

Отчет по лабораторной работе №2

Дискреционное разграничение прав в Linux. Основные атрибуты

Татьяна Александровна Буллер

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Создание пользователей	6
2.2	Вход в систему от имени созданного пользователя	7
2.3	Пользователь в системе	7
2.4	Права директорий и файлов	9
3	Выводы	13

Список таблиц

2.1 Права на каталог и файл в нем	11
---	----

Список иллюстраций

2.1	Создание пользователя	6
2.2	Пароль для пользователя	6
2.3	Вход в систему и рабочая директория	7
2.4	whoami	7
2.5	Группы пользователя	8
2.6	Содержимое файла /etc/passwd	8
2.7	Домашние директории пользователей	8
2.8	Атрибуты директорий	9
2.9	Создание директории и права на нее	9
2.10	Атрибуты директорий внутри домашнего каталога пользователя .	10
2.11	У гостей нет прав	10
2.12	У гостей нет прав создать файл	10
2.13	У гостей нет прав читать директорию	11

1 Цель работы

Получение практических навыков работы в консоли с атрибутами файлов, закрепление теоретических основ дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux

2 Выполнение лабораторной работы

2.1 Создание пользователей

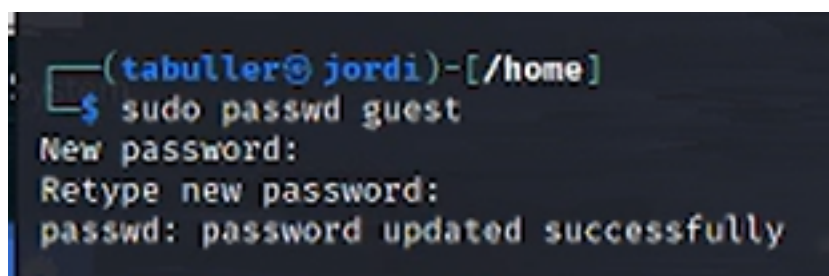
Для создания пользователей используем команду `useradd` с опцией `-m`, чтобы вместе с пользователем создать его домашнюю директорию и в дальнейшем иметь возможность зайти на рабочий стол под этой учетной записью.



```
(tabuller@jordi)-[/home]
$ sudo useradd -m guest
```

Рис. 2.1: Создание пользователя

Далее задаем пароль для пользователя, который необходимо подтвердить, введя повторно.



```
(tabuller@jordi)-[/home]
$ sudo passwd guest
New password:
Retype new password:
passwd: password updated successfully
```

Рис. 2.2: Пароль для пользователя

2.2 Вход в систему от имени созданного пользователя

С помощью команды `su guest` получаем неинтерактивную оболочку пользователя и переводим ее в интерактивную: в таком виде с ней удобнее работать. Команда `pwd` помогает определить, что в данный момент мы находимся в домашней директории созданного ранее пользователя.

```
$ pwd
/home/guest
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
(guest@jordi)-[~]
```

Рис. 2.3: Вход в систему и рабочая директория

Команда `whoami` выводит в терминал имя пользователя, от имени которого она была запущена. В нашем случае - `guest`.

```
(guest@jordi)-[~]
$ whoami
guest
```

Рис. 2.4: `whoami`

2.3 Пользователь в системе

Узнать, в каких группах состоит пользователь, можно двумя способами: командой `id` (выводит `id` пользователя и его имя, `id` группы пользователя и ее названия, а также `id` и названия прочих групп, куда входит данный пользователь) либо командой `groups`. Последняя, в отличие от первой, выводит на экран только названия групп.

```

(guest@jordi)-[~]
$ groups
guest

(guest@jordi)-[~]
$ id
uid=1002(guest) gid=1002(guest) groups=1002(guest)

```

Рис. 2.5: Группы пользователя

Запросив содержимое файла `/etc/passwd` командой `cat`, мы можем видеть всех пользователей машины, их `uid` и `gid`. Для созданного пользователя `uid` и `gid` совпадают: 1002.

```

jordi:x:1000:1000:jordi,,,:/home/jordi:/usr/bin/zsh
tabuller:x:1001:1001:::/home/tabuller:/bin/bash
guest:x:1002:1002::/home/guest:/bin/sh

```

Рис. 2.6: Содержимое файла `/etc/passwd`

Командой `ls -l /home/` определим существующие в системе домашние директории пользователей. Видим, что их три: `guest`, созданный по ходу выполнения лабораторной работы, `tabuller` - основной пользователь для выполнения лабораторных работ и `jordi`. Кто он и зачем существует - дело десятое. `jordi`. В каждой из директорий владелец имеет право на чтение, запись и исполнение. В директориях `tabuller` и `jordi` право на чтение и исполнение файлов также имеют участники группы и остальные пользователи машины.

```

(guest@jordi)-[~]
$ ls -l /home/
total 12
drwx----- 15 guest    guest    4096 Feb 14 05:26 guest
drwxr-xr-x 20 jordi    jordi    4096 Feb 14 03:32 jordi
drwxr-xr-x 19 tabuller tabuller 4096 Feb 14 05:26 tabuller

```

Рис. 2.7: Домашние директории пользователей

С помощью `lsattr` можем увидеть, что у каждой из домашних директорий установлен атрибут `e`.


```
(guest@jordi)-[~]
$ lsattr /home
-----e----- /home/jordi
-----e----- /home/tabuller
-----e----- /home/guest
```

Рис. 2.8: Атрибуты директорий

2.4 Права директорий и файлов

В домашней директории пользователя guest создадим директорию dir1. Можно увидеть, что сразу после создания директория выдает все права владельцу и права на чтение и исполнение членам группы и прочим пользователям.

```
(guest@jordi)-[~]
$ mkdir dir1

(guest@jordi)-[~]
$ ls -l dir1/
total 0

(guest@jordi)-[~]
$ ls -l dir1
total 0

(guest@jordi)-[~]
$ ls -l
total 36
drwxr-xr-x 2 guest guest 4096 Feb 14 05:26 Desktop
drwxr-xr-x 2 guest guest 4096 Feb 14 05:30 dir1
drwxr-xr-x 2 guest guest 4096 Feb 14 05:26 Documents
drwxr-xr-x 2 guest guest 4096 Feb 14 05:26 Downloads
drwxr-xr-x 2 guest guest 4096 Feb 14 05:26 Music
drwxr-xr-x 2 guest guest 4096 Feb 14 05:26 Pictures
drwxr-xr-x 2 guest guest 4096 Feb 14 05:26 Public
drwxr-xr-x 2 guest guest 4096 Feb 14 05:26 Templates
drwxr-xr-x 2 guest guest 4096 Feb 14 05:26 Videos
```

Рис. 2.9: Создание директории и права на нее

По умолчанию также у всех директорий внутри домашней установлен атрибут е: то же самое, что мы видели и в директории /home

```
(guest@jordi)-[~]
$ lsattr
-----e----- ./Pictures
-----e----- ./Music
-----e----- ./Videos
-----e----- ./Desktop
-----e----- ./Public
-----e----- ./dir1
```

Рис. 2.10: Атрибуты директорий внутри домашнего каталога пользователя

Снимем все права с созданной нами директории. Можем видеть, что вывод команды `ls -l` теперь показывает только то, что перед нами директория. Никто не имеет права ни на чтение, ни на запись, ни на исполнение файлов. Владелец в том числе.

```
(guest@jordi)-[~]
$ chmod 000 dir1/

(guest@jordi)-[~]
$ ls -l
total 36
drwxr-xr-x 2 guest guest 4096 Feb 14 05:26 Desktop
d----- 2 guest guest 4096 Feb 14 05:30 dir1
```

Рис. 2.11: У гостей нет прав

При попытке создать файл внутри директории и перенаправить в него вывод команды `echo` сталкиваемся с ошибкой: на эту директорию у пользователя больше нет прав.

```
(guest@jordi)-[~]
$ echo "test" > dir1/file
bash: dir1/file: Permission denied
```

Рис. 2.12: У гостей нет прав создать файл

При попытке прочитать содержимое директории тоже ничего не увидим: права на чтение сняты вместе со всеми остальными.

```
(guest@jordi)-[~]
$ ls -l dir1/
ls: cannot open directory 'dir1/': Permission denied
```

Рис. 2.13: У гостей нет прав читать директорию

Заполним таблицу о правах доступа владельца на основе полученной информации:

Таблица 2.1: Права на каталог и файл в нем

Права дирек- тории	Права файла	Созда- ние файла	Удале- ние файла	За- пись в файл	Чте- ние файла	Смена дирек- тории	Про- смотр фай- лов в дирек- тории	Пере- имено- вание файла
000	000	-	-	-	-	-	-	-
100	000	-	-	-	-	+	-	-
200	000	+	+	-	-	-	-	+
300	000	+	+	-	-	+	-	+
400	000	-	-	-	-	+	-	-
500	000	+	+	-	-	+	-	+
600	000	+	+	-	-	-	+	+
700	000	+	+	-	-	+	+	-
100	200	-	-	+	-	+	-	-
200	200	+	+	+	-	-	-	+
300	200	+	+	+	-	+	-	+
400	200	-	-	+	-	+	+	-
500	200	+	+	+	-	-	+	-
600	200	+	+	+	-	-	+	+
700	200	+	+	+	-	+	+	+

Права дирек- тории	Права файла	Созда- ние файла	Удале- ние файла	За- пись в файл	Чте- ние файла	Смена дирек- тории	Про- смотр фай- лов в дирек- тории	Пере- имено- вание файла
100	400	-	-	-	-	+	-	-
200	400	+	+	-	-	-	-	+
300	400	+	+	-	-	+	-	+
400	400	-	-	-	+	+	-	-
500	400	+	+	-	+	+	-	+
600	400	+	+	-	+	-	+	+
700	400	+	+	-	+	+	+	-
100	700	-	-	-	-	+	-	-
200	700	+	+	+	+	-	-	+
300	700	+	+	+	+	+	-	+
400	700	-	-	+	+	+	-	-
500	700	-	-	+	+	+	+	-
600	700	+	+	+	-	+	+	+
700	700	+	+	+	+	+	+	+

3 Выводы

Получены практические навыки работы в консоли с атрибутами файлов, закреплены теоретические основы дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux