Лабораторная работа №5

Дисциплина: Информационная безопасность

Дорофеева Алёна Тимофеевна

Содержание

# 1 Цель работы

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.

# 2 Задание

* Изучить на практике применение SetUID- и Sticky-битов.

# 3 Теоретическое введение

**Setuid** – это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо. Классический пример использования этого бита в операционной системе это команда sudo.

Там, где обычно установлен классический бит **x** (на исполнение), выставляется специальный бит **s**. Это позволяет обычному пользователю системы выполнять команды с повышенными привилегиями без необходимости входа в систему как root, разумеется зная пароль пользователя root. Принцип работы **Setgid** очень похож на setuid с отличием, что файл будет запускаться пользователем от имени группы, которая владеет файлом.

Биты **setuid** и **setgid** выставляются с помощью команд chmod u+s и chmod g+s соответственно.[**01?**]

**Sticky-bit** — дополнительный атрибут файлов или каталогов в операционных системах семейства UNIX. В настоящее время sticky bit используется в основном для каталогов, чтобы защитить в них файлы. Из такого каталога пользователь может удалить только те файлы, владельцем которых он является. Примером может служить каталог /tmp, в который запись открыта для всех пользователей, но нежелательно удаление чужих файлов. Установка атрибута производится утилитой chmod.

В операционной системе Solaris для файлов, не являющихся программами, имеет строго противоположное действие — запрещает сохранение данных этого файла в системном кэше.[**02?**]

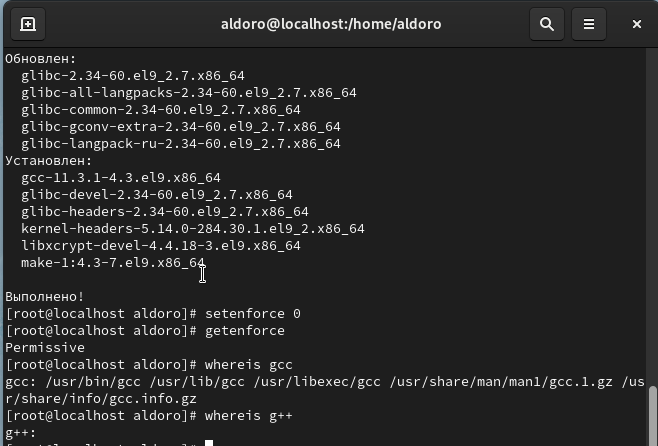
Это разрешение полезно для защиты файлов от случайного удаления в среде, где несколько пользователей имеют права на запись в один и тот же каталог. Если применяется закрепленный sticky bit, пользователь может удалить файл, только если он является пользователем-владельцем файла или каталога, в котором содержится файл. По этой причине он применяется в качестве разрешения по умолчанию для каталога /tmp и может быть полезен также для каталогов общих групп.

Когда вы применяете sticky bit, пользователь может удалять файлы, только если выполняется одно из следующих условий:

* Пользователь является владельцем файла;
* Пользователь является владельцем каталога, в котором находится файл.[**03?**]

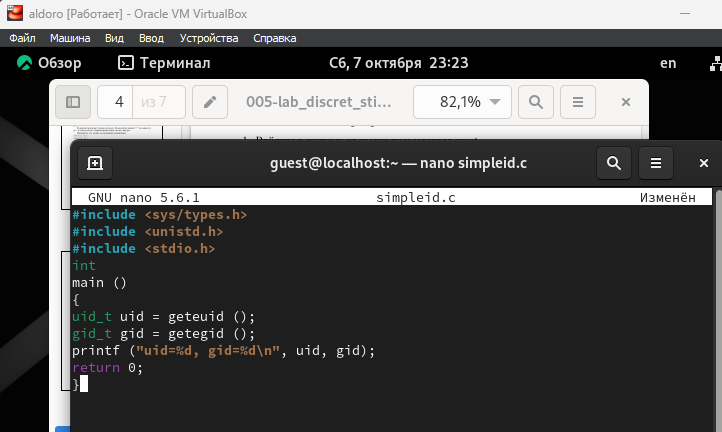
# 4 Выполнение лабораторной работы

1. Вошла в систему от имени пользователя guest (рис. ??). Убедилась, что в системе установлен компилятор gcc, проверив версию командой gcc -v (рис. ??). Видим, что он установлен. Отключила систему запретов до очередной перезагрузки командой setenforce 0 (рис. ??). После этого команда getenforce выводит “Permissive” (рис. ??).



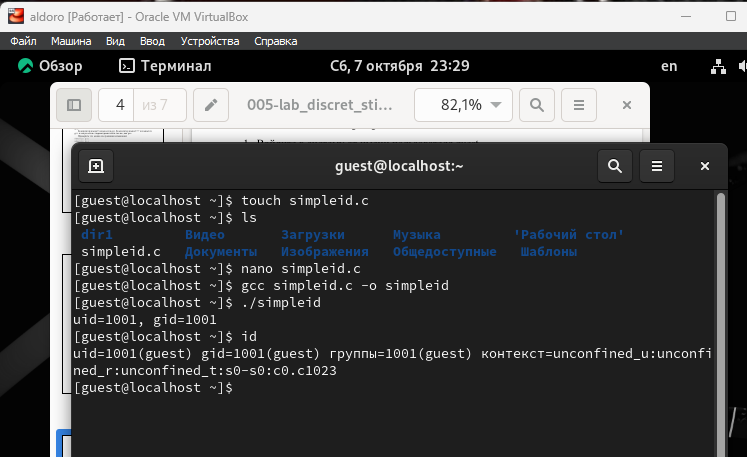
Подготовка лабораторного стенда

1. Создала программу simpleid.c в текстовом редакторе nano (рис. ??).



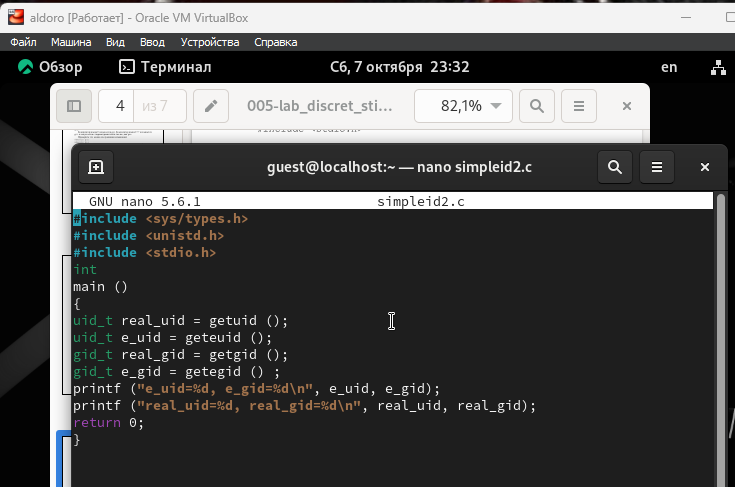
Программа simpleid

1. Cкомпилировала программу и убедилась, что файл программы создан, командой gcc simpleid.c -o simpleid (рис. ??).
2. Выполнила программу simpleid, введя ./simpleid (рис. ??).
3. Выполнила системную программу id (рис. ??). Видим, что выводимая информация о uid и gid идентична, но команда id также выводит groups=1001(guest).



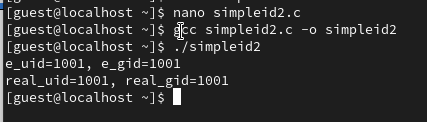
Программа simpleid

1. Усложнила программу, добавив вывод действительных идентификаторов (рис. ??). Получившуюся программу назвала simpleid2.c (рис. ??).



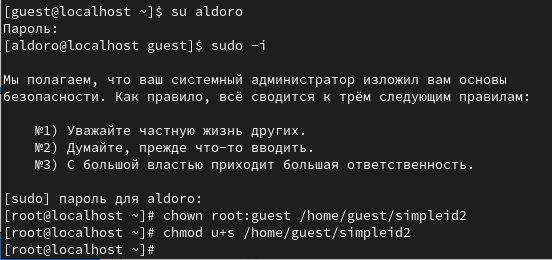
Усложнение программы

1. Cкомпилировала программу командой gcc simpleid2.c -o simpleid2 (рис. ??). Запустила программу, введя ./simpleid2 (рис. ??).



Программа simpleid2

1. Выполнила команды sudo chown root:guest /home/guest/simpleid2 и sudo chmod u+s /home/guest/simpleid2 (рис. ??).
2. Первая команда меняет у файла владельца и группу. Вторая команда ставит SetUID-бит.
3. Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2 с помощью команды ls -l simpleid2 (рис. ??). Видим, что все установлено верно.
4. Запустила файл simpleid2 командой ./simpleid2 (рис. ??). Также выполнила системную команду id (рис. ??). Видим, что файл выводит информацию не только о реальном идентификаторе, но и эффективных ID - EUID и EGID. Фактические ID пользователя и группы соответствуютт ID пользователя, который вызвал процесс (пользователь guest, группа guest). Эффективный ID пользователя соответствует установленному SetUid биту на исполняемом файле (привелегии суперпользователя). Эффективный ID группы соответствует установленному SetGid биту на исполняемом файле.Команда id выводит информацию только о реальном идентификаторе.
5. Проделала то же самое относительно SetGID-бита (рис. ??). Видим, что теперь при исполнении файла выводится EUID = 1001, EGID = 0, так как мы установили SetGid бит.



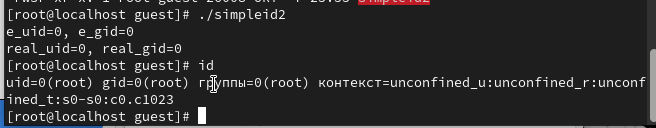
SetUID- и SetGID-биты

1. Создала программу readfile с помощью тектового редактора nano (рис. ??).
2. Откомпилировала её командой gcc readfile.c -o readfile (рис. ??).

Файл readfile.c

Файл readfile.c

1. Сменила владельца у файла readfile.c на root командой sudo chown root /home/guest/readfile и изменила права так, чтобы только суперпользователь (в моем случае владелец) мог прочитать его, a guest (член группы) не мог командой sudo chmod 733 /home/guest/readfile (рис. ??).
2. Проверила, что пользователь guest не может прочитать файл readfile.c командой cat readfile.c (рис. ??). Видим, что нам действильно отказано в доступе.
3. Сменила у программы readfile владельца обратно на guest и установила SetUID-бит командами sudo chown guest:guest /home/guest/readfile и sudo chmod u+s /home/guest/readfile (рис. ??).



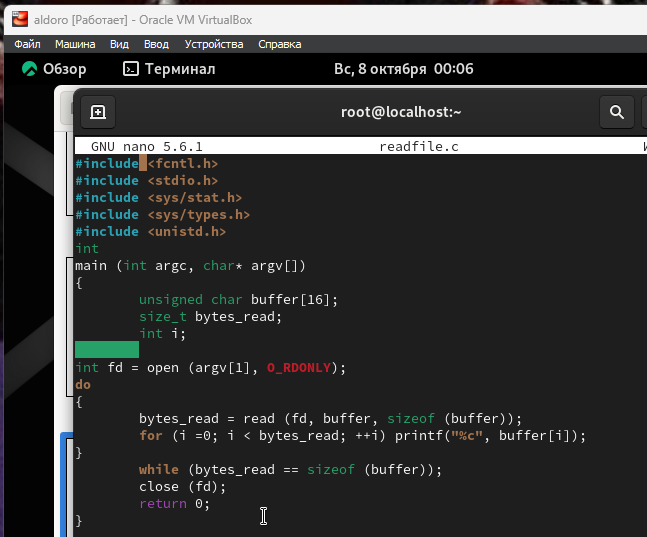
Смена владельца файла readfile.c

1. Проверила, может ли программа readfile прочитать файл readfile.c (рис. ??). Видим, что может, так как мы установили SetUID-бит.

Чтение файла readfile.c

Чтение файла readfile.c

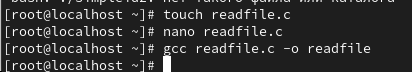
1. Проверилп, может ли программа readfile прочитать файл /etc/shadow (рис. ??). Видим, что может, так как мы установили SetUID-бит.



Чтение файла /etc/shadow

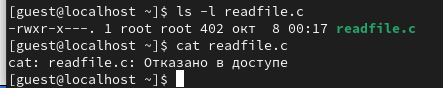
## 4.1 Исследование Sticky-бита

1. Выяснила, установлен ли атрибут Sticky на директории /tmp, для чего выполнила команду ls -l / | grep tmp (рис. ??). Видим, что он установлен (буква t в конце).
2. От имени пользователя guest создала файл file01.txt в директории /tmp со словом test командой echo "test" > /tmp/file01.txt (рис. ??).
3. Просмотрела атрибуты у только что созданного файла (команда ls -l /tmp/file01.txt) и разрешила чтение и запись для категории пользователей «все остальные» командой chmod o+rw /tmp/file01.txt (рис. ??). Проверила атрибуты (команда ls -l /tmp/file01.txt) (рис. ??).



Проверка наличия Sticky-бита на директории /tmp

1. От пользователя guest2 (не являющегося владельцем) попробовала прочитать файл /tmp/file01.txt командой cat /tmp/file01.txt (рис. ??). Прочитать файл удалось.
2. От пользователя guest2 попробовала дозаписать в файл /tmp/file01.txt слово test2 командой echo "test2" >> /tmp/file01.txt (рис. ??). Выполнить данную операцию не удалось.
3. Проверила содержимое файла командой cat /tmp/file01.txt (рис. ??). Действие по сути бессмысленное, так как файл не менялся.
4. От пользователя guest2 попробовала записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo "test3" > /tmp/file01.txt (рис. ??). Выполнить эту операцию не удалось.
5. Проверила содержимое файла командой cat /tmp/file01.txt (рис. ??). Действие по сути бессмысленное, так как файл не менялся.
6. От пользователя guest2 попробовала удалить файл /tmp/file01.txt командой rm /tmp/fileOl.txt (рис. ??).



Проверка возможности действий при наличии Sticky-бита

1. Повысила свои права до суперпользователя следующей командой su - и выполнида после этого команду chmod -t /tmp, снимающую атрибут t (Sticky-бит) с директории /tmp (рис. ??).
2. Покинула режим суперпользователя командой exit (рис. ??)

Снятие Sticky-бита

Снятие Sticky-бита

1. От пользователя guest2 проверила, что атрибута t у директории /tmp нет командой ls -l / | grep tmp (рис. ??).
2. Повторила предыдущие шаги (рис. ??). Теперь мы смогли удалить файл.
3. Нам удалось удалить файл, когда мы сняли Sticky-бит.

Проверка возможности действий при отсутствии Sticky-бита

Проверка возможности действий при отсутствии Sticky-бита

# 5 Выводы

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

# Список литературы