Отчет по лабораторной работе №5

по дисциплине: Информационная безопасность

Го Чаопэн

Содержание

# Цели 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.

# Задание

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

# 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].

# Выполнение лабораторной работы

1. От имени пользователя guest создадим программу simpleid.c, скомпилируем ее и убедимся, что файл создан. Выполним команды ./simpleid и id и убедимся, что полученные данные совпадают ([1](#fig:001)).

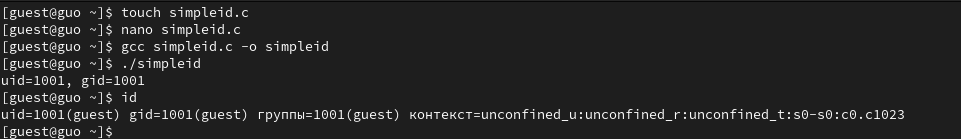


Figure 1: Использование команд ./simpleid и id

Код программы simpleid.c:

#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
int  
main ()  
{  
 uid\_t uid = geteuid ();  
 gid\_t gid = getegid ();  
 printf ("uid=%d, gid=%d\n", uid, gid);  
 return 0;  
}

1. Усложним программу и запишем ее в файл simpleid2.c. Запустим получившуюся программу ([2](#fig:002)).

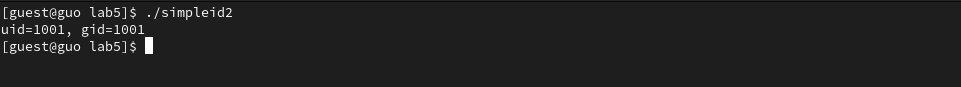


Figure 2: Запуск программы simpleid2

Код программы simpleid2.c:

#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
int  
main ()  
{  
 uid\_t real\_uid = getuid ();  
 uid\_t e\_uid = geteuid ();  
 gid\_t real\_gid = getgid ();  
 gid\_t e\_gid = getegid () ;  
 printf ("e\_uid=%d, e\_gid=%d\n", e\_uid, e\_gid);  
 printf ("real\_uid=%d, real\_gid=%d\n", real\_uid, real\_gid);  
 return 0;  
}

1. От имени суперпользователя установим новые атрибуты и сменим владельца файла simpleid2 ([3](#fig:003)).



Figure 3: Установки новых атрибутов и смена владельца файла simpleid2

1. Выполним команду ./simpleid2 ([4](#fig:004)).



Figure 4: Использование команд ./simpleid2

1. Проделаем то же самое относительно SetGID-бита ([5](#fig:005)).

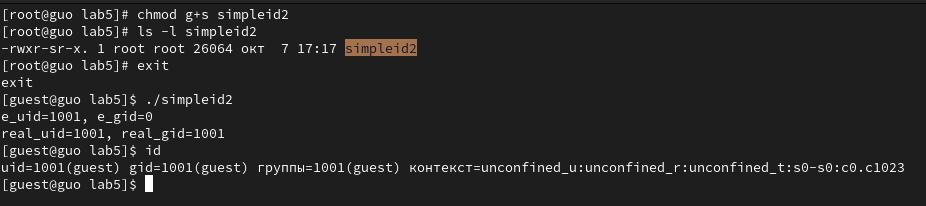


Figure 5: Операции с SetGID-битом

1. Создадим и скомпилируем программу readfile.c. Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог ([6](#fig:006)).

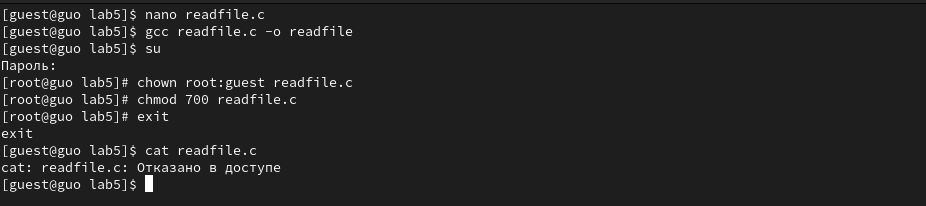


Figure 6: Изменение владельца и прав файла readfile.c

Пользователь guest не может прочитать файл readfile.c

1. Сменим у программы readfile владельца и установим SetUID-бит ([7](#fig:07)).

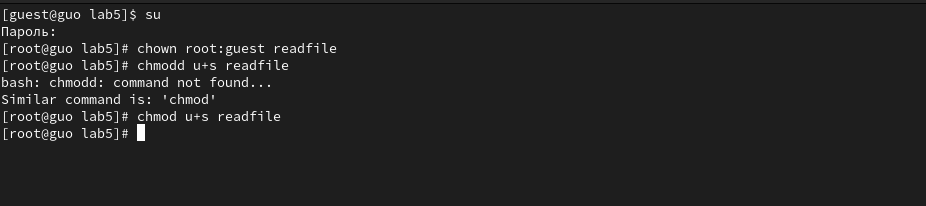


Figure 7: Работа с параметрами readfile

1. Проверим, может ли программа readfile прочитать файл readfile.c ([8](#fig:008)).



Figure 8: Попытка прочитать файл readfile.c программой readfile

1. Проверим, может ли программа readfile прочитать файл /etc/shadow ([9](#fig:009)).

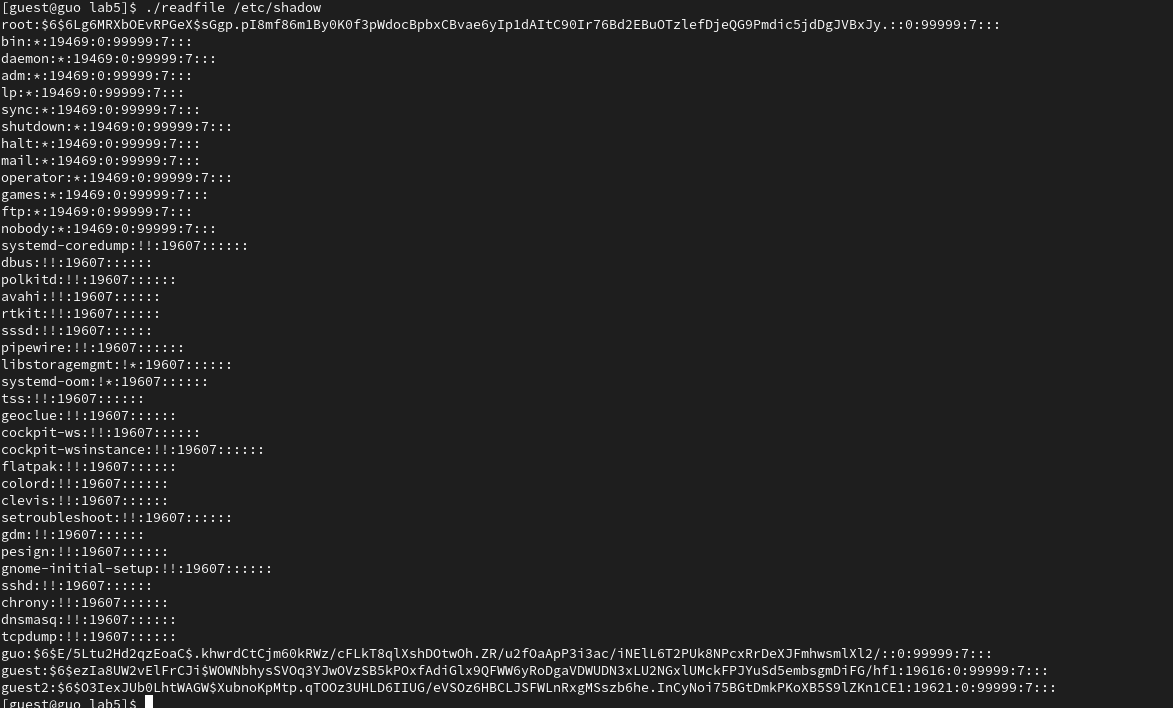


Figure 9: Попытка прочитать файл /etc/shadow программой readfile

1. Выясним, установлен ли атрибут Sticky на директории /tmp ([10](#fig:010)).

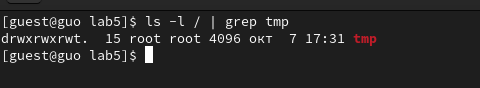


Figure 10: Чтение атрибутов директории /tmp

1. 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» ([11](#fig:011)).

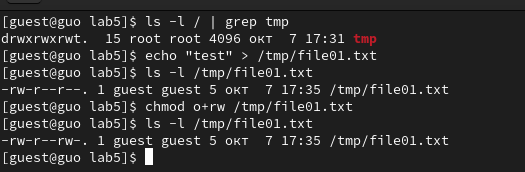


Figure 11: Создание файла /tmp/file01.txt

1. От пользователя guest2 попробуем прочитать файл /tmp/file01.txt. Далее попробуем дозаписать в файл /tmp/file01.txt слово test2, записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию. После этого попробуем удалить данных файл ([12](#fig:012)).

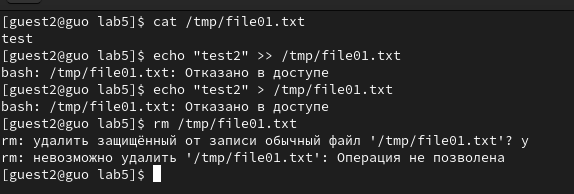


Figure 12: Работа с файлом /tmp/file01.txt

Пользователь guest2 принадлежит группе guest, поэтому у него нет доступа к вышеописанным действиям, так как у группы нет права доступа на запись для данного файла.

1. От имени суперпользователя снимем атрибут t с директории /tmp. От пользователя guest2 проверим, что атрибута t у директории /tmp нет. Повторим предыдущие шаги. Теперь мы можем удалить файл. ([13](#fig:013)).

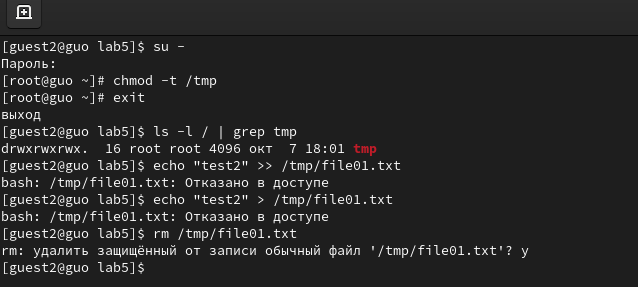


Figure 13: Удаление атрибута t директории /tmp

# 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. Операционные системы [Электронный ресурс]. URL: <https://softline.tm/solutions/programmnoe-obespechenie/operating-system>.

2. Права доступа [Электронный ресурс]. URL: <https://w.wiki/7UBB>.