

Отчет по лабораторной работе №6

Основы информационной безопасности

Петрова Алевтина Александровна НКА-бд-04-23

Содержание

1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux¹. Проверить работу SELinux на практике совместно с веб-сервером Apache.
[course?]

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. Более подробно см. в [f?].

1. **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:

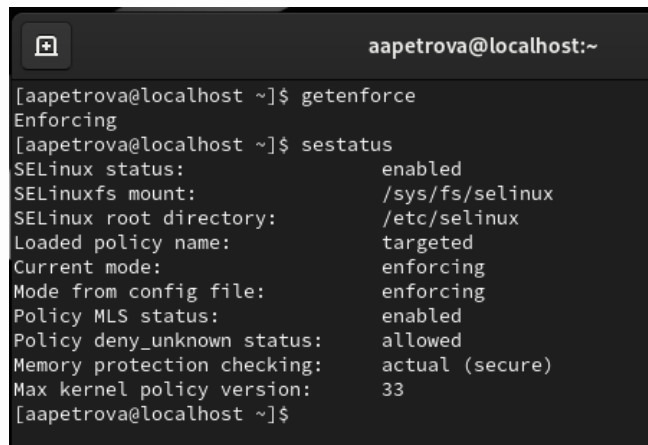
- чтобы открывать динамические PHP-страницы,
- для распределения поступающей на сервер нагрузки,
- для обеспечения отказоустойчивости сервера,
- чтобы потренироваться в настройке сервера и запуске PHP-скриптов.

Apache является кроссплатформенным ПО и поддерживает такие операционные системы, как Linux, BSD, MacOS, Microsoft, BeOS и другие.

Более подробно см. в [s?].

3 Выполнение лабораторной работы

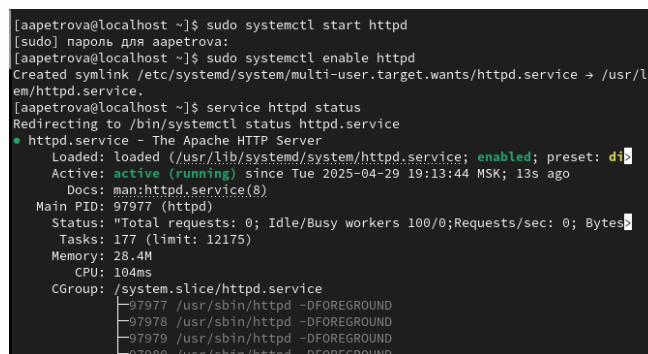
Вошла в систему под своей учетной записью. Убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus` (рис. [fig:001?]).



```
aapetrova@localhost:~  
[aapetrova@localhost ~]$ getenforce  
Enforcing  
[aapetrova@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  
[aapetrova@localhost ~]$
```

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. [fig:002?]).



```
[aapetrova@localhost ~]$ sudo systemctl start httpd  
[sudo] пароль для aapetrova:  
[aapetrova@localhost ~]$ sudo systemctl enable httpd  
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.  
[aapetrova@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; preset: disabled)  
   Active: active (running) since Tue 2025-04-29 19:13:44 MSK; 13s ago  
     Docs: man:httpd.service(8)  
  Main PID: 97977 (httpd)  
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served: 0"  
    Tasks: 177 (limit: 12175)  
  Memory: 28.4M  
    CPU: 104ms  
   CGroup: /system.slice/httpd.service  
            └─97977 /usr/sbin/httpd -DFOREGROUND  
            └─97978 /usr/sbin/httpd -DFOREGROUND  
            └─97979 /usr/sbin/httpd -DFOREGROUND  
            └─97980 /usr/sbin/httpd -DFOREGROUND
```

Проверка работы Apache

С помощью команды `ps auxZ | grep httpd` нашла веб-сервер Apache в списке процессов. Его контекст безопасности - `httpd_t` (рис. [fig:003?]).

```
[aapetrova@localhost ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 97977 0.0 0.5 21024 11272 ?
0 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 97978 0.0 0.3 22900 7136 ?
0 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 97979 0.0 0.6 2358644 13192 ?
0 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 97980 0.0 0.6 2161972 12952 ?
0 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 97981 0.0 0.7 2161972 14868 ?
0 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 aapetro+ 98218 0.0 0.
+ 19:17 0:00 grep --color=auto httpd
[aapetrova@localhost ~]$
```

Контекст безопасности Apache

Просмотрела текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -bigrep httpd` (рис. [fig:004?]).

```
[aapetrova@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
```

Состояние переключателей SELinux

Просмотрела статистику по политике с помощью команды `seinfo`. Множество пользователей - 8, ролей - 39, типов - 5135. (рис. [fig:005?]).

```
[aapetrova@localhost ~]$ 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: 5169 Attributes: 259
Users: 8 Roles: 15
Booleans: 358 Cond. Expr.: 390
Allow: 65633 Neverallow: 0
Auditallow: 176 Dontaudit: 8703
Type_trans: 271851 Type_change: 94
Type_member: 37 Range_trans: 5931
Role allow: 40 Role_trans: 417
Constraints: 70 Validatetrans: 0
MLS Constrains: 72 MLS Val. Tran: 0
```

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. [fig:006?]).

```
[aapetrova@localhost ~]$ ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 янв 22 03:25 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 янв 22 03:25 html
[aapetrova@localhost ~]$
```

Типы поддиректорий

В директории /var/www/html нет файлов. (рис. [fig:007?]).

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

Типы файлов

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

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

(рис. [fig:008?]).

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

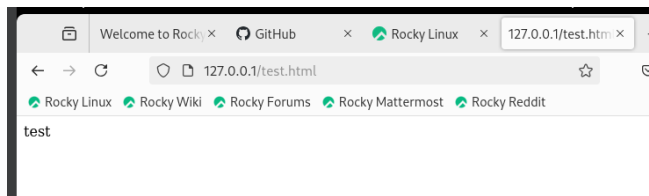
Создание файла

Проверяю контекст созданного файла. По умолчанию это `httpd_sys_content_t` (рис. [fig:009?]).

```
[aapetrova@localhost ~]$ ls -lZ /var/www/html/
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys
[aapetrova@localhost ~]$
```

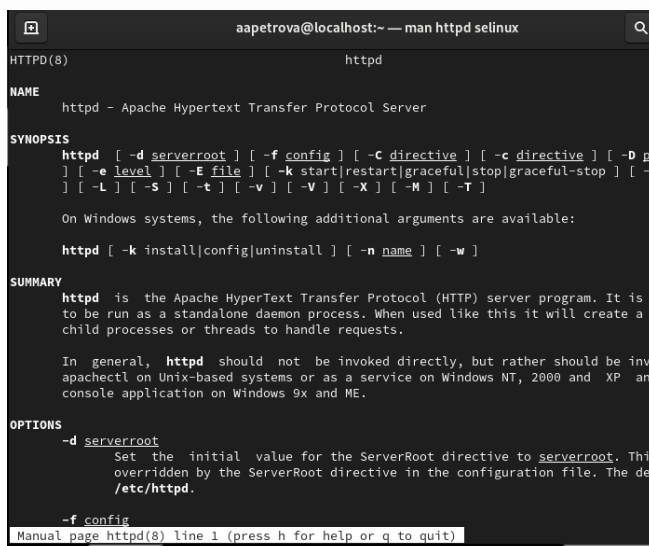
Контекст файла

Обращаюсь к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Файл был успешно отображён (рис. [fig:010?]).



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. [fig:011?]).



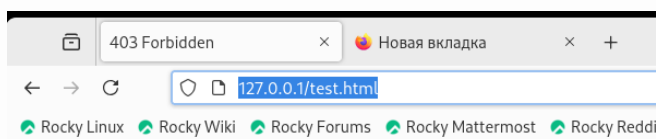
Изучение справки по команде

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. [fig:012?]).

```
[aapetrova@localhost ~]$ sudo chcon -t samba_share_t /var/www/html/test.html
[sudo] пароль для aapetrova:
[aapetrova@localhost ~]$ ls -lZ /var/www/html/test.html
-rw-r--r--. 1 root root unconfined_u:object_r:samba_share_t:s0 33 anp 29 19:24 /var/www/html/test.html
[aapetrova@localhost ~]$
```

Изменение контекста

При попытке отображения файла в браузере получаем сообщение об ошибке (рис. [fig:013?]).



Forbidden

You don't have permission to access this resource.

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 `log`-файл: `tail /var/log/messages`. Если в системе окажутся запущенными процессы `setroubleshootd` и `auditd`, то вы также сможете увидеть ошибки, аналогичные указанным выше, в файле `/var/log/audit/audit.log`. (рис. [fig:014?]).

```
[aapetrova@localhost ~]$ ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 33 anp 29 19:24 /var/www/html/test.html
[aapetrova@localhost ~]$ tail /var/log/audit/audit.log
tail: невозможно открыть '/var/log/audit/audit.log' для чтения: Отказано в доступе
[aapetrova@localhost ~]$ sudo tail /var/log/audit/audit.log
type=SERVICE_STOP msg=audit(1745944482.729:785): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=
system_u:system_r:init_t:s0 msg='unit=dbus-1:1-org.fedoraproject.setroubleshootPrivileged@0 comm
="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'UID="root" AUI
D="unset"
type=SERVICE_STOP msg=audit(1745944482.787:786): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=
system_u:system_r:init_t:s0 msg='unit=setroubleshootd comm="systemd" exe="/usr/lib/systemd/system
d" hostname=? addr=? terminal=? res=success'UID="root" AUID="unset"
type=SERVICE_STOP msg=audit(1745944492.037:787): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=
system_u:system_r:init_t:s0 msg='unit=systemd-timedated comm="systemd" exe="/usr/lib/systemd/syst
emd" hostname=? addr=? terminal=? res=success'UID="root" AUID="unset"
type=BPF msg=audit(1745944492.046:788): prog-id=169 op=UNLOAD
type=BPF msg=audit(1745944492.046:789): prog-id=168 op=UNLOAD
type=USER_ACCT msg=audit(1745944611.050:791): pid=99703 uid=1000 auid=1000 ses=3 subj=unconfined
```

Попытка прочесть `log`-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. [fig:015?]).

```
[aapetrova@localhost ~]$ sudo nano /etc/httpd/conf/httpd.conf
[aapetrova@localhost ~]$
```

Изменение файла

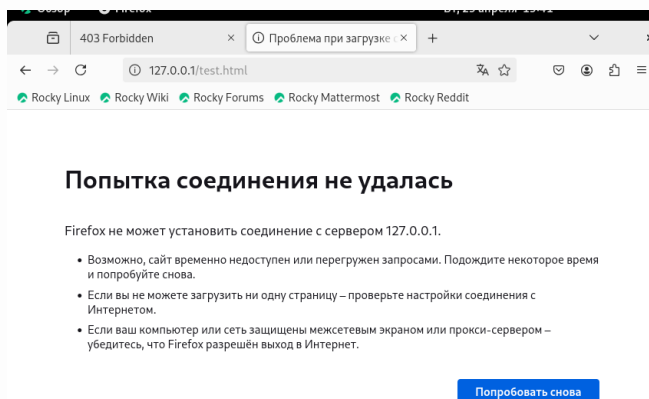
Нахожу строчку Listen 80 и заменяю её на Listen 81. (рис. [fig:016?]).

```
GNU nano 5.6.1 /etc/httpd/conf/httpd.conf
#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
# shared library (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
# this.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
# Include conf.modules.d/*.conf
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
```

Изменение порта

Выполняю перезапуск веб-сервера Apache. Произошёл сбой, потому что порт 80 для локальной сети, а 81 нет (рис. [fig:017?]).



Попытка прослушивания другого порта

Проанализируйте лог-файлы: tail -n1 /var/log/messages (рис. [fig:018?]).

```
[aapetrova@localhost ~]$ sudo tail -n1 /var/log/messages
Apr 29 19:41:40 localhost httpd[99846]: Server configured, listening on: port 81
[aapetrova@localhost ~]$
```


Проверка лог-файлов

Просмотрите файлы `/var/log/httpd/error_log`, `/var/log/httpd/access_log` и `/var/log/audit/audit.log` и выясните, в каких файлах появились записи. Запись появилась в файлу `error_log` (рис. [fig:019?]).

```
[aaetrova@localhost ~]$ sudo cat /var/log/httpd/error_log
[Thu Apr 29 19:13:44.206075 2025] [core:notice] [pid 97977:tid 97977] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Tue Apr 29 19:13:44.207890 2025] [suexec:notice] [pid 97977:tid 97977] 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 suppress this message
[Tue Apr 29 19:13:44.222436 2025] [lbmethod_heartbeat:notice] [pid 97977:tid 97977] AH02282: No lotmem from mod_heartbeat
[Tue Apr 29 19:13:44.231733 2025] [mpm_event:notice] [pid 97977:tid 97977] AH00489: Apache/2.4.18 (Rocky Linux) configured -- resuming normal operations
[Tue Apr 29 19:13:44.231789 2025] [core:notice] [pid 97977:tid 97977] AH00994: Command line: '/usr/sbin/httpd -D FOREGROUND'
[Tue Apr 29 19:34:24.457297 2025] [core:error] [pid 97979:tid 98147] (13)Permission denied: [client 127.0.0.1:59844] AH00035: access to /test.html denied (filesystem path '/var/www/html/test.html') because search permissions are missing on a component of the path
[Tue Apr 29 19:34:31.617661 2025] [core:error] [pid 97981:tid 98089] (13)Permission denied: [client 127.0.0.1:37060] AH00035: access to /test.html denied (filesystem path '/var/www/html/test.html') because search permissions are missing on a component of the path
[Tue Apr 29 19:39:44.440598 2025] [core:error] [pid 97980:tid 98109] (13)Permission denied: [client 127.0.0.1:59844] AH00035: access to /test.html denied (filesystem path '/var/www/html/test.html') because search permissions are missing on a component of the path
```

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. [fig:020?]).

```
[aaetrova@localhost ~]$ sudo semanage port -a -t http_port_t -p tcp 81
Port tcp/81 already defined, modifying instead
[aaetrova@localhost ~]$ semanage port -l | grep http_port_t
ValueError: Политика SELinux не задана, или нет доступа к хранилищу.
[aaetrova@localhost ~]$ sudo semanage port -l | grep http_port_t
http_port_t      tcp      81, 80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t tcp      5988
```

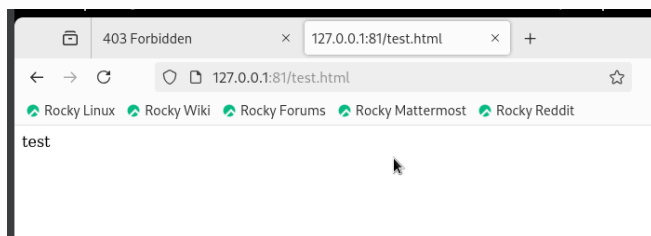
Проверка портов

Перезапускаю сервер Apache (рис. [fig:021?]).

```
pegasus_http_port_t tcp 5988
[aaetrova@localhost ~]$ sudo systemctl restart httpd
[aaetrova@localhost ~]$ sudo chcon -t httpd_sys_content_t /var/www/html/test.html
[aaetrova@localhost ~]$ sudo systemctl restart httpd
[aaetrova@localhost ~]$
```

Перезапуск сервера

Теперь он работает, ведь мы внесли порт 81 в список портов `httpd_port_t` (рис. [fig:022?]).



Проверка сервера

Возвращаю в файле `/etc/httpd/httpd.conf` порт 80, вместо 81. Проверяю, что порт 81 удален, это правда. (рис. [fig:023?]).

```
[aapetrova@localhost ~]$ sudo semanage port -d -t http_port_t -p tcp 81
[aapetrova@localhost ~]$
```

Проверка порта 81

Далее удаляю файл test.html, проверяю, что он удален(рис. [fig:024?]).

```
[aapetrova@localhost ~]$ rm /var/www/html/test.html
rm: удалить защищенный от записи обычный файл '/var/www/html/test.html'? y
rm: невозможно удалить '/var/www/html/test.html': Отказано в доступе
[aapetrova@localhost ~]$ sudo rm /var/www/html/test.html
[aapetrova@localhost ~]$ ls -lZ /var/www/html
итого 0
[aapetrova@localhost ~]$
```

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