

Лабораторная работа №5

Дискреционное разграничение прав в Linux. Основные атрибуты

Выполнила: Накова Амина Михайловна, НПИбд-02-23

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 ###
Выполнение работы

1. Войдите в систему от имени пользователя guest. Создайте программу simpleid.c: /home/guest/dir1/file1 командой ([рис. @fig-001]).

```
[amina@localhost ~]$ nano simpleid.c
[amina@localhost ~]$ gcc simpleid.c -o simpleid
bash: gcc: команда не найдена...
Установить пакет «gcc», предоставляющий команду «gcc»? [N/y] y

* Ожидание в очереди...
* Загрузка списка пакетов...
Следующие пакеты должны быть установлены:
gcc-11.5.0-5.el9_5.x86_64      Various compilers (C, C++, Objective-C, ...)
glibc-devel-2.34-168.el9_6.23.x86_64  Object files for development using stand
ard C libraries.
glibc-headers-2.34-168.el9_6.23.x86_64  Additional internal header files for gli
bc-devel.
kernel-headers-5.14.0-570.39.1.el9_6.x86_64  Header files for the Linux kerne
l for use by glibc
libxcrypt-devel-4.4.18-3.el9.x86_64  Development files for libxcrypt
make-1:4.3-8.el9.x86_64      A GNU tool which simplifies the build process fo
r users
```

Figure 1: рис.1.1

```
[amina@localhost ~]$ gcc simpleid.c -o simpleid
[amina@localhost ~]$ ./simpleid
uid=1000, gid=1000
[amina@localhost ~]$ id
uid=1000(amina) gid=1000(amina) группы=1000(amina),10(wheel) контекст=unconfined
u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Figure 2: рис.2.1

2. Скомпилируйте программу и убедитесь, что файл программы создан: gcc simpleid.c -o simpleid. Выполните программу simpleid: ./simpleid. Выполните системную программу id: id и сравните полученный вами результат

с данными предыдущего пункта задания. Усложните программу, добавив вывод действительных идентификаторов ([рис. @fig-002]).

```
[amina@localhost ~]$ gcc simpleid2.c -o simpleid2
[amina@localhost ~]$ ./simpleid2
e_uid=1000, e_gid=1000
real_uid=1000, real_gid=1000
[amina@localhost ~]$ chown root:guest /home/guest/simpleid2
chown: невозможно получить доступ к '/home/guest/simpleid2': Нет такого файла и
и каталога
[amina@localhost ~]$ sudo chown root:guest /home/guest/simpleid2
[sudo] пароль для амина:
chown: невозможно получить доступ к '/home/guest/simpleid2': Нет такого файла и
и каталога
[amina@localhost ~]$ sudo -i
[root@localhost ~]# sudo chown root:guest /home/guest/simpleid2
chown: невозможно получить доступ к '/home/guest/simpleid2': Нет такого файла и
и каталога
[root@localhost ~]# sudo chown root:amina /home/amina/simpleid2
[root@localhost ~]# sudo chmod u+s /home/amina/simpleid2
[root@localhost ~]# ls -l simpleid2
ls: невозможно получить доступ к 'simpleid2': Нет такого файла или каталога
```

Figure 3: рис.3.1

3. Получившуюся программу назовите `simpleid2.c`. Скомпилируйте и запустите `simpleid2.c`: `gcc simpleid2.c -o simpleid2 ./simpleid2`. От имени суперпользователя выполните команды: `chown root:guest /home/guest/simpleid2`, `chmod u+s /home/guest/simpleid2`. Используйте `sudo` или повысьте временно свои права с помощью `su`. Поясните, что делают эти команды. Выполните проверку правильности установки новых атрибутов и смены владельца файла `simpleid2`: `ls -l simpleid2`. Запустите `simpleid2` и `id`: `./simpleid2 id`. Сравните результаты. Прodelайте тоже самое относительно SetGID-бита([рис. @fig-003]).

```
ls: невозможно получить доступ к 'simpleid2': Нет такого файла или каталога
[root@localhost ~]# su - amina
[amina@localhost ~]$ ls -l simpleid2
-rwsr-xr-x. 1 root amina 17656 сен 12 20:04 simpleid2
[amina@localhost ~]$ ./simpleid2
e_uid=0, e_gid=1000
real_uid=1000, real_gid=1000
[amina@localhost ~]$ id
uid=1000(amina) gid=1000(amina) группы=1000(amina),10(wheel) контекст=unconfined
u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[amina@localhost ~]$ nano readfile.c
[amina@localhost ~]$ sudo -i
[root@localhost ~]# su - amina
[amina@localhost ~]$ gcc readfile.c -o readfile
[amina@localhost ~]$ sudo chown root:root /home/amina/readfile
[amina@localhost ~]$ ls -l readfile
-rwxr-xr-x. 1 root root 17600 сен 12 20:09 readfile
[amina@localhost ~]$ sudo chown root:guest /home/amina/readfile
[amina@localhost ~]$ ls -l readfile
```

Figure 4: рис.4.1

4. Откомпилируйте её. gcc readfile.c -o readfile 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. Проверьте, что пользователь guest не может прочитать файл readfile.c. Смените у программы readfile владельца и установите SetU'D-бит. Проверьте, может ли программа readfile прочитать файл readfile.c? (рис. @fig-004)).

```
[amina@localhost ~]$ echo "test" > /tmp/file01.txt
[amina@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 amina amina 5 сен 12 20:13 /tmp/file01.txt
[amina@localhost ~]$ chmod o+rw /tmp/file01.txt
[amina@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 amina amina 5 сен 12 20:13 /tmp/file01.txt
[amina@localhost ~]$ su - guest
Пароль:
[guest@localhost ~]$ cat /tmp/file01.txt
test
[guest@localhost ~]$ echo "test2" > /tmp/file01.txt
[guest@localhost ~]$ ls -l file01.txt
ls: невозможно получить доступ к 'file01.txt': Нет такого файла или каталога
[guest@localhost ~]$ cat /tmp/file01.txt
test2
[guest@localhost ~]$
```

Figure 5: рис.5.1

5. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду ls -l / | grep tmp От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test: echo "test" > /tmp/file01.txt Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»: ls -l /tmp/file01.txt, chmod o+rw /tmp/file01.txt, ls -l /tmp/file01.txt. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt: cat /tmp/file01.txt. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой echo "test2" > /tmp/file01.txt. Удалось ли вам выполнить операцию? Проверьте содержимое файла командой cat /tmp/file01.txt. 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. Удалось ли вам выполнить операцию? Проверьте содержимое файла командой cat /tmp/file01.txt (рис. @fig-005)).

Выводы: В ходе лабораторной работы были изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Были получены практические навыки работы с дополнительными

атрибутами файлов в Linux.

Исследование показало, что:

SetUID-бит позволяет программе выполняться с правами владельца файла

Sticky-бит предотвращает удаление файлов в общих директориях пользователями, не являющимися владельцами этих файлов

Эти механизмы обеспечивают дополнительный уровень безопасности в системе

Работа позволила лучше понять механизмы дискреционного разграничения доступа в Linux и их практическое применение.