Отчёт по лабораторной работе №6

Дисциплина: Основы информационной безопасности

Полиенко Анастасия Николаевна, НПМбд-01-19

Содержание

Список иллюстраций

1 Цель работы

Получить практические навыки адмирирования в ОС Linux и ознакомиться с технологией SELinux совместно с веб-сервером Apache.

2 Теоретическое введение

SELinux, или Security Enhanced Linux, — это продвинутый механизм управления доступом, разработанный Агентством национальной безопасности (АНБ) США для предотвращения злонамеренных вторжений. Он реализует мандатную модель управления доступом (MAC — Mandatory Access control) в дополнение к уже существующей в Linux дискреционной модели (DAC — Discretionary Access Control), то есть разрешениям на чтение, запись, выполнение.

У SELinux есть три режима работы:

- Enforcing — ограничение доступа в соответствии с политикой. Запрещено все, что не разрешено в явном виде. Режим по умолчанию.
- Permissive — ведёт лог действий, нарушающих политику, которые в режиме enforcing были бы запрещены, но не запрещает сами действия.
- Disabled — полное отключение SELinux.

В основе структуры безопасности SELinux лежат политики. Политика — это набор правил, определяющих ограничения и права доступа для всего, что есть в системе. Под “всем” в данном случае понимаются пользователи, роли, процессы и файлы. Политика определяет связь этих категорий друг с другом. |

Более подробно см. в [1].

Apache — это свободное программное обеспечение, с помощью которого можно создать веб-сервер. Несмотря на то, что Apache чаще всего называют сервером (более того, его официальное название — Apache HTTP Server) — это всё-таки программа, которую устанавливают на сервер, чтобы добиться определённых результатов.

Для чего нужен Apache сервер:

- чтобы открывать динамические PHP-страницы,
- для распределения поступающей на сервер нагрузки,
- для обеспечения отказоустойчивости сервера,
- чтобы потренироваться в настройке сервера и запуске PHP-скриптов.

Apache является кроссплатформенным ПО и поддерживает такие операционные системы, как Linux, BSD, MacOS, Microsoft, BeOS и другие.

Более подробно см. в [2].

3 Выполнение лабораторной работы

С помощью команды getenforce убеждаемся, что SELinux работает в режиме enforcing, а с помощью команды sestatus устанавливаем политику targeted (рис. 3.1).

```
[anpolienko@anpolienko ~]$ getenforce
Enforcing
[anpolienko@anpolienko ~]$ sestatus targeted
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
[anpolienko@anpolienko ~]$
```

Рис. 3.1: getenforce и sestatus

Убеждаемся, что сервер работает с помощью команды service httpd status (рис. 3.2).

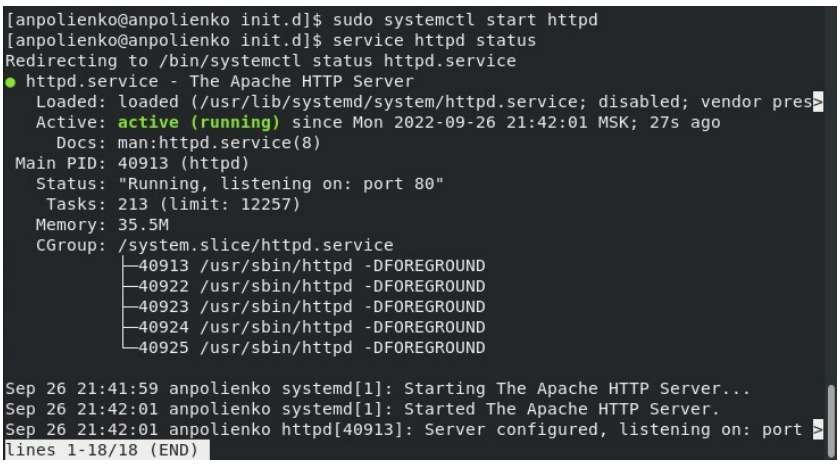

Рис. 3.2: Работающий сервер

С помощью команды ps -eZ находим, что контекст безопасности Apache — httpd_t (рис. 3.3).

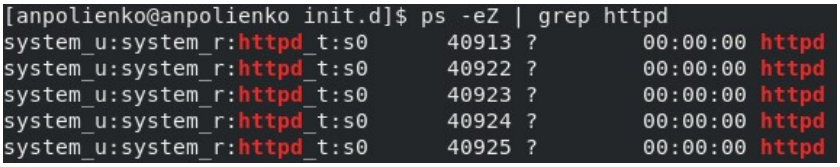

Рис. 3.3: Контекст безопасности Apache

Смотрим текущее состояние переключателей командой sestatus -b httpd (рис. 3.4).

```
[anpolienko@anpolienko init.d]$ 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
authlogin_radius                            off
authlogin_yubikey                           off
```

Рис. 3.4: Состояние переключателей

Смотрим статистику по политике командой seinfo. Узнаём, что множество пользователей — 8, ролей — 14, типов — 4989 (рис. 3.5).

```
[anpolienko@anpolienko init.d]$ seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:             31 (MLS enabled)
Target Policy:              selinux
Handle unknown classes:     allow
  Classes:             132    Permissions:         464
  Sensitivities:         1    Categories:         1024
  Types:              4989    Attributes:          255
  Users:                 8    Roles:                14
  Booleans:            339    Cond. Expr.:         387
  Allow:            113819    Neverallow:            0
  Auditallow:          168    Dontaudit:         10402
  Type_trans:       255386    Type_change:          87
  Type_member:          35    Range_trans:        5784
  Role allow:           38    Role_trans:          422
  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. 3.5: Статистика seinfo

Определяем тип файлов и круг пользователей с правой на создание и поддиректорий в директориях /var/www и /var/www/html командой ls -lZ (рис. 3.6).

```
[anpolienko@anpolienko init.d]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Jun 22 17:18
 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0     6 Jun 22 17:18
 html
[anpolienko@anpolienko init.d]$ ls -lZ /var/www/html
total 0
[anpolienko@anpolienko init.d]$ ls -l /var/www/html
total 0
[anpolienko@anpolienko init.d]$ ls -l /var/www
total 0
drwxr-xr-x. 2 root root 6 Jun 22 17:18 cgi-bin
drwxr-xr-x. 2 root root 6 Jun 22 17:18 html
[anpolienko@anpolienko init.d]$
```

Рис. 3.6: Данные директорий /var/www и /var/www/html

От имени суперпользователя создаём файл /var/www/html/test.html (рис. 3.7).

```
[anpolienko@anpolienko init.d]$ su -
Password:
[root@anpolienko ~]# touch /var/www/html/test.html
[root@anpolienko ~]# nano /var/www/html/test.html
[root@anpolienko ~]# cat /var/www/html/test.html
<html>
<body>test</body>
</html>
```

Рис. 3.7: Файл test.html

Командой matchpathcon узнаём контекст файла test.html и директории /var/www/html — это httpd_sys_content_t (рис. [-#fig:008]).

```
[root@anpolienko ~]# matchpathcon /var/www/html/test.html
/var/www/html/test.html system_u:object_r:httpd_sys_content_t:s0
[root@anpolienko ~]# matchpathcon -V /var/www/html.
/var/www/html. error: No such file or directory
[root@anpolienko ~]# matchpathcon -V /var/www/html
/var/www/html verified.
[root@anpolienko ~]# matchpathcon  /var/www/html
/var/www/html   system_u:object_r:httpd_sys_content_t:s0
[root@anpolienko ~]#
```

Рис. 3.8: Констекст файла test.html

Обращаемся к файлу через ссылку в веб-браузере. Контент отображён корректно (рис. 3.9).

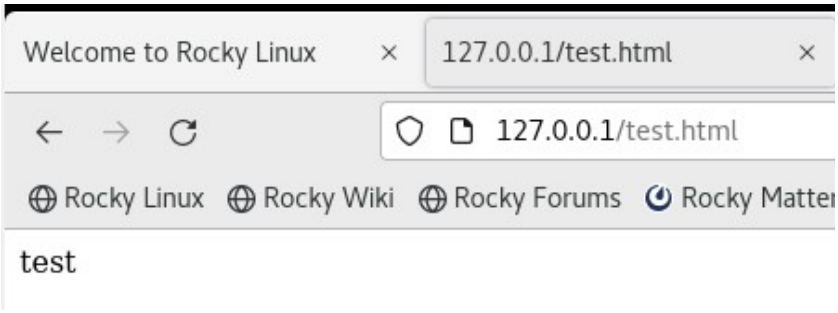

Рис. 3.9: Просмотр файла в веб-браузере

Изучая справку man httpd_selunix узнаём, что для httpd определены следующие контексты: httpd_sys_content_t, httpd_sys_script_exec_t, httpd_sys_script_ro_t, httpd_sys_script_rw_t, httpd_sys_script_ra_t, httpd_unconfined_script_exec_t. Меняем контекст файла test.html командой chcon -t (рис. 3.10).

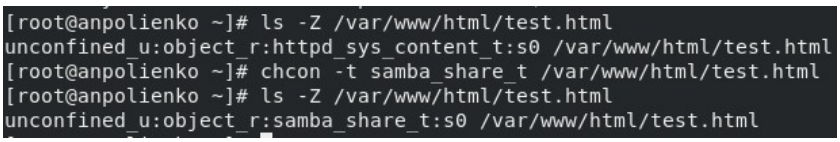

Рис. 3.10: Смена контекста

При повторной попытке открыть файл через веб-браузер получаем ошибку доступа (рис. 3.11).

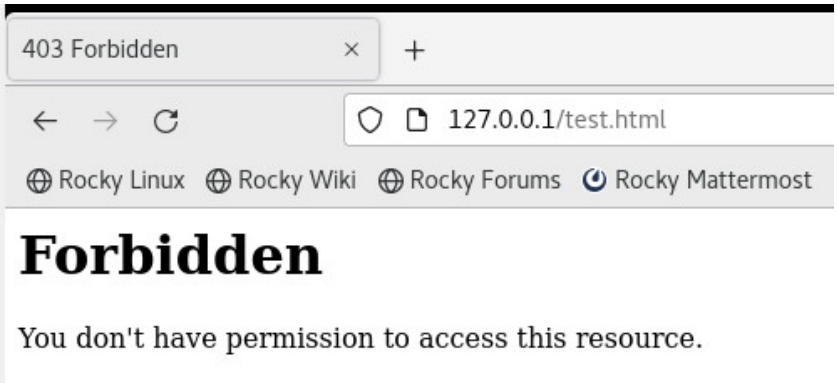

Рис. 3.11: Ошибка доступа

Убеждаемся, что файл доступен для чтения всем пользователям командой ls -l. Далее смотрим log-файлы веб-сервера Apache командой tail, где показаны ошибки (рис. 3.12).

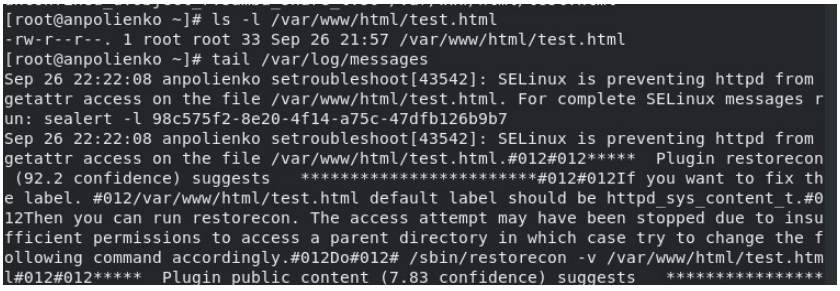

Рис. 3.12: Ошибки в log-файлах

Устанавливаем веб-сервер Apache на прослушивание TCP-порта 81, изменяя строку Listen в файле /etc/httpd/conf/httpd.conf (рис. 3.13).

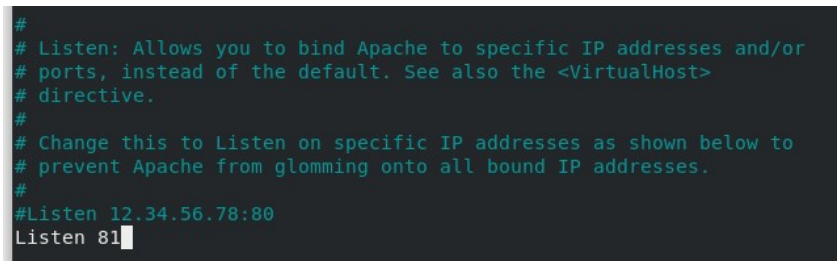

Рис. 3.13: Прослушивание 81 порта

Перезапускаем сервер и смотри данные log-файлов веб-сервера Apache (рис. 3.14).

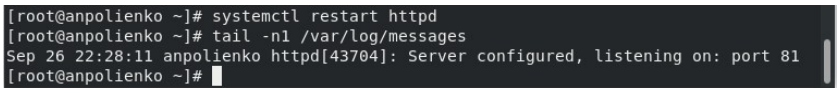

Рис. 3.14: Перезапуск сервера

Устанавлием для веб-сервера Apache порт TCP-81 и проверяем его наличие в списке портов командой semanage (рис. 3.15).

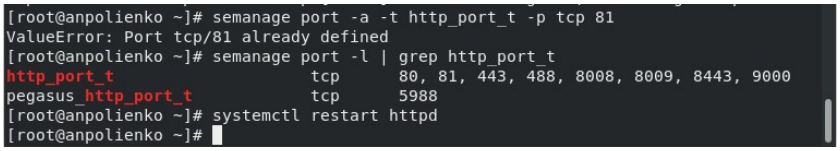

Рис. 3.15: Установка порта

Возвращаем файлу test.html контекст httpd_sys_content_t и снова успешно просматриваем страницу в веб-браузере (рис. 3.16).

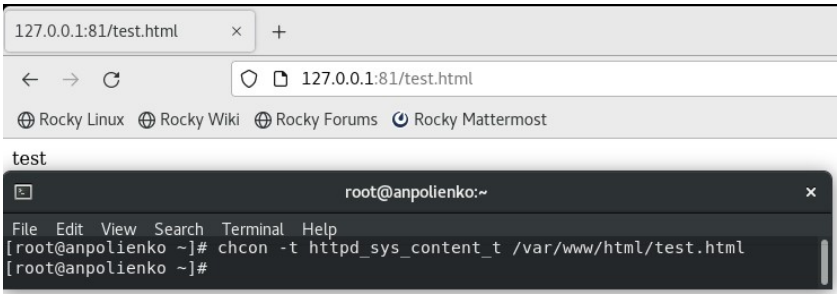

Рис. 3.16: Повторный просмотр файла в веб-браузере

Возвращаем в конфигурационный файл прослушивание порта 80 и удаляем порт 81 из списка портов (рис. 3.17).

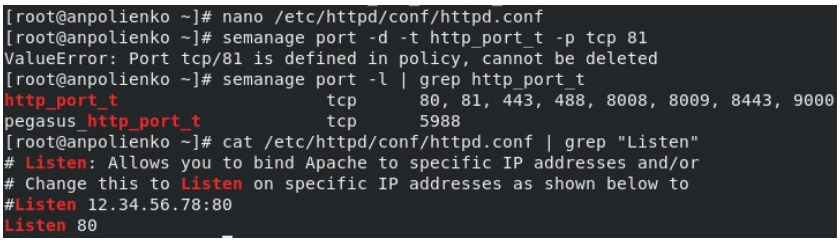

Рис. 3.17: Удаление порта

Удаляем файл test.html (рис. 3.18).

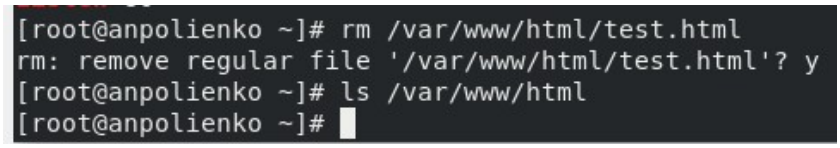

Рис. 3.18: Удаление файла

4 Выводы

Я получила основные навыки администрирования в ОС Linux и проверила работу SELinux на практике совместно с веб-сервером Apache.

Список литературы

1. What is SELinux [Электронный ресурс]. ITProffi, 2021. URL: https://itproffi.ru/chto-takoe-selinux-nastrojka-vklyuchenie-i-otklyuchenie/.
2. What us Apache [Электронный ресурс]. 2domains, 2021. URL: https://2domains.ru/support/vps-i-servery/shto-takoye-apache.