Отчёта по лабораторной работе №2

Основы информационной безопасности

Надежда Александровна Рогожина

Содержание

# 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 OC Linux.

# 2 Задание

1. В установленной при выполнении предыдущей лабораторной работы операционной системе создайте учётную запись пользователя guest (использую учётную запись администратора): useradd guest
2. Задайте пароль для пользователя guest (использую учётную запись администратора): passwd guest
3. Войдите в систему от имени пользователя guest.
4. Определите директорию, в которой вы находитесь, командой pwd. Сравните её с приглашением командной строки. Определите, является ли она вашей домашней директорией? Если нет, зайдите в домашнюю директорию.
5. Уточните имя вашего пользователя командой whoami.
6. Уточните имя вашего пользователя, его группу, а также группы, куда входит пользователь, командой id. Выведенные значения uid, gid и др. запомните. Сравните вывод id с выводом команды groups.
7. Сравните полученную информацию об имени пользователя с данными, выводимыми в приглашении командной строки.
8. Просмотрите файл /etc/passwd командой cat /etc/passwd. Найдите в нём свою учётную запись. Определите uid пользователя. Определите gid пользователя. Сравните найденные значения с полученными в предыдущих пунктах.
9. Определите существующие в системе директории командой ls -l /home/. Удалось ли вам получить список поддиректорий директории /home? Какие права установлены на директориях?
10. Проверьте, какие расширенные атрибуты установлены на поддиректориях, находящихся в директории /home, командой: lsattr /home. Удалось ли вам увидеть расширенные атрибуты директории? Удалось ли вам увидеть расширенные атрибуты директорий других пользователей?
11. Создайте в домашней директории поддиректорию dir1 командой mkdir dir1. Определите командами ls -l и lsattr, какие права доступа и расширенные атрибуты были выставлены на директорию dir1.
12. Снимите с директории dir1 все атрибуты командой chmod 000 dir1 и проверьте с её помощью правильность выполнения команды ls -l
13. Попытайтесь создать в директории dir1 файл file1 командой echo "test" > /home/guest/dir1/file1. Объясните, почему вы получили отказ в выполнении операции по созданию файла? Оцените, как сообщение об ошибке отразилось на создании файла? Проверьте командой ls -l /home/guest/dir1 действительно ли файл file1 не находится внутри директории dir1.
14. Заполните таблицу «Установленные права и разрешённые действия» (см. табл. 2.1), выполняя действия от имени владельца директории (файлов), определив опытным путём, какие операции разрешены, а какие нет. Если операция разрешена, занесите в таблицу знак «+», если не разрешена, знак «-».
15. На основании заполненной таблицы определите те или иные минимально необходимые права для выполнения операций внутри директории dir1, заполните табл. 2.2

# 3 Теоретическое введение

Есть 3 вида разрешений. Они определяют права пользователя на 3 действия: чтение, запись и выполнение. В Linux эти действия обозначаются вот так:

* r — read (чтение) — право просматривать содержимое файла;
* w — write (запись) — право изменять содержимое файла;
* x — execute (выполнение) — право запускать файл, если это программа или скрипт.

У каждого файла есть 3 группы пользователей, для которых можно устанавливать права доступа.

* owner (владелец) — отдельный человек, который владеет файлом. Обычно это тот, кто создал файл, но владельцем можно сделать и кого-то другого.
* group (группа) — пользователи с общими заданными правами.
* others (другие) — все остальные пользователи, не относящиеся к группе и не являющиеся владельцами.

**Как узнать разрешения файла?**

Чтобы посмотреть права доступа к файлу, нужно вызвать команду ls с опцией -l. Эта опция отвечает за вывода списка в длинном формате.

ls -l <путь>

user@bash: ls -l /home/karpaff/linuxtutorialwork/chick.png  
-rwxr----x 1 harry users 2.7K Jan 4 07:32 /home/karpaff/linuxtutorialwork/chick.png  
user@bash:

В приведенном выше примере первые 10 символов вывода показывают разрешения файла.

* Первый символ определяет тип файла. Если первый символ — прочерк -, то это обычный файл. Если первый символ d, то это каталог.
* Следующие 3 символа показывают разрешения для владельца. Буква означает наличие разрешения, а прочерк - — его отсутствие. В нашем примере у владельца есть все разрешения (чтение, запись и выполнение).
* Следующие 3 символа показывают разрешения для группы. В этом примере у членов группы есть разрешение на чтение, но нет разрешений на запись и выполнение. Обратите внимание, порядок записи разрешений всегда такой: чтение, запись, выполнение.
* Последние 3 символа показывают разрешения для всех остальных пользователей. В этом примере у них есть только разрешение на выполнение.

**Как изменить права доступа?**

Для изменения прав доступа к файлу или каталога используется команда chmod (от англ. change mode). Эта команда меняет биты режима файла — если совсем просто, это индикатор разрешений.

chmod [разрешение] [путь]

Аргументы команды chmod, отвечающие за разрешение, состоят из 3 компонентов:

* Для кого мы меняем разрешение? Обозначается первыми буквами слов: [ugoa] — user (пользователь, он же владелец), group (группа), others (другие), all (все).
* Мы предоставляет или отзываем разрешения? Обозначается плюсом +, если предоставляем, минусом -, если отзываем.
* Какое разрешение мы хотим изменить? Чтение (r), запись (w), выполнение (x).

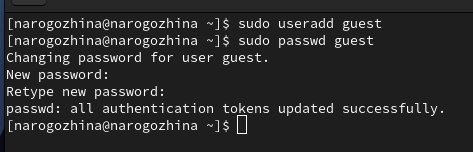
В табл. [[1](#tbl:std-dir)] приведено краткое описание основных команд, использованных в лабораторной работе.

Table 1: Описание некоторых команд Linux

| Имя каталога | Описание каталога |
| --- | --- |
| cd | Смена директории |
| clear | Очищение окна терминала |
| id | Отображение системного идентификатора указанного пользователя |
| whoami | Имя пользователя |
| group | Группа, к которой принадлежит пользователь |
| sudo | Права суперпользователя |
| cat | Отражение содержимого текстового файла в терминале |
| chmod | Изменение прав доступа к файлу |
| ls | Просмотр содержимого директории |
| touch | Создание файла |
| mkdir | Создание директории |
| rm | Удаление файла |
| mv | Перемещение/переименование файла |

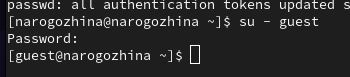
# 4 Выполнение лабораторной работы

В первую очередь добавить нового пользователя и установим для него пароль (рис.[??]):



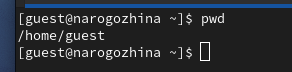
Добавление нового пользователя

Далее зайдем от имени пользователя guest (рис.[??]):



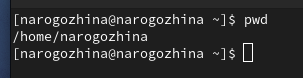
Вход от имени guest

Определим директорию у guest (рис.[??]):



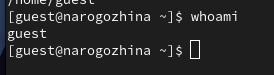
guest:pwd

Также наш домашний каталог (рис.[??]):



narogozhina:pwd

Видим, что домашний каталог пользователя guest **не совпадает** с домашним каталогом пользователя narogozhina. Уточним имя пользователя у обоих пользователей(рис.[??], рис.[??]):

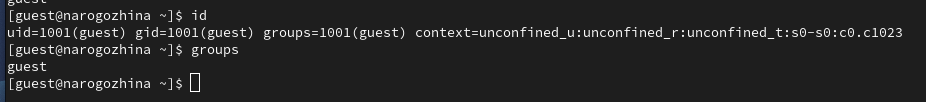


guest:whoami

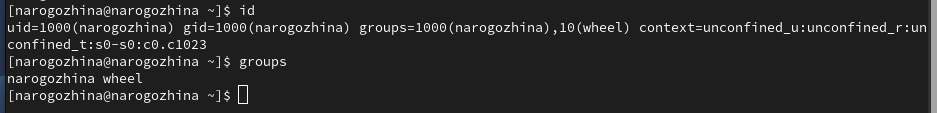
narogozhina:whoami

narogozhina:whoami

Уточним имя пользователя и группы командами id и groups(рис.[??], рис.[??]):

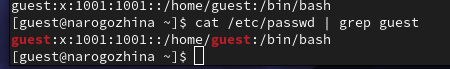


guest:id+groups

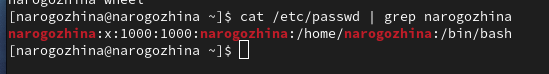


narogozhina:id+groups

Видно, что вывод разный (id также выводит id этого пользователя), однако сами группы показываются одинаково. Выведем на экран терминала файл /etc/passwd (рис.[??], рис.[??]):

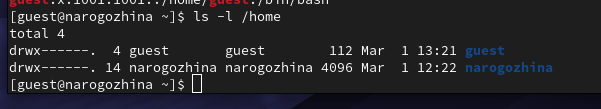


guest:cat /etc/passwd

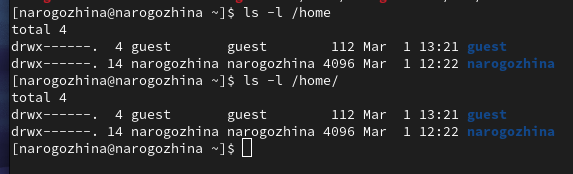


narogozhina:cat /etc/passwd

Видим, что uid и gid у обоих пользователей совпадают. Посмотрим на директории (рис.[??], рис.[??]):

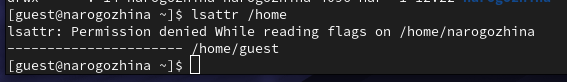


guest:ls -l /home

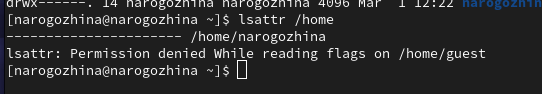


narogozhina:ls -l /home

Как мы видим, для пользователя *guest* не доступен каталог *narogozhina*, и наоборот. На обоих каталогах установлены полные права только для владельцев этих каталогов. Все остальные пользователи, как и пользователи их группы, просматривать, читать и выполнять эти файлы/каталоги не могут. Просмотрим атрибуты поддиректорий этих каталогов (рис.[??], рис.[??]):

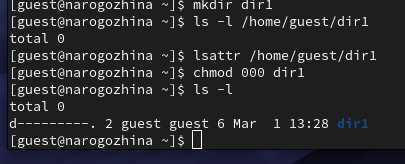


guest:lsattr /home



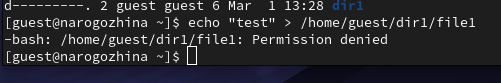
narogozhina:lsattr /home

Как мы видим, для пользователя *guest* не доступны даже для просмотра подкаталоги *narogozhina*, и наоборот (о чем было сказано в предыдущем пункте). Однако для своего каталога каждый из пользователей посмотреть атрибуты может. Создадим подкаталог dir1 и посмотрим на его атрибуты (рис.[??]):



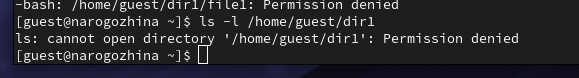
Создание каталога dir1, его атрибуты

Изначально атрибутов у dir1 не было, однако это не мешало нам подтвердить возможность просмотра. chmod 000 означает, что мы закрыли каталог и для чтения, и для записи, и для выполнения для всех пользователей (d———). Попробовать создать файл конечно можно, но тем не менее, это не получится (рис.[??]):



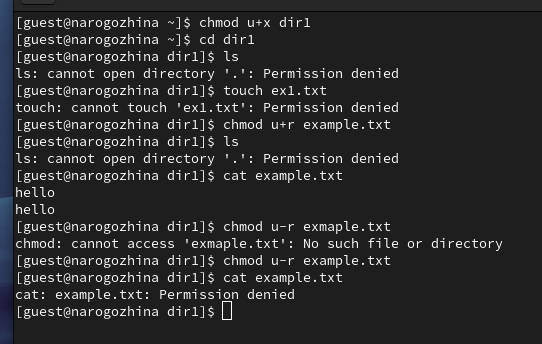
Попытка создать файл

Отказ мы получили по вышеописанной причине - у нас нет прав для взаимодействия с каталогом dir1, в который мы пытаемся добавить файл. Также, мы не может и посмотреть все, что у нас есть в этой директории (рис.[??]):

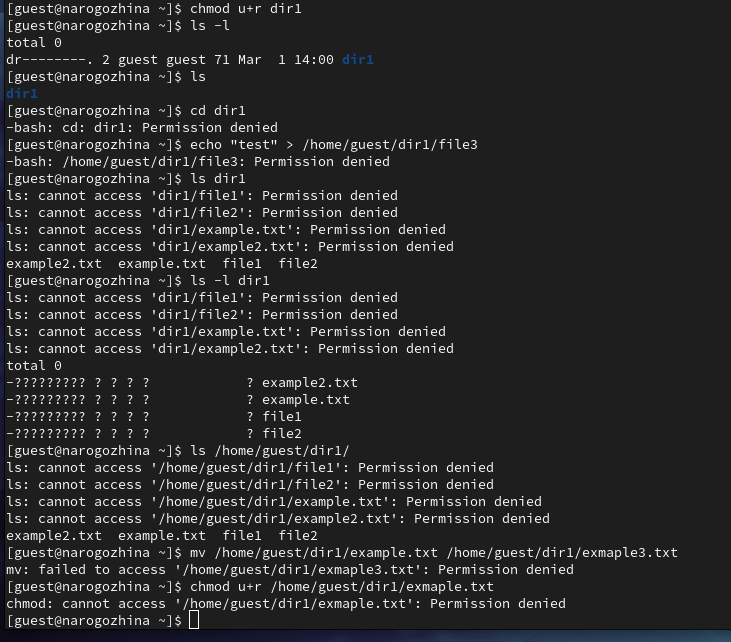


Проверка командой

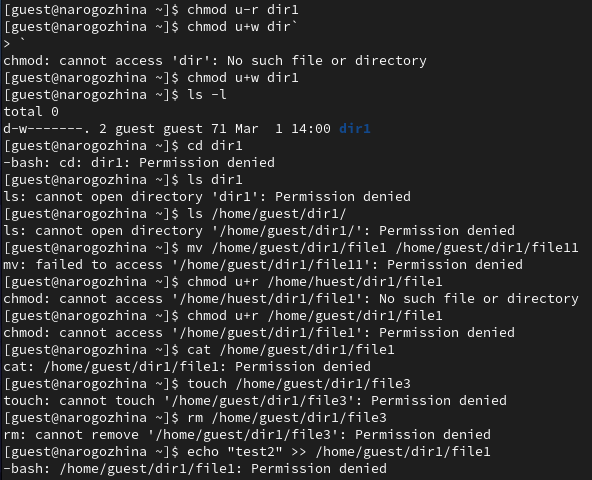
Далее я приведу скриншоты, по которым я заполняла таблицу 2.1 (рис.[??], рис.[??], рис.[??], рис.[??], рис.[??], рис.[??]):



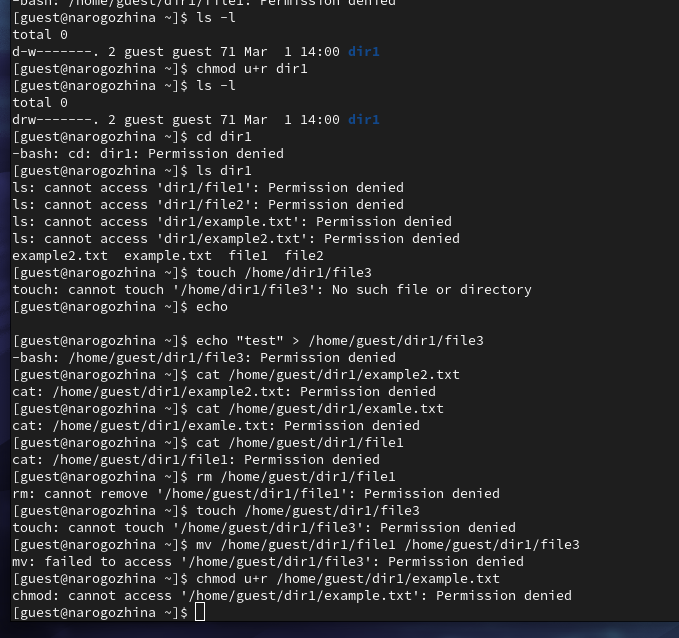
d–x——



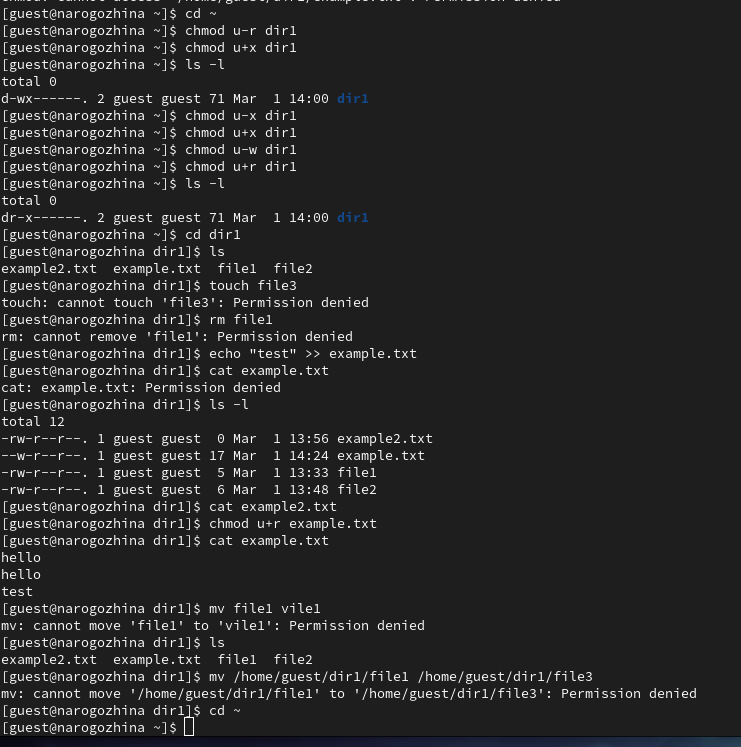
dr——–



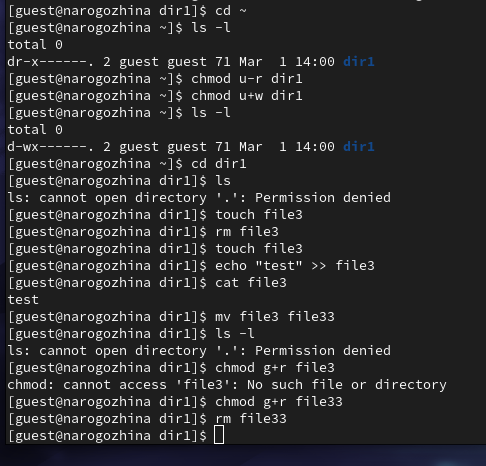
d-w——-



drw——-



dr-x——



d-wx——

После проведенных тестов, итоговая таблица[[2](#tbl:std-dir1)] заполнена:

Table 2: Установленные права и разрешённые действия

| Права директории | Права файла | Создание файла | Удаление файла | Запись в файл | Чтение файла | Смена директории | Просмотр файлов в директории | Переименование файла | Смена атрибутов файла |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| d(000) | 000 | - | - | - | - | - | - | - | - |
| d--x------ | 001 | - | - | - | - | + | - | - | + |
| dr-------- | 100 | - | - | - | + | - | + | - | - |
| d-w------- | 010 | + | - | - | - | - | - | - | - |
| drw------- | 110 | - | - | - | - | - | + | - | - |
| dr-x------ | 101 | - | - | + | + | + | + | - | + |
| d-wx------ | 011 | + | + | + | + | + | - | + | + |
| drwx------ | 111 | + | + | + | + | + | + | + | + |

После создания предыдущей таблицы, можем заполнить минимальные права для совершения операций : |||| | Операция | Минимальные права на директорию | Минимальные права на файл | | Создание файла | x | w | | Удаление файла | x | w | | Чтение файла | r | r | | Запись в файл | wx | w | | Переименование файла | wx | x | | Создание поддиректории | wx | wx | | Удаление поддиректории | wx | wx | ||||

# 5 Выводы

В ходе выполнения работы мы получили практические навыки работы в консоли с атрибутами файлов и закрепили теоретические основы разграничения доступа в современных системах с открытым кодом на базе OC Linux.

# Список литературы