Отчет по лабораторной работе №2

Дискреционное разграничение прав в Linux. Основные атрибуты

Исаханян Эдуард Тигранович

2022 Sep 13th

Содержание

List of Tables

List of Figures

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 Теоретическое введение[1]

В Линуксе существует 3 основынх права доступа: - Чтение - разрешает получать содержимое файла, но на запись нет. Для каталога позволяет получить список файлов и каталогов, расположенных в нем; - Запись - разрешает записывать новые данные в файл или изменять существующие, а также позволяет создавать и изменять файлы и каталоги; - Выполнение - вы не можете выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу.

Но все эти права были бы бессмысленными, если бы применялись сразу для всех пользователей. Поэтому каждый файл имеет три категории пользователей, для которых можно устанавливать различные сочетания прав доступа:

- Владелец - набор прав для владельца файла, пользователя, который его создал или сейчас установлен его владельцем. Обычно владелец имеет все права, чтение, запись и выполнение.
- Группа - любая группа пользователей, существующая в системе и привязанная к файлу. Но это может быть только одна группа и обычно это группа владельца, хотя для файла можно назначить и другую группу.
- Остальные - все пользователи, кроме владельца и пользователей, входящих в группу файла.

[1] Открытый источник информации

3 Выполнение лабораторной работы

В установленной при выполнении предыдущей лабораторной работы операционной системе создим учётную запись пользователя guest (используя учётную запись администратора) и зададим пароль (рис. 3.1).

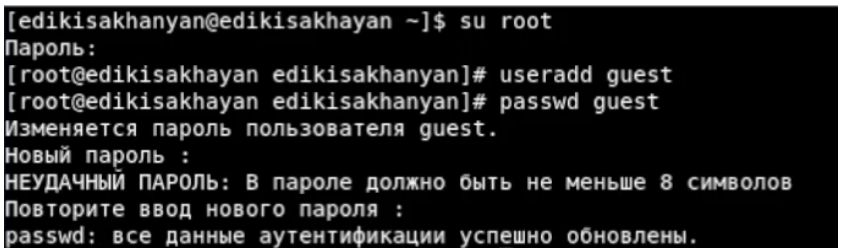

Figure 3.1: Создание пользователя guest и задание ему пароля

Войдем в систему от имени пользователя guest (рис. 3.2).

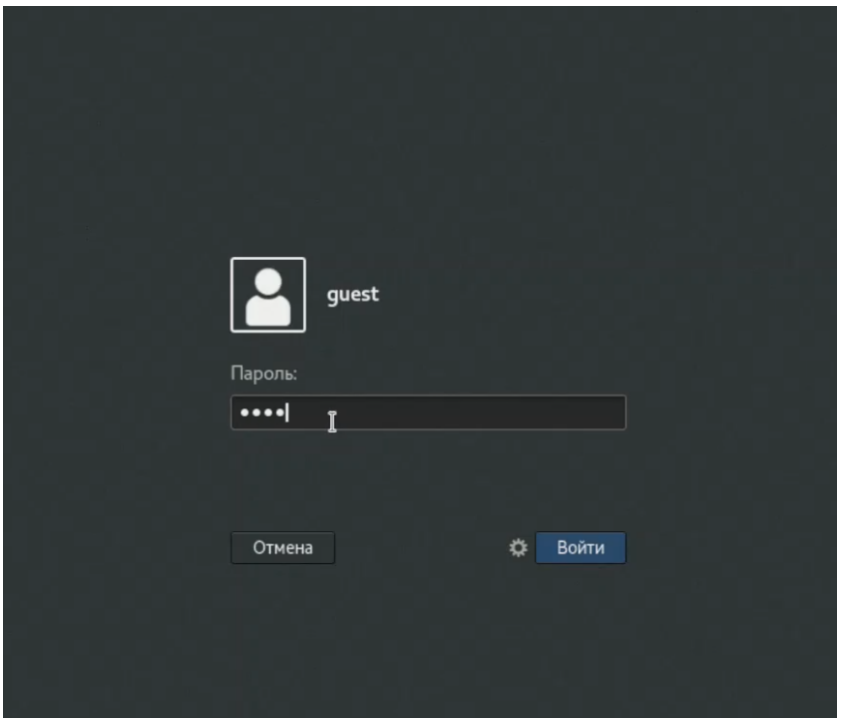

Figure 3.2: Вход в систему под guest

Определим директорию, в которой мы находимся (рис. 3.3).

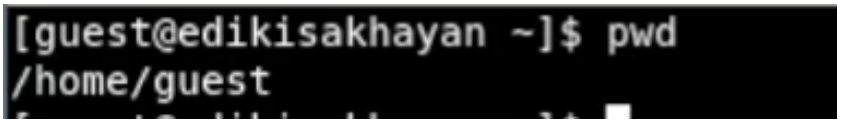

Figure 3.3: Проверка директории

Как видно это наша домашняя директория. Она в целом совпадает с приглашением командной строки: в командной строке есть guest (пользователь) и ~ (указывает на то, что мы находимя в домашней директории).

Уточним имя нашего пользователя (рис. 3.4).

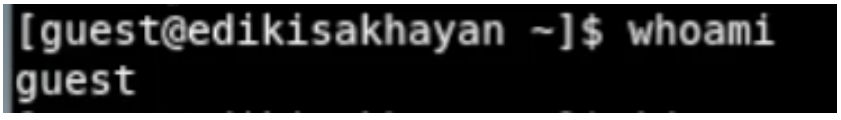

Figure 3.4: Проверка имени пользователя

Уточним имя нашего пользователя, его группу, а также группы, куда он входит,

командой id. Выведенные значения uid, gid и др. запомнилим. Выполним команду groups. Полученные значения совпадают с тем, что выдала id (рис. 3.5).

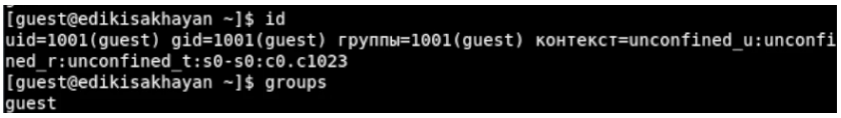

Figure 3.5: Проверка ID

Полученная информация об имени пользователя частично совпадает с данными, выводимыми в приглашении командной строки, но является более подробной.

Посмторим файл /etc/passwd (рис. 3.6).

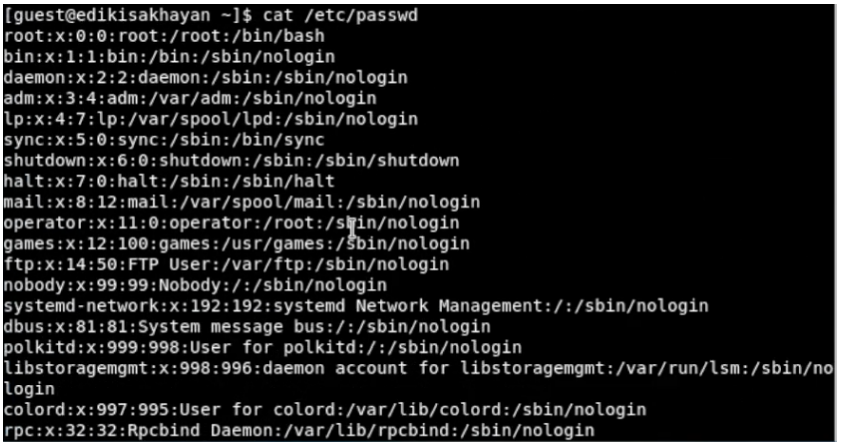

Figure 3.6: Файл /etc/passwd

Найдем в нем нашу учётную запись (рис. 3.7).

Figure 3.7: Учётная запись guest в файле /etc/passwd

Определим uid пользователя: 1001 и gid пользователя: 1001. Эти значения совпадают с полученными в ранее значениями.

Определим существующие в системе директории (рис. 3.8).

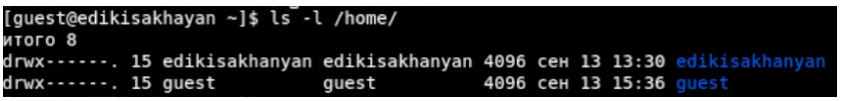

Figure 3.8: Существующие директории

Как видим владельцы директорий имеют на них полные права, а группы и другие пользователи не имеют никаких прав на эти директории.

Проверим, какие расширенные атрибуты установлены на поддиректориях, находящихся в директории (рис. 3.9).

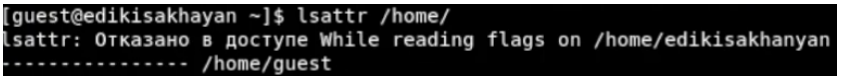

Figure 3.9: Расширенные атрибуты поддиректорий

Мы можем посмотреть только расширенные атрибуты директории guest, а расширенные атрибуты директорий других пользователей нам не доступны.

Создадим в домашней директории поддиректорию dir1: mkdir dir1. Определим командами ls -l и lsattr, какие права доступа и расширенные атрибуты были выставлены на директорию dir1 (рис. 3.10 - 3.11).

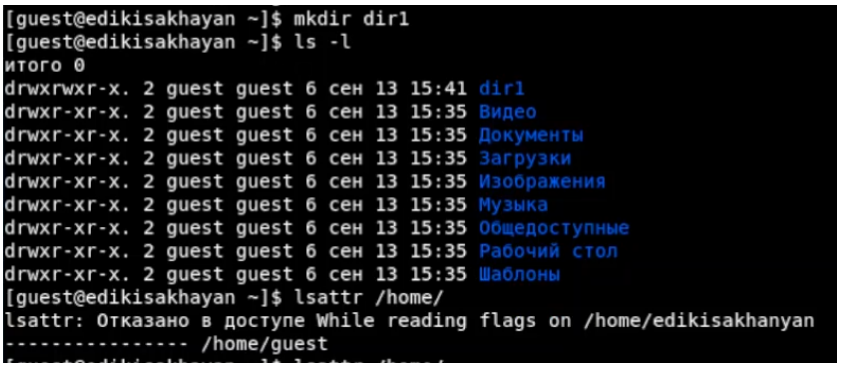

Figure 3.10: Поддиректория dir1

Figure 3.11: Расширенные атрибуты поддиректорий

Снимем с директории dir1 все атрибуты: chmod 000 dir1, и проверим с её помощью правильность выполнения команды ls -l (рис. 3.12).

Figure 3.12: Снятие с директорий все атрибуты

Попытаемся создать в директории dir1 файл file1: echo “test” > /home/guest/dir1/file1 (рис. 3.13).

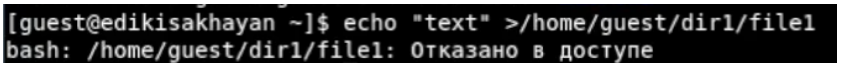

Figure 3.13: Попытка создать файл

Я получим отказ в выполнении операции по созданию файла, т. к. мы сняли с директории все атрибуты (даже для владельцев). Сообщение об ошибке никак не отразилось на создании файла, потому что он не был создан.

Заполним таблицу «Установленные права и разрешённые действия», выполняя действия от имени владельца директории (файлов), определив опытным путём,

какие операции разрешены, а какие нет. Если операция разрешена, занесем в таблицу знак «+», если не разрешена – знак «-» (рис. 3.14 - 3.15).

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)-(700)	-	-	-	-	-	-	-	-
d--x------ (100)	(000)	-	-	-	-	+	-	-	+
d--x------ (100)	(100)	-	-	-	-	+	-	-	+
d--x------ (100)	(200)	-	-	+	-	+	-	-	+
d--x------ (100)	(300)	-	-	+	-	+	-	-	+
d--x------ (100)	(400)	-	-	-	+	+	-	-	+
d--x------ (100)	(500)	-	-	-	+	+	-	-	+
d--x------ (100)	(600)	-	-	+	+	+	-	-	+
d--x------ (100)	(700)	-	-	+	+	+	-	-	+
d-w------- (200)	(000)-(700)	-	-	-	-	-	-	-	-
d-wx------ (300)	(000)	+	+	-	-	+	-	+	+
d-wx------ (300)	(100)	+	+	-	-	+	-	+	+
d-wx------ (300)	(200)	+	+	+	-	+	-	+	+
d-wx------ (300)	(300)	+	+	+	-	+	-	+	+
d-wx------ (300)	(400)	+	+	-	+	+	-	+	+
d-wx------ (300)	(500)	+	+	-	+	+	-	+	+
d-wx------ (300)	(600)	+	+	+	+	+	-	+	+
d-wx------ (300)	(700)	+	+	+	+	+	-	+	+
dr--------- (400)	(000)-(700)	-	-	-	-	-	+	-	-
dr-x------ (500)	(000)	-	-	-	-	+	+	-	+
dr-x------ (500)	(100)	-	-	-	-	+	+	-	+
dr-x------ (500)	(200)	-	-	+	-	+	+	-	+
dr-x------ (500)	(300)	-	-	+	+	+	+	-	+
dr-x------ (500)	(400)	-	-	-	+	+	+	-	+
dr-x------ (500)	(500)	-	-	-	+	+	+	-	+
dr-x------ (500)	(600)	-	-	+	+	+	+	-	+
dr-x------ (500)	(700)	-	-	+	+	+	+	-	+
drw------- (600)	(000)-(700)	-	-	-	-	-	+	-	-
drwx------ (700)	(000)	+	+	-	-	+	+	+	+
drwx------ (700)	(100)	+	+	-	-	+	+	+	+
drwx------ (700)	(200)	+	+	+	-	+	+	+	+
drwx------ (700)	(300)	+	+	+	-	+	+	+	+
drwx------ (700)	(400)	+	+	-	+	+	+	+	+

Figure 3.14: Установленные права и разрешённые действия 1

drwx------ (700)	(300)	+	+	+	-	+	+	+	+
drwx------ (700)	(400)	+	+	-	+	+	+	+	+
drwx------ (700)	(500)	+	+	-	+	+	+	+	+
drwx------ (700)	(600)	+	+	+	+	+	+	+	+
drwx------ (700)	(700)	+	+	+	+	+	+	+	+

Figure 3.15: Установленные права и разрешённые действия 2

На основании заполненной таблицы определим те или иные минимально необходимые права для выполнения операций внутри директории dir1, заполним таблицу «Минимальные права для совершения операций» (рис. 3.16).

Операции	Минимаотные права на дитекторию	Минимальные права на файл
Создание файла	d-wx------ (300)	(000)
Удаление файла	d-wx------ (300)	(000)
Чтение файла	d--x------ (100)	(400)
Запись в файл	d--x------ (100)	(200)
Переименование файла	d-wx------ (300)	(000)
Создание поддиректорию	d-wx------ (300)	(000)
удаление поддиректории	d-wx------ (300)	(000)

Figure 3.16: Минимальные права для совершения операций

4 Выводы

Входе работы, мы получии практические навыки работы в консоли с атрибутами файлов, закрепили теоретические основы дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux.

Список литературы

1. Методические материалы к лабораторной работе, представленные на сайте “ТУИС РУДН” https://esystem.rudn.ru/

::: {#refs} :::