Основы информационной безопасности

Лабораторная работа № 6. Мандатное разграничение прав в Linux

Сунгурова М.

Информация

Докладчик

- Сунгурова Мариян Мухсиновна
- НКНбд-01-21
- Российский университет дружбы народов

Цель работы

Цель работы

Целью данной работы является приобретение практических навыков администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinx на практике совместно с веб-сервером Apache.

Теоретические сведения

SELinux (SELinux) — это система принудительного контроля доступа, реализованная на уровне ядра. Впервые эта система появилась в четвертой версии CentOS, а в 5 и 6 версии реализация была существенно дополнена и улучшена. Эти улучшения позволили SELinux стать универсальной системой, способной эффективно решать массу актуальных задач. Стоит помнить, что классическая система прав Unix применяется первой, и управление перейдет к SELinux только в том случае, если эта первичная проверка будет успешно пройдена.

Домен – список действий, которые может выполнять процесс. Обычно в качестве домена определяется минимально-возможный набор действий, при помощи которых процесс способен функционировать. Таким образом, если процесс дискредитирован, злоумышленнику не удастся нанести большого вреда.

Роль – список доменов, которые могут быть применены. Если какого-то домена нет в списке доменов какой-то роли, то действия из этого домена не могут быть применены.

Тип – набор действий, которые допустимы по отношения к объекту. Тип отличается от домена тем, что он может применяться к пайпам, каталогам и файлам, в то время как домен применяется к процессам.

Контекст безопасности – все атрибуты SELinux — роли, типы и домены.

Выполнение лабораторной работы

Выполнение лабораторной работы

В конфигурационном файле /etc/httpd/httpd.conf зададим параметр ServerName. Также необходимо проследить, чтобы пакетный фильтр был отключён или в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола tcp. Отключим фильтр командами(рис. (fig:001?))

```
[mmsungurova@mmsungurova httpd]$ sudo nano /etc/httpd/httpd.conf
[mmsungurova@mmsungurova httpd]$ iptables -F
iptables v1.8.10 (nf_tables): Could not fetch rule set generation id: Permission denied (you must be root)
[mmsungurova@mmsungurova httpd]$ sudo iptables -F
[mmsungurova@mmsungurova httpd]$ sudo iptables -P INPUT ACCEPT iptables -P OUTPUT ACCEPT
Bad argument `iptables'
Try `iptables -h' or 'iptables --help' for more information.
[mmsungurova@mmsungurova httpd]$ sudo iptables -P INPUT ACCEPT
[mmsungurova@mmsungurova httpd]$ sudo iptables -P INPUT ACCEPT iptables -P OUTPUT ACCEPT
Bad argument `iptables'
Try `iptables -h' or 'iptables --help' for more information.
[mmsungurova@mmsungurova httpd]$ sudo iptables -P OUTPUT ACCEPT
[mmsungurova@mmsungurova httpd]$ sudo iptables -I INPUT -p tcp --dport 80 -j ACCEPT
[mmsungurova@mmsungurova httpd]$ sudo iptables -I INPUT -p tcp --dport 81 -j ACCEPT
[mmsungurova@mmsungurova httpd]$ sudo iptables -I OUTPUT -p tcp --sport 80 -j ACCEPT
[mmsungurova@mmsungurova httpd]$ sudo iptables -I OUTPUT -p tcp --sport 81 -j ACCEPT
[mmsungurova@mmsungurova httpd]$
```

Рис. 1: Подготовка лабораторного стенда

Выполнение лабораторной работы

Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus(рис. (fig:002?)).

```
[mmsungurova@mmsungurova httpd]$ getenforce
Enforcing
[mmsungurova@mmsungurova httpd]$ 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
[mmsungurova@mmsungurova httpd]$ 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: inactive (dead)
       Docs: man:httpd.service(8)
[mmsungurova@mmsungurova httpd]$ service httpd start
Redirecting to /bin/systemctl start httpd.service
[mmsungurova@mmsungurova httpd]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0    root      110620  0.9  0.4  20364 11692 ?        Ss   19:23   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    110630  0.0  0.2  22096  7528 ?        S    19:23   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    110635  0.4  0.6 1112656 19796 ?       Sl   19:23   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    110636  0.3  0.4 981520 13472 ?        Sl   19:23   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    110637  0.6  0.6 981520 17648 ?        Sl   19:23   0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 mmsungu+ 110830 0.0  0.0 221688 2560 pts/0 S+ 19:23   0:00 grep --color=auto httpd
[mmsungurova@mmsungurova httpd]$ 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.
[mmsungurova@mmsungurova httpd]$ sestatus httpd
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
```

Найдите веб-сервер Apache в списке процессов, определим его контекст безопасности(рис. (fig:002?))

Мы можем видеть контекст безопасности SELinux: system_u:system_r:httpd_t.

Также просмотрим множество пользователей, ролей, типов(рис. (fig:007?)):

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

Рис. 3: Множества пользователей, ролей, типов

Определив тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды ls -lZ /var/www, увидим, что есть директория, содержащая cgi-скрипты, и директория /var/www/html, содержащая все скрипты httpd(в данный момент пустая)(рис. (fig:008?)):

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

Рис. 4: Просмотр типов директорий в /var/www

Можно увидеть, что создание файлов в директории /var/www/html разрешено только владельцу – root.

Создадим от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания(рис. (fig:009?)):

Затем посмотрим контекст безопасности, который был задан по умолчанию этому файлу((**fig:010?**)):

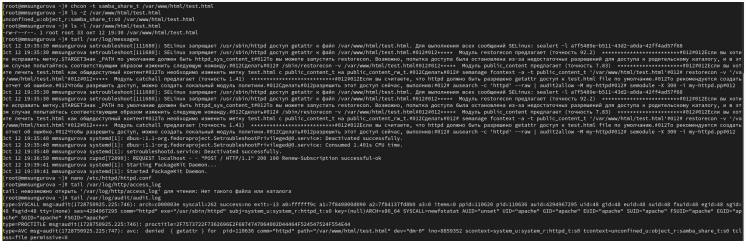

Рис. 6: Установка пароля для пользователя с правами администратора

Увидим, что файлам по умолчанию сопоставляется свободный пользователь SELinux unconfined_u, указана роль object_r используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах и тип httpd_sys_content_t, который позволяет процессу httpd получить доступ к файлу

Обратимся к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html, убедимся, что файл был успешно отображён.(рис. (**fig:011?**)):

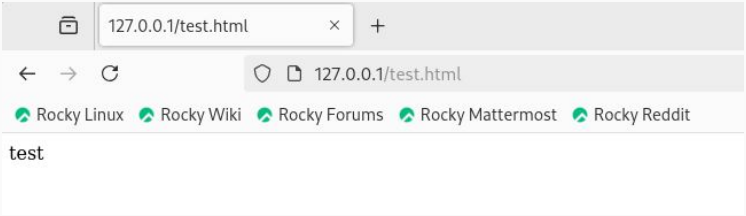

Рис. 7: Открытие html-страницы через браузер

Изучив справку man httpd_selinux, выясним, какие контексты файлов определены для httpd. Сопоставив их с типом файла test.html увидим, что его контекст httpd_sys_content_t для содержимого, которое должно быть доступно для всех скриптов httpd и для самого демона.

Изменим контекст файла /var/www/html/test.html с httpd_sys_content_t на тот, к которому процесс httpd не должен иметь доступа – samba_share_t(рис. (fig:012?)):

Рис. 8: Изменение контекста файла /var/www/html/test.html

Теперь снова попробуем получить доступ к файлу через браузер и получим отказ(рис. (fig:013?)):

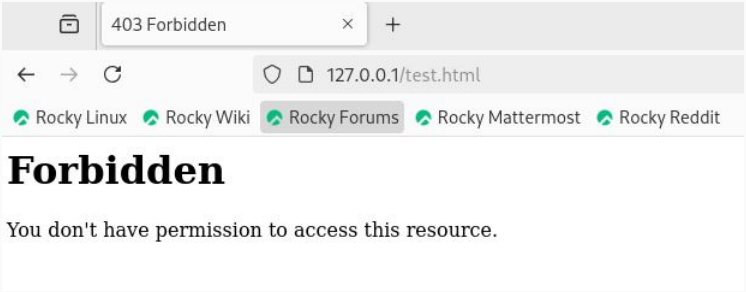

Рис. 9: Отказ в доступе к html-странице через браузер

Посмотрим log-файлы веб-сервера Apache и системный лог-файл и увидим, что отказ происходит, так как доступ запрещен SELinux именно к веб-серверу(на просто просмтр текстовых файлов это не влияет)(рис. (fig:014?)):

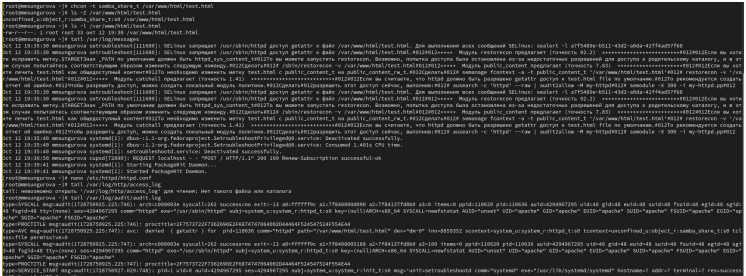

Рис. 10: Просмотр лог-файлов

Выполнение лабораторной работы

Запустим веб-сервер Apache на прослушивание TCP-порта 81. Для этого в файле /etc/httpd/httpd.conf найдем строчку Listen 80 и заменим её на Listen 81(рис. (fig:015?)):

Рис. 11: Замена прослушиваемого порта

Просмотрев лог-файлы увидим, что порт для прослушивания был сменен(рис. (fig:017?)):

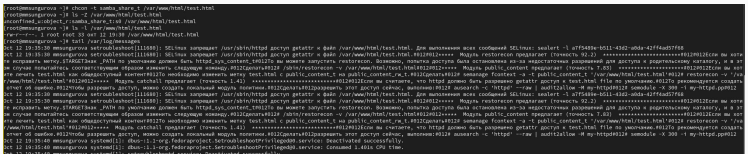

Рис. 12: Просмотр лог-файлов

Также этот порт мог быть отклчен, тогда мы бы совсем не видели страницу, добавлять порты и просматривать актуальные можно с помощью команды seamanage(рис. (fig:018?)):

Рис. 13: Просмотр портов с помощью seamnage

В конце работы вернем все сделанные изменения в файлах конфигурации веб-сервера.

```
[root@mmsungurova ~]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@mmsungurova ~]# nano /etc/httpd/httpd.conf
[root@mmsungurova ~]# semanage port -d -t http_port_t -p tcp 81
ValueError: Port tcp/81 is defined in policy, cannot be deleted
[root@mmsungurova ~]# rm /var/www/html/test.html
rm: удалить обычный файл '/var/www/html/test.html'? y
[root@mmsungurova ~]#
```

Рис. 14: Окончание работы

Выводы

Выводы

В результате выполнения работы были приобретены практические навыки администрирования ОС Linux. Получено первое практическое знакомство с технологией SELinux. Проверена работа SELinx на практике совместно с веб-сервером Apache.