Отчет по лабораторной работе №5

Дискреционное разграничение прав в Linux. Исследование влияния расширенных атрибутов

Евсеева Дарья Олеговна

8 октября, 2022

Содержание

Список иллюстраций

Цель работы

Целью данной работы является изучение механизмов изменения идентификаторов и 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.

Задание

1. Провести работу с SetUID- и SetGID-битами.
2. Провести работу со Sticky-битом.

Теоретическое введение

Дискреционное разграничение доступа — подход к разграничению доступа, предполагающий назначение владельцев объектов, которые по собственному усмотрению определяют права доступа субъектов (других пользователей) к объектам (файлам), которыми владеют.

Дискреционные механизмы разграничения доступа используются для разграничения прав доступа процессов как обычных пользователей, так и для ограничения прав системных программ (например, служб операционной системы), которые работают от лица псевдопользовательских учетных записей.

Выполнение лабораторной работы

Выполнять работу будем в операционной системе, установленной при выполнении первой лабораторной работы.

1. Работа с SetUID- и SetGID-битами

Для начала войдем в систему от имени пользователя guest.

Рис. 1: Вход в систему от имени пользователя guest

Далее создадим файл simpleid.c и запишем в него код программы.

Рис. 2: Создание файла

Рис. 3: Код программы

Скомпилируем программу и выполним ее, сравнив результаты с выводом команды id.

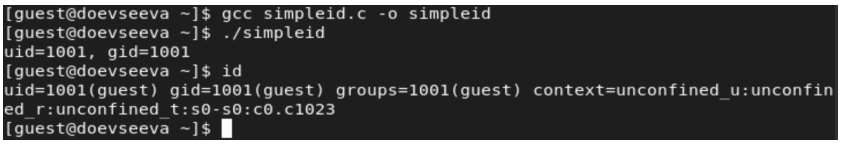

Рис. 4: Выполнение программы и команды id

Мы видим, что результаты вывода программы и команды совпадают и соответствуют действительности.

Далее создадим файл simpleid2.c и запишем в него ранее написанную программу, добавив вывод действительных идентификаторов.

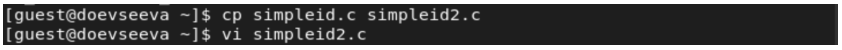

Рис. 5: Создание файла

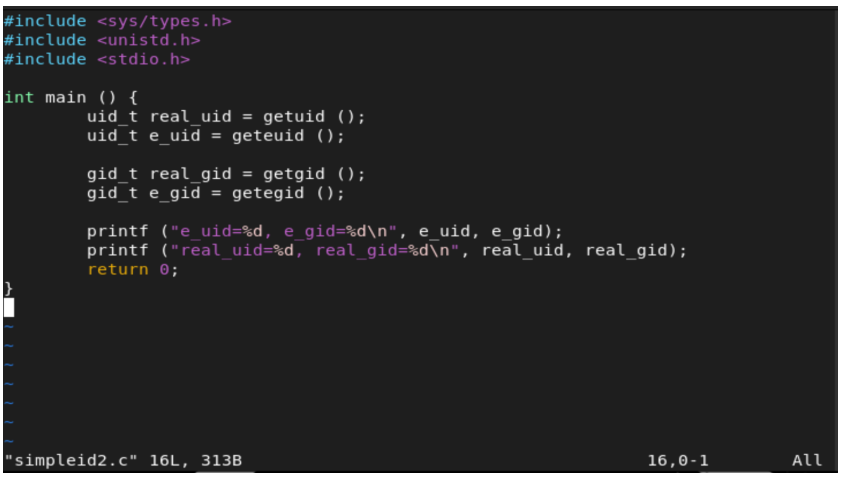

Рис. 6: Код программы

Скомпилируем и запустим программу.

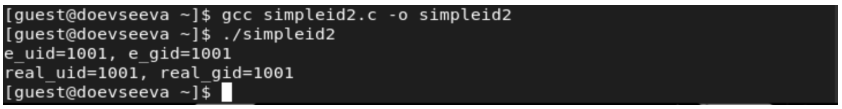

Рис. 7: Результаты запуска программы

Здесь мы видим, что полученные значения для пар uid и gid совпадают.

Далее от имени суперпользователя сменим владельца файла simpleid2 и установим на него атрибут s для пользователя, после чего проверим правильность выполненных команд и сравним результат запуска simpleid2 с выводом команды id для суперпользователя и для пользователя guest.

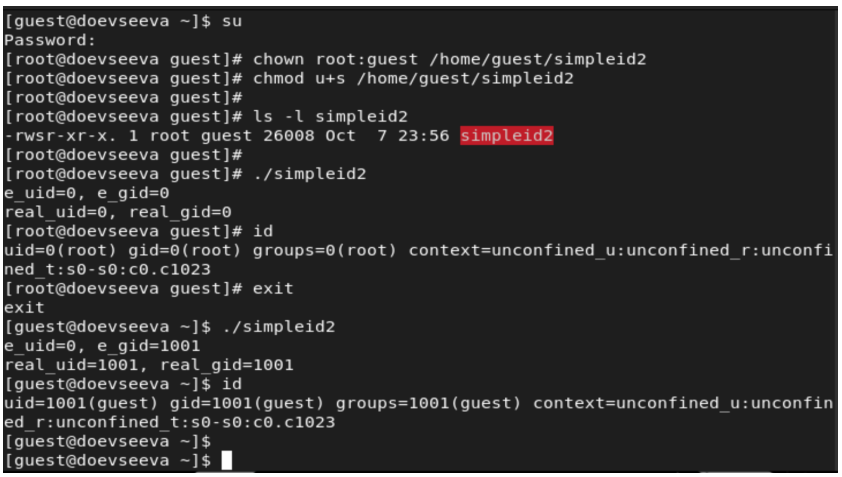

Рис. 8: Установка атрибута s для пользователя

Мы видим, что при выводе результатов от имени суперпользователя все результаты совпадают по значениям, однако при выводе результатов от имени пользователя guest можно заметить, что значение e_uid остается соответствующим суперпользователю, а остальные значения соответствуют значениям для пользователя guest.

Далее проделаем те же действия, установив на файл от имени суперпользователя атрибут s для группы.

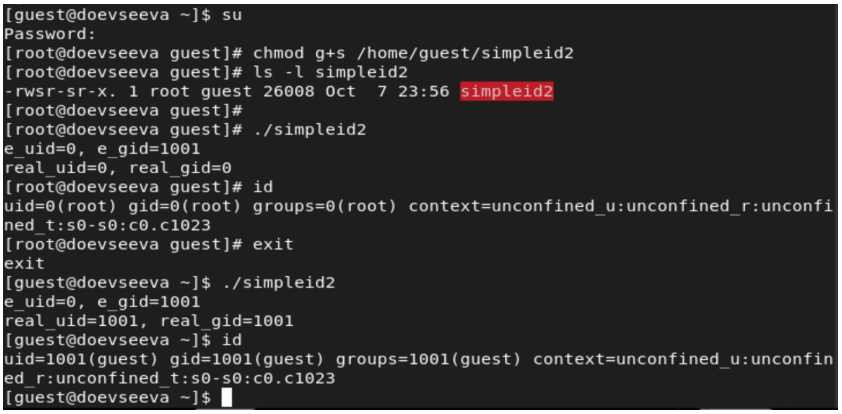

Рис. 9: Установка атрибута s для группы

Здесь мы видим, что при выводе результатов от имени суперпользователя все результаты совпадают по значениям, кроме значения g_uid, которое соответствует

группе guest, а при выводе результатов от имени пользователя guest значение e_uid остается соответствующим суперпользователю, а значение g_uid с остальными результатами также соответствует группе и пользователю guest.

Теперь создадим программу readfile.c и откомпилируем ее.

```
[guest@doevseeva ~]$ touch readfile.c
[guest@doevseeva ~]$ vi readfile.c
```

Рис. 10: Создание файла

```
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

int main (int argc, char* argv[]) {
        unsigned char buffer[16];
        size_t bytes_read;
        int i;

        int fd = open (argv[1], O_RDONLY);
        do {
                bytes_read = read (fd, buffer, sizeof (buffer));
                for (i = 0; i < bytes_read; ++i)
                        printf("%c", buffer[i]);
        }
        while (bytes_read == sizeof (buffer));
        close (fd);
        return 0;
}

~
"readfile.c" 22L, 422B                                    5,16          All
```

Рис. 11: Код программы

```
[guest@doevseeva ~]$ gcc readfile.c -o readfile
```

Рис. 12: Компиляция программы

Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь мог прочитать его, а guest не мог, и проверим успешность изменения прав.

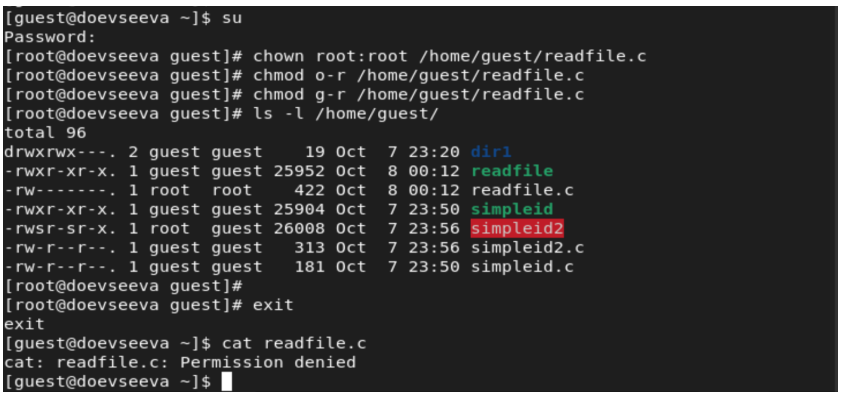

Рис. 13: Смена владельца файла и изменение прав

Далее сменим у программы readfile владельца и установим SetUID-бит.

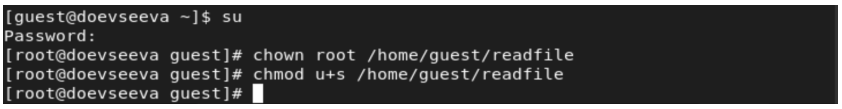

Рис. 14: Смена владельца и установка SetUID-бита

Проверим возможность чтения программой readfile файлов readfile.c и /etc/shadow от имени суперпользователя и пользователя guest.

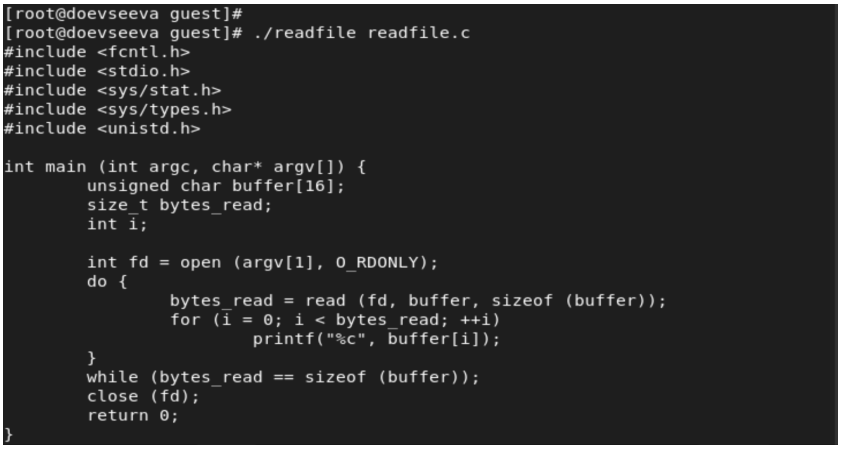

Рис. 15: Чтение readfile.c от имени суперпользователя

```
[root@doevseeva guest]# ./readfile /etc/shadow
root:$6$WBTn0LvqyZ30L6dS$vQtWyhTUOY9ABTAVak9ZDgcwlHWG03eq4/wi5KVUZVj5m9Lx9PDFGeb
D.n8UuzEOS8z9lS3lMvm4dh6C9QihF/::0:99999:7:::
bin:*:19123:0:99999:7:::
daemon:*:19123:0:99999:7:::
adm:*:19123:0:99999:7:::
lp:*:19123:0:99999:7:::
sync:*:19123:0:99999:7:::
shutdown:*:19123:0:99999:7:::
halt:*:19123:0:99999:7:::
mail:*:19123:0:99999:7:::
operator:*:19123:0:99999:7:::
games:*:19123:0:99999:7:::
ftp:*:19123:0:99999:7:::
nobody:*:19123:0:99999:7:::
systemd-coredump:!!:19245::::::
dbus:!!:19245::::::
polkitd:!!:19245::::::
rtkit:!!:19245::::::
sssd:!!:19245::::::
avahi:!!:19245::::::
pipewire:!!:19245::::::
libstoragemgmt:!!:19245::::::
```

Рис. 16: Чтение /etc/shadow от имени суперпользователя

```
[root@doevseeva guest]# exit
exit
[guest@doevseeva ~]$ ./readfile readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

int main (int argc, char* argv[]) {
        unsigned char buffer[16];
        size_t bytes_read;
        int i;

        int fd = open (argv[1], O_RDONLY);
        do {
                bytes_read = read (fd, buffer, sizeof (buffer));
                for (i = 0; i < bytes_read; ++i)
                        printf("%c", buffer[i]);
        }
        while (bytes_read == sizeof (buffer));
        close (fd);
        return 0;
}
```

Рис. 17: Чтение readfile.c от имени пользователя guest

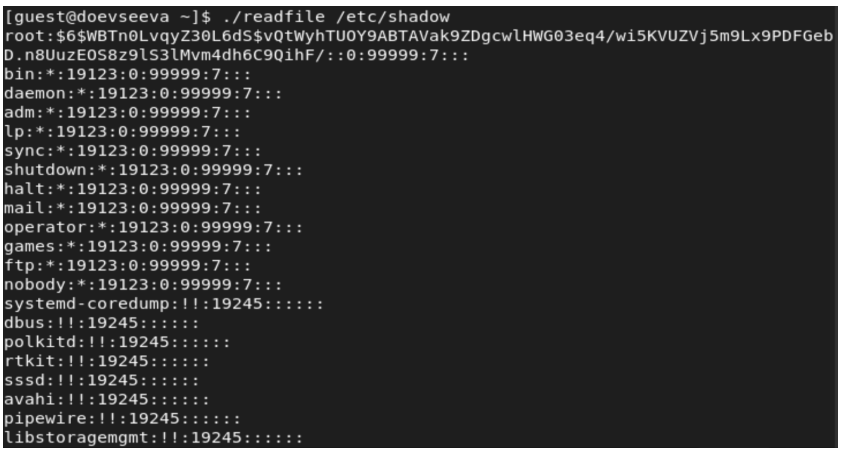

Рис. 18: Чтение /etc/shadow от имени пользователя guest

Как мы видим, файлы были успешно прочитаны в обоих случаях.

2. Работа со Sticky-битом

Выясним, установлен ли атрибут Sticky на директории /tmp.

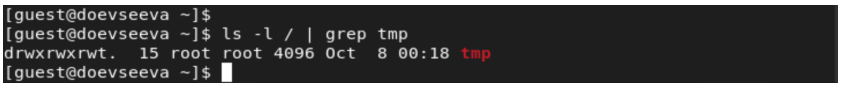

Рис. 19: Проверка наличия атрибута Sticky

Мы видим, что атрибут t установлен на директории.

От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test, просмотрим его атрибуты и разрешим чтение и запись для категории пользователей ‘все остальные’.

Рис. 20: Создание файла и изменение прав

От имени пользователя guest2 попробуем прочитать созданный файл.

Рис. 21: Чтение файла от имени пользователя guest2

Файл был успешно прочитан.

Далее попробуем сделать дозапись в файл, перезаписать его содержимое и удалить файл.

Рис. 22: Попытка дозаписи в файл

Рис. 23: Попытка перезаписи файла

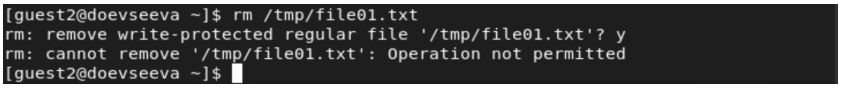

Рис. 24: Попытка удаления файла

При попытке выполнения каждой из команд было отказано в доступе, соответственно выполнить их не удалось.

Повысим свои права до суперпользователя и снимем атрибут t с директории /tmp.

Рис. 25: Снятие атрибута Sticky

Покинем режим суперпользователя и проверим успешность снятия атрибута.

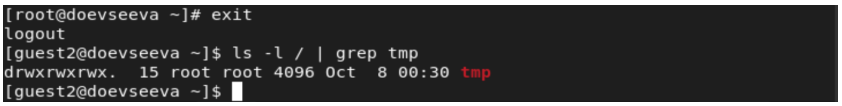

Рис. 26: Проверка атрибутов

Далее попробуем выполнить команды, которые не удалось выполнить ранее.

Рис. 27: Попытка повторного выполнения команд

Итак, мы можем видеть, что при попытке дозаписи в файл или перезаписи его содержимого в доступе снова было отказано, однако на этот раз удалось успешно удалить файл от имени пользователя, не являющегося его владельцем.

Повысим свои права до суперпользователя и вернем атрибут t на директорию /tmp.

Рис. 28: Возвращение атрибута Sticky

Выводы

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

Список литературы

- Методические материалы к лабораторной работе, представленные на сайте “ТУИС РУДН” https://esystem.rudn.ru/