

Лабораторная работа № 6

Мандатное разграничение прав в Linux

Абдуллаев Сайидазизхон Шухратович

Содержание

Цель работы	5
Задание	6
Теоретическое введение	7
Ход работы	8
Выводы	16

Список иллюстраций

1	Проверка режима и политики SELinux	8
2	Проверка статуса веб-сервера	9
3	Определение контекста безопасности	9
4	Состояние переключателей	10
5	Статистика по политике	10
6	Определяем типы	11
7	Создание html-файла	11
8	Проверка контекста файла	12
9	Обращение через веб-сервер	12
10	Тип файла	12
11	Изменение контекста	13
12	Повторный просмотр файла в браузере	13
13	Запуск веб-сервера	14
14	Анализ лог-файлов	14
15	Запуск веб-сервера	14
16	Доступ к файлу	15
17	Исправление конфигурационного файла	15
18	Удаление привезки и файла html	15

Список таблиц

Цель работы

Развить навыки администрирования ОС Linux. Проверить работу SELinux на практике совместно с веб-сервером Apache.

Задание

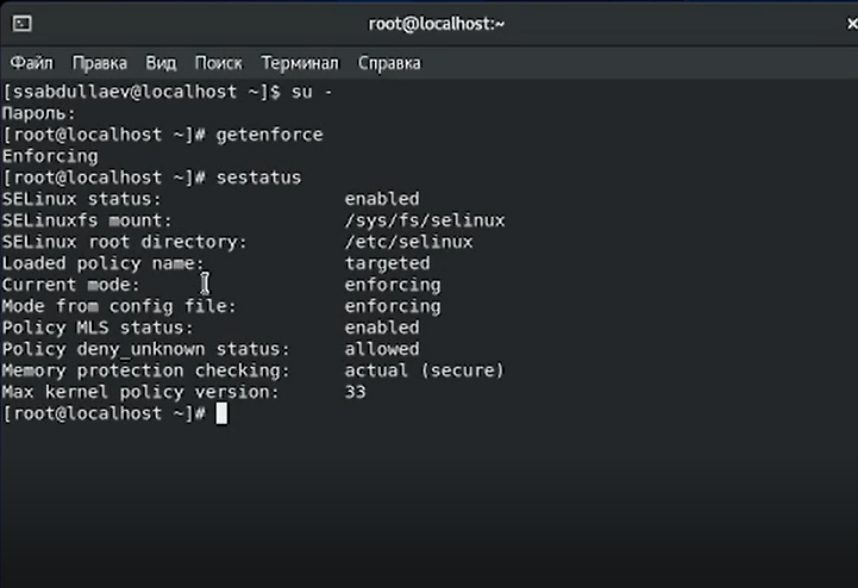
Получить первое практическое знакомство с технологией SELinux

Теоретическое введение

Для разграничения доступа субъектов — программ к объектам — файлам дерева каталогов используют так называемый мандатный (от англ, mandatory — обязательный или принудительный) подход (MAC, mandatory access control), предполагающий следование обязательным правилам доступа к файлам, назначаемым администраторами системы. Правила доступа строятся на основе знания о внутреннем устройстве программ и представляют собой описание набора минимально необходимых условий их целевого функционирования. Таким образом, в мандатных правилах, ограничивающих доступ к SSH-ключам пользователя, только программе ssh должен быть разрешен доступ для непосредственного выполнения своих прямых функций, а программам firefox и skype в доступе к SSH-ключам должно быть отказано.

Ход работы

1. Входим в систему и убеждаемся, что SELinux работает в режиме enforcing политики targeted. (Рис. [-@fig:001]).



```
root@localhost:~  
Файл Правка Вид Поиск Терминал Справка  
[ssabdullaev@localhost ~]$ su -  
Пароль:  
[root@localhost ~]# getenforce  
Enforcing  
[root@localhost ~]# 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  
[root@localhost ~]#
```

Рис. 1: Проверка режима и политики SELinux

2. Проверяем работу веб-сервера. (Рис. [-@fig:002]).


```
root@localhost:~  
Файл Правка Вид Поиск Терминал Справка  
[root@localhost ~]# 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 pres  
   Active: active (running) since Sat 2022-02-12 09:58:57 MSK; 31min ago  
     Docs: man:httpd.service(8)  
  Main PID: 3937 (httpd)  
    Status: "Running, listening on: port 80"  
   Tasks: 213 (limit: 11233)  
  Memory: 28.9M  
   CGroup: /system.slice/httpd.service  
           └─3937 /usr/sbin/httpd -DFOREGROUND  
             └─3938 /usr/sbin/httpd -DFOREGROUND  
               └─3939 /usr/sbin/httpd -DFOREGROUND  
                 └─3940 /usr/sbin/httpd -DFOREGROUND  
                   └─3941 /usr/sbin/httpd -DFOREGROUND  
фев 12 09:58:57 localhost.localdomain systemd[1]: Starting The Apache HTTP Serv  
фев 12 09:58:57 localhost.localdomain httpd[3937]: AH00558: httpd: Could not re  
фев 12 09:58:57 localhost.localdomain systemd[1]: Started The Apache HTTP Serve  
фев 12 09:58:57 localhost.localdomain httpd[3937]: Server configured, listening  
lines 1-19/19 (END)
```

Рис. 2: Проверка статуса веб-сервера

3. Определяем контекст безопасности веб-сервера.(Рис. [-@fig:003]).

```
root@localhost:~  
Файл Правка Вид Поиск Терминал Справка  
[root@localhost ~]# ps -eZ | grep httpd  
system_u:system_r:httpd_t:s0      3937 ?        00:00:00 httpd  
system_u:system_r:httpd_t:s0      3938 ?        00:00:00 httpd  
system_u:system_r:httpd_t:s0      3939 ?        00:00:00 httpd  
system_u:system_r:httpd_t:s0      3940 ?        00:00:00 httpd  
system_u:system_r:httpd_t:s0      3941 ?        00:00:00 httpd  
[root@localhost ~]#
```

Рис. 3: Определение контекста безопасности

4. Теперь посмотрим текущее состояние SELinux переключателей. Как можно заметить, практически все переключатели выключены. (Рис. [-@fig:004]).

```

root@localhost:~
Файл Правка Вид Поиск Терминал Справка
[root@localhost ~]# 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_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

```

Рис. 4: Состояние переключателей

- Посмотрели статистику по политике. Кроме того, определили множество пользователей, ролей, типов (Рис. [-@fig:005]).

```

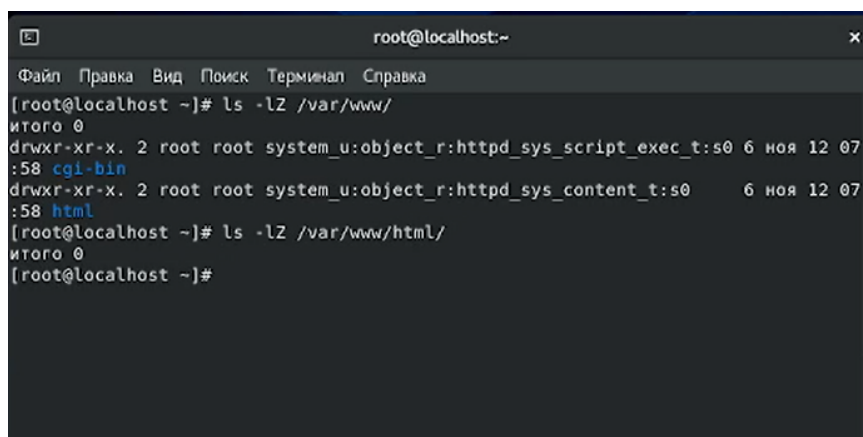
root@localhost:~
Файл Правка Вид Поиск Терминал Справка
Policy Version: 31 (MLS enabled)
Target Policy: selinux
Handle unknown classes: allow
Classes: 132 Permissions: 464
Sensitivities: 1 Categories: 1024
Types: 4971 Attributes: 255
Users: 8 Roles: 14
Booleans: 338 Cond. Expr.: 386
Allow: 112752 Neverallow: 0
Auditallow: 166 Dontaudit: 10365
Type_trans: 252815 Type_change: 87
Type_member: 35 Range_trans: 5781
Role_allow: 38 Role_trans: 421
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: 34
Genfscon: 107 Portcon: 646
Netifcon: 0 Nodecon: 0

```

Рис. 5: Статистика по политике

- Определили тип файлов и поддиректорий, находящихся в директории /var/www. Тип файлов, находящихся в директории /var/www/html, определить

не удалось, т.к. директория не соержжит файлов. Кроме того, определили круг пользователей, которым разрешено создание файлов в данной директории. Оказалось, что только суперпользователь имеет такое право. (Рис. [-@fig:006]).



```
root@localhost:~  
Файл Правка Вид Поиск Терминал Справка  
[root@localhost ~]# ls -lZ /var/www/  
итого 0  
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 ноя 12 07  
:58 cgi-bin  
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 ноя 12 07  
:58 html  
[root@localhost ~]# ls -lZ /var/www/html/  
итого 0  
[root@localhost ~]#
```

Рис. 6: Определяем типы

7. Создали от имени суперпользователя html-файл. (Рис. [-@fig:007]).



```
root@localhost:/var/www/html  
Файл Правка Вид Поиск Терминал Справка  
GNU nano 2.9.8 test.html  
  
<html>  
<body> test </body>  
</html>  
  
[ Wrote 3 lines ]  
^G Помощь ^O Записать ^W Поиск ^K Вырезать ^J Вывернуть ^C ТекПозиц  
^X Выход ^R ЧитФайл ^_ Замена ^U Отмен. выр Т Словарь ^_ К строке
```

Рис. 7: Создание html-файла

8. Проверили контекст созданного файла и обратились к файлу через веб-

сервер (Рис. [-@fig:008], -@fig:009]).



Рис. 8: Проверка контекста файла



Рис. 9: Обращение через веб-сервер

9. Изучили справку `man httpd_selinux` и сопоставили их с типом файла `test`. (Рис. [-@fig:010]).

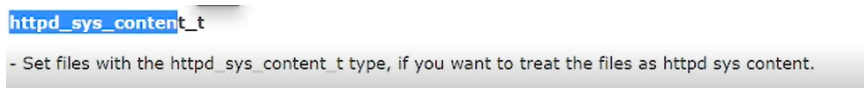
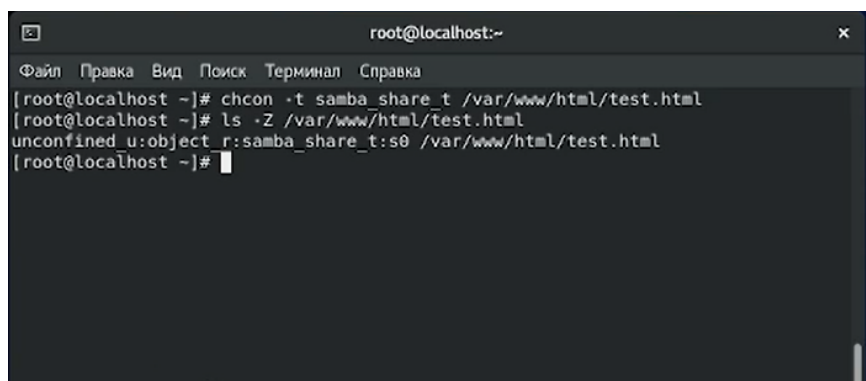


Рис. 10: Тип файла

10. Изменили контекст файла на `samba_share_t` и проверили, что контекст поменялся. После этого в браузере получили сообщение об ошибке. Это произошло, поскольку SELinux запретил доступ к файлу (Рис. [-@fig:011], [-@fig:012]).



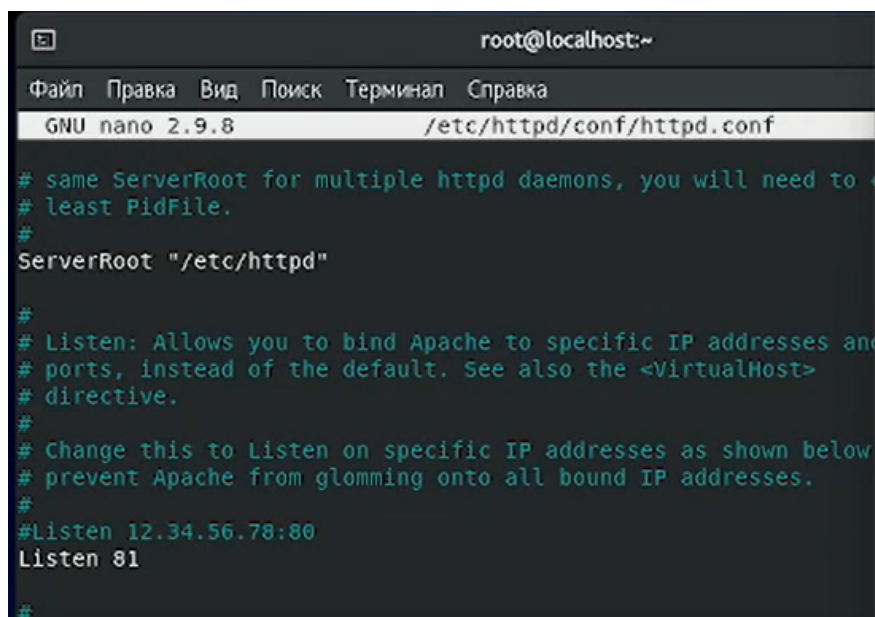
```
root@localhost:~  
Файл Правка Вид Поиск Терминал Справка  
[root@localhost ~]# chcon -t samba_share_t /var/www/html/test.html  
[root@localhost ~]# ls -Z /var/www/html/test.html  
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html  
[root@localhost ~]#
```

Рис. 11: Изменение контекста



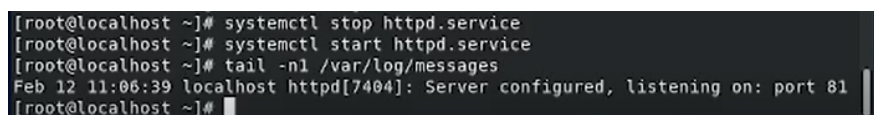
Рис. 12: Повторный просмотр файла в браузере

11. Запустили веб-сервер Apache на прослушивание TCP-порта 81. После чего перезапустили веб-сервер и проанализировали log-файлы. Также проверили список портов (Рис. [-@fig:013], [-@fig:014]).



```
root@localhost:~  
Файл Правка Вид Поиск Терминал Справка  
GNU nano 2.9.8 /etc/httpd/conf/httpd.conf  
  
# same ServerRoot for multiple httpd daemons, you will need to  
# least PidFile.  
#  
ServerRoot "/etc/httpd"  
  
#  
# Listen: Allows you to bind Apache to specific IP addresses and  
# ports, instead of the default. See also the <VirtualHost>  
# directive.  
#  
# Change this to Listen on specific IP addresses as shown below  
# prevent Apache from glomming onto all bound IP addresses.  
#  
#Listen 12.34.56.78:80  
Listen 81  
#
```

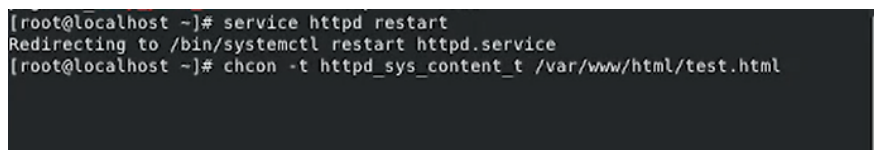
Рис. 13: Запуск веб-сервера



```
[root@localhost ~]# systemctl stop httpd.service  
[root@localhost ~]# systemctl start httpd.service  
[root@localhost ~]# tail -n1 /var/log/messages  
Feb 12 11:06:39 localhost httpd[7404]: Server configured, listening on: port 81  
[root@localhost ~]#
```

Рис. 14: Анализ лог-файлов

12. Снова запустили веб-сервер Apache и вернули контекст `httpd_sys_content_t` к файлу, а затем попробовали получить доступ к файлу через браузер. В результате увидели содержимое файла (Рис. [-@fig:015], [-@fig:016]).



```
[root@localhost ~]# service httpd restart  
Redirecting to /bin/systemctl restart httpd.service  
[root@localhost ~]# chcon -t httpd_sys_content_t /var/www/html/test.html
```

Рис. 15: Запуск веб-сервера

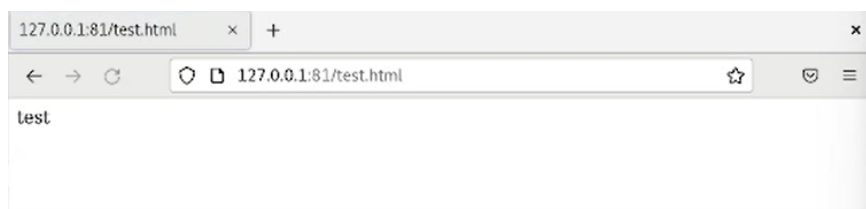


Рис. 16: Доступ к файлу

13. Исправили обратно конфигурационный файл apache, вернув Listen 80 и попытались удалить привязку http_port_t к 81 порту. Вылезла ошибка, поскольку порт 81 определен на уровне политики. После этого удалили html-файл (Рис. [-@fig:017], [-@fig:018]).

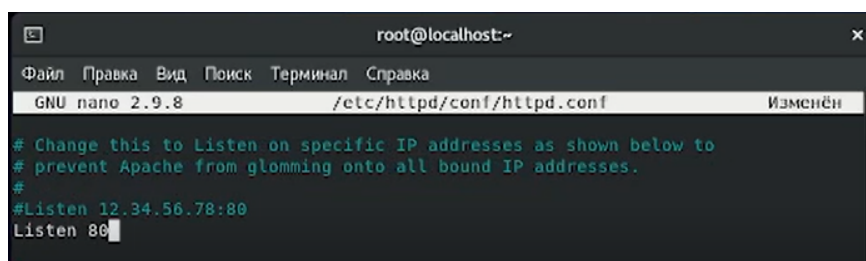


Рис. 17: Исправление конфигурационного файла

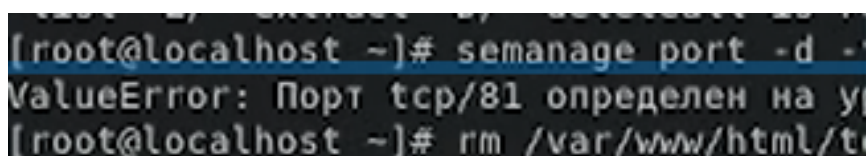


Рис. 18: Удаление привязки и файла html

Выводы

В результате выполнения данной работы была изучена технология SELinux, а также проверена работа SELinux с веб-сервером Apache.