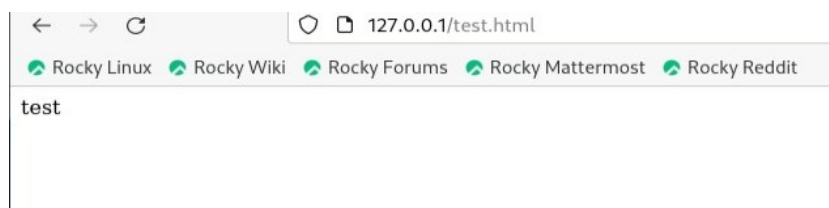


Рис. 3.9: Файл `/var/www/html`

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`. После этого проверьте, что контекст поменялся (рис. 3.10).

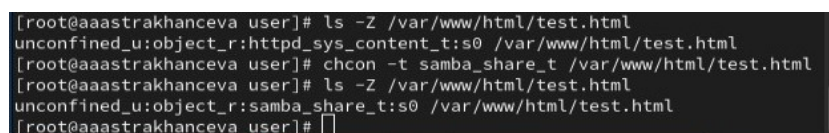


Рис. 3.10: Изменение контекста файла `/var/www/html/test.html`

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` (рис. 3.11).

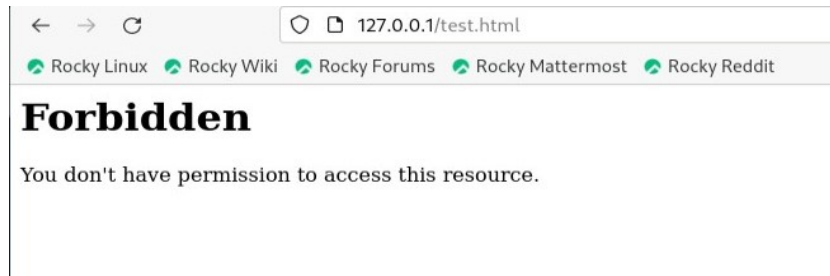


Рис. 3.11: Сообщение об ошибке

Просмотрите log-файлы веб-сервера Apache. Также просмотрите системный лог-файл: `tail /var/log/messages` (рис. 3.12).

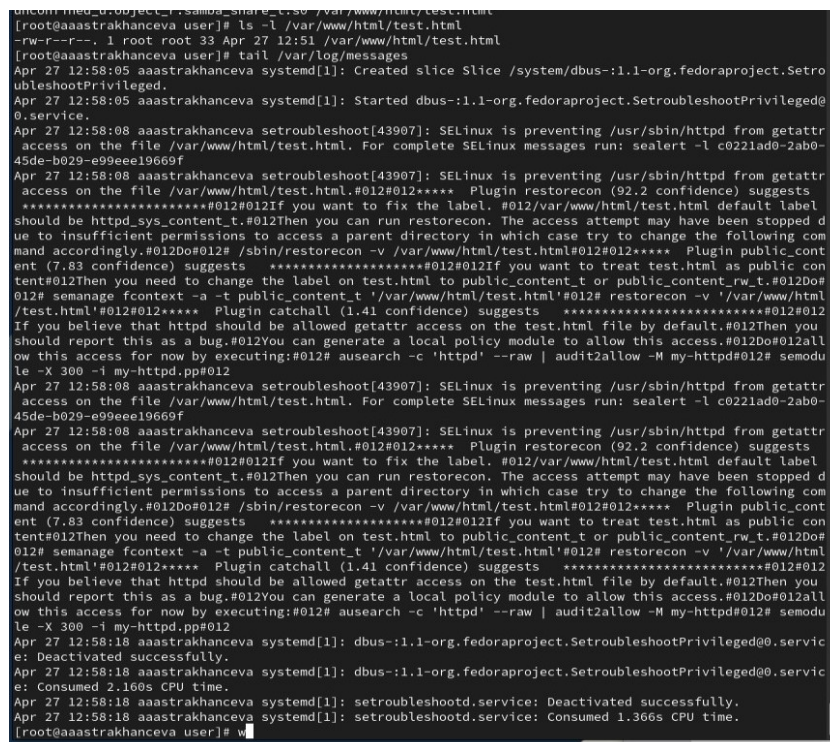
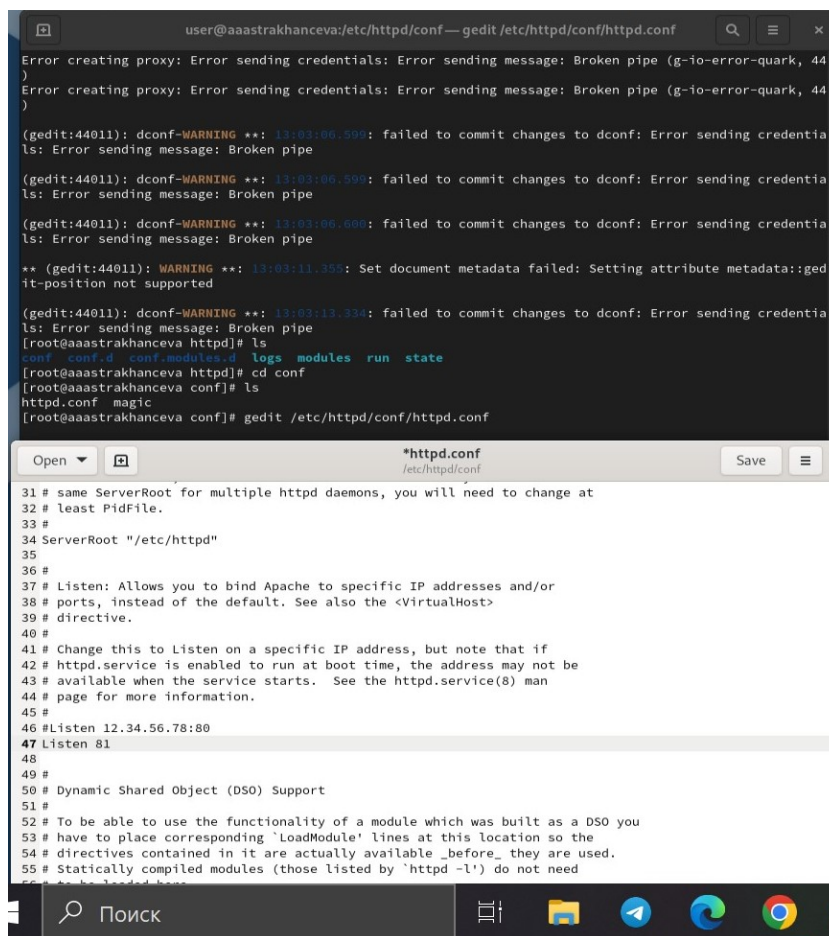


Рис. 3.12: Просмотр log-файлов

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. 3.13).



```
user@aaastrakhanceva:/etc/httpd/conf — gedit /etc/httpd/conf/httpd.conf
Error creating proxy: Error sending credentials: Error sending message: Broken pipe (g-io-error-quark, 44)
Error creating proxy: Error sending credentials: Error sending message: Broken pipe (g-io-error-quark, 44)
(gedit:44011): dconf-WARNING **: 13:03:00.599: failed to commit changes to dconf: Error sending credentials: Error sending message: Broken pipe
(gedit:44011): dconf-WARNING **: 13:03:00.599: failed to commit changes to dconf: Error sending credentials: Error sending message: Broken pipe
(gedit:44011): dconf-WARNING **: 13:03:00.600: failed to commit changes to dconf: Error sending credentials: Error sending message: Broken pipe
** (gedit:44011): WARNING **: 13:03:11.355: Set document metadata failed: Setting attribute metadata::gedit-position not supported
(gedit:44011): dconf-WARNING **: 13:03:13.334: failed to commit changes to dconf: Error sending credentials: Error sending message: Broken pipe
[root@aaastrakhanceva httpd]# ls
conf conf.d conf.modules.d logs modules run state
[root@aaastrakhanceva httpd]# cd conf
[root@aaastrakhanceva conf]# ls
httpd.conf magic
[root@aaastrakhanceva conf]# gedit /etc/httpd/conf/httpd.conf

*httpd.conf
/etc/httpd/conf

31 # same ServerRoot for multiple httpd daemons, you will need to change at
32 # least PidFile.
33 #
34 ServerRoot "/etc/httpd"
35 #
36 #
37 # Listen: Allows you to bind Apache to specific IP addresses and/or
38 # ports, instead of the default. See also the <VirtualHost>
39 # directive.
40 #
41 # Change this to Listen on a specific IP address, but note that if
42 # httpd.service is enabled to run at boot time, the address may not be
43 # available when the service starts. See the httpd.service(8) man
44 # page for more information.
45 #
46 #Listen 12.34.56.78:80
47 Listen 81
48 #
49 #
50 # Dynamic Shared Object (DSO) Support
51 #
52 # To be able to use the functionality of a module which was built as a DSO you
53 # have to place corresponding 'LoadModule' lines at this location so the
54 # directives contained in it are actually available _before_ they are used.
55 # Statically compiled modules (those listed by 'httpd -l') do not need
56 # to be loaded here
```

Рис. 3.13: Просмотр log-файлов

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 (рис. 3.14).

```

[root@aaastrakhanceva log]# cat /var/log/httpd/access_log
127.0.0.1 - - [27/Apr/2024:12:53:27 +0300] "GET /test.html HTTP/1.1" 200 33 "-" "Mozilla/5.0 (X11; Linux
x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - - [27/Apr/2024:12:53:27 +0300] "GET /favicon.ico HTTP/1.1" 404 196 "http://127.0.0.1/test.htm
l" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - - [27/Apr/2024:12:56:10 +0300] "GET /test.html HTTP/1.1" 200 33 "-" "Mozilla/5.0 (X11; Linux
x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - - [27/Apr/2024:12:58:03 +0300] "GET /test.html HTTP/1.1" 403 199 "-" "Mozilla/5.0 (X11; Linux
x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
127.0.0.1 - - [27/Apr/2024:13:04:46 +0300] "GET /test.html HTTP/1.1" 403 199 "-" "Mozilla/5.0 (X11; Linux
x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
[root@aaastrakhanceva log]# /var/log/httpd/error_log
bash: /var/log/httpd/error_log: Permission denied
[root@aaastrakhanceva log]# /var/log/audit/audit.log
bash: /var/log/audit/audit.log: Permission denied
[root@aaastrakhanceva log]# exit
exit
[user@aaastrakhanceva ~]$ cat /var/log/audit/audit.log
cat: /var/log/audit/audit.log: Permission denied

```

Рис. 3.14: Просмотр файлов /var/log/messages, /var/log/http/error_log

Выполните команду `semanage port -a -t http_port_t -p tcp 81`. После этого проверьте список портов командой `semanage port -l | grep http_port_t`. Убедитесь, что порт 81 появился в списке (рис. 3.15).

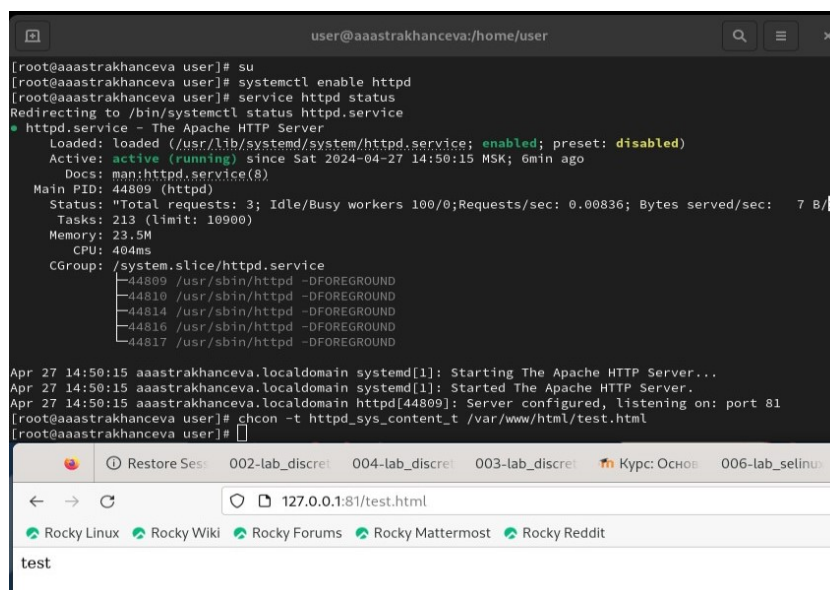
```

[root@aaastrakhanceva user]# 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,pe
rmissive,dontaudit}
               ...
semanage: error: unrecognized arguments: -p 81
[root@aaastrakhanceva user]# semanage port -a -t http_port_tp tcp 81
usage: semanage [-h]
               {import,export,login,user,port,ibpkey,ibendport,interface,module,node,fcontext,boolean,pe
rmissive,dontaudit}
               ...
semanage: error: unrecognized arguments: 81
[root@aaastrakhanceva user]# semanage port -a -t http_port_tp tcp 81
usage: semanage [-h]
               {import,export,login,user,port,ibpkey,ibendport,interface,module,node,fcontext,boolean,pe
rmissive,dontaudit}
               ...
semanage: error: unrecognized arguments: 81
[root@aaastrakhanceva user]# 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@aaastrakhanceva user]#

```

Рис. 3.15: Добавление порта 81

Попробуйте запустить веб-сервер Apache ещё раз. Поняли ли вы, почему он сейчас запустился, а в предыдущем случае не смог? Верните 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. 3.16).



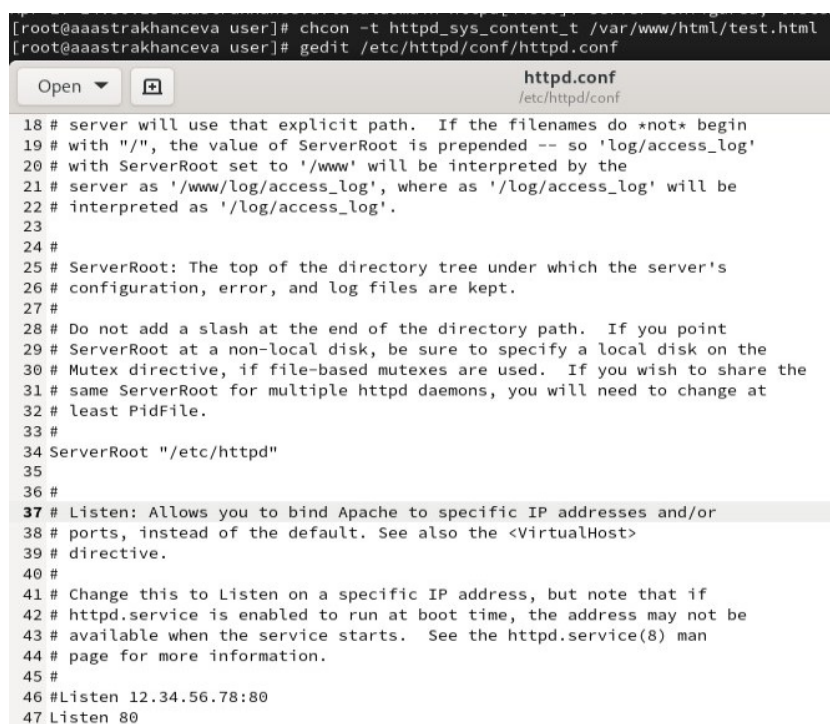
```
[root@aaastrakhanceva user]# su
[root@aaastrakhanceva user]# systemctl enable httpd
[root@aaastrakhanceva user]# systemctl status httpd.service
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-04-27 14:50:15 MSK; 6min ago
     Docs: man:httpd.service(8)
   Main PID: 44809 (httpd)
    Status: "Total requests: 3; Idle/Busy workers 100/0; Requests/sec: 0.00836; Bytes served/sec: 7 B/s"
     Tasks: 213 (limit: 10900)
    Memory: 23.5M
       CPU: 404ms
   CGroup: /system.slice/httpd.service
           └─44809 /usr/sbin/httpd -DFOREGROUND
             └─44810 /usr/sbin/httpd -DFOREGROUND
               └─44814 /usr/sbin/httpd -DFOREGROUND
                 └─44816 /usr/sbin/httpd -DFOREGROUND
                   └─44817 /usr/sbin/httpd -DFOREGROUND

Apr 27 14:50:15 aaastrakhanceva.localdomain systemd[1]: Starting The Apache HTTP Server...
Apr 27 14:50:15 aaastrakhanceva.localdomain systemd[1]: Started The Apache HTTP Server.
Apr 27 14:50:15 aaastrakhanceva.localdomain httpd[44809]: Server configured, listening on: port 81
[root@aaastrakhanceva user]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@aaastrakhanceva user]#
```

test

Рис. 3.16: Запуск веб-сервера Apache

Исправьте обратно конфигурационный файл apache, вернув Listen 80 (рис. 3.17).



```
[root@aaastrakhanceva user]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@aaastrakhanceva user]# gedit /etc/httpd/conf/httpd.conf
```

httpd.conf
/etc/httpd/conf

```
18 # server will use that explicit path. If the filenames do *not* begin
19 # with "/", the value of ServerRoot is prepended -- so 'log/access_log'
20 # with ServerRoot set to '/www' will be interpreted by the
21 # server as '/www/log/access_log', whereas '/log/access_log' will be
22 # interpreted as '/log/access_log'.
23
24 #
25 # ServerRoot: The top of the directory tree under which the server's
26 # configuration, error, and log files are kept.
27 #
28 # Do not add a slash at the end of the directory path. If you point
29 # ServerRoot at a non-local disk, be sure to specify a local disk on the
30 # Mutex directive, if file-based mutexes are used. If you wish to share the
31 # same ServerRoot for multiple httpd daemons, you will need to change at
32 # least PidFile.
33 #
34 ServerRoot "/etc/httpd"
35
36 #
37 # Listen: Allows you to bind Apache to specific IP addresses and/or
38 # ports, instead of the default. See also the <VirtualHost>
39 # directive.
40 #
41 # Change this to Listen on a specific IP address, but note that if
42 # httpd.service is enabled to run at boot time, the address may not be
43 # available when the service starts. See the httpd.service(8) man
44 # page for more information.
45 #
46 #Listen 12.34.56.78:80
47 Listen 80
```

Рис. 3.17: Исправление конфигурационного файла apache

Удалите привязку `http_port_t` к 81 порту: `semanage port -d -t http_port_t -p tcp 81`, проверьте, что порт 81 удалён. Удалите файл `/var/www/html/test.html`: `rm /var/www/html/test.html` (рис. 3.18)

```
[root@aaastrakhanceva user]# semanage port -d -t http_port_t -p tcp 81
ValueError: Port tcp/81 is defined in policy, cannot be deleted
[root@aaastrakhanceva user]# rm /var/www/html/test.html
rm: remove regular file '/var/www/html/test.html'? y
[root@aaastrakhanceva user]# 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@aaastrakhanceva user]#
```

Рис. 3.18: Удаление привязки к порту 81 и файла `/var/www/html/test.html`:

4 Выводы

В ходе выполнения ЛРН^{№6} я развила навыки администрирования ОС Linux. Получила первое практическое знакомство с технологией SELinux1. Проверила работу SELinx на практике совместно с веб-сервером Apache.

5 Список литературы. Библиография

[1] SELinux - система принудительного контроля доступа: <https://redos.red-soft.ru/base/manual/safe-redos/selinux/>

[2] Права в Linux (chown, chmod, SUID, GUID, sticky bit, ACL, umask): <https://habr.com/ru/articles/469667/>