Лабораторная работа №6

Мандатное разграничение прав в Linux

Аскеров Александр Эдуардович

Содержание

Список иллюстраций

1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux.

Проверить работу SELinx на практике совместно с веб-сервером Apache.

2 Выполнение лабораторной работы

Войдём в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus.

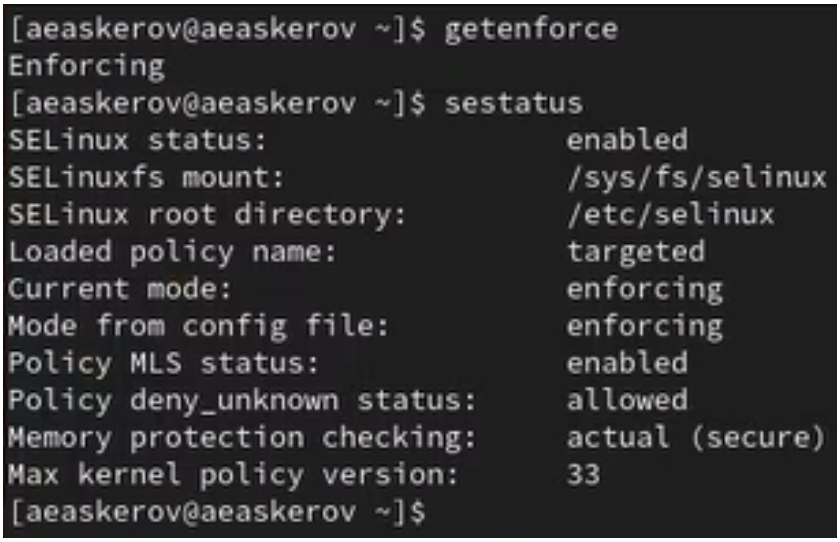

Рис. 2.1: Проверка SELinux

Обратимся с помощью браузера к веб-серверу, запущенному на компьютере, и убедимся, что последний работает.

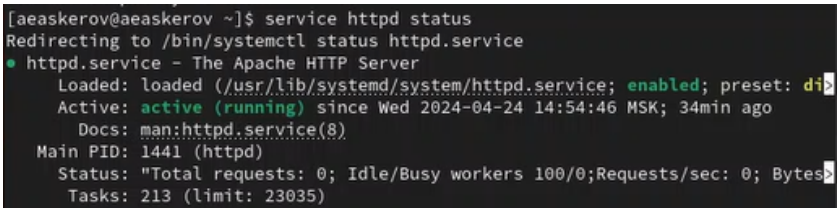

Рис. 2.2: Проверка работы веб-сервера

Найдём веб-сервер Apache в списке процессов, определим его контекст безопасности.

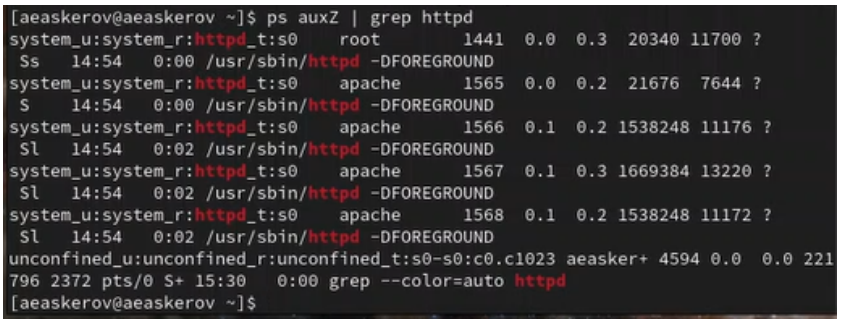

Рис. 2.3: Контекст безопасности веб-сервера

Посмотрим текущее состояние переключателей SELinux для Apache.

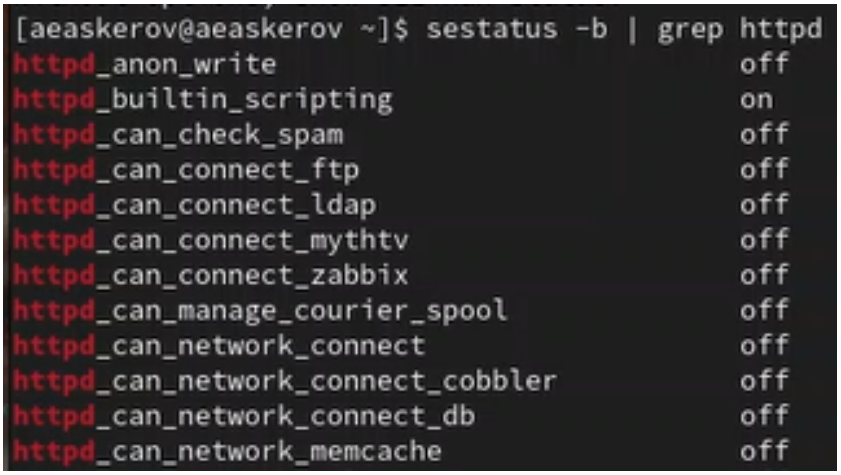

Рис. 2.4: Переключатели SELinux

Посмотрим статистику по политике с помощью команды seinfo, также определим множество пользователей, ролей, типов.

```
[aeaskerov@aeaskerov ~]$ seinfo
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:             65409    Neverallow:            0
  Auditallow:          172    Dontaudit:          8647
  Type_trans:       267813    Type_change:          94
  Type_member:          37    Range_trans:        6164
  Role allow:           39    Role_trans:          419
  Constraints:          70    Validatetrans:         0
  MLS Constrain:        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
  Netifcon:              0    Nodecon:               0
[aeaskerov@aeaskerov ~]$
```

Рис. 2.5: Статистика по политике

Определим тип файлов и поддиректорий, находящихся в директории /var/www.

```
[aeaskerov@aeaskerov ~]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Oct 28 12
:35 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0     6 Oct 28 12
:35 html
[aeaskerov@aeaskerov ~]$
```

Рис. 2.6: Тип файлов и поддиректорий

Определим тип файлов, находящихся в директории /var/www/html.

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

Рис. 2.7: Тип файлов

Определим круг пользователей, которым разрешено создание файлов в директории /var/www/html.

```
[aeaskerov@aeaskerov ~]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Oct 28 12
:35 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0     6 Oct 28 12
:35 html
[aeaskerov@aeaskerov ~]$
```

Рис. 2.8: Круг пользователей с разрешением

Создадим от имени суперпользователя html-файл /var/www/html/test.html следующего содержания.

```
1 <html>
2 <body>test</body>
3 </html>
```

Рис. 2.9: html-файл

Проверим контекст созданного файла. Запишем контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html.

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

Рис. 2.10: Контекст файла

Обратимся к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html.

Изучим справку man httpd_selinux.

Изменим 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.

```
[root@aeaskerov html]# chcon -t samba_share_t /var/www/html/test.html
[root@aeaskerov html]#
```

Рис. 2.11: Изменение контекста

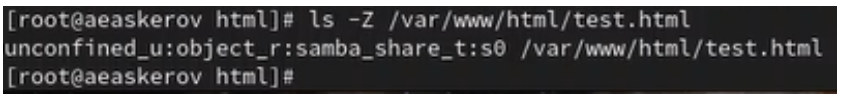

Рис. 2.12: Проверка контекста

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

Проанализируем ситуацию. Просмотрим log-файлы веб-сервера Apache. Также просмотрим системный лог-файл.

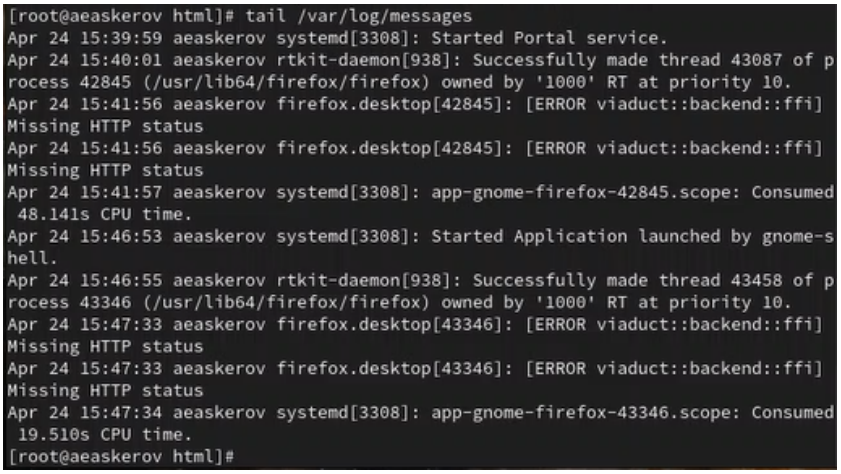

Рис. 2.13: Системный лог-файл

Посмотрим файл /var/log/audit/audit.log.

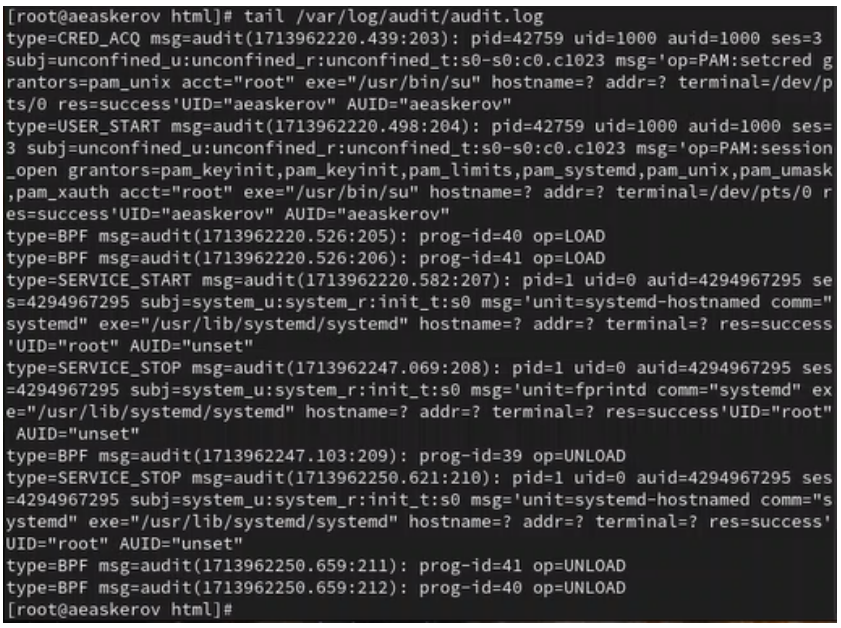

Рис. 2.14: Файл audit.log

Попробуем 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 найдём строчку Listen 80 и заменим её на Listen 81.

Рис. 2.15: Изменение порта

Выполним перезапуск веб-сервера Apache. Произошёл сбой из-за изменения порта.

Проанализируем лог-файлы.

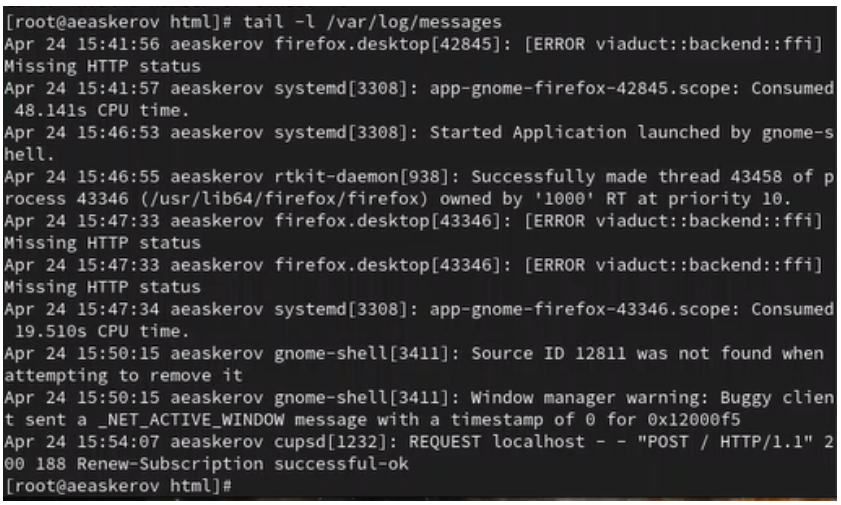

Рис. 2.16: Файл messages

Просмотрим файлы /var/log/http/error_log, /var/log/http/access_log и /var/log/audit/audit.log и выясним, в каких файлах появились записи.

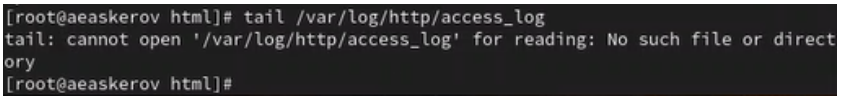

Рис. 2.17: Файл access_log

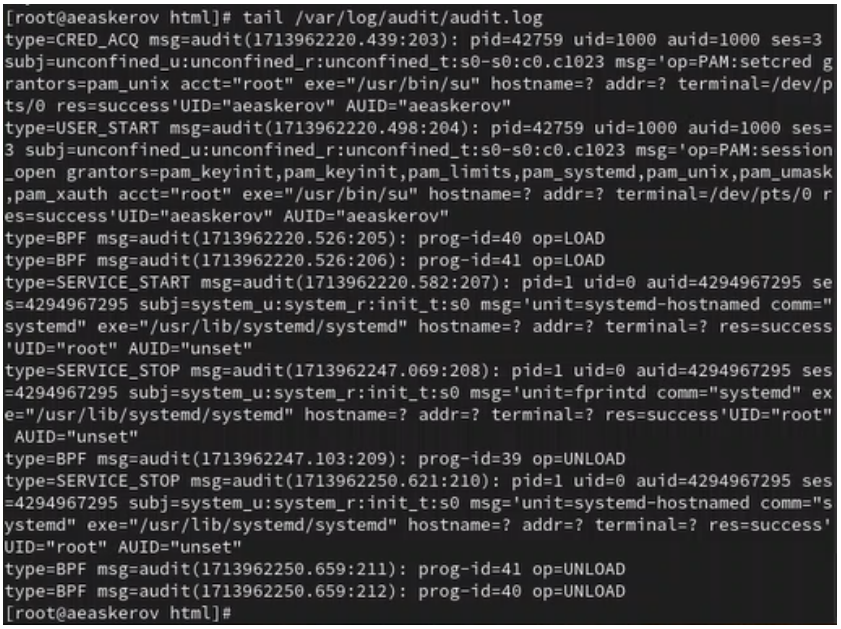

Рис. 2.18: Файл audit.log

Выполним следующую команду.

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

Рис. 2.19: Управление портами

```
[root@aeaskerov html]# 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
[root@aeaskerov html]#
```

Рис. 2.20: Управление портами

Порт 81 появился в списке.

Попробуем запустить веб-сервер Apache ещё раз. Теперь он запустился.

Вернём контекст httpd_sys_content__t к файлу /var/www/html/test.html.

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

Рис. 2.21: Возврат контекста

После этого попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html.

Мы видим содержимое файла – слово «test».

Исправим обратно конфигурационный файл apache, вернув Listen 80.

```
3 Listen 80
```

Рис. 2.22: Изменение порта

Удалим привязку http_port_t к 81-му порту.

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

Рис. 2.23: Удаление привязки

Удалим файл /var/www/html/test.html.

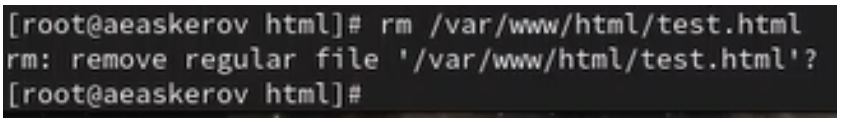

Рис. 2.24: Удаление файла

3 Выводы

Развиты навыки администрирования ОС Linux. Получено первое практическое знакомство с технологией SELinux.

Проверена работа SELinx на практике совместно с веб-сервером Apache.

Список литературы

1. Веб-сервер Apache