

Отчёт по лабораторной работе 6

Основы информационной безопасности

Нджову Нелиа

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	17
	Список литературы	18

Список иллюстраций

2.1	проверка режима работы SELinux	6
2.2	Проверка работы Apache	7
2.3	Контекст безопасности Apache	7
2.4	Состояние переключателей SELinux	8
2.5	Статистика по политике	9
2.6	Типы поддиректорий	9
2.7	Типы файлов	9
2.8	Создание файла	10
2.9	Содержание файла	10
2.10	Контекст файла	10
2.11	Отображение файла	11
2.12	Изучение справки по команде	11
2.13	Изменение контекста	12
2.14	Отображение файла	12
2.15	Попытка прочесть лог-файл	13
2.16	Изменение файла	13
2.17	Изменение порта	13
2.18	Попытка прослушивания другого порта	14
2.19	Проверка лог-файлов	14
2.20	Проверка лог-файлов	15
2.21	Проверка портов	15
2.22	Перезапуск сервера	15
2.23	Проверка сервера	16
2.24	Проверка порта 81	16
2.25	Удаление файла	16

Список таблиц

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.

2 Выполнение лабораторной работы

Я вошла в свою учетную запись и убедилась, что SELinux работает в режиме принудительного применения целевой политики, используя команды `getenforce` и `status`(рис.1)

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

Рис. 2.1: проверка режима работы SELinux

Я запускаю сервер `apache`, затем использую браузер для доступа к веб-серверу, запущенному на компьютере, он запущен, как видно из вывода команды `service httpd status`(рис.2)

```

[Nelianjovu@Nelianjovu ~]$ sudo systemctl start httpd
[Nelianjovu@Nelianjovu ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[Nelianjovu@Nelianjovu ~]$ 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 20:28:07 MSK; 27s ago
     Docs: man:httpd.service(8)
  Main PID: 40742 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0"
    Tasks: 177 (limit: 12114)
  Memory: 32.5M
    CPU: 74ms
  CGroup: /system.slice/httpd.service
          └─40742 /usr/sbin/httpd -DFOREGROUND
             └─40743 /usr/sbin/httpd -DFOREGROUND
                └─40744 /usr/sbin/httpd -DFOREGROUND
                   └─40745 /usr/sbin/httpd -DFOREGROUND
                      └─40746 /usr/sbin/httpd -DFOREGROUND

Apr 29 20:28:07 Nelianjovu systemd[1]: Starting The Apache HTTP Server...
Apr 29 20:28:07 Nelianjovu httpd[40742]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 instead.

```

Рис. 2.2: Проверка работы Apache

Я нашла веб-сервер Apache в списке процессов, используя команду `ps aux | grep httpd`. Его контекст безопасности - `http_t`(рис.3)

```

[Nelianjovu@Nelianjovu ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 40742 0.0 0.5 20312 11716 ?
ss 20:28 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 40743 0.0 0.3 21648 7372 ?
S 20:28 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 40744 0.0 0.8 2095376 17188 ?
sl 20:28 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 40745 0.0 0.5 1964240 11044 ?
sl 20:28 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 40746 0.0 0.8 1964240 17188 ?
sl 20:28 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 Nelianj+ 40996 0.0 0.1 22
1664 2260 pts/0 S+ 20:30 0:00 grep --color=auto httpd
[Nelianjovu@Nelianjovu ~]$

```

Рис. 2.3: Контекст безопасности Apache

Я просмотрела текущее состояние коммутаторов SELinux для Apache, используя команду `status -grep httpd`(рис.4)

```

[Nelianjovu@Nelianjovu ~]$ 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

```

Рис. 2.4: Состояние переключателей SELinux

Я просмотрела статистику по политике, используя команду `setinfo`. Всего 8 пользователей, 39 ролей и 5135 типов(рис.5)


```
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:	65380	Neverallow:	0
Auditallow:	172	Dontaudit:	8647
Type_trans:	267809	Type_change:	94
Type_member:	37	Range_trans:	6164
Role allow:	39	Role_trans:	419
Constraints:	70	Validatetrans:	0
MLS Constrains:	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

Рис. 2.5: Статистика по политике

Типы подкаталогов, расположенных в каталоге /var/www с помощью команды `ls -lZ /var/www`, следующие: владельцем является root, только у владельца есть права на изменение. В каталоге нет файлов(рис.6)

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

Рис. 2.6: Типы поддиректорий

В директории /var/www/html нет файлов(рис.7)

```
[Nelianjovu@Nelianjovu ~]$ ls -lZ /var/www/html
total 0
[Nelianjovu@Nelianjovu ~]$
```

Рис. 2.7: Типы файлов

Только суперпользователь может создать файл, поэтому я создала файл с по-

мощью команды `touch.html` и ввела в него код(рис.8 и рис.9)

```
[Nelianjovu@Nelianjovu ~]$ sudo touch /var/www/html/test.html
[sudo] password for Nelianjovu:
[Nelianjovu@Nelianjovu ~]$ sudo nano /var/www/html/test.html
[Nelianjovu@Nelianjovu ~]$ sudo nano /var/www/html/test.html
[Nelianjovu@Nelianjovu ~]$ sudo cat /var/www/html/test.html
<html>
<body>test</body>
</html>
[Nelianjovu@Nelianjovu ~]$
```

Рис. 2.8: Создание файла

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

Рис. 2.9: Содержание файла

Я проверяю контекст созданного файла. По умолчанию это `httpd_sys_content_type`(рис.10)

```
[Nelianjovu@Nelianjovu ~]$ ls -lZ /var/www/html/
total 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 Apr 29 20:48 test.html
[Nelianjovu@Nelianjovu ~]$
```

Рис. 2.10: Контекст файла

Я получаю доступ к файлу через веб-сервер, вводя адрес в браузере `http://127.0.0.1/test.html` . Файл был успешно отображен(рис.11)

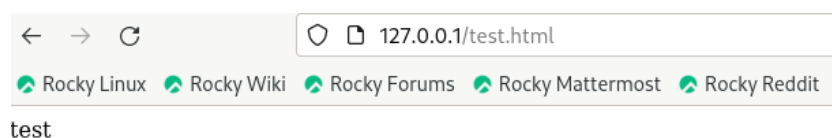


Рис. 2.11: Отображение файла

Я изучил справку man по httpd-selinux. Давайте рассмотрим полученный контекст подробнее. Поскольку по умолчанию пользователи CentOS не имеют доступа к типам, созданный нами файл test.html SELinux был сопоставлен, пользователь unconfined-u. Это первая часть контекста. Кроме того, политика совместного использования ролей RBAC используется процессами, но не файлами, поэтому роли не имеют значения для файлов. Роль object-r используется по умолчанию для файлов на “постоянных” носителях и в сетевых файловых системах. 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.12)

```
HTTPD(8)                                httpd                                HTTPD(8)
NAME
    httpd - Apache Hypertext Transfer Protocol Server

SYNOPSIS
    httpd [ -d serverroot ] [ -f config ] [ -C directive ] [ -c directive ]
    [ -D parameter ] [ -e level ] [ -E file ] [ -k start|restart|grace-
    ful|stop|graceful-stop ] [ -h ] [ -l ] [ -L ] [ -s ] [ -t ] [ -v ] [ -V
    ] [ -X ] [ -M ] [ -T ]

    On Windows systems, the following additional arguments are available:

    httpd [ -k install|config|uninstall ] [ -n name ] [ -w ]

SUMMARY
    httpd is the Apache HyperText Transfer Protocol (HTTP) server program.
    It is designed to be run as a standalone daemon process. When used like
    this it will create a pool of child processes or threads to handle re-
    quests.

    In general, httpd should not be invoked directly, but rather should be
    invoked via apachectl on Unix-based systems or as a service on Windows
    Manual page httpd(8) line 1 (press h for help or q to quit)
```

Рис. 2.12: Изучение справки по команде

I change the context of the file /var/www/html/test.html from httpd_sys_content_t to any other that the httpd process should not have access to, for example, to

samba_share_t:chcon -t samba_share_t /var/www/html/test.html css/var/www/html/test.html

The context has really changed(рис.13)

```
[Nelianjovu@Nelianjovu ~]$ sudo chcon -t samba_share_t /var/www/html/test.html
[sudo] password for Nelianjovu:
[Nelianjovu@Nelianjovu ~]$ ls -lZ /var/www/html
total 4
-rw-r--r--. 1 root root unconfined_u:object_r:samba_share_t:s0 33 Apr 29 20:48 test.html
[Nelianjovu@Nelianjovu ~]$
```

Рис. 2.13: Изменение контекста

Когда я пытаюсь отобразить файл в браузере, мы получаем сообщение об ошибке(рис.14)

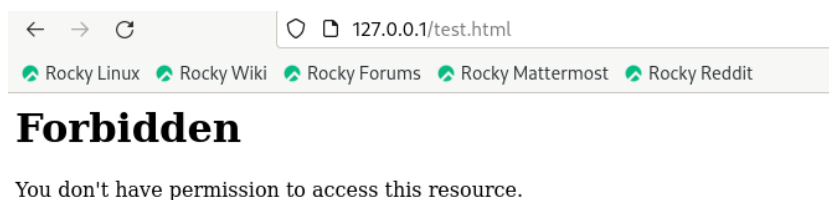


Рис. 2.14: Отображение файла

Файл не был отображен, хотя права доступа позволяют любому пользователю читать этот файл, поскольку был задан контекст, к которому процесс httpd не должен иметь доступа. Я просматриваю файлы журнала веб-сервера Apache и файл системного журнала: tail /var/log/messages. Если в системе запущены процессы setroubleshootd и audit, вы также можете увидеть ошибки, аналогичные перечисленным выше, в файле /var/log/audit/audit.log(рис.15)

```
[Nelianjovu@Nelianjovu ~]$ ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 33 Apr 29 20:48 /var/www/html/test.html
[Nelianjovu@Nelianjovu ~]$ tail /var/log/audit/audit.log
tail: cannot open '/var/log/audit/audit.log' for reading: Permission denied
[Nelianjovu@Nelianjovu ~]$ sudo tail /var/log/audit/audit.log
type=SYSCALL msg=audit(1745949381.902:301): arch=c000003e syscall=262 success=no
exit=-13 a0=ffffff9c a1=7f6edc00ac00 a2=7f6eda7fb8b0 a3=100 items=0 ppid=40742
pid=40746 audit=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48
fsgid=48 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd" subj=syste
m_u:system_r:httpd_t:s0 key=(null)ARCH=x86_64 SYSCALL=newfstatat AUID="unset" U
ID="apache" GID="apache" EUID="apache" SUID="apache" FSUID="apache" EGID="apache
" SGID="apache" FSGID="apache"
type=PROCTITLE msg=audit(1745949381.902:301): proctitle=2F7573722F7362696E2F6874
747064002D44464F524547524F554E44
type=SERVICE_START msg=audit(1745949382.091:302): pid=1 uid=0 audit=4294967295 se
s=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=setroubleshootd comm="sy
stemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'U
ID="root" AUID="unset"
type=SERVICE_START msg=audit(1745949382.228:303): pid=1 uid=0 audit=4294967295 se
s=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=dbus-1.1-org.fedoraproj
ect.SetroubleShootPrivileged@1 comm="systemd" exe="/usr/lib/systemd/systemd" hos
```

Рис. 2.15: Попытка прочесть лог-файл

Чтобы запустить веб-сервер Apache, прослушивающий TCP-порт 81, я открываю файл /etc/httpd/httpd.conf для внесения изменений(рис.16)

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

Рис. 2.16: Изменение файла

Нахожу строчку Listen 80 и заменяю её на Listen 81(рис.17)

```
GNU nano 5.6.1 /etc/httpd/conf/httpd.conf
# least PidFile.
#
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
#
# Dynamic Shared Object (DSO) Support
#
[ Wrote 358 lines ]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line
```

Рис. 2.17: Изменение порта

Я перезапускаю веб-сервер Apache. Сбой произошел из-за того, что порт 80 предназначен для локальной сети, а порт 81 - нет(рис.18)

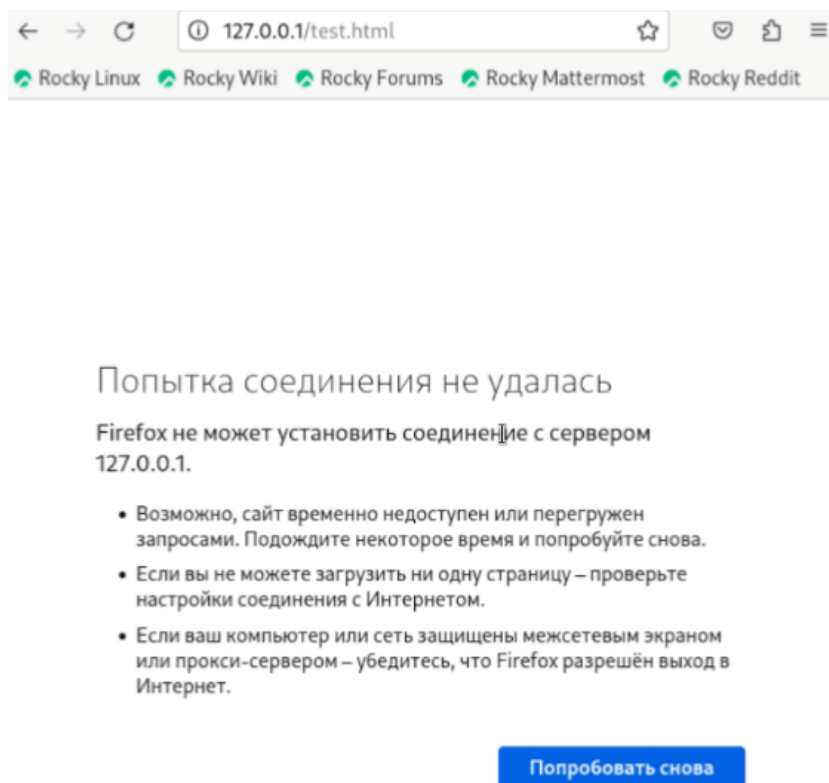


Рис. 2.18: Попытка прослушивания другого порта

Проанализируйте лог-файлы: `tail -nl /var/log/messages`(рис.19)

```
[Nelianjovu@Nelianjovu ~]$ sudo tail -nl /var/log/messages
Apr 29 21:05:18 Nelianjovu systemd[1]: setroubleshootd.service: Deactivated successfully.
[Nelianjovu@Nelianjovu ~]$
```

Рис. 2.19: Проверка лог-файлов

Я просматриваю файлы `/var/log/http/error_log`, `/var/log/httpd access_log` и `/var/log/audit/аудит.лог` и выясняю, в каких файлах появились записи. Запись появилась в файле `error_log`(рис.20)

```
[Nelianjovu@Nelianjovu ~]$ sudo cat /var/log/httpd/error_log
[Tue Apr 29 20:28:07.724642 2025] [core:notice] [pid 40742:tid 40742] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Tue Apr 29 20:28:07.725936 2025] [suexec:notice] [pid 40742:tid 40742] AH01232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using fe80::a00:27ff:feab:767f%enp0s3. Set the 'ServerName' directive globally to suppress this message
[Tue Apr 29 20:28:07.736768 2025] [lbmethod_heartbeat:notice] [pid 40742:tid 40742] AH02282: No slotmem from mod_heartbeat
[Tue Apr 29 20:28:07.740790 2025] [mpm_event:notice] [pid 40742:tid 40742] AH00489: Apache/2.4.62 (Rocky Linux) configured -- resuming normal operations
[Tue Apr 29 20:28:07.740816 2025] [core:notice] [pid 40742:tid 40742] AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'
[Tue Apr 29 20:56:02.521740 2025] [core:error] [pid 40746:tid 40904] (13)Permission denied: [client 127.0.0.1:55942] 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 20:56:21.904060 2025] [core:error] [pid 40746:tid 40907] (13)Permission denied: [client 127.0.0.1:34300] 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 21:03:30.155207 2025] [core:error] [pid 40746:tid 40909] (13)Permission denied: [client 127.0.0.1:55942] AH00035: access to /test.html denied (filesystem path '/var/www/html/test.html') because search permissions are missing on a component of the path
```

Рис. 2.20: Проверка лог-файлов

Я запускаю команду `semanage port -at http_port_t tcp 81`, после чего проверяю список портов командой `semanage port -l | grep http_port_t`. Порт 81 появился в списке (рис.21)

```
[Nelianjovu@Nelianjovu ~]$ sudo semanage port -a -t http_port_t -p tcp 81
ValueError: Port tcp/81 already defined
[Nelianjovu@Nelianjovu ~]$ 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
```

Рис. 2.21: Проверка портов

Перезапускаю сервер Apache (рис.22)

```
[Nelianjovu@Nelianjovu ~]$ sudo systemctl restart httpd
[Nelianjovu@Nelianjovu ~]$ sudo chcon -t httpd_sys_content_t /var/www/html/test.html
[Nelianjovu@Nelianjovu ~]$ sudo systemctl restart httpd
[Nelianjovu@Nelianjovu ~]$
```

Рис. 2.22: Перезапуск сервера

Теперь он работает, ведь мы внесли порт 81 в список портов `httpd_port_t` (рис.23)

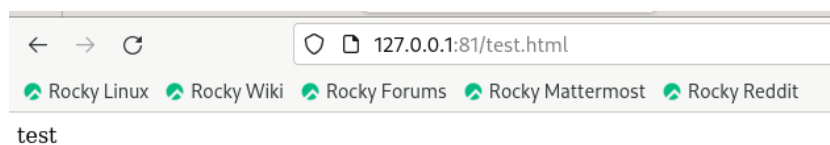


Рис. 2.23: Проверка сервера

Возвращаю в файле /etc/httpd/httpd.conf порт 80, вместо 81. Проверяю, что порт 81 удален, это правда(рис.24)

```
[Nelianjovu@Nelianjovu ~]$ sudo nano /etc/httpd/conf/httpd.conf
[Nelianjovu@Nelianjovu ~]$ semanage port -d -t http_port_t -p tcp 81
ValueError: SELinux policy is not managed or store cannot be accessed.
[Nelianjovu@Nelianjovu ~]$ sudo semanage port -d -t http_port_t -p tcp 81
ValueError: Port tcp/81 is defined in policy, cannot be deleted
[Nelianjovu@Nelianjovu ~]$
```

Рис. 2.24: Проверка порта 81

Далее удаляю файл test.html, проверяю, что он удален(рис.25)

```
[Nelianjovu@Nelianjovu ~]$ ls -lZ /var/www/html
total 0
[Nelianjovu@Nelianjovu ~]$
```

Рис. 2.25: Удаление файла

3 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

006-lab_selinux.pdf