

Лабораторная работа №7

Мандатное разграничение прав в Linux

Чванова Ангелина Дмитриевна

2024 год

Российский университет дружбы народов, Москва, Россия

Докладчик

- Чванова Ангелина Дмитриевна
- студент
- Российский университет дружбы народов
- angelinachdm@gmail.com
- <https://adchvanova-new.github.io/ru/>



Цель работы

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

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.

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).

Выполнение лабораторной работы

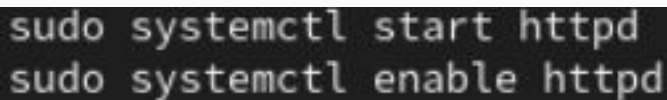
Для начала был выполнен вход в систему под своей учетной записью. После чего необходимо было проверить, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`

```
[root@localhost ~]# getenforce
Permissive
[root@localhost ~]# sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
Current mode:                  permissive
Mode from config file:        enforcing
Policy MLS status:            enabled
Policy deny_unknown status:    allowed
Memory protection checking:    actual (secure)
Max kernel policy version:     33
[root@localhost ~]# _
```

Рис. 1: проверка режима работы SELinux

Выполнение лабораторной работы

Запускаем сервер apache, далее обращаемся с помощью браузера к веб-серверу, запущенному на компьютере, он работает, что видно из вывода команды `service httpd status`

A terminal window with a dark background and light-colored text. It contains two lines of commands: 'sudo systemctl start httpd' and 'sudo systemctl enable httpd'.

```
sudo systemctl start httpd
sudo systemctl enable httpd
```

Рис. 2: Проверка работы Apache

Выполнение лабораторной работы

С помощью команды `ps auxZ | grep httpd` находим веб-сервер Apache в списке процессов. Его контекст безопасности - `httpd_t`

```
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 Sat 2024-04-20 04:52:10 MSK; 31s ago
     Docs: man:httpd.service(8)
   Main PID: 30093 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0"
   Tasks: 213 (limit: 10899)
  Memory: 37.9M
    CPU: 301ms
   CGroup: /system.slice/httpd.service
           └─30093 /usr/sbin/httpd -DFOREGROUND
             └─30133 /usr/sbin/httpd -DFOREGROUND
               └─30134 /usr/sbin/httpd -DFOREGROUND
                 └─30135 /usr/sbin/httpd -DFOREGROUND
                   └─30136 /usr/sbin/httpd -DFOREGROUND
```

Рис. 3: Контекст безопасности Apache

Выполнение лабораторной работы

Просмотрим текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -bigrep httpd`

```
system_u:system_r:httpd_t:s0 root 30093 0.1 0.6 20340 11624 ?
Ss 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 30133 0.0 0.4 21676 7436 ?
S 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 30134 0.0 1.0 2193664 19320 ?
Sl 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 30135 0.0 0.8 2062528 15228 ?
Sl 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 30136 0.0 0.8 2062528 15228 ?
Sl 04:52 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 evdwork+ 42224 0.0 0.1 22
1688 2388 pts/0 S+ 04:53 0:00 grep --color=auto httpd
```

Рис. 4: Состояние переключателей SELinux

Выполнение лабораторной работы

Просмотрим статистику по политике с помощью команды `seinfo`.
Множество пользователей - 8, ролей - 39, типов - 5135.

```
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
cdrecord_read_content          off
cluster_can_network_connect    off
```

Рис. 5: Статистика по политике

Выполнение лабораторной работы

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

```
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:                   5135
Users:                    8
Booleans:                 357
Allow:                    65409
Auditallow:               172
Type_trans:               267813
Type_member:              37
Role_allow:               39
Constraints:              70
MLS Constrain:            72
Permissives:              2
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.:              390
Neverallow:               0
Dontaudit:                8647
Type_change:              94
Range_trans:              6164
Role_trans:               419
Validatetrans:            0
MLS Val. Tran:            0
Polcap:                   6
Typebounds:               0
Neverallowxperm:          0
Dontauditxperm:           0
Ibpkeycon:                0
Fs_use:                   35
Portcon:                   665
Nodecon:                  0
```

Рис. 6: Типы поддиректорий

Выполнение лабораторной работы

В директории `/var/www/html` нет файлов.

```
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 окт 28 12:35 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0      6 окт 28 12:35 html
```

Рис. 7: Типы файлов

Выполнение лабораторной работы

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

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

```
sudo touch /var/www/html/test.html
```

Рис. 8: Создание файла

Выполнение лабораторной работы

Проверяем контекст созданного файла. По умолчанию это httpd_sys_content_t

A terminal window with a dark background and light gray text. It contains two lines of commands: 'sudo nano /var/www/html/test.html' and 'sudo cat /var/www/html/test.html'.

```
sudo nano /var/www/html/test.html  
sudo cat /var/www/html/test.html
```

Рис. 9: Контекст файла

Выполнение лабораторной работы

Проверяем контекст созданного файла. По умолчанию это httpd_sys_content_t

```
итого 4  
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 anp 20 05:01 test.html
```

Рис. 10: Контекст файла

Выполнение лабораторной работы

Обращаемся к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Файл успешно отображается

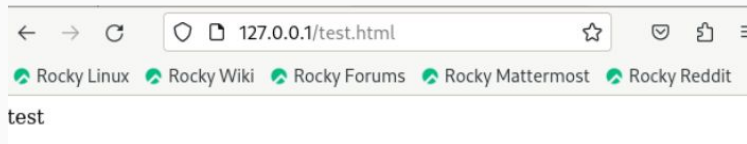


Рис. 11: Отображение файла

Выполнение лабораторной работы

```
HTTPD(8)                                httpd

NAME
    httpd - Apache Hypertext Transfer Protocol Server

SYNOPSIS
    httpd [ -d serverroot ] [ -f config ] [ -C directive ] [ -c directive ]
    [ -e level ] [ -E file ] [ -k start|restart|graceful|stop|graceful-stop ]
    [ -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 can be run as a standalone daemon process. When used like this it will create
    child processes or threads to handle requests.
```

Рис. 12: Изучение справки по команде

Изменяем контекст файла на любой другой, к которому процесс httpd не должен иметь доступа

```
итого 4  
-rw-r--r--. 1 root root unconfined_u:object_r:samba_share_t:s0 33 anp 20 05:01 test.html
```

Рис. 13: Изменение контекста

Выполнение лабораторной работы

При попытке отображения файла в браузере получаем сообщение об ошибке

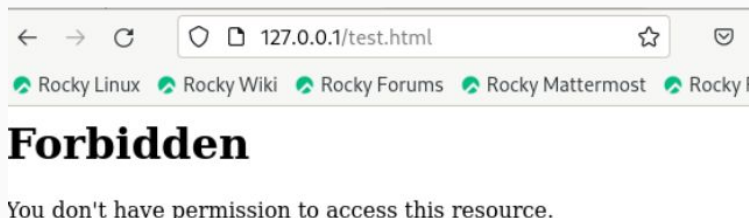


Рис. 14: Отображение файла

Выполнение лабораторной работы

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. Просматриваем log-файлы веб-сервера Apache и системный лог-файл и если в системе окажутся запущенными процессы setroubleshootd и auditd, то также можно увидеть ошибки, аналогичные указанным выше, в файле /var/log/audit/audit.log.

```
type=SYSCALL msg=audit(1713578987.972:279): arch=c000003e syscall=262 success=no exit=-13 a0=ffffff9c a1=7f97cc004c10 a2=7f97c2ffc8b0 a3=100 items=0 p
pid=30893 ppid=30136 auid=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/s
bin/httpd" subj=system_u:system_r:httpd_t:s0 key=(null)ARCH=x86_64 SYSCALL=newfstatat AUID="unset" UID="apache" GID="apache" EUID="apache" SUID="apach
e" FSUID="apache" EGID="apache" SGID="apache" FSGID="apache"
type=PROCTITLE msg=audit(1713578987.972:279): proctitle=2F7573722F73626962F6874747064002044464F524547524F554E44
type=SERVICE_START msg=audit(1713578989.341:280): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=setroubleshoot
d comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'UID="root" AUID="unset"
type=SERVICE_START msg=audit(1713578990.334:281): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=dbus-1.1-org.
fedoraproject.SetroubleshootPrivileged@1 comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'UID="root" AUID="unset"
```

Рис. 15: Попытка прочесть лог-файл

Выполнение лабораторной работы

Чтобы запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80) открываем файл `/etc/httpd/httpd.conf` для изменения.

```
sudo nano /etc/httpd/conf/httpd.conf
```

Рис. 16: Изменение файла

Выполнение лабораторной работы

Находим строчку Listen 80 и заменяем её на Listen 81.

```
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used. If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# 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
#
# To be able to use the functionality of a module which was built as a 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
```

Рис. 17: Изменение порта

Выполнение лабораторной работы

После чего выполняем перезапуск веб-сервера Apache. Произошёл сбой, потому что порт 80 для локальной сети, а 81 нет

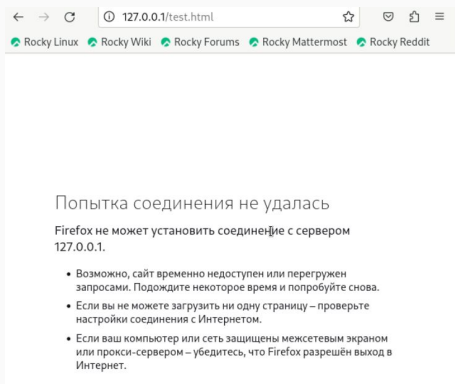


Рис. 18: Попытка прослушивания другого порта

Выполнение лабораторной работы

```
sudo tail -n1 /var/log/messages  
systemd[1]: Started The Apache HTTP Server.
```

Рис. 19: Проверка лог-файлов

Выполнение лабораторной работы

Запись появилась в файлу error_log

```
[Sat Apr 20 04:52:10.304359 2024] [core:notice] [pid 30093:tid 30093] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Sat Apr 20 04:52:10.307330 2024] [suexec:notice] [pid 30093:tid 30093] AH0122: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using fe80::a00:27ff:fe98:bdea%enp0s3. Set the 'ServerName' directive globally to suppress this message
[Sat Apr 20 04:52:10.371973 2024] [lbmethod_heartbeat:notice] [pid 30093:tid 30093] AH02282: No slotmem from mod_heartbeat
[Sat Apr 20 04:52:10.389422 2024] [mpm_event:notice] [pid 30093:tid 30093] AH0489: Apache/2.4.57 (Rocky Linux) configured -- resuming normal operations
[Sat Apr 20 04:52:10.389524 2024] [core:notice] [pid 30093:tid 30093] AH0094: Command line: '/usr/sbin/httpd -D FOREGROUND'
[Sat Apr 20 05:09:47.974451 2024] [core:error] [pid 30136:tid 30312] (13)Permission denied: [client 127.0.0.1:44098] AH00035: access to /test.html denied (filesystem path '/var/www/html/test.html') because search permissions are missing on a component of the path
[Sat Apr 20 05:15:41.743945 2024] [core:error] [pid 30134:tid 30322] (13)Permission denied: [client 127.0.0.1:58006] AH00035: access to /test.html denied (filesystem path '/var/www/html/test.html') because search permissions are missing on a component of the path
[Sat Apr 20 05:16:30.614988 2024] [mpm_event:notice] [pid 30093:tid 30093] AH
```

Рис. 20: Проверка лог-файлов

Выполнение лабораторной работы

Выполняем команду `semanage port -a -t http_port_t -p tcp 81` После этого проверяем список портов командой `semanage port -l | grep http_port_t` Порт 81 появился в списке (рис. [-@fig:021]).



```
sudo semanage port -l | grep http_port_t
tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
```

Рис. 21: Проверка портов

Перезапускаем сервер Apache

```
sudo systemctl restart httpd  
sudo chcon -t httpd_sys_content_t /var/www/html/test.html  
sudo systemctl restart httpd
```

Рис. 22: Перезапуск сервера

Выполнение лабораторной работы

Теперь он работает, ведь порт 81 влючен в список портов 'httpd_port_t

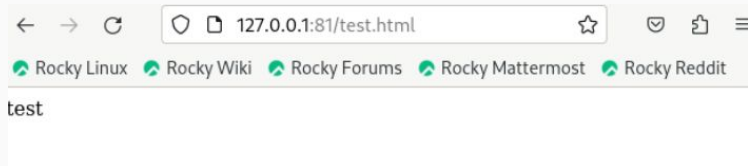


Рис. 23: Проверка сервера

Выполнение лабораторной работы

Возвращаем в файле /etc/httpd/httpd.conf порт 80, вместо 81.
Проверяем, что порт 81 удален

```
sudo nano /etc/httpd/conf/httpd.conf  
semanage port -d -t http_port_t -p tcp 81  
к не задана, или нет доступа к хранилищу.  
sudo semanage port -d -t http_port_t -p tcp 81  
defined in policy, cannot be deleted
```

Рис. 24: Проверка порта 81

Далее чего удаляем файл test.html

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

[1] Документация по Virtual Box:

<https://www.virtualbox.org/wiki/Documentation>

[2] Документация по Git: <https://git-scm.com/book/ru/v2>

[3] Документация по Markdown:

<https://learn.microsoft.com/ru-ru/contribute/markdown-reference>