

Лабораторная работа №6

Дискреционное разграничение прав в Linux. Основные атрибуты

Выполнила: Накова Амина Михайловна, НПИбд-02-23

Цель работы

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

Выполнение работы

1 . Для проведения указанной лабораторной работы на одно рабочее место требуется компьютер с установленной операционной системой Linux, поддерживающей технологию SELinux. Предполагается использовать стандартный дистрибутив Linux CentOS с включённой политикой SELinux targeted и режимом enforcing. Для выполнения заданий требуется наличие учётной записи администратора (root) и учётной записи обычного пользователя. Постоянно работать от учётной записи root неправильно с точки зрения безопасности /home/guest/dir1/file1 командой ([рис. @fig-001]).

```
[amina@localhost ~]$ sudo systemctl start httpd
[amina@localhost ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[amina@localhost ~]$ sestatus -bigrep httpd
sestatus: invalid option -- 'i'

Usage: sestatus [OPTION]

  -v  Verbose check of process and file contexts.
  -b  Display current state of booleans.

Without options, show SELinux status.
[amina@localhost ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
```

Figure 1: рис.1.1

2. Войдите в систему с полученными учётными данными и убедитесь, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus. Обратитесь с помощью браузера к вебсерверу, запущенному на вашем компьютере, и убедитесь, что последний работает: service httpd status или /etc/rc.d/init.d/httpd status. Если не работает, запустите его так же, но с параметром start ([рис. @fig-002]).

```
without options, show SELinux status.
[amina@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; enabled; preset: disabled)
   Active: active (running) since Fri 2025-09-12 21:25:01 MSK; 1min 40s ago
     Docs: man:httpd.service(8)
  Main PID: 37428 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0;Uptime: 0:00:01"
   Tasks: 177 (limit: 22876)
  Memory: 33.6M
    CPU: 576ms
  CGroup: /system.slice/httpd.service
          └─37428 /usr/sbin/httpd -DFOREGROUND
            └─37429 /usr/sbin/httpd -DFOREGROUND
              └─37430 /usr/sbin/httpd -DFOREGROUND
                └─37431 /usr/sbin/httpd -DFOREGROUND
                  └─37432 /usr/sbin/httpd -DFOREGROUND
```

Figure 2: рис.2.1

3. Найдите веб-сервер Apache в списке процессов, определите его контекст безопасности и занесите эту информацию в отчёт. Например, можно использовать команду `ps auxZ | grep httpd` или `ps -eZ | grep httpd`. Посмотрите текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -bigrep httpd`. Обратите внимание, что многие из них находятся в положении «off». (рис. @fig-003).

```
└─37428 /usr/sbin/httpd -DFOREGROUND
  └─37429 /usr/sbin/httpd -DFOREGROUND
    └─37430 /usr/sbin/httpd -DFOREGROUND
      └─37431 /usr/sbin/httpd -DFOREGROUND
        └─37432 /usr/sbin/httpd -DFOREGROUND

сен 12 21:25:01 localhost.localdomain systemd[1]: Starting The Apache HTTP Server: httpd.
сен 12 21:25:01 localhost.localdomain httpd[37428]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 instead. Please set the 'ServerName' directive globally to suppress this message
сен 12 21:25:01 localhost.localdomain httpd[37428]: Server configured, listening on *
сен 12 21:25:01 localhost.localdomain systemd[1]: Started The Apache HTTP Server: httpd.
ESC0D
Main PID: 37428 (httpd)
Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0;Uptime: 0:00:01"
Tasks: 177 (limit: 22876)
Memory: 33.6M
CPU: 576ms
CGroup: /system.slice/httpd.service
```

Figure 3: рис.3.1

4. Посмотрите статистику по политике с помощью команды `seinfo`, также определите множество пользователей, ролей, типов. Определите тип файлов и поддиректорий, находящихся в директории `/var/www`, с помощью команды `ls -lZ /var/www`. Определите тип файлов, находящихся в директории `/var/www/html`: `ls -lZ /var/www/html`. Определите круг пользователей, которым разрешено создание файлов в директории `/var/www/html` (рис. @fig-004).

Выводы: В ходе лабораторной работы было получено практическое знакомство с технологией SELinux. Были изучены

```

-v Verbose check of process and file contexts.
-b Display current state of booleans.

Without options, show SELinux status.
[amina@localhost ~]$ ls -LZ /var/www
system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
system_u:object_r:httpd_sys_content_t:s0 html
[amina@localhost ~]$ ls -LZ /var/www/html
[amina@localhost ~]$ cd /var/www/html
[amina@localhost html]$ nano test.html
[amina@localhost html]$ sudo nano test.html
[sudo] пароль для амина:
[amina@localhost html]$ http://127.0.0.1/test.html.
-bash: http://127.0.0.1/test.html.: Нет такого файла или каталога
[amina@localhost html]$ http://127.0.0.1/test.html
-bash: http://127.0.0.1/test.html: Нет такого файла или каталога
[amina@localhost html]$

```

Figure 4: рис.4.1

основные механизмы мандатного разграничения доступа в Linux, работа с контекстами безопасности, портами и политиками. Исследование показало, что: SELinux обеспечивает дополнительный уровень безопасности поверх стандартных прав доступа. Контексты безопасности определяют, какие процессы могут обращаться к каким ресурсам. Изменение контекста файла может заблокировать доступ к нему, даже если стандартные права доступа разрешают чтение. Для работы служб на нестандартных портах необходимо явно добавлять эти порты в политику SELinux. Работа позволила получить практические навыки администрирования SELinux и понимание важности мандатного контроля доступа в современных операционных системах.