Работа с SELinux и веб-сервером Apache

LAB 6

Чилеше Лупупа

Содержание

Список иллюстраций

Список таблиц

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

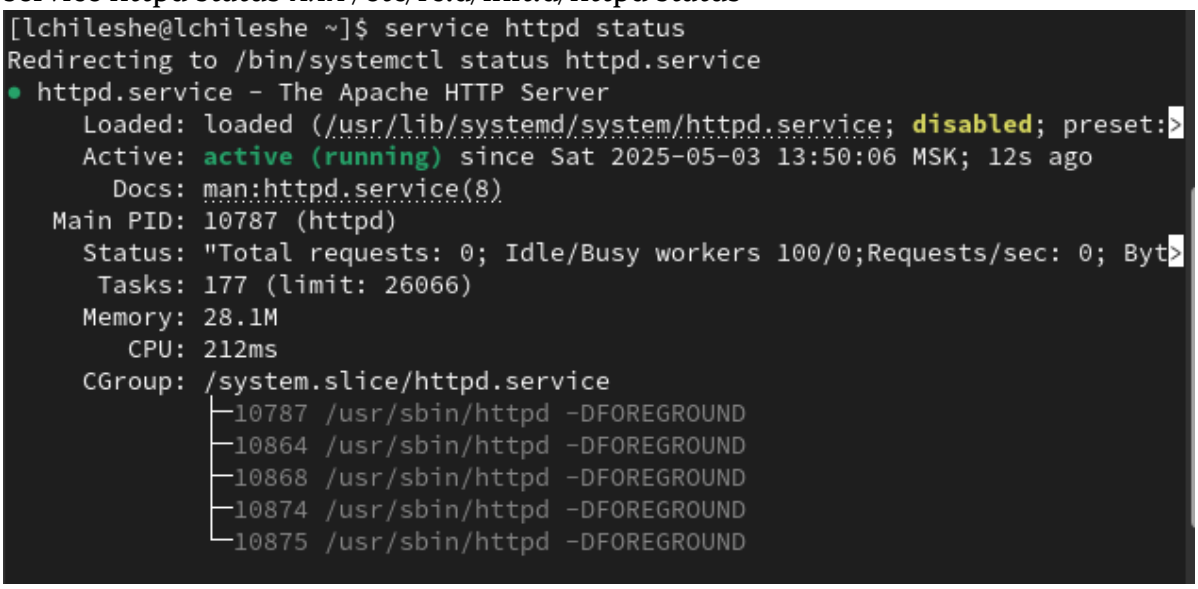
getenforce sestatus

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

```
[lchileshe@lchileshe ~]$ getenforce
Enforcing
[lchileshe@lchileshe ~]$ sestatus
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
[lchileshe@lchileshe ~]$
```

1.2 2. Проверка статуса веб-сервера Apache

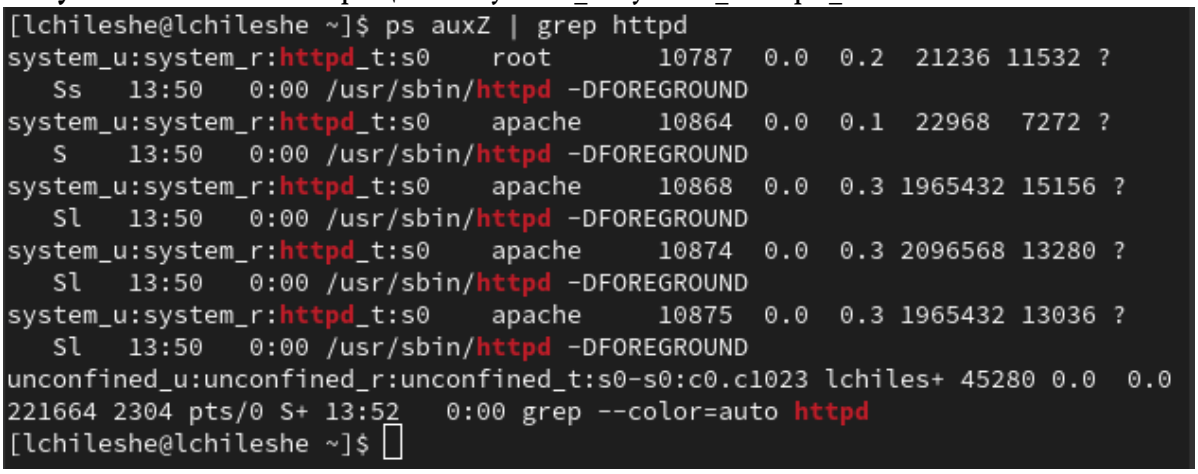
service httpd status *или* /etc/rc.d/init.d/httpd status



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

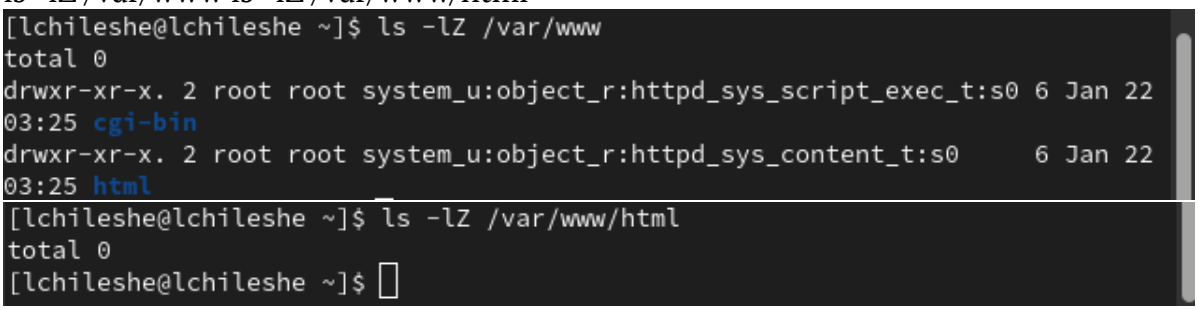
```
[lchileshe@lchileshe ~]$ seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:             33 (MLS enabled)
Target Policy:              selinux
Handle unknown classes:     allow
  Classes:             135    Permissions:         457
  Sensitivities:         1    Categories:         1024
  Types:              5169    Attributes:          259
  Users:                 8    Roles:                15
  Booleans:            358    Cond. Expr.:         390
  Allow:             65633    Neverallow:            0
  Auditallow:          176    Dontaudit:          8703
  Type_trans:       271851    Type_change:          94
  Type_member:          37    Range_trans:        5931
  Role allow:           40    Role_trans:          417
  Constraints:          70    Validatetrans:         0
  MLS Constrain:        72    MLS Val. Tran:         0
  Permissives:           1    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
[lchileshe@lchileshe ~]$
```

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



1.7 8. Проверка прав пользователей на создание файлов

Вывод команды ls -ldZ /var/www/html

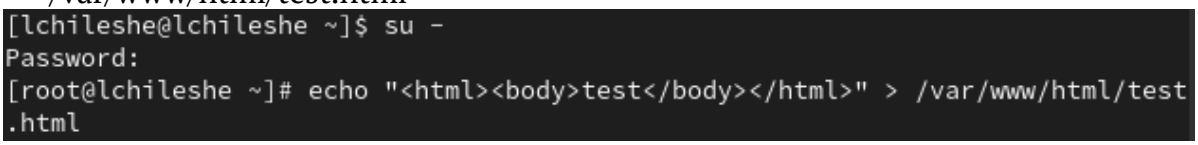
Анализ: Только root имеет права записи. Пользователи без расширенных политик не могут записывать файлы без смены контекста.

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

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

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

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

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

B 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.