Лабораторная работа № 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Юрченко Артём Алексеевич

Содержание

Список иллюстраций

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

1. Исследовать SetUID- и SetGID-биты.
2. Исследовать Sticky-бит.

3 Теоретическое введение

- Операционная система — это комплекс программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем [1].
- Права доступа определяют, какие действия конкретный пользователь может или не может совершать с определенным файлами и каталогами. С помощью разрешений можно создать надежную среду — такую, в которой никто не может поменять содержимое ваших документов или повредить системные файлы. [2].

4 Выполнение лабораторной работы

1. От имени пользователя guest создадим программу simpleid.c, скомпилируем ее и убедимся, что файл создан (4.1).

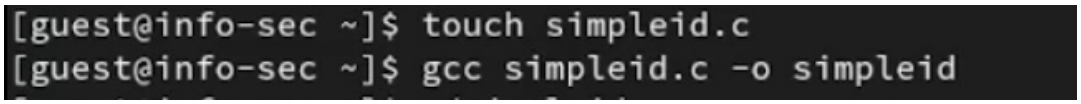

Рис. 4.1: Создание файла simleid.c

2. Выполним команды ./simpleid и id и убедимся, что полученные данные совпадают (4.2).

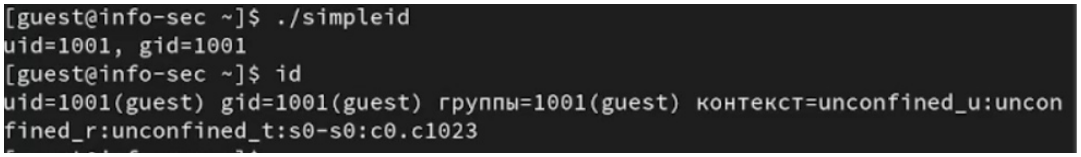

Рис. 4.2: Использование команд ./simpleid и id

3. Усложним программу и запишем ее в файл simpleid2.c. Запустим получившуюся программу (4.3).

```
[guest@info-sec ~]$ touch simpleid2.c
[guest@info-sec ~]$ gcc simpleid2.c -o simpleid2
[guest@info-sec ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Рис. 4.3: Создание и запуск программы simpleid2

4. От имени суперпользователя установим новые атрибуты и сменим владельца файла simpleid2 (4.4).

```
[guest@info-sec ~]$ su
Пароль:
[root@info-sec guest]# chown root:guest /home/guest/simpleid2
[root@info-sec guest]# chmod u+s /home/guest/simpleid2
[root@info-sec guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26064 окт  7 19:36 simpleid2
```

Рис. 4.4: Установки новых атрибутов и смена владельца файла simpleid2

5. Выполним команды ./simpleid2 и id и убедимся, что полученные данные совпадают (4.5).

```
[root@info-sec guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@info-sec guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unco
nfined_t:s0-s0:c0.c1023
[root@info-sec guest]#
```

Рис. 4.5: Использование команд ./simpleid2 и id

6. Проделаем то же самое относительно SetGID-бита (4.6).

Рис. 4.6: Операции с SetGID-битом

7. Создадим и скомпилируем программу readfile.c (4.7).

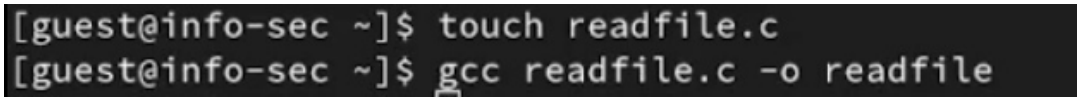

Рис. 4.7: Создание и компиляция программы readfile.c

8. Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог(4.8).

Рис. 4.8: Изменение владельца и прав файла readfile.c

9. Проверим, что пользователь guest не может прочитать файл readfile.c. (4.9).

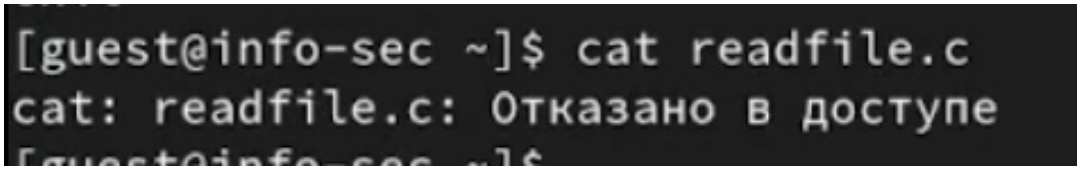

Рис. 4.9: Проверка, что пользователь guest не может прочитать файл readfile.c.

10. Сменим у программы readfile владельца и установим SetUID-бит (4.10).

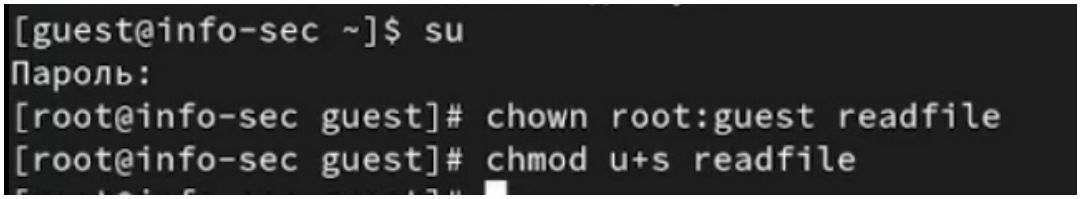

Рис. 4.10: Работа с параметрами readfile

11. Проверим, может ли программа readfile прочитать файл readfile.c (4.11).

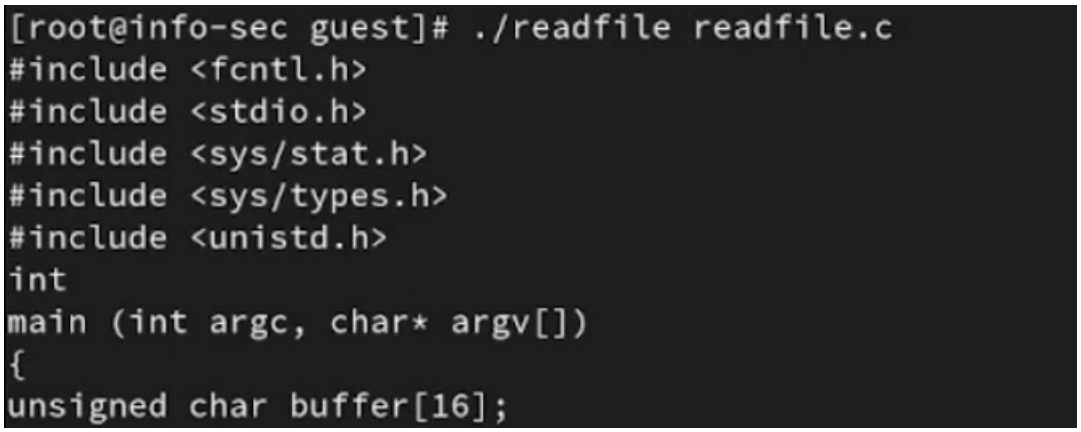

Рис. 4.11: Попытка прочитать файл readfile.c программой readfile

12. Проверим, может ли программа readfile прочитать файл /etc/shadow (4.12).

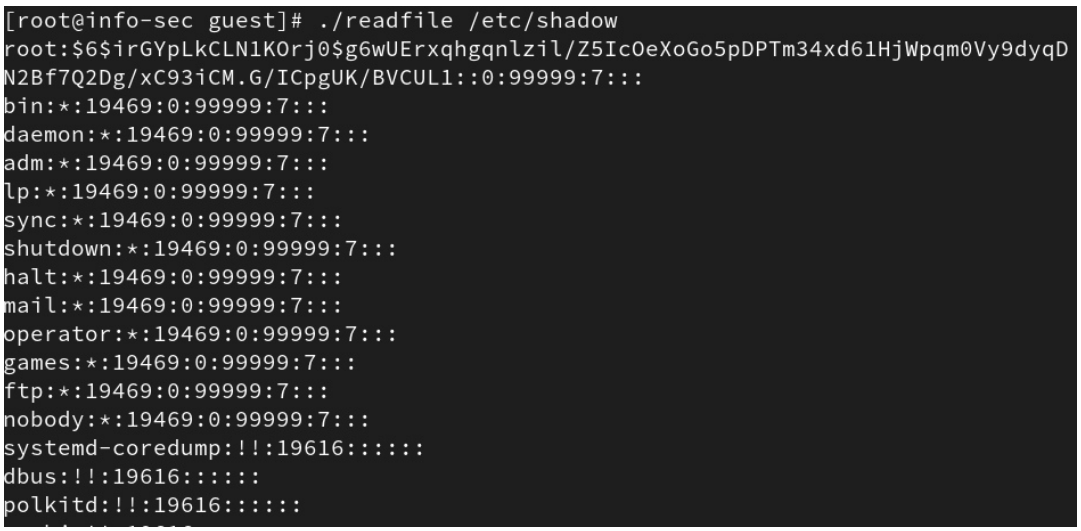

Рис. 4.12: Попытка прочитать файл /etc/shadow программой readfile

13. Выясним, установлен ли атрибут Sticky на директории /tmp (4.13).

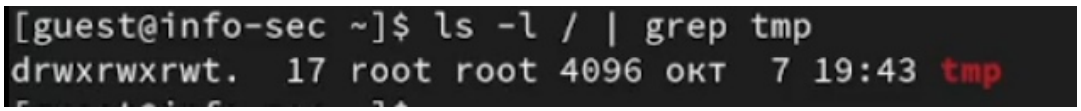

Рис. 4.13: Чтение атрибутов директории /tmp

14. От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные» (4.14).

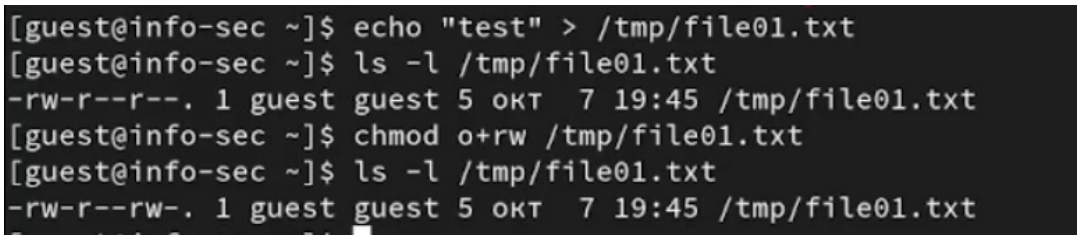

Рис. 4.14: Чтение атрибутов директории /tmp

15. От пользователя guest2 попробуем прочитать файл /tmp/file01.txt (4.15).

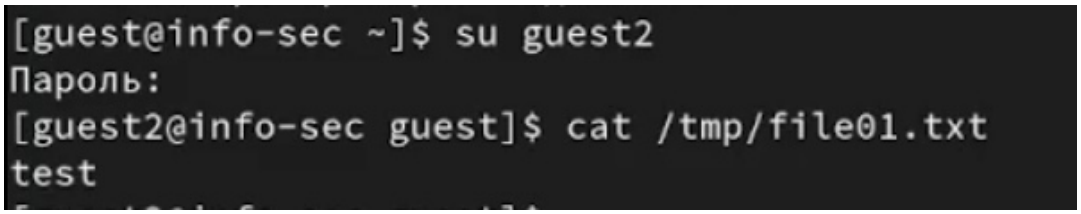

Рис. 4.15: Попытка прочтения файла /tmp/file01.txt

16. От пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2 (4.16).

Рис. 4.16: Попытка дозаписи в файл /tmp/file01.txt

17. От пользователя guest2 попробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию (4.17).

Рис. 4.17: Попытка записи в файл /tmp/file01.txt

18. От пользователя guest2 попробуем удалить файл /tmp/file01.txt (4.18).

Рис. 4.18: Попытка удаления файла /tmp/file01.txt

19. От имени суперпользователя снимем атрибут t с директории /tmp. От пользователя guest2 проверим, что атрибута t у директории /tmp нет (4.19).

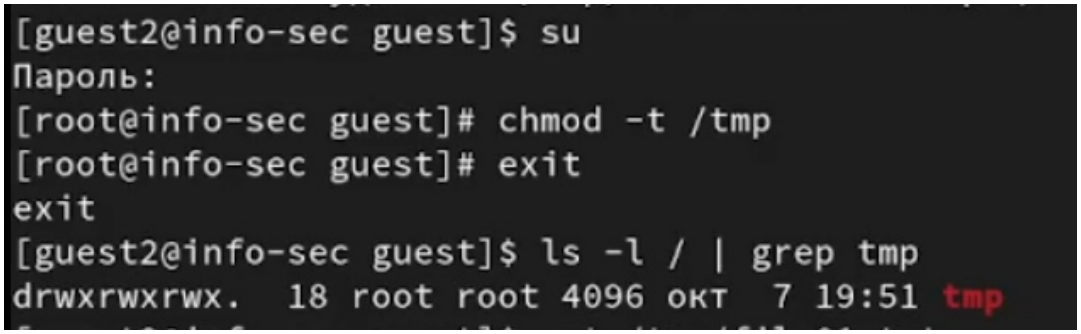

Рис. 4.19: Удаление атрибута t директории /tmp

20. Повторим предыдущие шаги. Теперь мы можем удалить файл (4.20).

Рис. 4.20: Повторение предыдущих шагов

21. Повысим свои права до суперпользователя и вернем атрибут t на директорию /tmp (4.20).

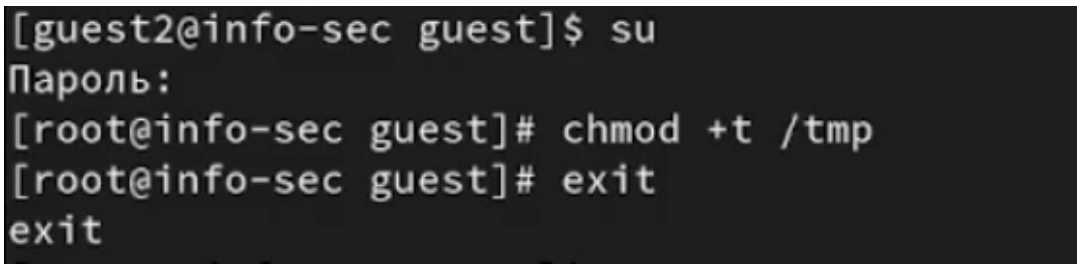

Рис. 4.21: Возвращение атрибута t директории /tmp

5 Выводы

В рамках данной лабораторной работы были изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получены практические навыков работы в консоли с дополнительными атрибутами. Рассмотрены принципы работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Список литературы

[1] https://blog.skillfactory.ru/glossary/operacztionnaya-sistema/

[2] https://codechick.io/tutorials/unix-linux/unix-linux-permissions