

Отчёт по лабораторной работе №6

Информационная безопасность

Мандатное разграничение прав в Linux

Выполнила: Данзанова Саяна, НПИбд-01-21, 1032217624

Содержание

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

Список иллюстраций

1	(рис. 1. Проверка режима enforcing политики targeted)	7
2	(рис. 2. Проверка работы веб-сервера)	8
3	(рис. 3. Контекст безопасности веб-сервера Apache)	8
4	(рис. 4. Текущее состояние переключателей SELinux)	9
5	(рис. 5. Статистика по политике)	9
6	(рис. 6. Просмотр файлов и поддиректорий в директории /var/www) . .	10
7	(рис. 7. Создание файла /var/www/html/test.html. Обращение к файлу через веб-сервер)	10
8	(рис. 8. Изменение контекста. Обращение к файлу через веб-сервер) . . .	11
9	(рис. 9. Просмотр log-файла)	12
10	(рис. 10. Установка веб-сервера Apache на прослушивание TCP-порта 81)	12
11	(рис. 11. Перезапуск веб-сервера и анализ лог-файлов)	13
12	(рис. 12. Проверка установки порта 81)	13
13	(рис. 13. Возвращение исходного контекста файлу. Обращение к файлу через веб-сервер)	13
14	(рис. 14. Возвращение Listen 80 и попытка удалить порт 81)	14
15	(рис. 15. Удаление файла test.html)	14

Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux¹. Проверить работу SELinx на практике совместно с веб-сервером Apache.

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].

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”

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

Рис. 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”

```

Выполнено!
[szdzanazanova@szdzanazanova ~]$ sudo systemctl start httpd
[szdzanazanova@szdzanazanova ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[szdzanazanova@szdzanazanova ~]$ 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 Mon 2024-10-07 00:17:41 MSK; 38s ago
     Docs: man:httpd.service(8)
    Main PID: 6829 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0"
      Tasks: 177 (limit: 17412)
    Memory: 26.0M
       CPU: 130ms
    CGroup: /system.slice/httpd.service
            └─6829 /usr/sbin/httpd -DFOREGROUND
              └─6830 /usr/sbin/httpd -DFOREGROUND
                └─6831 /usr/sbin/httpd -DFOREGROUND
                  └─6835 /usr/sbin/httpd -DFOREGROUND
                    └─6837 /usr/sbin/httpd -DFOREGROUND

окт 07 00:17:41 szdzanazanova systemd[1]: Starting The Apache HTTP Server...
окт 07 00:17:41 szdzanazanova httpd[6829]: 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
окт 07 00:17:41 szdzanazanova systemd[1]: Started The Apache HTTP Server.
окт 07 00:17:41 szdzanazanova httpd[6829]: Server configured, listening on: port 80
lines 1-20/20 (END)...skipping...

```

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

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

```

[szdzanazanova@szdzanazanova ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 6829 0.0 0.4 20364 11476 ?
Ss 00:17 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 6830 0.0 0.2 22096 7256 ?
S 00:17 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 6831 0.0 0.6 1112656 17560 ?
Sl 00:17 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 6835 0.0 0.3 981652 11172 ?
Sl 00:17 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 6837 0.0 0.3 981520 11172 ?
Sl 00:17 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 szdzanazanova 7064 0.0 0.0 221
688 2432 pts/0 S+ 00:19 0:00 grep --color=auto httpd

```

Рис. 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”


```
[szdanzanova@szdanzanova ~]$ 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. 4: (рис. 4. Текущее состояние переключателей SELinux)

Посмотрели статистику по политике с помощью команды “seinfo”. Множество пользователей - 8, ролей - 15, типов 5145

```
[szdanzanova@szdanzanova ~]$ seinfo
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:                        5145
Users:                        8
Booleans:                     356
Allow:                        65504
Auditallow:                   176
Type_trans:                   271770
Type_member:                  37
Role allow:                   40
Constraints:                  70
MLS Constrain:               72
Permissives:                  4
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.:                  388
Neverallow:                   0
Dontaudit:                    8682
Type_change:                  94
Range_trans:                  5931
Role_trans:                   417
Validatetrans:                0
MLS Val. Tran:                0
Polcap:                       6
Typebounds:                   0
Neverallowxperm:              0
Dontauditxperm:               0
Ibpkeycon:                    0
Fs_use:                       35
Portcon:                      665
Nodecon:                      0
```

Рис. 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

```
[szdanzanova@szdanzanova ~]$ ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 авг 8 19
:30 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 авг 8 19
:30 html
[szdanzanova@szdanzanova ~]$ ls -lZ /var/www/html
итого 0
```

Рис. 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.

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

```
[root@szdanzanova szdanzanova]# touch /var/www/html/test.html
[root@szdanzanova szdanzanova]# nano /var/www/html/test.html
[root@szdanzanova szdanzanova]# cat /var/www/html/test.html
<html>
<body> test </body>
</html>
```

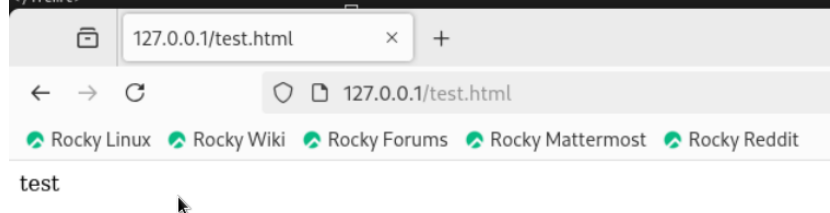


Рис. 7: (рис. 7. Создание файла /var/www/html/test.html. Обращение к файлу через веб-сервер)

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

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)

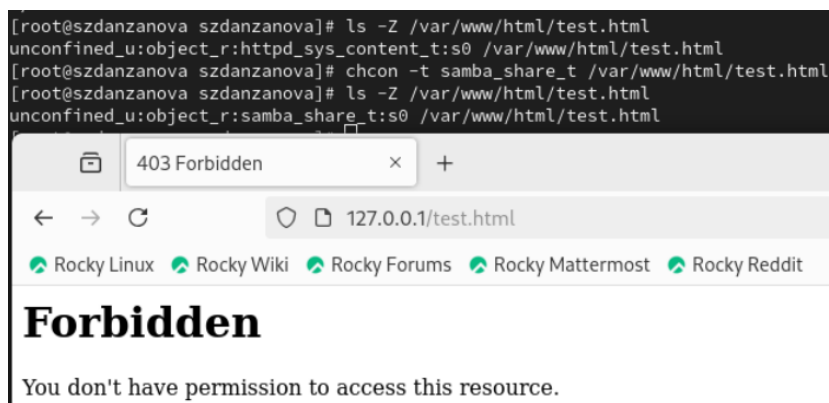


Рис. 8: (рис. 8. Изменение контекста. Обращение к файлу через веб-сервер)

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

```
[root@szdanzanova szdanzanova]# ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 32 окт  7 00:31 /var/www/html/test.html
[root@szdanzanova szdanzanova]# tail /var/log/messages
Oct  7 00:37:48 szdanzanova systemd[1]: Started dbus-1.1-org.fedoraproject.Setr
oubleshootPrivileged@0.service.
Oct  7 00:37:51 szdanzanova setroubleshoot[8262]: SELinux запрещает /usr/sbin/ht
tpd доступ getattr к файл /var/www/html/test.html. Для выполнения всех сообщений
SELinux: sealert -l 42385416-fae5-4b19-b8a5-b8c69ca05aa7
Oct  7 00:37:51 szdanzanova setroubleshoot[8262]: SELinux запрещает /usr/sbin/ht
tpd доступ getattr к файл /var/www/html/test.html.#012#012***** Модуль restorec
on предлагает (точность 92.2) *****#012#012Если вы хотите и
справить метку.$TARGETЗнак _PATH по умолчанию должен быть httpd_sys_content_t#01
2То вы можете запустить restorecon. Возможно, попытка доступа была остановлена и
з-за недостаточных разрешений для доступа к родительскому каталогу, и в этом слу
чае попытайтесь соответствующим образом изменить следующую команду.#012Сделать#0
12# /sbin/restorecon -v /var/www/html/test.html#012#012***** Модуль public_cont
ent предлагает (точность 7.83) *****#012#012Если вы хотите лечи
ть test.html как общедоступный контент#012То необходимо изменить метку test.html
с public_content_t на public_content_rw_t.#012Сделать#012# semanage fcontext -a
-t public_content_t '/var/www/html/test.html'#012# restorecon -v '/var/www/html
/test.html'#012#012***** Модуль catchall предлагает (точность 1.41) *****
*****#012#012Если вы считаете, что httpd должно быть разрешено getat
tr доступ к test.html file по умолчанию.#012То рекомендуется создать отчет об ош
ибке.#012Чтобы разрешить доступ, можно создать локальный модуль политики.#012Сде
лать#012разрешить этот доступ сейчас, выполнив:#012# ausearch -c 'httpd' --raw |
audit2allow -M my-httpd#012# semodule -X 300 -i my-httpd.pp#012
Oct  7 00:37:51 szdanzanova setroubleshoot[8262]: SELinux запрещает /usr/sbin/ht
tpd доступ getattr к файл /var/www/html/test.html. Для выполнения всех сообщений
SELinux: sealert -l 42385416-fae5-4b19-b8a5-b8c69ca05aa7
Oct  7 00:37:52 szdanzanova setroubleshoot[8262]: SELinux запрещает /usr/sbin/ht
tpd доступ getattr к файл /var/www/html/test.html.#012#012***** Модуль restorec
on предлагает (точность 92.2) *****#012#012Если вы хотите и
справить метку.$TARGETЗнак _PATH по умолчанию должен быть httpd_sys_content_t#01
2То вы можете запустить restorecon. Возможно, попытка доступа была остановлена и
```

Рис. 9: (рис. 9. 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

```
GNU nano 5.6.1 /etc/httpd/conf/httpd.conf Изменён
#
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
```

Рис. 10: (рис. 10. Установка веб-сервера Apache на прослушивание TCP-порта 81)

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

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

```
[root@szdanzanova szdanzanova]# systemctl restart httpd
[root@szdanzanova szdanzanova]# tail -n1 /var/log/messages
Oct 7 00:47:43 szdanzanova httpd[8692]: Server configured, listening on: port 81
```

Рис. 11: (рис. 11. Перезапуск веб-сервера и анализ лог-файлов)

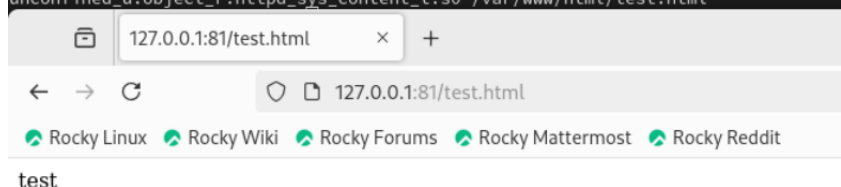
Выполнили команду “semanage port -a -t http_port_t -p tcp 81” и убедились, что порт TCP-81 установлен. Проверили список портов командой “semanage port -l | grep http_port_t”, убедились, что порт 81 есть в списке и запускаем веб-сервер Apache снова

```
[szdanzanova@szdanzanova ~]$ sudo semanage port -a -t http_port_t -p tcp 81
[sudo] пароль для szdanzanova:
usage: semanage [-h]
                {import,export,login,user,port,ibpkey,ibendport,interface,module,
                ,node,fcontext,boolean,permissive,dontaudit}
                ...
semanage: error: unrecognized arguments: -p 81
[szdanzanova@szdanzanova ~]$ 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
[szdanzanova@szdanzanova ~]$ systemctl restart httpd
```

Рис. 12: (рис. 12. Проверка установки порта 81)

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”

```
[szdanzanova@szdanzanova ~]$ sudo chcon -t httpd_sys_content_t /var/www/html/test.html
[szdanzanova@szdanzanova ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
```



The screenshot shows a web browser window with the address bar set to "127.0.0.1:81/test.html". The browser displays the word "test" in a simple font. The browser's address bar and tabs are visible, showing the URL and the page title.

Рис. 13: (рис. 13. Возвращение исходного контекста файлу. 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

```
#
#Listen 12.34.56.78:80
Listen 80
```

Рис. 14: (рис. 14. Возвращение Listen 80 и попытка удалить порт 81)

Удалили файл “`/var/www/html/test.html`” командой “`rm /var/www/html/test.html`”

```
[root@szdanzanova szdanzanova]# sudo rm /var/www/html/test.html
[root@szdanzanova szdanzanova]# ls /var/www/html/test.html
ls: невозможно получить доступ к '/var/www/html/test.html': Нет такого файла или каталога
```

Рис. 15: (рис. 15. Удаление файла test.html)

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.

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>