Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Буллет Т. А.

15 февраля 2025

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Буллер Татьяна Александровна
- студент направления Бизнес-информатика
- Российский университет дружбы народов

Вводная часть

Объект и предмет исследования

- Операционная система linux, дисрибутив Rocky
- Среда виртуализации VirtualBox

Цели и задачи

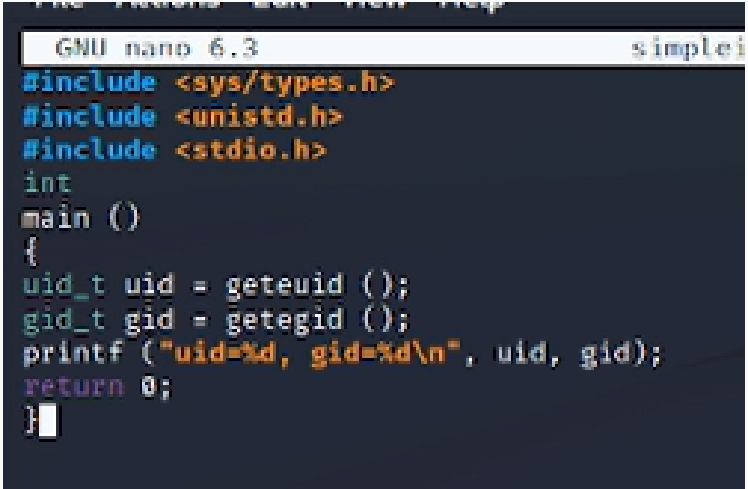
- Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов.
- Получение практических навыков работы в консоли с дополнительными атрибутами.
- Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Материалы и методы

- Процессор pandoc для входного формата Markdown
- Среда виртуализации VirtualBox

Выполнение лабораторной работы

Создадим программу simpleid.c. Эта программа с помощью функций geteuid и getegid получает uid и gid пользователя соответственно, после чего выводит их на экран.



Скомпилировав программу, получим вывод, в значениях совпадающий с выводом команды id.

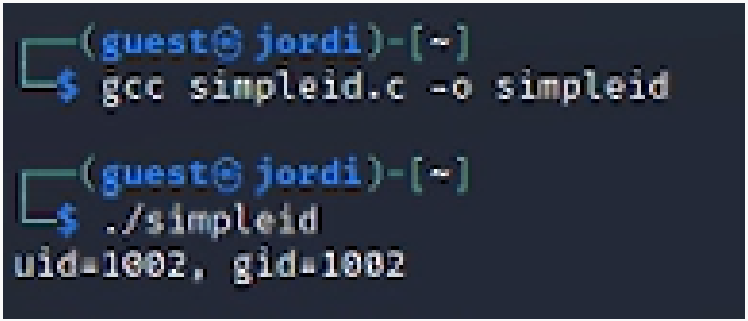


Рис. 2: Скомпилированная программа

После этого усложним программу, как показано на скриншоте:

Рис. 4: simpleid2

Теперь вывод дополнен и все еще совпадает с выводом id.

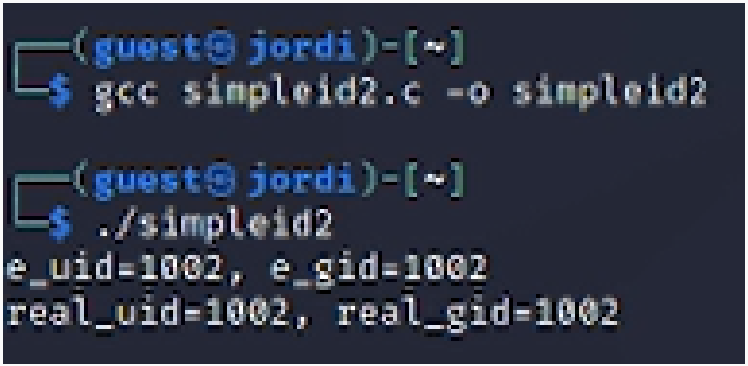


Рис. 5: Дополненный вывод

Следующим шагом от имени суперпользователя назначим владельцам файла программы суперпользователя и добавим ей SUID-бит.

Рис. 6: Переназначение владельца и добавление SUID-бита

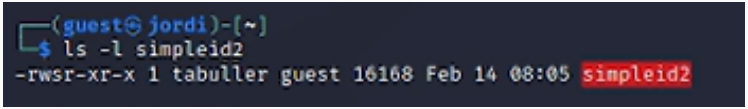


Рис. 7: Новые права программы

Теперь при запуске этой программы видим, что она выводит e_uid: идентификатор пользователя, от имени которого она была запущена; real_uid - идентификатор пользователя, от имени которого она исполняется.

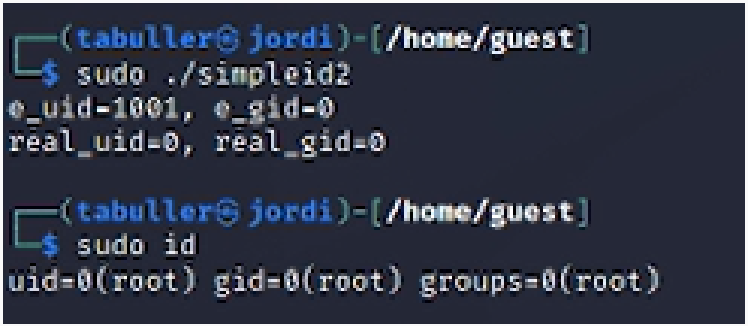
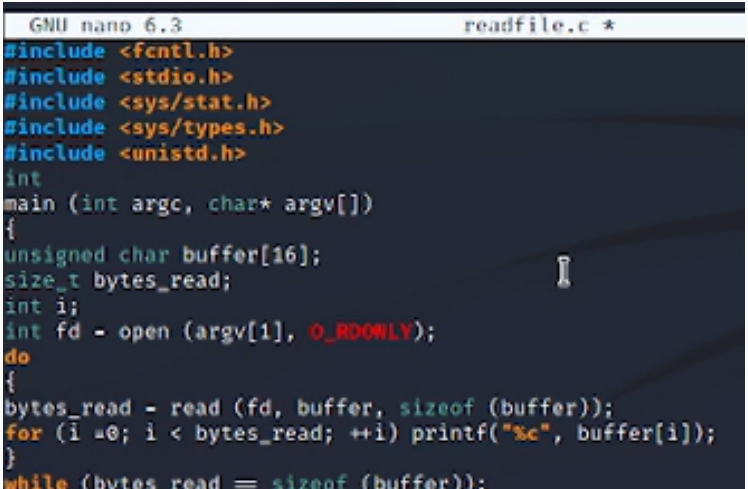


Рис. 8: Вывод simpleid2 с SUID-битом

Исследование SUID-бита

Создадим еще одну программу: аналог cat readfile, которая будет получать содержимое файла, название которого передано ей в аргументе, и выводить его на экран.

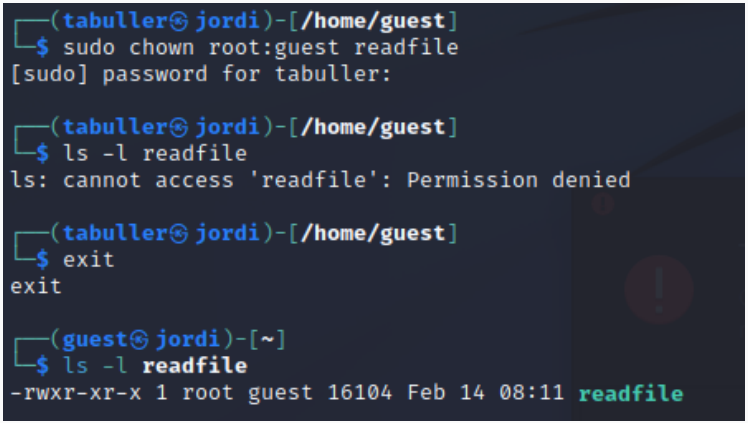


Скомпилируем ее, передадим файл кода во владение суперпользователю и добавим SUID-бит, после чего заберем у всех остальных пользователей все права на него. Видим, что теперь пользователь guest не может прочитать содержимое.

Рис. 10: Изменение прав файла кода

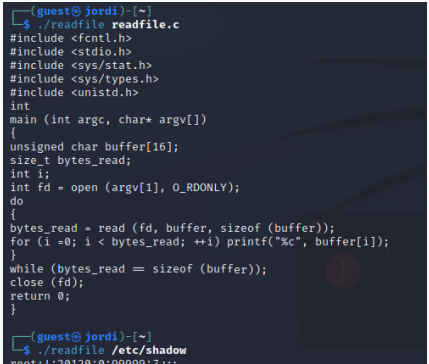
Исследование SUID-бита

Следующим шагом изменим права на программу, которая была скомпилирована по коду readfile.c. Добавим тот же SUID-бит и передадим во владение суперпользователю. Вилим, что теперь название программы подсвечено красным.



Исследование SUID-бита

Теперь при попытке прочитать readfile.c с помощью скомпилированной им программы мы получаем содержимое файла. Это происходит потому, что программа исполняется от имени суперпользователя и наделена всеми теми же правами, что и root. Она в том числе может прочитать и системный файл /etc/shadow, в котором по умолчанию хранится список пользователей и их хэшированные пароли.



Просмотрев в корневой директории системы права поддиректорий, видим, что на директории tmp установлен бит t. Полезно отметить также, что писать и читать эту директорию может кто угодно.

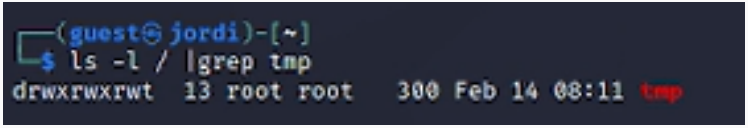


Рис. 15: Sticky-бит на директории tmp

Далее создадим в этой директории файл file01. Изначально права на него позволяют пользователям кроме владельца только читать его, владельцу - записывать и читать. Добавим для остальных пользователей права на запись.

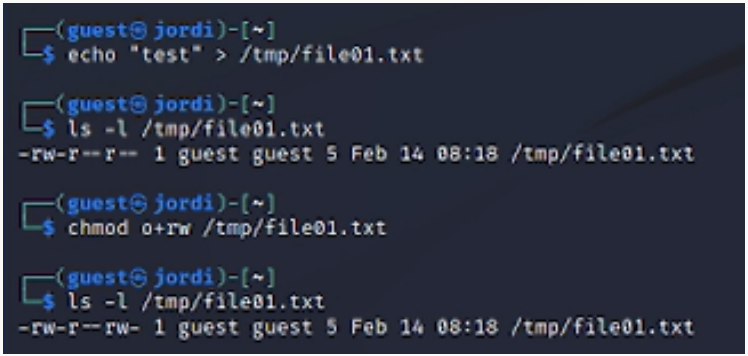
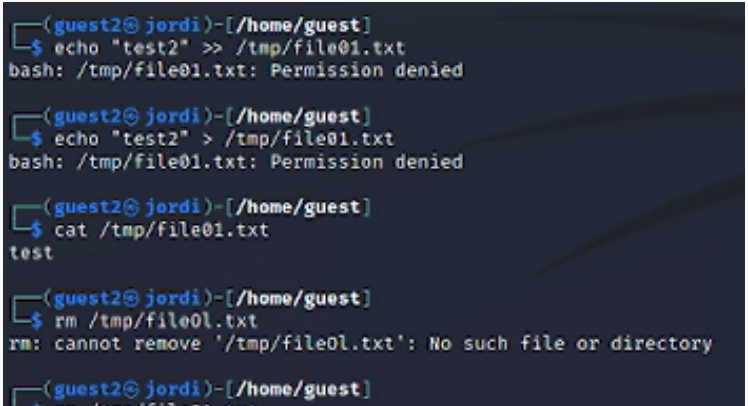


Рис. 16: Созданный в директории tmp файл

Переключившись на пользователя guest2, однако, мы все еще не можем сделать с этим файлом ничего, кроме прочтения, потому что пользователь guest2 входит в группу пользователя guest, а группе не были добавлены права на запись. Файл, кроме прочего, “защищен” Sticky-битом, установленным на директории.



От имени суперпользователя снимем Sticky-бит и снова проверим права на директорию tmp: символ t пропал.

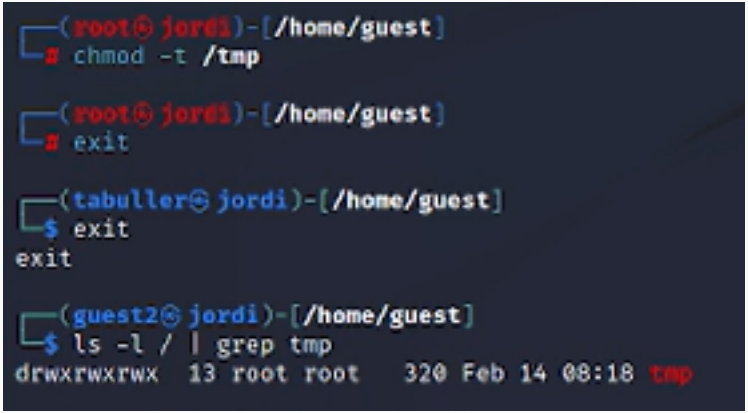


Рис. 18: Снятие Sticky-бита

Теперь уже, все еще не являясь владельцем файла, мы можем его переименовывать и удалять.

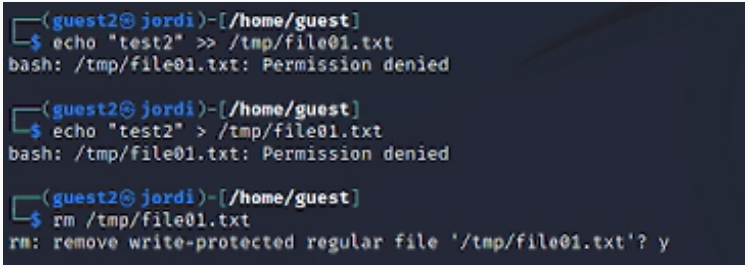


Рис. 19: Работа с файлом от лица стороннего пользователя

Выводы

Выводы

Изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами. Рассмотрена работа механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.