

Отчет по лабораторной работе №5

-

Овениязов Артур

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	ВАЖНО!!! Пунктов в отчете было бы необычайного много если от- вечать скриншотом на каждый, поэтому многие пункты были объединены для более легкого составления отчета	8
4.2	Часть первая	8
4.3	Часть вторая	10
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Результат компилирования simpleid и id	8
4.2	Результат компилирования simpleid2	9
4.3	Результат chmod, chown, simpleid2, id	9
4.4	Результат компилирования readfile.c	9
4.5	Смена прав и владельца readfile.c	9
4.6	Проверка чтения файла	10
4.7	Обратная смена владельца файла	10
4.8	Проверка чтения	10
4.9	Результат chmod и ls -l /tmp/file01.txt	11
4.10	Результат cat и echo	12
4.11	Результат chmod -t /tmp	12
4.12	Результат su/chmod +t	13

Список таблиц

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

Лабораторная работа подразумевает практическое исследование дискреционных разграничений в современных системах с открытым кодом на базе ОС Linux, а именно изучение атрибутов для групп пользователей.

3 Теоретическое введение

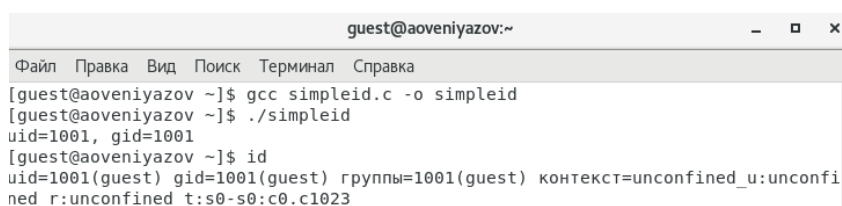
В Linux, как и в любой многопользовательской системе, абсолютно естественным образом возникает задача разграничения доступа субъектов — пользователей к объектам — файлам дерева каталогов. Один из подходов к разграничению доступа — так называемый дискреционный (от англ, discretion — чье-либо усмотрение) — 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. [1]

4 Выполнение лабораторной работы

4.1. ВАЖНО!!! Пунктов в отчете было бы необычайного много если отвечать скриншотом на каждый, поэтому многие пункты были объединены для более легкого составления отчета .

4.2. Часть первая

Войдите в систему от имени пользователя guest.Создайте программу simpleid.c. 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. Программа и команда выдают одинаковые результаты.



```
guest@aoveniyazov:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@aoveniyazov ~]$ gcc simpleid.c -o simpleid  
[guest@aoveniyazov ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@aoveniyazov ~]$ id  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 4.1: Результат компилирования simpleid и id

Усложните программу, добавив вывод действительных идентификаторов,скомпилируйте и запустите simpleid2.c: gcc simpleid2.c -o simpleid2

./simpleid2

```
[guest@aoveniyazov ~]$ gcc simpleid2.c -o simpleid2
[guest@aoveniyazov ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aoveniyazov ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aoveniyazov ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 4.2: Результат компилирования simpleid2

От имени суперпользователя выполните команды `chown root:guest /home/guest/simpleid2` `chmod u+s /home/guest/simpleid2` `Chmod u+s` наделяет каждого пользователя правами владельца(кто пытается получить к нему доступ), а `chown` меняет владельца. Выполните проверку правильности установки новых атрибутов и смены владельца файла `simpleid2`: `ls -l simpleid2` Запустите `simpleid2` и `id`: `./simpleid2 id` Результаты совпадают.

```
[guest@aoveniyazov ~]$ su
Пароль:
[root@aoveniyazov guest]# chown root:guest /home/guest/simpleid2
[root@aoveniyazov guest]# chmod u+s /home/guest/simpleid2
[root@aoveniyazov guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8576 ноя 12 04:36 simpleid2
[root@aoveniyazov guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@aoveniyazov guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconf
ined_t:s0-s0:c0.c1023
```

Рис. 4.3: Результат chmod, chown, simpleid2, id

Создайте программу `readfile.c`.Откомпилируйте её. `gcc readfile.c -o readfile`

```
[guest@aoveniyazov ~]$ gcc readfile.c -o readfile
[guest@aoveniyazov ~]$ chown root:guest /home/guest/readfile.c
chown: изменение владельца «/home/guest/readfile.c»: Операция не позволена
```

Рис. 4.4: Результат компилирования readfile.c

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

```
[root@aoveniyazov guest]# chown root /home/guest/readfile.c
[root@aoveniyazov guest]# chmod 700 /home/guest/readfile.c
```

Рис. 4.5: Смена прав и владельца readfile.c

Проверьте, что пользователь guest не может прочитать файл readfile.c.

```
[guest@aoveniyazov ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
```

Рис. 4.6: Проверка чтения файла

Смените у программы readfile владельца и установите SetU'D-бит

```
[root@aoveniyazov guest]# chown guest /home/guest/readfile.c
[root@aoveniyazov guest]#
```

Рис. 4.7: Обратная смена владельца файла

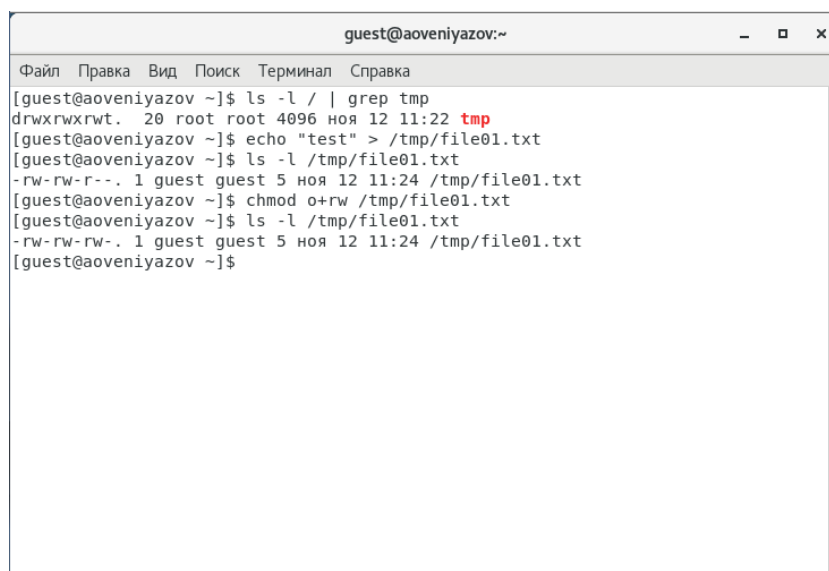
Проверьте, может ли программа readfile прочитать файл readfile.c? Проверьте, может ли программа readfile прочитать файл /etc/shadow? Чтение обоих файлов не удалось.

```
[guest@aoveniyazov ~]$ ./readfile readfile.c
bash: ./readfile: Отказано в доступе
[guest@aoveniyazov ~]$ ./readfile /etc/shadow
bash: ./readfile: Отказано в доступе
```

Рис. 4.8: Проверка чтения

4.3. Часть вторая

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`



```
guest@aoveniyazov:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@aoveniyazov ~]$ ls -l / | grep tmp  
drwxrwxrwt. 20 root root 4096 ноя 12 11:22 tmp  
[guest@aoveniyazov ~]$ echo "test" > /tmp/file01.txt  
[guest@aoveniyazov ~]$ ls -l /tmp/file01.txt  
-rw-rw-r--. 1 guest guest 5 ноя 12 11:24 /tmp/file01.txt  
[guest@aoveniyazov ~]$ chmod o+rw /tmp/file01.txt  
[guest@aoveniyazov ~]$ ls -l /tmp/file01.txt  
-rw-rw-rw-. 1 guest guest 5 ноя 12 11:24 /tmp/file01.txt  
[guest@aoveniyazov ~]$
```

Рис. 4.9: Результат chmod и ls -l /tmp/file01.txt

От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt: cat /tmp/file01.txt5., попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой echo "test2" > /tmp/file01.txt. 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 Проверьте содержимое файла командой cat /tmp/file01.txt От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой rm /tmp/file01.txt Все операции были выполнены успешно.

```

[guest2@aoveniyazov guest]$ cat /tmp/file01.txt
test
[guest2@aoveniyazov guest]$ echo "test2" > /tmp/file01.txt
[guest2@aoveniyazov guest]$ cat /tmp/file01.txt
test2
[guest2@aoveniyazov guest]$ echo "test3" > /tmp/file01.txt
[guest2@aoveniyazov guest]$ cat /tmp/file01.txt
test3
[guest2@aoveniyazov guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Нет такого файла или каталога
[guest2@aoveniyazov guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Нет такого файла или каталога
[guest2@aoveniyazov guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Нет такого файла или каталога
[guest2@aoveniyazov guest]$ cat /tmp/file01.txt
test3
[guest2@aoveniyazov guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена

```

Рис. 4.10: Результат cat и echo

Повысьте свои права до суперпользователя следующей командой `su -` и выполните после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`: `chmod -t /tmp`. Покиньте режим суперпользователя командой `exit`. От пользователя `guest2` проверьте, что атрибута `t` у директории `/tmp` нет: `ls -l / | grep tmp`. Повторите предыдущие шаги. Какие наблюдаются изменения? Ваши наблюдения занесите в отчёт - все удалось.

```

guest2@aoveniyazov:/home/guest
Файл  Правка  Вид  Поиск  Терминал  Справка
[guest2@aoveniyazov guest]$ su
Пароль:
[root@aoveniyazov guest]# chmod -t /tmp
[root@aoveniyazov guest]# exit
exit
[guest2@aoveniyazov guest]$ ls -l / | grep tmp
drwxrwxrwx. 20 root root 4096 ноя 12 04:58 tmp
[guest2@aoveniyazov guest]$ cat /tmp/file01.txt
test3
[guest2@aoveniyazov guest]$ echo "test4" > /tmp/file01.txt
[guest2@aoveniyazov guest]$ cat /tmp/file01.txt
test4
[guest2@aoveniyazov guest]$ rm /tmp/file01.txt
[guest2@aoveniyazov guest]$ ls /tmp
snap.onlyoffice-desktopeditors
ssh-VHUF\ztFH2Ae
ssh-W6QY0ZHNRsMV
systemd-private-9c1603ff491746ac86750c450d3d8eb3-bolt.service-qEMtCw
systemd-private-9c1603ff491746ac86750c450d3d8eb3-chronyd.service-eWDz0v
systemd-private-9c1603ff491746ac86750c450d3d8eb3-colord.service-8zqaWq
systemd-private-9c1603ff491746ac86750c450d3d8eb3-cups.service-CXuK2e
systemd-private-9c1603ff491746ac86750c450d3d8eb3-fwupd.service-ELYFf7
systemd-private-9c1603ff491746ac86750c450d3d8eb3-rtkit-daemon.service-18Qg02
tracker-extract-files.1000
tracker-extract-files.1001

```

Рис. 4.11: Результат `chmod -t /tmp`

Повысьте свои права до суперпользователя и верните атрибут `t` на директо-

рию /tmp: su - chmod +t /tmp exit

```
[guest2@aoveniyazov guest]$ su
Пароль:
[root@aoveniyazov guest]# chmod +t /tmp
[root@aoveniyazov guest]# exit
exit
[guest2@aoveniyazov guest]$ ls -l / | grep tmp
drwxrwxrwt. 20 root root 4096 ноя 12 05:03 tmp
```

Рис. 4.12: Результат su/chmod +t

5 Выводы

Сегодня я приобрел практические навыки работы в консоли с атрибутами файлов, закрепил теоретические основы дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux и изучил механизмы изменения идентификаторов применения SetUID- и Sticky-битов.

Список литературы

1. Дискреционное разграничение доступа Linux [Электронный ресурс]. Сайт, 2021. URL: <http://debianinstall.ru/diskretionnoe-razgranichenie-dostupa-linux/>.