Отчёт по лабораторной работе №5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

дисциплина: Информационная безопасность

Рыбалко Элина Павловна

Содержание

Цель работы **5**

Техническое обеспечение **6**

Объект/Предмет исследования **7**

Теоретическое введение **8**

Выполнение лабораторной работы **9**

1. Подготовка лабораторного стенда 9
2. Создание программы 9
3. Исследование Sticky-бита 14

Вывод **18**

Список литературы **19**

Список иллюстраций

1 Подготовка лабораторного стенда 9
2 Создание программы simpleid.c 10
3 Компиляция программы simpleid.c 10
4 Создание программы simpleid2.c 11
5 Создание программы simpleid2.c 11
6 Компиляция simpleid2.c и изменение прав 12
7 Создание программы readfile.c 12
8 Компиляция readfile.c и изменение прав 13
9 Проверка чтения файла readfile.c 14
10 Проверка чтения файла /etc/shadow 14
11 Чтение и запись файла с атрибутом Sticky-бит 16
12 Чтение и запись файла без атрибута Sticky-бит 17

Список таблиц

Цель работы

Изучить механизмы 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.

Техническое обеспечение

Лабораторная работа подразумевает наличие на виртуальной машине VirtualBox операционной системы Linux (дистрибутив Rocky или CentOS). Выполнение работы возможно как в дисплейном классе факультета физико-математических и естественных наук РУДН, так и дома. Описание выполнения работы приведено для дисплейного класса со следующими характеристиками: – Intel Core i3-550 3.2 GHz, 4 GB оперативной памяти, 20 GB свободного места на жёстком диске; – ОС Linux Gentoo (http://www.gentoo.ru/); – VirtualBox верс. 6.1 или старше; – каталог с образами ОС для работающих в дисплейном классе: /afs/dk.sci.pfu.edu.ru/common/files/iso/.

Объект/Предмет исследования

Операционная система Linux и расширенные атрибуты.

Теоретическое введение

В Linux, как и в любой многопользовательской системе, абсолютно естественным образом возникает задача разграничения доступа субъектов — пользователей к объектам — файлам дерева каталогов.

Один из подходов к разграничению доступа — так называемый дискреционный (от англ, discretion — чье-либо усмотрение) — предполагает 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 в (например, служб операционной системы), которые работают от лица псевдопользовательских учетных записей. [2].

Выполнение лабораторной работы

1. Подготовка лабораторного стенда

1. Убедиться, что в системе установлен компилятор gcc (см. рис. -@fig:001).

2. Отключите систему запретов до очередной перезагрузки системы командой setenforce 0. После этого команда getenforce должна выводить Permissive (см. рис. -@fig:001).

```
[eprybalko@eprybalko ~]$ su
Пароль:
[root@eprybalko eprybalko]# yum install gcc
Загружены модули: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.corbina.net
 * extras: mirror.corbina.net
 * updates: mirror.corbina.net
Пакет gcc-4.8.5-44.el7.x86_64 уже установлен, и это последняя версия.
Выполнять нечего
[root@eprybalko eprybalko]# setenforce 0
[root@eprybalko eprybalko]# getenforce
Permissive
```

Рис. 1: Подготовка лабораторного стенда

2. Создание программы

1. Войдите в систему от имени пользователя guest (см. рис. -@fig:002).

2. Создайте программу simpleid.c (см. рис. -@fig:002).

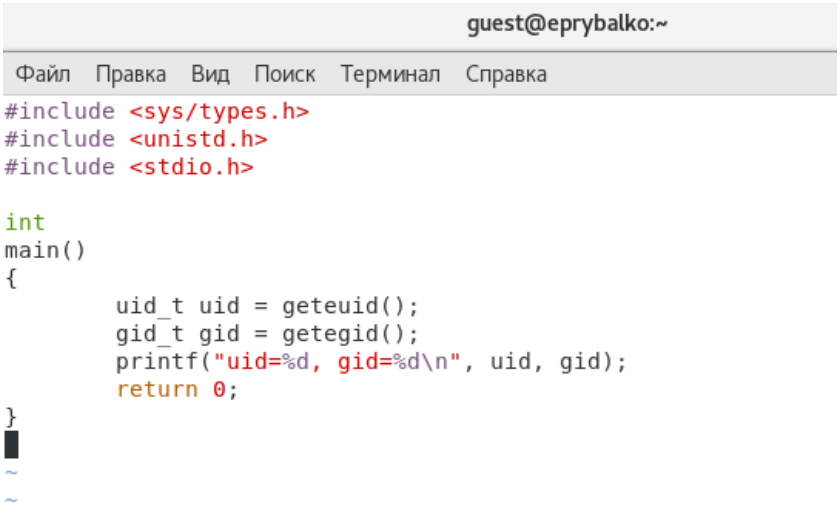

Рис. 2: Создание программы simpleid.c

3. Скомплилируйте программу и убедитесь, что файл программы создан: gcc simpleid.c -o simpleid (см. рис. -@fig:003).

4. Выполните программу simpleid: ./simpleid (см. рис. -@fig:003).

5. Выполните системную программу id: id и сравните полученный вами результат с данными предыдущего пункта задания (см. рис. -@fig:003).

Результаты совпадают.

```
[guest@eprybalko ~]$ vim simpleid.c
[guest@eprybalko ~]$ gcc simpleid.c -o simpleid
[guest@eprybalko ~]$ ./simpleid
uid=1001, gid=1001
[guest@eprybalko ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned r:unconfined t:s0-s0:c0.c1023
```

Рис. 3: Компиляция программы simpleid.c

6. Усложните программу, добавив вывод действительных идентификаторов. Получившуюся программу назовите simpleid2.c (см. рис. -@fig:004 и рис. -@fig:005).

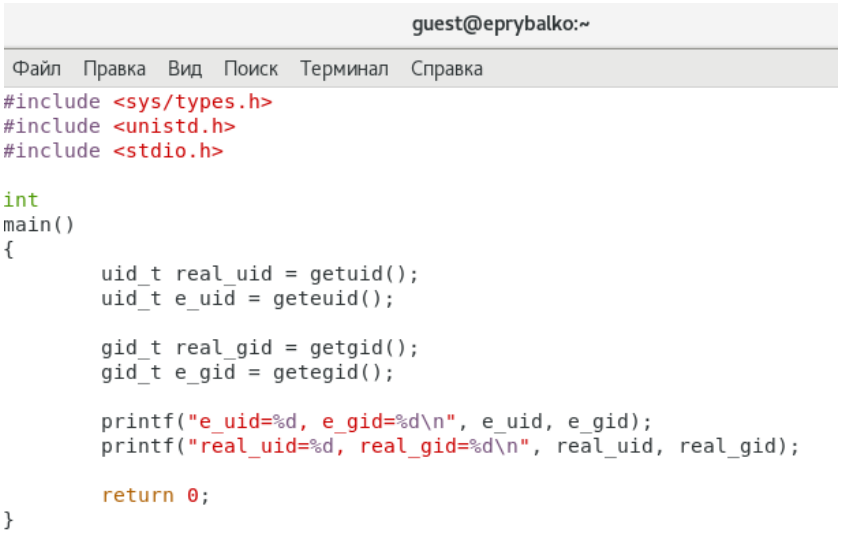

Рис. 4: Создание программы simpleid2.c

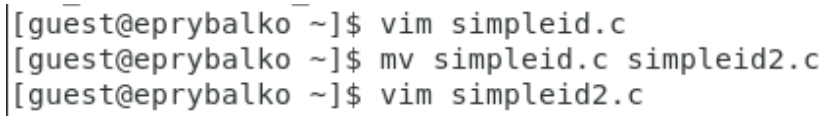

Рис. 5: Создание программы simpleid2.c

7. Скомпилируйте и запустите simpleid2.c (см. рис. -@fig:006).

8. От имени суперпользователя выполните команды (см. рис. -@fig:006).

9. Используйте sudo или повысьте временно свои права с помощью su (см. рис. -@fig:006).

Первая команда меняет владельца файла simpleid2 на группу guest. Вторая команда меняет права доступа к файлу simpleid2 для пользователя и установленные атрибуты SUID или SGID позволяют запускать файл на выполнение с правами владельца файла или группы соответственно.

10. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2 (см. рис. -@fig:006).

11. Запустите simpleid2 и id (см. рис. -@fig:006).

Результаты совпадают.

12. Проделайте тоже самое относительно SetGID-бита (см. рис. -@fig:006).

```
[guest@eprybalko ~]$ gcc simpleid2.c -o simpleid2
[guest@eprybalko ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@eprybalko ~]$ su
Пароль:
[root@eprybalko guest]# chown root:guest /home/guest/simpleid2
[root@eprybalko guest]# chmod u+s /home/guest/simpleid2
[root@eprybalko guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8616 окт  2 22:20 simpleid2
[root@eprybalko guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@eprybalko guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconf
ined_t:s0-s0:c0.c1023
[root@eprybalko guest]# chmod g+s /home/guest/simpleid2
[root@eprybalko guest]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 8616 окт  2 22:20 simpleid2
[root@eprybalko guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
```

Рис. 6: Компиляция simpleid2.c и изменение прав

13. Создайте программу readfile.c (см. рис. -@fig:007).

guest@eprybalko:/home/guest

Файл Правка Вид Поиск Терминал Справка

```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. 7: Создание программы readfile.c

14. Откомпилируйте её (см. рис. -@fig:008).

15. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (см. рис. -@fig:008).

16. Проверьте, что пользователь guest не может прочитать файл readfile.c (см. рис. -@fig:008).

17. Смените у программы readfile владельца и установите SetU'D-бит (см. рис. -@fig:008).

```
[root@eprybalko guest]# vim readfile.c
[root@eprybalko guest]# gcc readfile.c -o readfile
[root@eprybalko guest]# chown root readfile.c
[root@eprybalko guest]# chmod og-rwx readfile.c
[root@eprybalko guest]# exit
exit
[guest@eprybalko ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@eprybalko ~]$ su
Пароль:
[root@eprybalko guest]# chmod u+s /home/guest/readfile
```

Рис. 8: Компиляция readfile.c и изменение прав

18. Проверьте, может ли программа readfile прочитать файл readfile.c (см. рис. -@fig:009).

```
[guest@eprybalko ~]$ ./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. 9: Проверка чтения файла readfile.c

19. Проверьте, может ли программа readfile прочитать файл /etc/shadow (см. рис. -@fig:010).

Программа может прочитать файлы.

```
[guest@eprybalko ~]$ ./readfile /etc/shadow
root:$6$.sApTts4CMn/3CU9$1IfwWbcdWXKy2bhxOFeDg7M3ss7K9bfXM8fcTea.3ksu.LXtFIb0byj
m/ssiTWsgboUyU7pSuOX9IX4eucvcZ0::0:99999:7:::
bin:*:18353:0:99999:7:::
daemon:*:18353:0:99999:7:::
adm:*:18353:0:99999:7:::
lp:*:18353:0:99999:7:::
sync:*:18353:0:99999:7:::
shutdown:*:18353:0:99999:7:::
```

Рис. 10: Проверка чтения файла /etc/shadow

3. Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду ls -l / | grep tmp (см. рис. -@fig:011).

2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test (см. рис. -@fig:011).

3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные» (см. рис. -@fig:011).

4. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt (см. рис. -@fig:011).

5. От пользователя guest2 попробуйте дозаписать в файл (см. рис. -@fig:011).

Выполнить операцию удалось.

6. Проверьте содержимое файла командой cat /tmp/file01.txt (см. рис. -@fig:011).

7. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo “test3” > /tmp/file01.txt (см. рис. -@fig:011).

Выполнить операцию удалось.

8. Проверьте содержимое файла командой cat /tmp/file01.txt (см. рис. -@fig:011).

9. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой rm /tmp/fileOl.txt (см. рис. -@fig:011).

Выполнить операцию не удалось.

10. Повысьте свои права до суперпользователя следующей командой su и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp (см. рис. -@fig:011).

11. Покиньте режим суперпользователя командой exit (см. рис. -@fig:011).

```
[guest@eprybalko ~]$ ls -l / | grep tmp
drwxrwxrwt.  18 root root 4096 окт  2 23:54 tmp
[guest@eprybalko ~]$ echo "test" > /tmp/file01.txt
[guest@eprybalko ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт  2 23:55 /tmp/file01.txt
[guest@eprybalko ~]$ chmod o+rw /tmp/file01.txt
[guest@eprybalko ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 окт  2 23:55 /tmp/file01.txt
[guest@eprybalko ~]$ su guest2
Пароль:
[guest2@eprybalko guest]$ cat /tmp/file01.txt
test
[guest2@eprybalko guest]$ echo "test2" > /tmp/file01.txt
[guest2@eprybalko guest]$ cat /tmp/file01.txt
test2
[guest2@eprybalko guest]$ echo "test3" > /tmp/file01.txt
[guest2@eprybalko guest]$ cat /tmp/file01.txt
test3
[guest2@eprybalko guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
[guest2@eprybalko guest]$ su -
Пароль:
Последний вход в систему:Вс окт  2 22:38:33 MSK 2022на pts/1
[root@eprybalko ~]# chmod -t /tmp
[root@eprybalko ~]# exit
logout
```

Рис. 11: Чтение и запись файла с атрибутом Sticky-бит

12. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет (см. рис. -@fig:012).

13. Повторите предыдущие шаги (см. рис. -@fig:012).

Удалось перезаписать файл и также удалить его.

14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? (см. рис. -@fig:012).

Выполнить операцию удалось.

15. Повысьте свои права до суперпользователя и верните атрибут t на директорию /tmp (см. рис. -@fig:012).

```
[guest2@eprybalko guest]$ ls -l / | grep tmp
drwxrwxrwx.  18 root root 4096 окт  2 23:59 tmp
[guest2@eprybalko guest]$ cat /tmp/file01.txt
test3
[guest2@eprybalko guest]$ echo "test2" > /tmp/file01.txt
[guest2@eprybalko guest]$ cat /tmp/file01.txt
test2
[guest2@eprybalko guest]$ echo "test3" > /tmp/file01.txt
[guest2@eprybalko guest]$ cat /tmp/file01.txt
test3
[guest2@eprybalko guest]$ rm /tmp/file01.txt
[guest2@eprybalko guest]$ su -
Пароль:
Последний вход в систему:Вс окт  2 23:59:30 MSK 2022на pts/1
[root@eprybalko ~]# chmod +t /tmp
[root@eprybalko ~]# exit
logout
[guest2@eprybalko guest]$ ls -l / | grep tmp
drwxrwxrwt.  19 root root 4096 окт  3 00:01 tmp
```

Рис. 12: Чтение и запись файла без атрибута Sticky-бит

Вывод

Изучили механизмы изменения идентификаторов, применения SetUID- 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

1. Лабораторная работа №5
2. Дискреционное разграничение доступа Linux
3. Руководство по формуле Cmd Markdown
4. Руководство по оформлению Markdown файлов