Лабораторная работа №4

Дискреционное разграничение прав в Linux. Расширенные атрибуты

Парфенова Елизавета Евгеньевна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Получение практических навыков работы в консоли с расширенными атрибутами файлов

2 Теоретическое введение

Права доступа в операционной системе Linux представляют собой ключевой элемент безопасности, определяющий, какой доступ имеют пользователи и программы к файлам и каталогам. [1]

Расширенные атрибуты файловых объектов - поддерживаемая некоторыми файловыми системами возможность ассоциировать с файловыми объектами произвольные метаданные. [2]

Команда *chattr* изменяет атрибуты файлов в файловой системе Linux. Оператор «+» вызывает добавление выбранных атрибутов к существующим атрибутам файлов; «—» заставляет их удалить; и «=» делает их единственными атрибутами файлов.

Полня команда может выглядеть следующим образом: *chattr +a*

Команда *lsattr* перечисляет атрибуты файлов в файловой системе Linux. Например, *lsstar* , позволит просмотреть расширенные атрибуты, которые имеет определенный файл

Буквы **«aAcCdDeFijmPsStTux»** выбирают новые атрибуты для файлов:

- только добавление (a),
- без обновлений времени (A),
- сжатие (c),
- без копирования при записи (C),
- без дампа (d),
- синхронные обновления каталогов (D),
- формат экстента (e),

- поиск в каталогах без учёта регистра (F),
- неизменяемый (i),
- ведение журнала данных (j),
- без сжатия (m),
- иерархия проекта (P),
- безопасное удаление (s),
- синхронные обновления (S),
- без слияния хвостов (t),
- вершина иерархии каталогов (T),
- возможность восстановления после удаления (u)
- прямой доступ к файлам (x).

Следующие атрибуты доступны только для чтения и могут быть перечислены lsattr, но не могут быть изменены chattr:

- зашифрованный (E),
- индексированный каталог (I),
- встроенные данные (N)
- достоверность (V).

Подробне рассмотрим ращиренные атрибуты, которые мы будем использовать в лабораторной работе:

- Файл с установленным атрибутом «a» можно открыть только в режиме добавления для записи. Только суперпользователь или процесс, обладающий возможностью CAP_LINUX_IMMUTABLE, может установить или очистить этот атрибут.

- Файл с атрибутом «i» не может быть изменён: его нельзя удалить или переименовать, нельзя создать ссылку на этот файл, большую часть метаданных файла нельзя изменить, и файл нельзя открыть в режиме записи. Только суперпользователь или процесс, обладающий возможностью CAP_LINUX_IMMUTABLE, может установить или очистить этот атрибут. [3]

3 Выполнение лабораторной работы

От имени пользователя guest определим расширенные атрибуты файла /home/guest/dir1/file1 командой *lsattr /home/guest/dir1/file1*.(рис. 3.1) Видим, что файл не имеет никаких раширенных атрибутов

Рис. 3.1: Расширенные атрибуты file1

Установим командой *chmod 600 file1* на файл file1 права, разрешающие чтение и запись для владельца файла. Проверим, что права действительно были установлены (рис. 3.2)

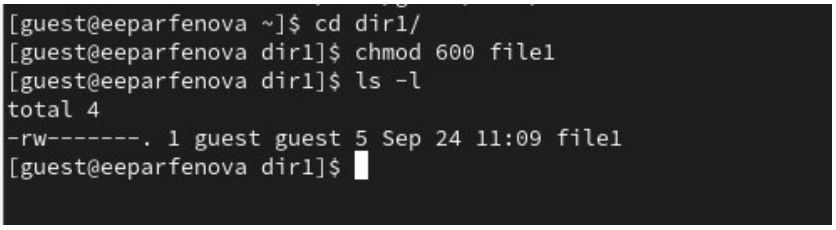

Рис. 3.2: Установленные атрибуты file1

Попробуем установить на файл /home/guest/dir1/file1 расширенный атрибут a от имени пользователя guest командой *chattr +a /home/guest/dir1/file1*. Видим, что мы не смогли это сделать и получили отказ от выполнения операции (рис. 3.3)

Рис. 3.3: Отказ от выполнения операции установки расширенного атрибута

Откроем второе окно терминала и, войдя с правами суперпользователя, попробуем ввести ту же команду и установить на файл ращиренный атрибут a (рис. 3.4). Проверим, что все успешно получилось от имени пользовтаеля guest командой *lsattr /home/guest/dir1/file1*. (рис. 3.5)

Рис. 3.4: Установка раширенного атрибута на file1

Рис. 3.5: Новые расширенные атрибуты file1

Выполним дозапись в файл file1 слова «test» командой *echo "test" /home/guest/dir1/file1*. После этого выполним чтение файла file1 командой *cat /home/guest/dir1/file1*, убеждаясь, что слово test было успешно записано в file1. (рис. 3.6)

Рис. 3.6: Дозапись и чтение текста в file1

Попробуем стереть имеющуюся в файле информацию командой *echo "abcd" > /home/guest/dirl/file1*. Видим, что нам отказано в операции. Попробуем переименовать файл, что также не получается. (рис. 3.7) Далее попробуем установить на файл file1 права, например, запрещающие чтение и запись для владельца файла командой *chmod 000 file1*, однако снова получаем отказ от выполнения операции. (рис. 3.8)

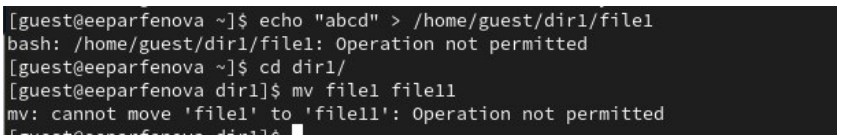

Рис. 3.7: Попытка стереть информацию в файле

Рис. 3.8: Попытка изменения атрибутов файла

Снимем раширенный атрибут a с file1 от имени суперпользователя (рис. 3.9) и попробуем выполнить те же операции. Видим, что каждая операция выполнена успешно. (рис. 3.10) (рис. 3.11)

Рис. 3.9: Снятие раширенного атрибута a

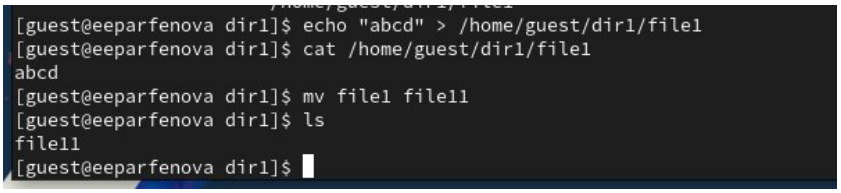

Рис. 3.10: Выполнение команд для файла без атрибута

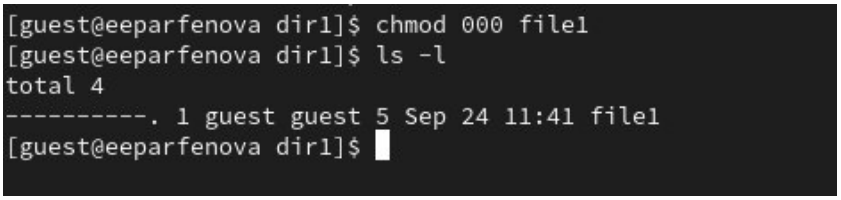

Рис. 3.11: Изменение прав для файла без атрибута

Далее проделаем те же самые действия, установив на файл атрибут i. Сделаем это командой *chattr +i /home/guest/dir1/file1* от имени суперпользователя (рис. 3.12) и проверим от имени пользователя guest, все ли получилось (рис. 3.13).

Рис. 3.12: Установка атрибута i

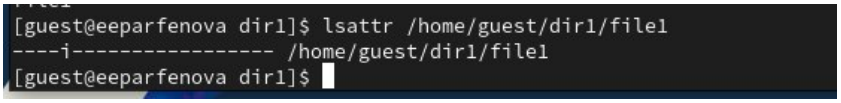

Рис. 3.13: Новые расширенные атрибуты file1

Проверим выполнение всех команд, опробованных выше, на файле с новым атрибутом. Видим, что тперь даже дозапись в файл нам недоступна, что логично, ведь атрибут i делает файл полностью неизменяемым для владельца. (рис. 3.14) (рис. 3.15)

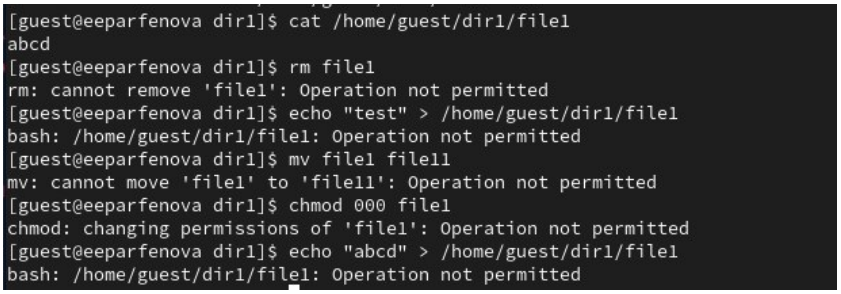

Рис. 3.14: Повторение команд для файла с атрибутом i

Рис. 3.15: Установка прав для файла с атрибутом i

4 Выводы

Мы получили практические навыки работы в консоли с расширенными атрибутами файлов

Список литературы

1. Как дать права пользователю Linux: инструкция [Электронный ресурс]. ООО «ТАЙМВЭБ.КЛАУД»., 2024. URL: https://timeweb.cloud/tutorials/linux/kak-dat-prava-polzovatelyu-linux.
2. Работа с расширенными атрибутами: attr, getfattr/setfattr, xattr [Электронный ресурс]. © 2003 – 2024 Компания Atlassian Corporation, 2023. URL: https://wiki.astralinux.ru/pages/viewpage.action?pageId=149063848.
3. Атрибуты файлов в Linux [Электронный ресурс]. 2021. URL: https://zalinux.ru/?p=6440.