

# **Отчёт по лабораторной работе №6**

## **Информационная безопасность**

**Мандатное разграничение прав в Linux**

Выполнила: Павлова Полина Алексеевна,  
НПИБд-02-21, 1032212967

# Содержание

1	Цель работы	4
2	Теоретическое введение	5
3	Выполнение лабораторной работы	7
4	Вывод	15
5	Список литературы. Библиография	16

## Список иллюстраций

3.1	(рис. 1. Проверка режима enforcing политики targeted)	7
3.2	(рис. 2. Проверка работы веб-сервера)	8
3.3	(рис. 3. Контекст безопасности веб-сервера Apache)	8
3.4	(рис. 4. Текущее состояние переключателей SELinux)	9
3.5	(рис. 5. Статистика по политике)	10
3.6	(рис. 6. Просмотр файлов и поддиректорий в директории /var/www)	10
3.7	(рис. 7. Создание файла /var/www/html/test.html)	11
3.8	(рис. 8. Обращение к файлу через веб-сервер)	11
3.9	(рис. 9. Изменение контекста)	11
3.10	(рис. 10. Обращение к файлу через веб-сервер)	12
3.11	(рис. 11. Просмотр log-файла)	12
3.12	(рис. 12. Установка веб-сервера Apache на прослушивание TCP-порта 81)	12
3.13	(рис. 13. Перезапуск веб-сервера и анализ лог-файлов)	13
3.14	(рис. 14. Содержание файла var/log/audit/audit.log)	13
3.15	(рис. 15. Обращение к файлу через веб-сервер)	13
3.16	(рис. 16. Возвращение Listen 80 и попытка удалить порт 81)	14

# 1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux<sup>1</sup>. Проверить работу SELinux 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. Более подробно см. в [1].

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.

Более подробно см. в [2].

### 3 Выполнение лабораторной работы

Вошли в систему под своей учетной записью и убедились, что SELinux работает в режиме enforcing политики targeted с помощью команд “getenforce” и “sestatus”

```
[papavloval2@papavloval2 ~]$ getenforce
sestatus
Enforcing
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
[papavloval2@papavloval2 ~]$
```

Рис. 3.1: (рис. 1. Проверка режима enforcing политики targeted)

Обратились с помощью браузера к веб-серверу, запущенному на компьютере, и убедились, что последний работает с помощью команды “service httpd status”

```
[papavloval2@papavloval2 ~]$ 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 Wed 2024-09-11 11:00:16 MSK; 3min 18s ago
     Docs: man:httpd.service(8)
  Main PID: 5336 (httpd)
    Status: "Total requests: 1; Idle/Busy workers 100/0; Requests/sec: 0.00529; Bytes served/sec:
           Tasks: 177 (limit: 10961)
           Memory: 30.5M
           CPU: 286ms
    CGroup: /system.slice/httpd.service
            └─5336 /usr/sbin/httpd -DFOREGROUND
              └─5338 /usr/sbin/httpd -DFOREGROUND
                └─5339 /usr/sbin/httpd -DFOREGROUND
                  └─5340 /usr/sbin/httpd -DFOREGROUND
                    └─5341 /usr/sbin/httpd -DFOREGROUND

сен 11 11:00:15 papavloval2 systemd[1]: Starting The Apache HTTP Server...
сен 11 11:00:16 papavloval2 httpd[5336]: Server configured, listening on: port 80
сен 11 11:00:16 papavloval2 systemd[1]: Started The Apache HTTP Server.
```

Рис. 3.2: (рис. 2. Проверка работы веб-сервера)

С помощью команды “ps auxZ | grep httpd” определили контекст безопасности веб-сервера Apache - httpd\_t

```
[papavloval2@papavloval2 ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root      5336  0.0  0.6 21104 11368 ?        Ss   11:00   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  5338  0.0  0.3 22980  7136 ?        S    11:00   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  5339  0.0  0.6 1441152 11752 ?       Sl   11:00   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  5340  0.0  0.8 1572288 15240 ?       Sl   11:00   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  5341  0.0  0.8 1441152 15002 ?       Sl   11:00   0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-c0:c1023 papavlo+ 5607 0.0  0.1 221820 2432 pts/0 $+ 11:03  0:00 grep --color=auto httpd
```

Рис. 3.3: (рис. 3. Контекст безопасности веб-сервера Apache)

Посмотрели текущее состояние переключателей SELinux для Apache с помощью команды “sestatus -bigrep httpd”, многие из переключателей находятся в положении “off”



```

[papavlova12@papavlova12 ~]$ 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_sssd off
httpd_dontaudit_search_dirs off
httpd_enable_cgi on
httpd_enable_ftp_server off
httpd_enable_homedirs off
httpd_execmem off
httpd_graceful_shutdown off
httpd_manage_ipa off
httpd_mod_auth_ntlm_winbind off
httpd_mod_auth_pam off
httpd_read_user_content off
httpd_run_ipa off
httpd_run_preupgrade off
httpd_run_stickshift off

```

Рис. 3.4: (рис. 4. Текущее состояние переключателей SELinux)

Посмотрели статистику по политике с помощью команды “seinfo”. Множество пользователей - 8, ролей - 14, типов 5100

```
[papavlova12@papavlova12 ~]$ 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:            65631    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 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: (рис. 5. Статистика по политике)

С помощью команды “ls -lZ /var/www” посмотрели файлы и поддиректории, находящиеся в директории /var/www. Используя команду “ls -lZ /var/www/html”, определили, что в данной директории файлов нет. Только владелец/суперпользователь может создавать файлы в директории /var/www/html

```
[papavlova12@papavlova12 ~]$ ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0  6 авг 12 16:20 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0      23 сен 11 10:39 html
[papavlova12@papavlova12 ~]$ ls -lZ /var/www/html
итого 4
```

Рис. 3.6: (рис. 6. Просмотр файлов и поддиректорий в директории /var/www)

От имени суперпользователя создали html-файл /var/www/html/test.html. Контекст созданного файла - httpd\_sys\_content\_t

```
[papavlova12@papavlova12 ~]$ vi /var/www/html/test.html
[papavlova12@papavlova12 ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
```

Рис. 3.7: (рис. 7. Создание файла /var/www/html/test.html)

Обратились к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html”.  
Файл был успешно отображен

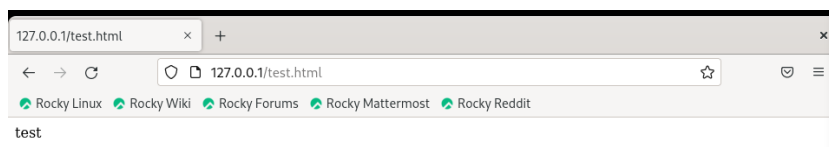


Рис. 3.8: (рис. 8. Обращение к файлу через веб-сервер)

Изучив справку `man httpd_selinux`, выяснили, что для `httpd` определены следующие контексты файлов:

`httpd_sys_content_t`, `httpd_sys_script_exec_t`,  
`httpd_sys_script_ro_t`, `httpd_sys_script_rw_t`,  
`httpd_sys_script_ra_t`, `httpd_unconfined_script_exec_t`.

Контекст моего файла - `httpd_sys_content_t` (в таком случае содержимое должно быть доступно для всех скриптов `httpd` и для самого демона). Изменили контекст файла на `samba_share_t` командой “`sudo chcon -t samba_share_t /var/www/html/test.html`” и проверили, что контекст поменялся

```
[papavlova12@papavlova12 ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[papavlova12@papavlova12 ~]$ sudo chcon -t samba_share_t /var/www/html/test.html
ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[papavlova12@papavlova12 ~]$ ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 33 сен 11 10:39 /var/www/html/test.html
```

Рис. 3.9: (рис. 9. Изменение контекста)

Попробовали еще раз получить доступ к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html” и получили сообщение об ошибке (т.к. к установленному ранее контексту процесс `httpd` не имеет доступа)

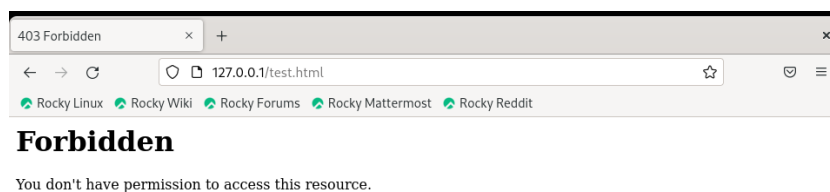


Рис. 3.10: (рис. 10. Обращение к файлу через веб-сервер)

Командой “`ls -l /var/www/html/test.html`” убедились, что читать данный файл может любой пользователь. Просмотрели системный лог-файл веб-сервера Apache командой “`sudo tail /var/log/messages`”, отображающий ошибки

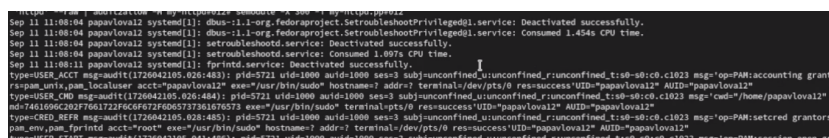


Рис. 3.11: (рис. 11. Просмотр log-файла)

В файле `/etc/httpd/conf/httpd.conf` заменили строчку “Listen 80” на “Listen 81”, чтобы установить веб-сервер Apache на прослушивание TCP-порта 81

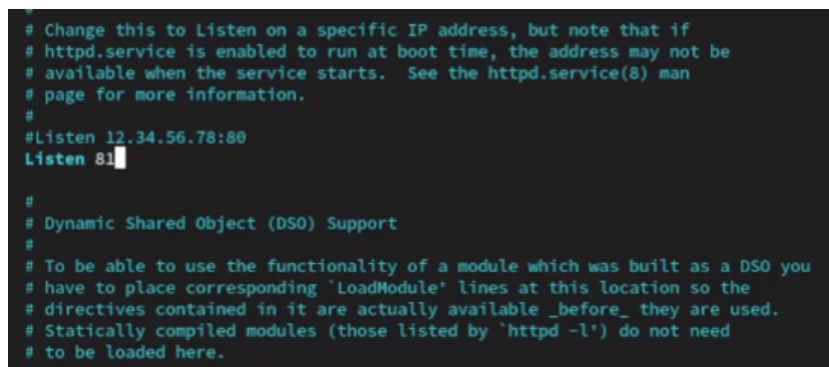


Рис. 3.12: (рис. 12. Установка веб-сервера Apache на прослушивание TCP-порта 81)

Перезапускаем веб-сервер Apache и анализируем лог-файлы командой “`tail -nl /var/log/messages`”

```

[papavlova12@papavlova12 ~]$ sudo systemctl restart httpd
[papavlova12@papavlova12 ~]$ tail -n1 /var/log/messages
tail: /var/log/httpd/error_log
tail: /var/log/httpd/access_log
tail: некорректное количество строк: -1
tail: невозможно открыть '/var/log/httpd/error_log' для чтения: Отказано в доступе
tail: невозможно открыть '/var/log/httpd/access_log' для чтения: Отказано в доступе
[papavlova12@papavlova12 ~]$ sudo -i
[root@papavlova12 ~]# tail -n1 /var/log/messages
tail: /var/log/httpd/error_log
tail: /var/log/httpd/access_log
tail: некорректное количество строк: -1
[Wed Sep 11 11:08:36.037837 2024] [core:notice] [pid 5336:t1d 5336] AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'
[Wed Sep 11 11:08:44.818341 2024] [autoindex:error] [pid 5339:t1d 5490] [client ::1:40650] AH01278: Cannot serve directory /var/www/html/: No matching DirectoryIndex (index.html) found, and server-generated directory index forbidden by Options directive
[Wed Sep 11 11:09:36.182634 2024] [core:error] [pid 5340:t1d 5452] (13)Permission denied: [client 127.0.0.1:41422] AH00035: access to /test.html denied (filesystem path '/var/www/html/test.html') because search permissions are missing on a component of the path
[Wed Sep 11 11:07:51.540908 2024] [core:error] [pid 5339:t1d 5487] (13)Permission denied: [client 127.0.0.1:49038] AH00035: access to /test.html denied (filesystem path '/var/www/html/test.html') because search permissions are missing on a component of the path
[Wed Sep 11 11:09:05.786365 2024] [mpm_event:notice] [pid 5336:t1d 5336] AH00492: caught SIGWINCH, shutting down gracefully
[Wed Sep 11 11:09:07.007080 2024] [core:notice] [pid 5742:t1d 5742] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Wed Sep 11 11:09:07.008093 2024] [suexec:notice] [pid 5742:t1d 5742] AH02231: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Wed Sep 11 11:09:07.020880 2024] [lbmethod_heartbeat:notice] [pid 5742:t1d 5742] AH02282: No slotmem from mod_heartbeat
[Wed Sep 11 11:09:07.041841 2024] [mpm_event:notice] [pid 5742:t1d 5742] AH00489: Apache/2.4.62 (CentOS Stream) configured -- resuming normal operations
[Wed Sep 11 11:09:07.041894 2024] [core:notice] [pid 5742:t1d 5742] AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'

```

Рис. 3.13: (рис. 13. Перезапуск веб-сервера и анализ лог-файлов)

Просмотрели файлы “var/log/http/error\_log”, “/var/log/http/access\_log” и “/var/log/audit/audit.log” и выяснили, что запись появилась в последнем файле

```

[Wed Sep 11 11:09:36.182634 2024] [core:error] [pid 5340:t1d 5452] (13)Permission denied: [client 127.0.0.1:41422] AH00035: access to /test.html denied (filesystem path '/var/www/html/test.html') because search permissions are missing on a component of the path
[Wed Sep 11 11:07:51.540908 2024] [core:error] [pid 5339:t1d 5487] (13)Permission denied: [client 127.0.0.1:49038] AH00035: access to /test.html denied (filesystem path '/var/www/html/test.html') because search permissions are missing on a component of the path
[Wed Sep 11 11:09:05.786365 2024] [mpm_event:notice] [pid 5336:t1d 5336] AH00492: caught SIGWINCH, shutting down gracefully
[Wed Sep 11 11:09:07.007080 2024] [core:notice] [pid 5742:t1d 5742] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Wed Sep 11 11:09:07.008093 2024] [suexec:notice] [pid 5742:t1d 5742] AH02231: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Wed Sep 11 11:09:07.020880 2024] [lbmethod_heartbeat:notice] [pid 5742:t1d 5742] AH02282: No slotmem from mod_heartbeat
[Wed Sep 11 11:09:07.041841 2024] [mpm_event:notice] [pid 5742:t1d 5742] AH00489: Apache/2.4.62 (CentOS Stream) configured -- resuming normal operations
[Wed Sep 11 11:09:07.041894 2024] [core:notice] [pid 5742:t1d 5742] AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'

```

Рис. 3.14: (рис. 14. Содержание файла var/log/audit/audit.log)

Вернули контекст “httpd\_sys\_content\_t” файлу “/var/www/html/test.html” командой “chcon -t httpd\_sys\_content\_t /var/www/html/test.html” и после этого попробовали получить доступ к файлу через веб-сервер, введя адрес “http://127.0.0.1:81/test.html”, в результате чего увидели содержимое файла - слово “test”

```

[root@papavlova12 ~]# sudo semanage port -a -t http_port_t -p tcp 81
semanage port -l | grep http_port_t
Port tcp/81 already defined, modifying instead
http_port_t          tcp      81, 80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t  tcp      5988

```

Рис. 3.15: (рис. 15. Обращение к файлу через веб-сервер)

Исправили обратно конфигурационный файл apache, вернув “Listen 80”. Попытались удалить привязку http\_port к 81 порту командой “semanage port -d -t http\_port\_t -p tcp 81”, но этот порт определен на уровне политики, поэтому его нельзя удалить. Удалили файл “/var/www/html/test.html” командой “rm /var/www/html/test.html”

```
[root@papavlova12 ~]# sudo systemctl restart httpd
[root@papavlova12 ~]# sudo chcon -t httpd_sys_content_t /var/www/html/test.html
[root@papavlova12 ~]# sudo semanage port -d -t http_port_t -p tcp 81
[root@papavlova12 ~]# sudo vi /etc/httpd/conf/httpd.conf
[root@papavlova12 ~]# sudo rm /var/www/html/test.html
```

Рис. 3.16: (рис. 16. Возвращение Listen 80 и попытка удалить порт 81)

## 4 Вывод

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.

## 5 Список литературы. Библиография

[0] Методические материалы курса

[1] SELinux: <https://habr.com/ru/companies/kingservers/articles/209644/>

[2] Apache: <https://2domains.ru/support/vps-i-servery/shto-takoye-apache>