Работа с SELinux и веб-сервером Apache

LAB 6

Чилеше Лупупа

Содержание

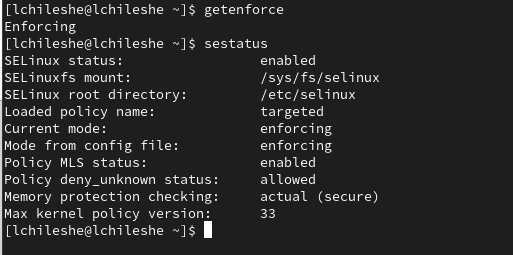
# 1 Цель работы

Освоить практическую работу с механизмами безопасности SELinux в режиме политики enforcing, изучить работу веб-сервера Apache в условиях SELinux, провести манипуляции с контекстами безопасности и портами, определить влияние SELinux на доступность файлов и служб.

## 1.1 1. Проверка режима работы SELinux

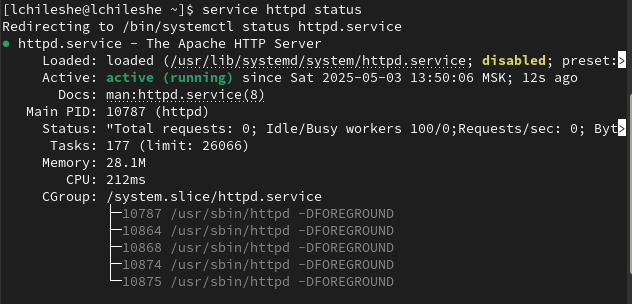
getenforce sestatus

**Описание:** Убедились, что SELinux работает в режиме enforcing с политикой targeted.



## 1.2 2. Проверка статуса веб-сервера Apache

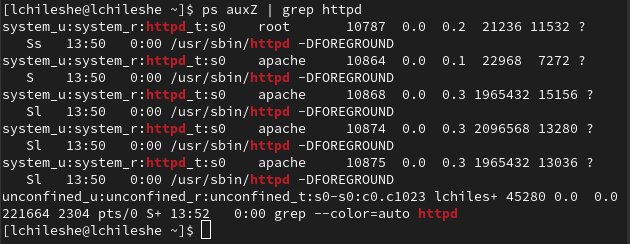
service httpd status или /etc/rc.d/init.d/httpd status



## 1.3 3. Поиск процесса Apache и его контекста SELinux

ps auxZ | grep httpd

**Результат:** Контекст процесса: system\_u:system\_r:httpd\_t:s0

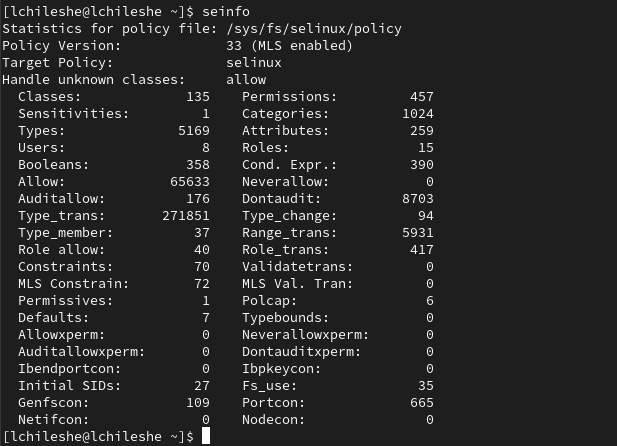


## 1.4 4. Просмотр переключателей SELinux для Apache

sestatus -b | grep httpd

## 1.5 5. Информация о политике SELinux

seinfo



**Результат:** Получены сведения о пользователях, ролях, типах, например:

* Пользователи: unconfined\_u, system\_u
* Роли: object\_r, system\_r
* Типы: httpd\_t, httpd\_sys\_content\_t

## 1.6 6–7. Типы файлов в /var/www и /var/www/html

ls -lZ /var/www ls -lZ /var/www/html

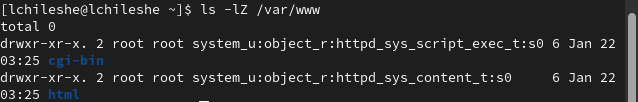


fig:

## 1.7 8. Проверка прав пользователей на создание файлов

Вывод команды ls -ldZ /var/www/html

fig:

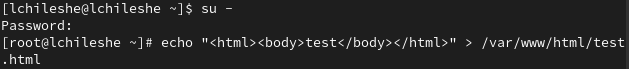
Анализ: Только root имеет права записи. Пользователи без расширенных политик не могут записывать файлы без смены контекста.

## 1.8 9. Создание HTML-файла

echo “

test

” > /var/www/html/test.html



## 1.9 10. Проверка контекста файла

ls -Z /var/www/html/test.html

Результат: unconfined\_u:object\_r:httpd\_sys\_content\_t:s0

fig:

## 1.10 11. Проверка доступа к файлу через браузер

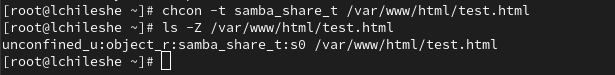
URL: http://127.0.0.1/test.html Результат: Файл успешно отображён.

## 1.11 12. Анализ справки man httpd\_selinux

man httpd\_selinux

**Результат:** Тип httpd\_sys\_content\_t разрешает доступ для httpd. Контекст файла соответствует этому типу, поэтому доступ был разрешён.

## 1.12 13. Смена типа контекста на samba\_share\_t

chcon -t samba\_share\_t /var/www/html/test.html ls -Z /var/www/html/test.html 

## 1.13 14. Попытка открыть файл в браузере

Результат: Ошибка: 403 Forbidden ##15. Анализ проблемы

**Причина:** SELinux запрещает httpd доступ к файлам с типом samba\_share\_t.

**Команды:**

ls -l /var/www/html/test.html tail /var/log/messages tail /var/log/audit/audit.log

**Вывод:** Запись в audit.log подтверждает блокировку доступа.

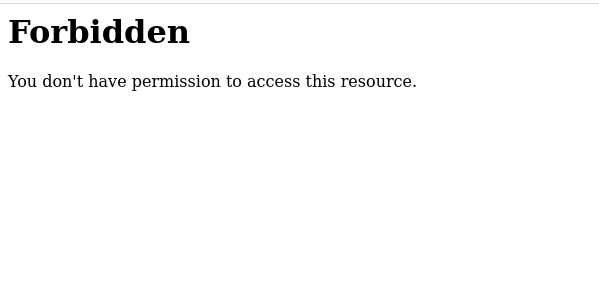
## 1.14 16. Изменение порта Apache

Файл конфигурации:

/etc/httpd/conf/httpd.conf

Изменение: Listen 80 → Listen 81 ## 17. Перезапуск Apache

service httpd restart



Результат: Ошибка запуска — SELinux блокирует прослушивание несогласованного порта. ## 18. Анализ логов

tail -n 50 /var/log/messages tail -n 50 /var/log/httpd/error\_log tail -n 50 /var/log/audit/audit.log

## 1.15 19. Разрешение порта 81

semanage port -a -t http\_port\_t -p tcp 81 semanage port -l | grep http\_port\_t

**Результат:** Порт 81 добавлен в список разрешённых.

## 1.16 20. Перезапуск Apache снова

service httpd restart

**Результат:** Успешный запуск, т.к. SELinux теперь разрешает httpd слушать порт 81. ## 21. Возврат контекста файла

chcon -t httpd\_sys\_content\_t /var/www/html/test.html

Проверка: URL: http://127.0.0.1:81/test.html Результат: Файл успешно отображён. 22–23. Возврат конфигурации и удаление порта

# В httpd.conf Listen 80

semanage port -d -t http\_port\_t -p tcp 81 semanage port -l | grep http\_port\_t

## 1.17 24. Удаление файла

rm /var/www/html/test.html

# 2 Выводы

В ходе лабораторной работы был изучен механизм SELinux и его взаимодействие с веб-сервером Apache. Были выполнены практические действия по изменению контекстов безопасности, настройке доступа к файлам и смене портов. Установлено, что SELinux эффективно ограничивает доступ к ресурсам, даже при наличии стандартных прав доступа, и требует явной настройки безопасности для каждого компонента. Работа позволила глубже понять внутренние механизмы безопасности Linux.