Лабораторная работа № 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Абакумов Егор Александрович

Содержание

# Цель работы

Изучение механизмов изменения идентификаторов, применения SetUID-, SetGID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

# Теоретическое описание

В операционной системе Linux есть много отличных функций безопасности, но она из самых важных - это система прав доступа к файлам. Linux, как последователь идеологии ядра Linux в отличие от Windows, изначально проектировался как многопользовательская система, поэтому права доступа к файлам в linux продуманы очень хорошо.

У каждого файла имеется определённый набор свойств в файловой системе. Например, это права доступа, владелец, имя, метки времени. В Linux каждый файл имеет довольно много свойств, например, права доступа устанавливаются трижды (для владельца, группы и всех прочих), метки времени также бывают трёх разных видов (время создание, доступа и изменения) [1].

Кроме того файлам и директориям могут быть установлены расширенные атрибуты доступа. Файловые атрибуты могут использовать администраторы и пользователи для защиты файлов от случайных удалений и изменений, а также их применяют злоумышленники, делая невозможным удаление вредоносного файла.

Для работы с этими атрибутами обычно используются утилиты chattr и lsattr, входящие в пакет e2fsprogs и предустановленные во всех современных дистрибутивах.

Для работы с правами используем команду chmod, синтаксис которой выглядит следующим образом:

chmod options permissions path\_to\_the\_file

Сначала рассмотрим какими бывают права доступа linux и как они устанавливаются. Есть три основных вида прав:

r - чтение; w - запись; x - выполнение; s - выполнение от имени суперпользователя (дополнительный);

Также есть три категории пользователей, для которых вы можете установить эти права на файл linux:

u - владелец файла; g - группа файла; o - все остальные пользователи; Синтаксис настройки прав такой:

группа\_пользователей действие вид\_прав

В качестве действий могут использоваться знаки “+” - включить или “-” - отключить [2].

Тем не менее, не стоит забывать, что вы не можете использовать chattr как меру безопасности так как атрибуты легко изменить. Один из способов решения этой проблемы - ограничить доступ к самой утилите chattr[3].

# Ход работы

1. Готовим систему и входим из-под guest’a. Пишем программу simpleid.c (иллюстр. 1). Компилируем программу, запускаем, видим вывод uid и gid пользователя, сравниваем вывод с id, всё так же (иллюстр. 2).

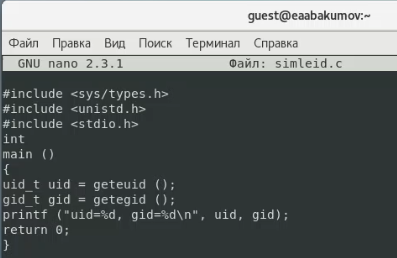


Figure 1: Листинг simpleid.c

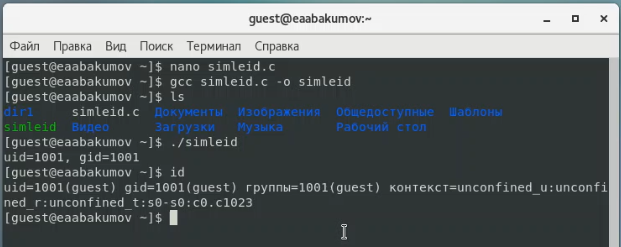


Figure 2: Запуск и сверка simpled.c и id

1. Усложняем программу и запускаем её (иллюстр. 3, 4). Из-под рута меняем владельца и добавляем SetUID бит на файл (иллюстр. 5). Проверяем правильность и запускаем программу еще раз. еuid возвращает id владельца, а real\_uid возвращает uid запускающего пользователя (иллюстр. 6).

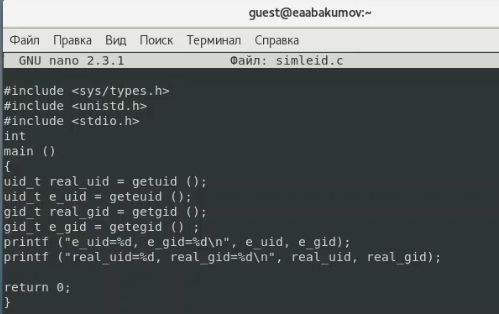


Figure 3: Листинг simpleid2.c

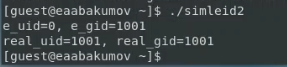


Figure 4: Запуск simpleid2.c

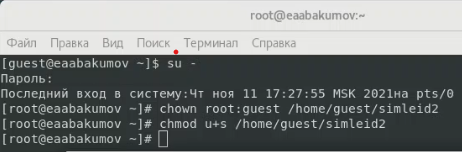


Figure 5: Команды под рутом

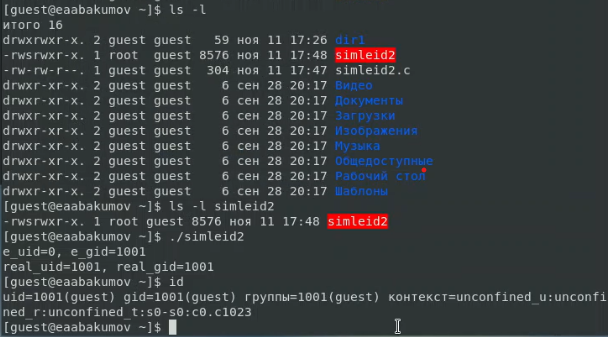


Figure 6: Повторный запуск simpleid2.c

1. Теперь добавим на файл SetGID бит с проделаем все то же самое (иллюстр. 7, 8).

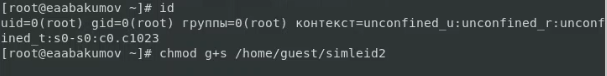


Figure 7: Команды под рутом для SetGID

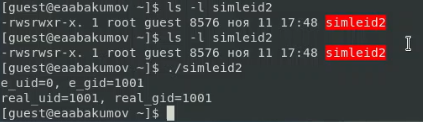


Figure 8: Третий запуск simpleid2.c

1. Пишем программу readfile.c (иллюстр. 9, 10).

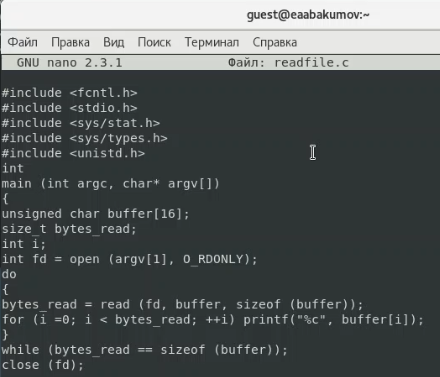


Figure 9: readfile.c (ч.1)

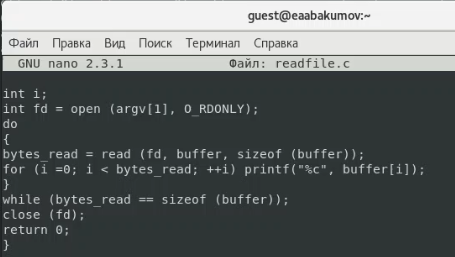


Figure 10: readfile.c (ч.2)

1. Меняем владельца у файла readfile.c и запрещаем чтение всем, кроме суперпользователя (иллюстр. 11). Проверяем, что guest не может читать (иллюстр. 12). Меняем владельца у программы readfile и добавляем SetUID бит на неё (иллюстр. 13).

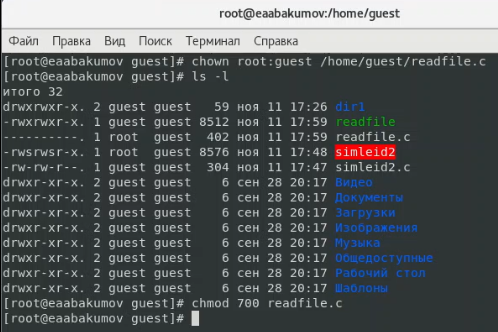


Figure 11: Смена владельца readfile.c

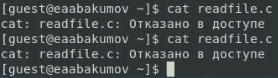


Figure 12: Проверка на cat из-под guest’a

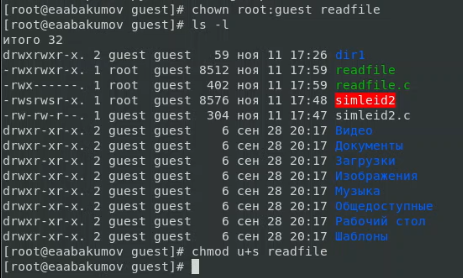


Figure 13: Смена владельца readfile и SetUID

1. Проверяем, может ли программа readfile прочитать файл readfile.c и файл /etc/shadow. Да, может (иллюстр. 14, 15). Хотя сам пользователь вручную не мог. Всё дело в том, что при вызове программы права пользователя повышаются SetUID битом до прав владельца, который может читать файлы (суперпользователь в нашем случае).

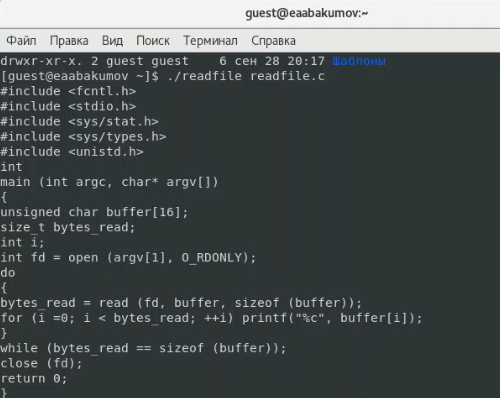


Figure 14: Проверка readfile.c программой readfile

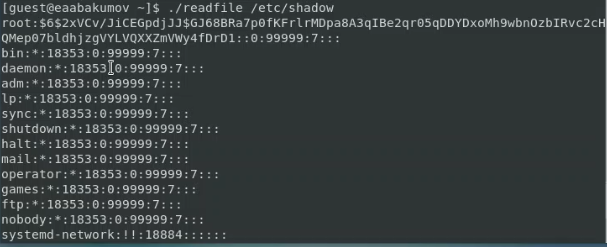


Figure 15: Проверка /etc/shadow программой readfile

1. Проверяем Sticky бит. Для этого создаем файл, которому даем rw права для others и пишем туда слово test (иллюстр. 16). Теперь пробуем выполнить дозапись в файл, перезапись файла и его удаление. Всё, кроме удаления, прошло успешно (иллюстр. 17).

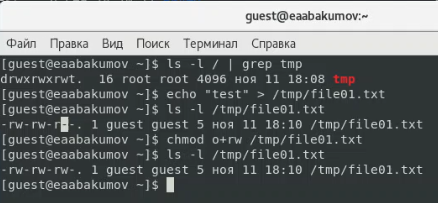


Figure 16: Создание файла и правка прав

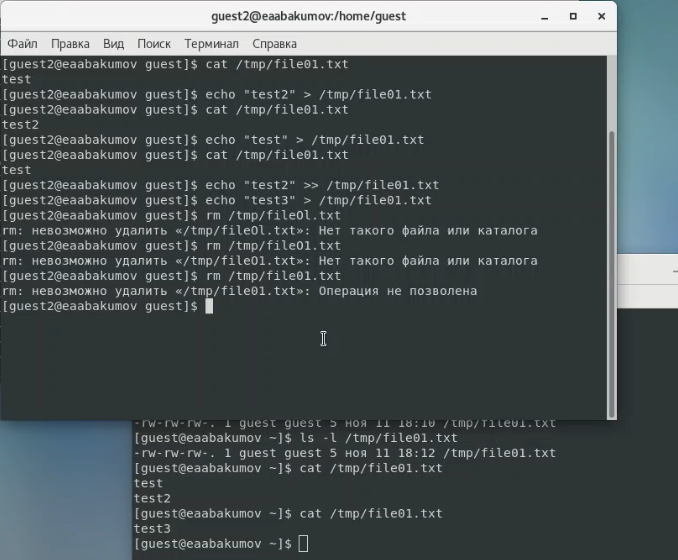


Figure 17: Тестирование файла

1. Повышаем права до суперпользователя и удаляем Sticky-бит с папки /tmp (иллюстр. 18). Повторяем наши тесты. Теперь прошли все команды, включая удаление файла (иллюстр. 19). Таким образом, пользователь, не являющийся владельцем файла, смог его удалить, так как Sticky-бит не был настроен. Возвращаем Sticky-бит на папку /tmp.

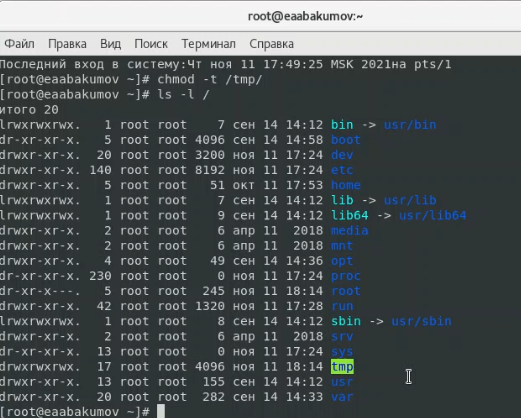


Figure 18: Удаление Sticky-бита

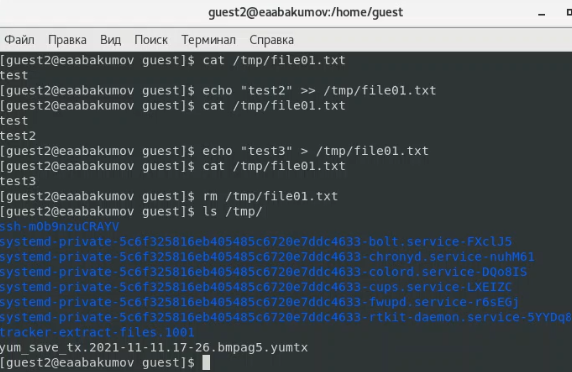


Figure 19: Повторное тестирование файла

# Выводы

В ходе работы мы успешно изучили механизмы изменения идентификаторов, применения SetUID-, SetGID- и Sticky-битов, получили практические навыки работы в консоли с дополнительными атрибутами, рассмотрели принципы работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

# Список литературы

1. Атрибуты файлов в Linux. // zalinux.ru. 2021. URL: https://zalinux.ru/?p=6440 (дата обращения 11.11.2021).
2. Команда chmod в Linux. // Losst. 2020. URL: https://losst.ru/komanda-chmod-linux (дата обращения 11.11.2021).
3. КОМАНДА CHATTR В LINUX. // Losst. 2020. URL: https://losst.ru/neizmenyaemye-fajly-v-linux (дата обращения 11.11.2021).
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..