

Отчёт по лабораторной работе №6

Мандатное разграничение прав в Linux

Тимур Дмитриевич Калинин

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	16
4	Библиография	17

List of Figures

2.1	Статус selinux	6
2.2	Статус веб-сервера	6
2.3	Запуск веб-сервера	7
2.4	Контекст службы Apache	7
2.5	Состояние переключателей SELinux для Apache	7
2.6	Статистика по политике	8
2.7	Тип файлов и директорий в /var/www	8
2.8	Тип файлов в /var/www/html	8
2.9	Содержание test.html	9
2.10	Контекст test.html	9
2.11	Обращение к файлу через веб-сервер	10
2.12	Обращение к файлу через веб-сервер	10
2.13	Контекст файла	10
2.14	Изменение контекста	10
2.15	Проверка доступа через веб-сервер	11
2.16	Системный лог-файл	11
2.17	Файл /var/log/audit/audit.log	11
2.18	Прослушивание 100 порта	12
2.19	Сбой перезапуска	12
2.20	Содержимое системного лог-файла	12
2.21	Содержимое /var/log/http/error_log	12
2.22	Содержимое /var/log/http/access_log	13
2.23	Содержимое /var/log/audit/audit.log	13
2.24	Добавление порта 100 для httpd	13
2.25	Перезапуск Apache	13
2.26	Возвращение контекста	14
2.27	Доступ через браузер	14
2.28	Исправление конфигурационного файла	14
2.29	Удаление 100 порта	14
2.30	Удаление test.html	15

List of Tables

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 Выполнение лабораторной работы

1. Войдите в систему с полученными учётными данными и убедитесь, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus` (Рис. 2.1)

```
[guest@tdkalinin ~]$ 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
```

Figure 2.1: Статус selinux

2. Обратитесь с помощью браузера к веб-серверу, запущенному на вашем компьютере, и убедитесь, что последний работает (Рис. 2.2). Если не работает, запустите его так же, но с параметром `start` (Рис. 2.3)

```
[root@tdkalinin guest]# service httpd status
Redirecting to /bin/systemctl status httpd.service
○ httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service;
   Active: inactive (dead)
   Docs: man:httpd.service(8)
lines 1-4/4 (END)
```

Figure 2.2: Статус веб-сервера

```
[root@tdkalinin guest]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@tdkalinin guest]# 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; vendor pr
   Active: active (running) since Mon 2022-10-10 11:28:04 MSK; 14s ago
     Docs: man:httpd.service(8)
   Main PID: 3910 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes
     Tasks: 213 (limit: 24677)
    Memory: 22.9M
       CPU: 44ms
    CGroup: /system.slice/httpd.service
           └─3910 /usr/sbin/httpd -DFOREGROUND
             └─3911 /usr/sbin/httpd -DFOREGROUND
               └─3915 /usr/sbin/httpd -DFOREGROUND
                 └─3916 /usr/sbin/httpd -DFOREGROUND
                   └─3918 /usr/sbin/httpd -DFOREGROUND

окт 10 11:28:04 tdkalinin.localdomain systemd[1]: Starting The Apache HTTP Serv
окт 10 11:28:04 tdkalinin.localdomain systemd[1]: Started The Apache HTTP Serv
окт 10 11:28:04 tdkalinin.localdomain httpd[3910]: Server configured, listening
lines 1-19/19 (END)
```

Figure 2.3: Запуск веб-сервера

3. Найдите веб-сервер Apache в списке процессов, определите его контекст безопасности и занесите эту информацию в отчёт (Рис. 2.4).

```
[root@tdkalinin guest]# ps aux | grep httpd
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 3860 0.0 0.2 235988 9012 pts/0 T 11:25 0:00 /bin/systemctl status httpd.service
system_u:system_r:httpd_t:s0 root 3910 0.0 0.2 20864 11564 ? Ss 11:28 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 3911 0.0 0.1 21516 7216 ? S 11:28 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 3915 0.0 0.2 1879216 10888 ? Sl 11:28 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 3916 0.0 0.3 1210352 12936 ? Sl 11:28 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 3918 0.0 0.2 1879216 10888 ? Sl 11:28 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 4148 0.0 0.0 221692 2412 pts/0 S+ 11:29 0:00 grep --color=auto httpd
[root@tdkalinin guest]#
```

Figure 2.4: Контекст службы Apache

4. Посмотрите текущее состояние переключателей SELinux для Apache. Обратите внимание, что многие из них находятся в положении «off» (Рис. 2.5).

```
[root@tdkalinin guest]# 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
```

Figure 2.5: Состояние переключателей SELinux для Apache

5. Посмотрите статистику по политике с помощью команды `seinfo`, также определите множество пользователей, ролей, типов (Рис. 2.6).

```
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:          33 (MLS enabled)
Target Policy:           selinux
Handle unknown classes:  allow
Classes:                 133      Permissions:          454
Sensitivities:           1        Categories:          1024
Types:                   5002     Attributes:           254
Users:                   8         Roles:                14
Booleans:                347      Cond. Expr.:         381
Allow:                   63996     Neverallow:           0
Auditallow:              168      Dontaudit:            8417
Type_trans:              258486   Type_change:          87
Type_member:              35       Range_trans:          5960
Role_allow:              38       Role_trans:           420
Constraints:              72      Validatetrans:        0
MLS Constrain:           72       MLS Val. Tran:        0
Permissives:             0        Polcap:               5
Defaults:                 7       Typebounds:           0
Allowxperm:               0       Neverallowxperm:      0
Auditallowxperm:         0       Dontauditxperm:       0
Ibendportcon:            0       Ibpkeycon:            0
Initial SIDs:            27       Fs_use:               33
Genfscon:                 106     Portcon:              651
Netifcon:                 0       Nodecon:              0

[root@tdkalinin guest]#
```

Figure 2.6: Статистика по политике

6. Определите тип файлов и поддиректорий, находящихся в директории `/var/www` (Рис. 2.7):

```
[root@tdkalinin guest]# ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 мая 16 15:10 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 мая 16 15:10 html
[root@tdkalinin guest]#
```

Figure 2.7: Тип файлов и директорий в `/var/www`

7. Определите тип файлов, находящихся в директории `/var/www/html` (Рис. 2.8):

```
[root@tdkalinin guest]# ls -lZ /var/www/html
итого 0
```

Figure 2.8: Тип файлов в `/var/www/html`

8. Определите круг пользователей, которым разрешено создание файлов в директории `/var/www/html`. Как видим, создание файлов разрешено только владельцу каталога.
9. 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. 2.9):

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

Figure 2.9: Содержание test.html

10. Проверьте контекст созданного вами файла. Занесите в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории `/var/www/html` (Рис. 2.10).

```
[root@tdkalinin html]# ls -Z test.html
unconfined u:object_r:httpd_sys_content_t:s0 test.html
[root@tdkalinin html]#
```

Figure 2.10: Контекст test.html

11. Обратитесь к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Убедитесь, что файл был успешно отображён (Рис. 2.11).



Figure 2.11: Обращение к файлу через веб-сервер

12. Изучите справку `man httpd_selinux` и выясните, какие контексты файлов определены для `httpd` (Рис. 2.12). Сопоставьте их с типом файла `test.html`. Проверить контекст файла можно командой `ls -Z` (Рис. 2.13).

```
The following process types are defined for httpd:
httpd_t, httpd_helper_t, httpd_php_t, httpd_rotatelog_t, httpd_suexec_t, httpd_sys_script_t, httpd_user_script_t, httpd_passwd_t, httpd_unconfined_script_t
```

Figure 2.12: Обращение к файлу через веб-сервер

```
[root@tdkalinin html]# ls -Z test.html
unconfined u:object r:httpd_sys_content_t:s0 test.html
[root@tdkalinin html]# mn
```

Figure 2.13: Контекст файла

13. 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`. После этого проверьте, что контекст поменялся (Рис. 2.14).

```
[root@tdkalinin html]# chcon -t samba_share_t test.html
[root@tdkalinin html]# ls -Z test.html
unconfined u:object r:samba_share_t:s0 test.html
[root@tdkalinin html]#
```

Figure 2.14: Изменение контекста

14. 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 (Рис. 2.15):

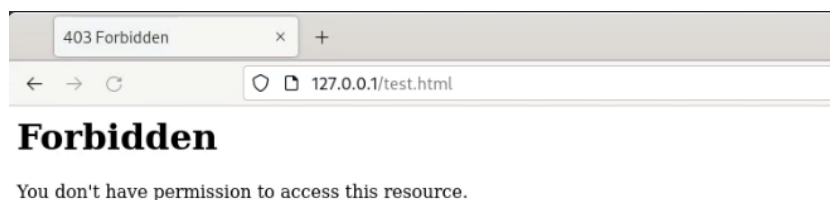


Figure 2.15: Проверка доступа через веб-сервер

15. Проанализируйте ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю? Потому что у процесса, который пытается запросить доступ к файлу, нет к нему доступа из-за метки.

Просмотрите log-файлы веб-сервера Apache. Также просмотрите системный лог-файл (Рис. 2.16):

```
Oct 10 12:19:03 tdkalinin setroubleshoot[5573]: SELinux запрещает /usr/sbin/httpd доступ getattr к файлу /var/www/html/test.html. Для выполнения всех сообщений SELinux: sealert -l d584e6c0-90fd-41af-9e8c-9402f50ee1d6
```

Figure 2.16: Системный лог-файл

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. Проверьте это утверждение самостоятельно (Рис. 2.17).

```
type=AVC msg=audit(1665393541.398:219): avc: denied { getattr } for pid=3918 comm="httpd" path="/var/www/html/test.html" dev="dm-0" ino=68945126 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file permissive=0
```

Figure 2.17: Файл /var/log/audit/audit.log

16. Попробуйте запустить веб-сервер Apache на прослушивание TCP-порта 100 (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 100 (Рис. 2.18).

```
#Listen 12.34.56.78:80
Listen 100
```

Figure 2.18: Прослушивание 100 порта

17. Выполните перезапуск веб-сервера Apache. Произошёл сбой? Поясните почему? Сбой произошёл из-за того, что у процесса нет доступа к 100 порту протокола TCP, так как он ограничивается selinux (Рис. 2.19).

```
[root@tdkalinin httpd]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" and "journalctl -xeu httpd.service"
for details.
```

Figure 2.19: Сбой перезапуска

18. Проанализируйте лог-файлы (Рис. 2.20):

```
[root@tdkalinin httpd]# tail -n1 /var/log/messages
Oct 10 12:35:20 tdkalinin systemd[1]: dbus-:1.10-org.fedoraproject.Setroubleshootd@2.service: Consumed 4.102s CPU time.
```

Figure 2.20: Содержимое системного лог-файла

Просмотрите файлы `/var/log/http/error_log` (Рис. 2.21), `/var/log/http/access_log` (Рис. 2.22) и `/var/log/audit/audit.log` (Рис. 2.23) и выясните, в каких файлах появились записи.

```
[root@tdkalinin log]# tail -n1 /var/log/httpd/error_log
[Mon Oct 10 12:35:03.829259 2022] [mpm_event:notice] [pid 6091:tid 6091]
AH00492: caught SIGWINCH, shutting down gracefully
```

Figure 2.21: Содержимое `/var/log/http/error_log`

```
[root@tdkalinin log]# tail -n3 /var/log/httpd/access_log
127.0.0.1 - - [10/Oct/2022:12:19:01 +0300] "GET /favicon.ico HTTP/1.1" 404 196 "http://127.0.0.1/test.html" "Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0"
127.0.0.1 - - [10/Oct/2022:12:27:13 +0300] "GET /test.html HTTP/1.1" 403 199 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0"
127.0.0.1 - - [10/Oct/2022:12:27:15 +0300] "GET /test.html HTTP/1.1" 403 199 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0"
[root@tdkalinin log]#
```

Figure 2.22: Содержимое /var/log/httpd/access_log

```
[root@tdkalinin httpd]# tail -n1 /var/log/audit/audit.log
type=SERVICE_STOP msg=audit(1665394520.204:244): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system u:system r:init t:s0 msg='unit=dbus-:1.10-org.fedoraproject.Setroubleshootd@2 comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=failed'UID="root" AUID="unset"
```

Figure 2.23: Содержимое /var/log/audit/audit.log

19. Выполните команду `semanage port -a -t http_port_t -p tcp 81` После этого проверьте список портов (Рис. 2.24)

```
[root@tdkalinin log]# semanage port -a -t http_port_t -p tcp 100
[root@tdkalinin log]# semanage port -l | grep http_port_t
http_port_t tcp 100, 80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus http_port_t tcp 5988
[root@tdkalinin log]#
```

Figure 2.24: Добавление порта 100 для httpd

Убедитесь, что порт 100 появился в списке.

20. Попробуйте запустить веб-сервер Apache ещё раз (Рис. 2.25). Поняли ли вы, почему он сейчас запустился, а в предыдущем случае не смог? Потому что в этот раз у процесса веб-сервера был доступ к порту 100.

```
[root@tdkalinin log]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@tdkalinin log]#
```

Figure 2.25: Перезапуск Apache

21. Верните контекст `httpd_sys_content_t` к файлу `/var/www/html/ test.html` (Рис. 2.26):

```
[root@tdkalinin log]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@tdkalinin log]#
```

Figure 2.26: Возвращение контекста

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. 2.27).

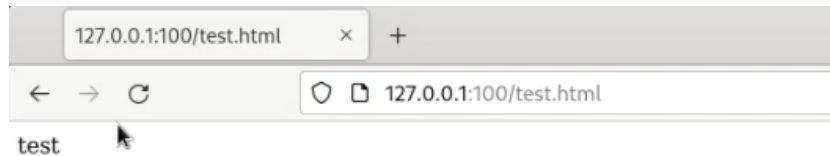


Figure 2.27: Доступ через браузер

22. Исправьте обратно конфигурационный файл `apache`, вернув `Listen 80` (Рис. 2.28).

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

Figure 2.28: Исправление конфигурационного файла

23. Удалите привязку `http_port_t` к 100 порту и проверьте, что порт 100 удалён (Рис. 2.29).

```
[root@tdkalinin conf]# semanage port -d -t http_port_t -p tcp 100
[root@tdkalinin log]#
```

Figure 2.29: Удаление 100 порта

24. Удалите файл `/var/www/html/test.html` (Рис. 2.30):

```
[root@tdkalinin log]# rm /var/www/html/test.html  
rm: удалить обычный файл '/var/www/html/test.html'? y  
[root@tdkalinin log]# █
```

Figure 2.30: Удаление test.html

3 Выводы

Мы развили навыки администрирования ОС Linux и получили первое практическое знакомство с технологией SELinux. Проверили работу SELinux на практике совместно с веб-сервером Apache.

4 Библиография

1. Лабораторная работа №6. - 5 с. URL: https://esystem.rudn.ru/pluginfile.php/1651891/mod_resource/content/2/006-lab_selinux.pdf