Лабораторная работа № 6

Мандатное разграничение прав в Linux

Абакумов Егор Александрович

Содержание

# Цель работы

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

# Теоретическое описание

SELinux представляет собой систему маркировки, каждый процесс имеет метку. Каждый файл, каталог или даже пользователь в системе имеет метку. Даже портам и устройствам и именам хостов в системе присвоены метки. SELinux определяет правила доступа процесса к объектам с определенными метками. Это и называется политикой. За соблюдением правил следит ядро. Иногда это еще называется обязательный контроль доступа (Mandatory Access Control, MAC). [1]

В дистрибутиве Linux MAC реализована поверх того, что мы называем моделью избирательного управления доступом (Discretionary Access Control), сокращённо DAC.

DAC – это управление доступом на основе списков управления доступом, где объектами доступа служат пользователь, группа и другие. Эти объекты имеют комбинацию полномочий чтение/запись/выполнение или r/w/x. SELinux позволяет ограничить доступ к объектам пользователя, который определён именно им так, что суперпользователь не может иметь суперпривилегии по отношению ко всем объектам нашего пользователя.

Другими словами, SELinux позволяет произвести точную настройку требований контроля доступа, с помощью которой определяется степень влияния пользователя на описанный настройкой процесс. Это в полной мере может предотвратить получение злоумышленником доступа ко всем объектам и процессам системы. [2]

Владелец файла не имеет полной свободы действий над атрибутами безопасности. Стандартные атрибуты контроля доступа, такие как группа и владелец ничего не значат для SELinux. Полностью все управляется метками. Значения атрибутов могут быть установлены и без прав root, но на это нужно иметь специальные полномочия SELinux.

# Ход работы

1. Готовим стенд. Для этого устанавливаем Apache, selinux command line tools и seinfo, дальше настраиваем ServerName в конфигурации httpd, разрешаем фильтру подключаться к 81-му порту. Делаем заранее за кадром.
2. Входим в систему и проверяем режим работы selinux (иллюстр. 1). Проверяем, что демон httpd включен и работает (иллюстр. 2). Находим процессы демона в списке и выписавем контекст: system\_u:system\_r:httpd\_t:s0 (иллюстр. 3). Просматриваем переключатели selinux (иллюстр. 4).

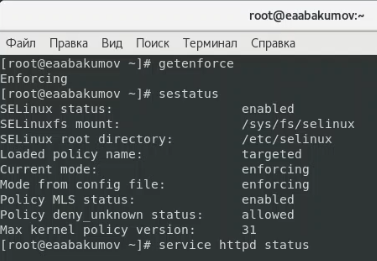


Figure 1: Режим работы selinux

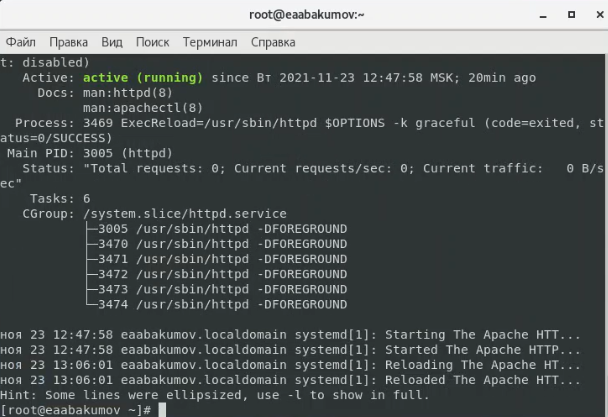


Figure 2: Проверка httpd

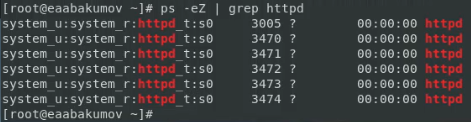


Figure 3: Процессы демона

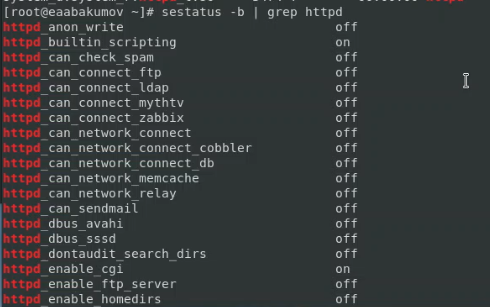


Figure 4: Переключатели selinux

1. Утилитой seinfo выводим статистику по политике, множество пользователей, ролей и типов (иллюстр. 5, 6, 7, 8).

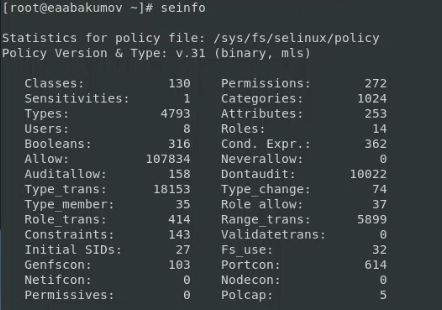


Figure 5: Статистика по политике

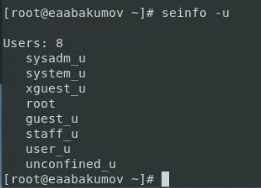


Figure 6: Множество пользователей

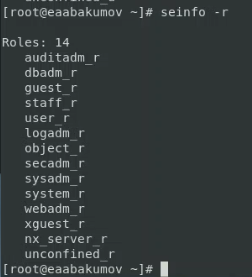


Figure 7: Множество ролей

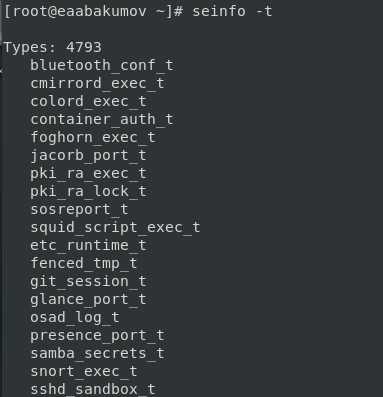


Figure 8: Множество типов

1. Проверяем тип файлов и поддиректорий, находящихся в директории /var/www. Так как в директории лежат только поддиректории, их типы: httpd\_sys\_script\_exec\_t и httpd\_sys\_content\_t. Пытаемся определить тип файлов в /var/www/html. Так как там нет файлов, определять нечего. Определяем круг пользователей, которым разрешено создание файлов в директории /var/www/html. Так как пользователь для директории установлен в system\_u, создавать файлы там может только суперпользователь (иллюстр. 9).

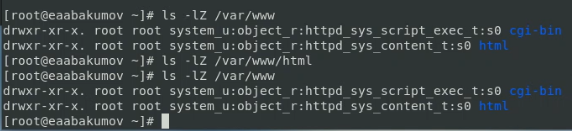


Figure 9: Проверка типов в /var/www

1. Из-под рута создаем файл в /var/www/html/ (иллюстр. 10). Проверяем контекст: unconfined\_u:object\_r:httpd\_sys\_content\_t:s0 (иллюстр. 11). Проверяем доступность файла в браузере (иллюстр. 12).

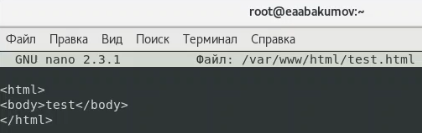


Figure 10: Файл /var/www/html/test.html

Figure 11: Контекст файла test.html

Figure 11: Контекст файла test.html

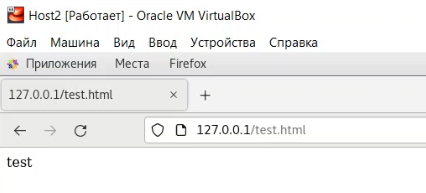


Figure 12: Проверка файла в браузере

1. Изучив контексты файлов для httpd\_selinux приходим к выводу, что контекст был выбран верно. Теперь изменим контекст (иллюстр. 13). Теперь браузер не имеет доступа к файлу, выдается ошибка (иллюстр. 14). Файл не был отображен, так как процесс httpd не имеет доступа к выбранному нами типу файла. Просматриваем логи, где говорится, что selinux запретил доступ из-за разницы в контекстах (иллюстр. 15).

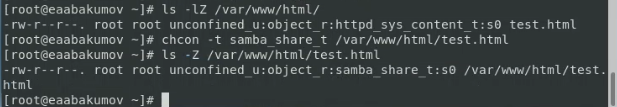


Figure 13: Изменение контекста

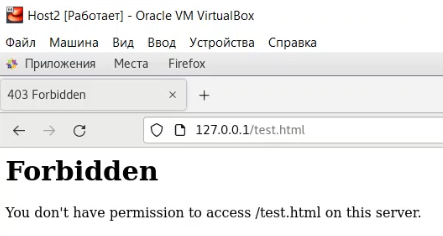


Figure 14: Ошибка доступа в браузере

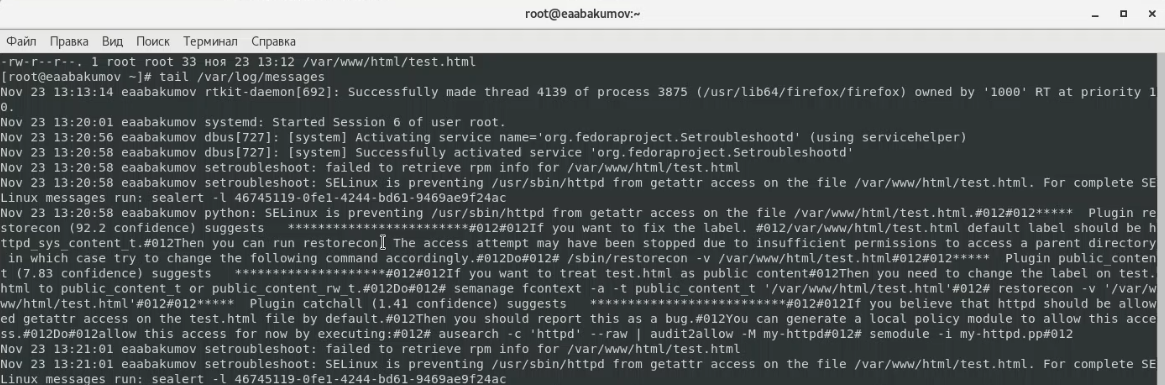


Figure 15: Логи

1. Меняем порт прослушивания httpd с 80 на 81 (иллюстр. 16). Пытаемся перезапустить демона, он перезапускается, так как 81 порт прописан по дефолту в политике. В логах никаких ошибок не видим (иллюстр. 17). Выполняем добавление порта, на что selinux отказывается это делать, посколько порт уже прописан, проверяем это в списке (иллюстр. 18). Собственно, поэтому демон и перезапустился, хотя по задумке лабораторной работы не должен был, так как процесс попытался бы получить доступ к запрещенному порту, на что selinux не дал бы разрешения, а httpd, не стерпев такого обращения с собой, отказался бы запускаться, и только прописывание порта в selinux позволило бы выполнить перезапуск. Короче говоря, возвращаем верный контекст на файл test.html (иллюстр. 19) и проверяем его через 81-ый порт браузером (иллюстр. 20). Удаляем привязку порта, selinux отвечает, что порт прописан в политике, так что удаляться он не будет. Ну, мы его туда и не добавляли, так что кто мы такие, чтобы удалять его оттуда. Удаляем файл /var/www/html/test.html.

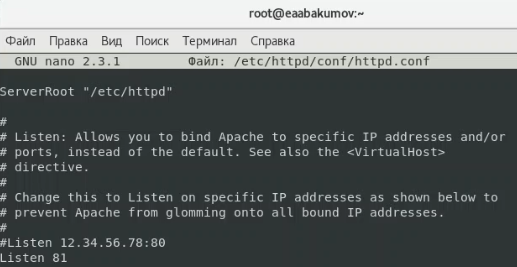


Figure 16: Замена порта прослушивания

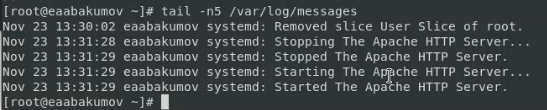


Figure 17: Отсутствие ошибок в логах

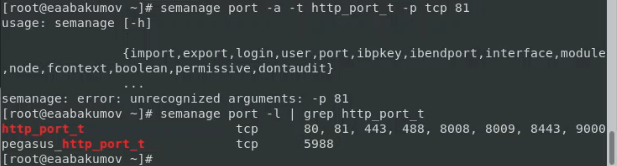


Figure 18: Список портов httpd в selinux

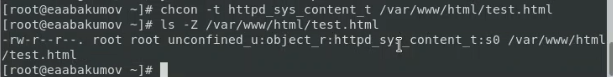


Figure 19: Возвращение верного контекста на файл test.html

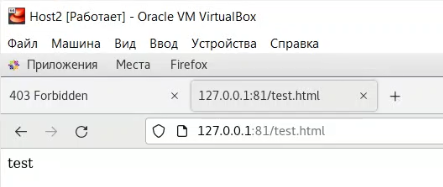


Figure 20: Проверка доступа к файлу браузером через 81-ый порт

# Выводы

В ходе работы мы успешно развили навыки администрирования ОС Linux, получили первое практическое знакомство с технологией SELinux и проверили работу SELinx на практике совместно с веб-сервером Apache.

# Список литературы

1. НАСТРОЙКА SELINUX. // Losst. 2021. URL: https://losst.ru/nastrojka-selinux (дата обращения 23.11.2021).
2. Введение в SELinux под CentOS Stream // RUVDS. 2021. URL: https://ruvds.com/ru/helpcenter/vvedenie-v-selinux-pod-centos-stream/ (дата обращения 23.11.2021).
3. seinfo (1) // OpenNET. 2021. URL: https://www.opennet.ru/man.shtml?topic=seinfo&category=1&russian=2 (дата обращения 23.11.2021).
4. Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян. Информационная безопасность компьютерных сетей: лабораторные работы. // Факультет физико-математических и естественных наук. M.: РУДН, 2015. 64 с..