Цель работы

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

Задание

Выполнить задания лабораторной работы и проанализировать полученные результаты.

Теоретическое введение

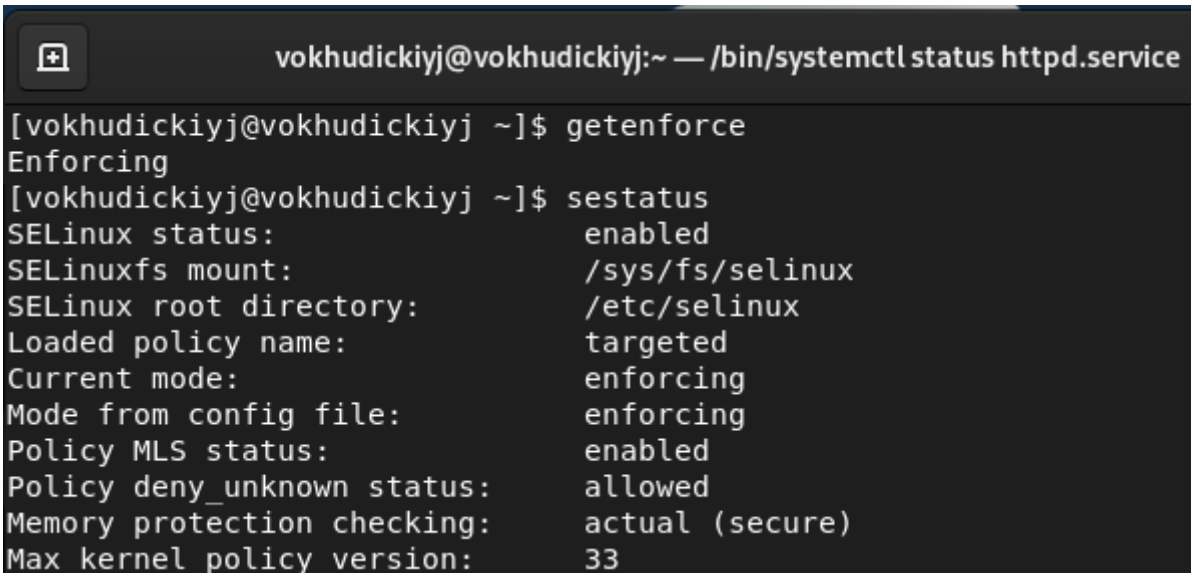
Мандатное управление доступом (англ. Mandatory access control, MAC) — разграничение доступа субъектов к объектам, основанное на назначении метки конфиденциальности для информации, содержащейся в объектах, и выдаче официальных разрешений (допуска) субъектам на обращение к информации такого уровня конфиденциальности. Также иногда переводится как Принудительный контроль доступа. Это способ, сочетающий защиту и ограничение прав, применяемый по отношению к компьютерным процессам, данным и системным устройствам и предназначенный для предотвращения их нежелательного использования.

Мандатная модель управления доступом, помимо дискреционной и ролевой, является основой реализации разграничительной политики доступа к ресурсам при защите информации ограниченного доступа. При этом данная модель доступа практически не используется «в чистом виде», обычно на практике она дополняется элементами других моделей доступа. Для файловых систем оно может расширять или заменять дискреционный контроль доступа и концепцию пользователей и групп. Самое важное достоинство заключается в том, что пользователь не может полностью управлять доступом к ресурсам, которые он создаёт.

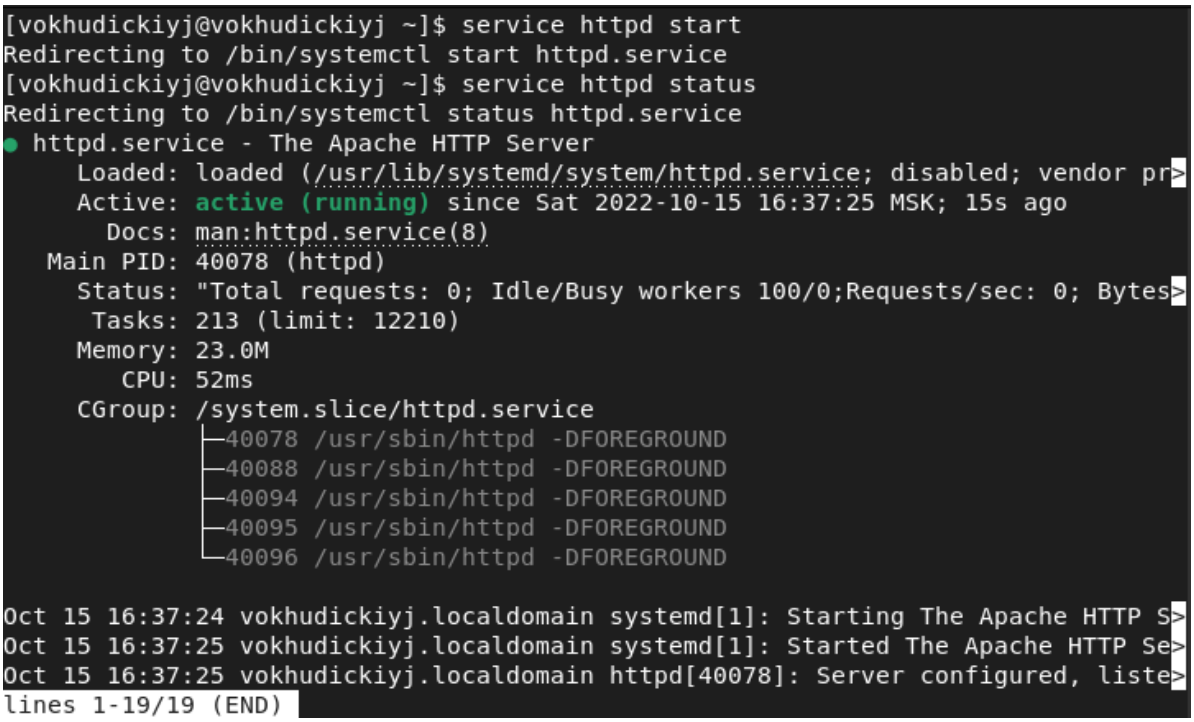
Политика безопасности системы, установленная администратором, полностью определяет доступ, и обычно пользователю не разрешается устанавливать более свободный доступ к своим ресурсам, чем тот, который установлен администратором пользователю. Системы с дискреционным контролем доступа разрешают пользователям полностью определять доступность своих ресурсов, что означает, что они могут случайно или преднамеренно передать доступ неавторизованным пользователям. Такая система запрещает пользователю или процессу, обладающему определённым уровнем доверия, получать доступ к информации, процессам или устройствам более защищённого уровня. Тем самым обеспечивается изоляция пользователей и процессов, как известных, так и не известных системе (неизвестная программа должна быть максимально лишена доверия, и её доступ к устройствам и файлам должен ограничиваться сильнее).

Выполнение лабораторной работы

Вошёл в систему с полученными учётными данными и убедился, что SELinux работает в режиме enforcing политики targeted с помощью команд *getenforce* и *sestatus*:



Обратился с помощью браузера к веб-серверу, запущенному на компьютере, и убедился, что последний работает командой *service httpd status*. Он не работал, поэтому пришлось запустить его так же, но с параметром командой *service httpd start*:



Нашёл веб-сервер Apache в списке процессов и определил его контекст безопасности, используя команду *ps auxZ | grep httpd*:

```
[vokhudickiyj@vokhudickiyj ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0    root       40078  0.0  0.5  20248 11640 ?
 Ss   16:37   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache     40088  0.0  0.3  21572  7376 ?
 S    16:37   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache     40094  0.0  0.5 1079376 11064 ?
 Sl   16:37   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache     40095  0.0  0.6 1210512 13112 ?
 Sl   16:37   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache     40096  0.0  0.5 1079376 11064 ?
 Sl   16:37   0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 vokhudi+ 40337 0.0  0.1 22
1668 2264 pts/0 S+ 16:40   0:00 grep --color=auto httpd
[vokhudickiyj@vokhudickiyj ~]$
```

Посмотрел текущее состояние переключателей SELinux для Apache с помощью команды *sestatus -b | grep httpd*:

```
[vokhudickiyj@vokhudickiyj ~]$ 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
httpd_can_network_memcache                  off
httpd_can_network_relay                     off
httpd_can_sendmail                          off
httpd_dbus_avahi                            off
```

Посмотрел статистику по политике с помощью команды seinfo:

```
vokhudickiyj@vokhudickiyj:~
  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
```

Определил тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды *ls -lZ /var/www*.

Определил тип файлов, находящихся в директории /var/www/html с помощью команды *ls -lZ /var/www/html*.

Определил круг пользователей, которым разрешено создание файлов в директории /var/www/html. Разрешено только владельцу.

```
[vokhudickiyj@vokhudickiyj ~]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 May 16 15
:10 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0     6 May 16 15
:10 html
[vokhudickiyj@vokhudickiyj ~]$ ls -lZ /var/www/html
total 0
```

Создал от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания:

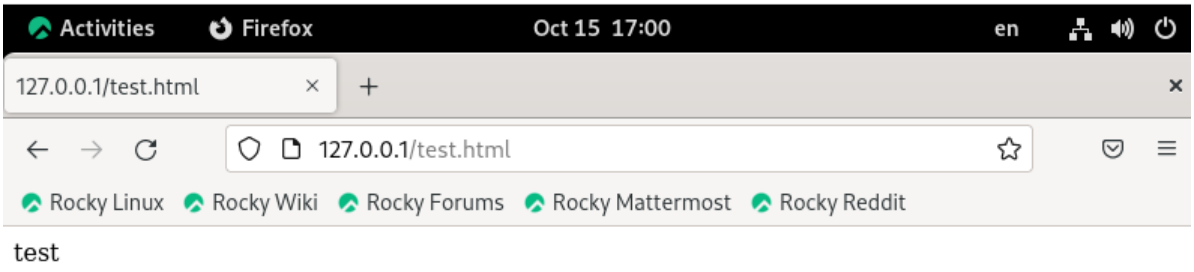
```
<html>
<body>test</body>
</html>
```

```
[vokhudickiyj@vokhudickiyj ~]$ su
Password:
[root@vokhudickiyj vokhudickiyj]# touch /var/www/html/test.html
[root@vokhudickiyj vokhudickiyj]# vim /var/www/html/test.html
```

Проверил контекст созданного мной файла:

```
[root@vokhudickiyj vokhudickiyj]# ps auxZ | grep test.html
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 40722 0.0  0.1 221800
 2260 pts/0 S+ 16:58   0:00 grep --color=auto test.html
[root@vokhudickiyj vokhudickiyj]#
```

Обратился к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Убедился, что файл был успешно отображён:



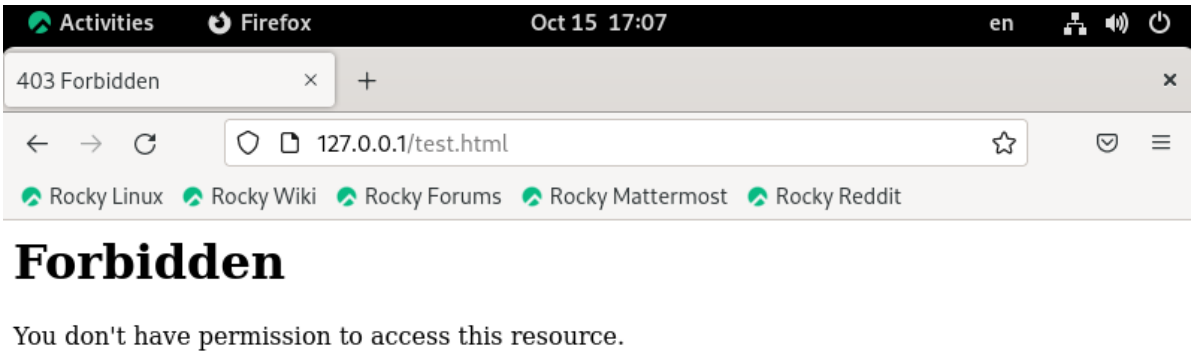
Изучил справку man httpd_selinux и выяснил, какие контексты файлов определены для httpd. Сопоставил их с типом файла test.html. Проверил контекст файла командой *ls -Z /var/www/html/test.html*:

```
[root@vokhudickiyj vokhudickiyj]# ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
```

Изменил контекст файла /var/www/html/test.html с httpd_sys_content_t на samba_share_t командами *chcon -t samba_share_t /var/www/html/test.html* и *ls -Z /var/www/html/test.html*. После этого проверил, что контекст поменялся:

```
[root@vokhudickiyj vokhudickiyj]# chcon -t samba_share_t /var/www/html/test.html
[root@vokhudickiyj vokhudickiyj]# ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[root@vokhudickiyj vokhudickiyj]#
```

Попробовал ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Получил сообщение об ошибке:



Forbidden

You don't have permission to access this resource.

Проверил, действительно ли права доступа позволяют читать этот файл любому пользователю, с помощью команды *ls -l /var/www/html/test.html*. Просмотрел системный лог-файл командой *tail /var/log/messages*:

```
[root@vokhudickiyj vokhudickiyj]# ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 34 Oct 15 16:57 /var/www/html/test.html
[root@vokhudickiyj vokhudickiyj]# tail /var/log/messages
Oct 15 17:07:32 vokhudickiyj setroubleshoot[41473]: failed to retrieve rpm info
for /var/www/html/test.html
Oct 15 17:07:32 vokhudickiyj setroubleshoot[41473]: SELinux is preventing /usr/s
bin/httpd from getattr access on the file /var/www/html/test.html. For complete
SELinux messages run: sealert -l 2973acf8-3c81-4c6c-88df-0e3cce75d783
Oct 15 17:07:32 vokhudickiyj setroubleshoot[41473]: SELinux is preventing /usr/s
bin/httpd from getattr access on the file /var/www/html/test.html.#012#012*****
 Plugin restorecon (92.2 confidence) suggests   **********************#012#012
If you want to fix the label. #012/var/www/html/test.html default label should b
e httpd_sys_content_t.#012Then you can run restorecon. The access attempt may ha
ve been stopped due to insufficient permissions to access a parent directory in
which case try to change the following command accordingly.#012Do#012# /sbin/res
torecon -v /var/www/html/test.html#012#012*****  Plugin public_content (7.83 con
fidence) suggests   ********************#012#012If you want to treat test.html a
s public content#012Then you need to change the label on test.html to public_con
tent_t or public_content_rw_t.#012Do#012# semanage fcontext -a -t public_content
_t '/var/www/html/test.html'#012# restorecon -v '/var/www/html/test.html'#012#01
2*****  Plugin catchall (1.41 confidence) suggests   **************************#
012#012If you believe that httpd should be allowed getattr access on the test.ht
ml file by default.#012Then you should report this as a bug.#012You can generate
 a local policy module to allow this access.#012Do#012allow this access for now
by executing:#012# ausearch -c 'httpd' --raw | audit2allow -M my-httpd#012# semo
dule -X 300 -i my-httpd.pp#012
```

Увидел ошибки, аналогичные указанным выше, в файле /var/log/audit/audit.log:

```
[root@vokhudickiyj vokhudickiyj]# tail /var/log/audit/audit.log
type=SERVICE_START msg=audit(1665842841.757:270): pid=1 uid=0 auid=4294967295 se
s=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=dbus-:1.10-org.fedorapro
ject.Setroubleshootd@0 comm="systemd" exe="/usr/lib/systemd/systemd" hostname=?
addr=? terminal=? res=success'UID="root" AUID="unset"
type=SERVICE_START msg=audit(1665842849.873:271): pid=1 uid=0 auid=4294967295 se
s=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=dbus-:1.10-org.fedorapro
ject.SetroubleshootPrivileged@0 comm="systemd" exe="/usr/lib/systemd/systemd" ho
stname=? addr=? terminal=? res=success'UID="root" AUID="unset"
type=AVC msg=audit(1665842851.577:272): avc:  denied  { getattr } for  pid=40096
 comm="httpd" path="/var/www/html/test.html" dev="dm-0" ino=816488 scontext=syst
em_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=
file permissive=0
type=SYSCALL msg=audit(1665842851.577:272): arch=c000003e syscall=262 success=no
 exit=-13 a0=ffffff9c a1=7f31c0041d18 a2=7f31bd7f9830 a3=0 items=0 ppid=40078 pi
d=40096 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 f
sgid=48 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/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="apache" FSUID="apache" EGID="apache"
SGID="apache" FSGID="apache"
type=PROCTITLE msg=audit(1665842851.577:272): proctitle=2F7573722F7362696E2F6874
747064002D44464F524547524F554E44
type=AVC msg=audit(1665842851.577:273): avc:  denied  { getattr } for  pid=40096
```

Попробовал запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в /etc/services). Для этого в файле /etc/httpd/httpd.conf нашёл строчку Listen 80 и заменил её на Listen 81. Выполнил перезапуск веб-сервера Apache. Сбоя не произошло.

```
[root@vokhudickiyj vokhudickiyj]# vim /etc/httpd/httpd.conf
[root@vokhudickiyj vokhudickiyj]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@vokhudickiyj vokhudickiyj]# 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 Sat 2022-10-15 16:37:25 MSK; 42min ago
       Docs: man:httpd.service(8)
   Main PID: 40078 (httpd)
     Status: "Total requests: 4; Idle/Busy workers 100/0;Requests/sec: 0.00157;>
      Tasks: 213 (limit: 12210)
     Memory: 24.2M
        CPU: 1.002s
     CGroup: /system.slice/httpd.service
             ├─40078 /usr/sbin/httpd -DFOREGROUND
             ├─40088 /usr/sbin/httpd -DFOREGROUND
             ├─40094 /usr/sbin/httpd -DFOREGROUND
             ├─40095 /usr/sbin/httpd -DFOREGROUND
             └─40096 /usr/sbin/httpd -DFOREGROUND

Oct 15 16:37:24 vokhudickiyj.localdomain systemd[1]: Starting The Apache HTTP S>
Oct 15 16:37:25 vokhudickiyj.localdomain systemd[1]: Started The Apache HTTP Se>
```

Проанализировал лог-файлы /var/log/http/error_log, /var/log/http/access_log и /var/log/audit/audit.log. Так как сбоя не было, ни в каких файлах не появились записи:

```
[root@vokhudickiyj vokhudickiyj]# cat /var/log/http/error_log
cat: /var/log/http/error_log: No such file or directory
[root@vokhudickiyj vokhudickiyj]# cat /var/log/http/access_log
cat: /var/log/http/access_log: No such file or directory
```

Выполнил команду *semanage port -a -t http_port_t -p tcp 81*. Порт 81 уже был определён. После этого проверил список портов командой *semanage port -l | grep http_port_t*. Убедился, что порт 81 в списке:

```
[root@vokhudickiyj vokhudickiyj]# semanage port -a -t http_port_t -p tcp 81
ValueError: Port tcp/81 already defined
[root@vokhudickiyj vokhudickiyj]# 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
```

Вернул контекст httpd_sys_content__t к файлу /var/www/html/ test.html командой *chcon -t httpd_sys_content_t /var/www/html/test.html*.
После этого попробовал получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html. Я увидел содержимое файла — слово «test».

```
[root@vokhudickiyj vokhudickiyj]# chcon -t httpd_sys_content_t /var/www/html/tes
t.html
[root@vokhudickiyj vokhudickiyj]#
```

Исправил обратно конфигурационный файл apache, вернув Listen 80. Попытался удалить привязку http_port_t к 81 порту командой *semanage port -d -t http_port_t -p tcp 81*. Совершить удаление не удалось:

```
[root@vokhudickiyj vokhudickiyj]# semanage port -d -t http_port_t -p tcp 81
ValueError: Port tcp/81 is defined in policy, cannot be deleted
```

Удалил файл /var/www/html/test.html командой *rm /var/www/html/test.html*:

```
[root@vokhudickiyj vokhudickiyj]# rm /var/www/html/test.html
rm: remove regular file '/var/www/html/test.html'? y
[root@vokhudickiyj vokhudickiyj]#
```

Выводы

В ходе выполнения работы я развил навыки администрирования ОС Linux, получил первое практическое знакомство с технологией SELinux, проверил работу SELinx на практике совместно с веб-сервером Apache.

Список литературы

- Кулябов Д. С., Королькова А. В., Геворкян М. Н Лабораторная работа №6. Мандатное разграничение прав в Linux