Лабораторная работа № 2

Дискреционное разграничение прав в Linux. Основные атрибуты

Абакумов Егор Александрович

Содержание

List of Figures

List of Tables

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.

Задание

Создать нового пользователя с нужными правами, провести эксперимент по выявлению минимально необходимых прав для действий над файловой структурой.

Теоретическое описание

В операционной системе Linux есть много отличных функций безопасности, но она из самых важных - это система прав доступа к файлам. Linux, как последователь идеологии ядра Linux в отличие от Windows, изначально проектировался как многопользовательская система, поэтому права доступа к файлам в linux продуманы очень хорошо.

Изначально каждый файл имел три параметра доступа [1]:

- 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.

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.

Для управления правами используется команда chmod. При использовании chmod вы можете устанавливать разрешения для пользователя (user), группы (group) и других (other). Вы можете использовать эту команду в двух режимах: относительный режим и абсолютный режим. В абсолютном режиме три цифры используются для установки основных разрешений [2].

Ход работы

1. Создаем нового пользователя guest и задаем для него пароль (иллюстр. 0.1). Входим на новую учетную запись, вводим pwd. Мы находимся в домашней директории, о чем говорит значок “тильда” (~) в приглашении командной строки, вывод команды pwd и тот факт, что мы еще никуда не переходили, а по дефолту пользователь стартует в домашней директории. Далее проверяем пользователя командой whoami - мы guest. Командой id проверяем группы и имя пользователя. Тут тоже имя пользователя guest, группа guest, uid и gid равны 1001. Команда groups выводит единственную группу guest, куда мы входим. Приглашение командной строки так же указывает на то, что мы работаем под пользователем guest (иллюстр. 0.2).

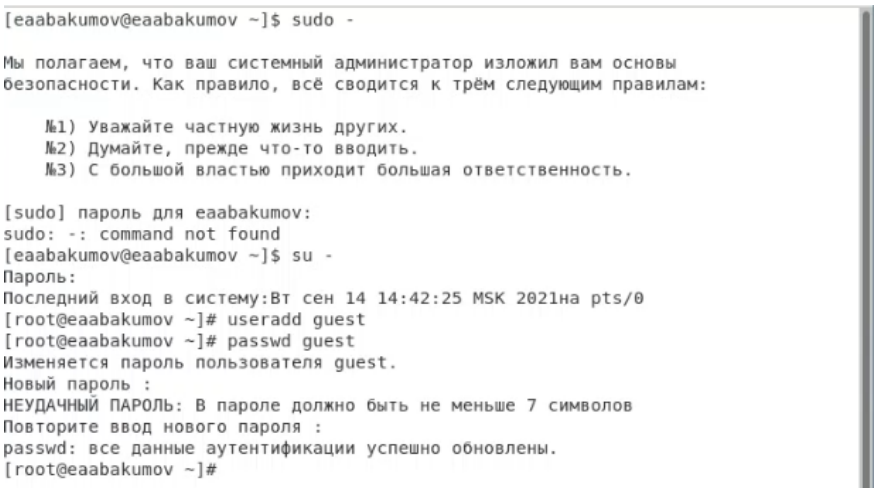

Figure 0.1: Создание пользователя

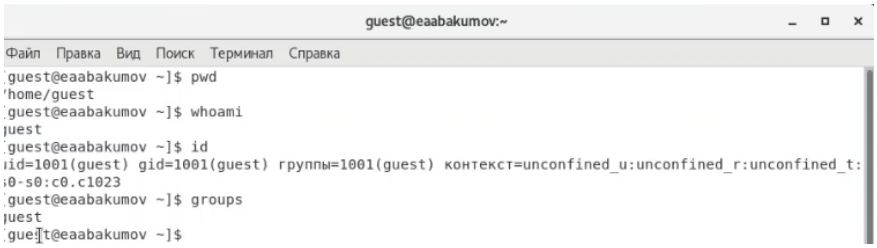

Figure 0.2: Лог консоли guest

2. Теперь проверим /etc/passwd командой cat. Найдем там себя в последней строчке. Наш uid равен gid и равен 1001, что совпадает с выводом id. Проверим существующие домашние директории командой ls -l /home. Успешно. Тут увидим две папки (по количеству пользователей), обе с правами 700. Проверим вывод команды lsattr на те же папки, тут увидим, что расширенных атрибутов нашей домашней директории нет, а просмотреть атрибуты папки другого пользователя нам не дает система (иллюстр. 0.3).

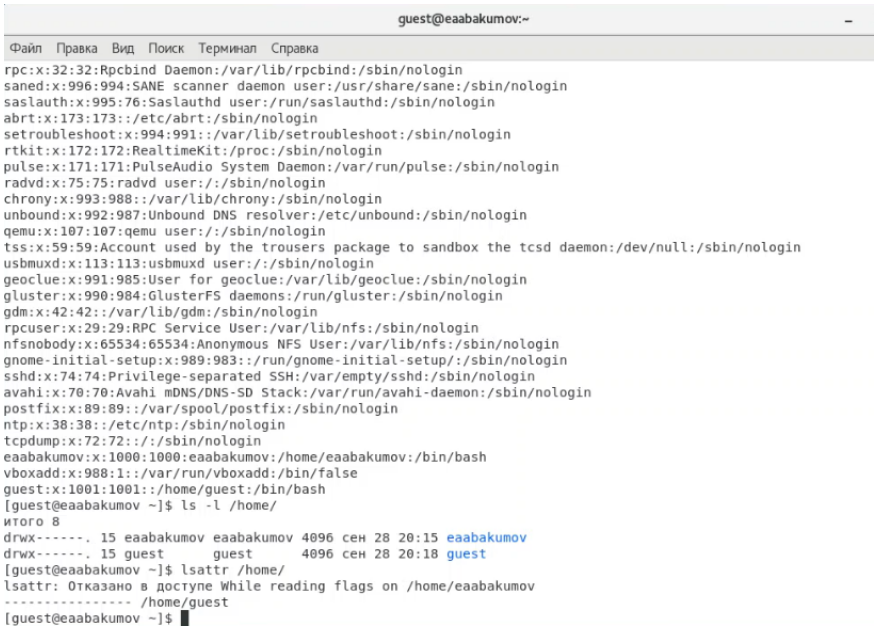

Figure 0.3: Лог консоли по проверке домашних директорий

3. Создадим в домашней директории guest папку dir1. Папка получила права 775 и не получила никаких расширенных атрибутов (иллюстр. 0.4).

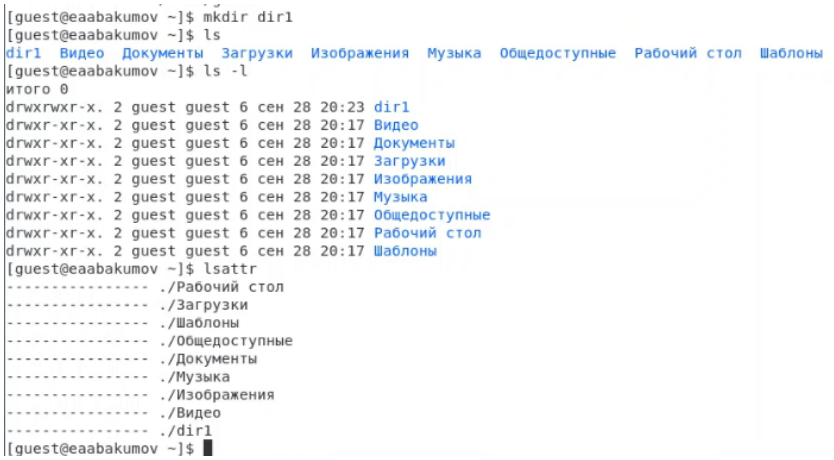

Figure 0.4: Лог консоли по созданию dir1

4. Командой chmod обнуляем права на dir1 и проверяем это. Далее пытаемся создать в папке файл file1, что у нас не выходит, так как система блокирует действие из-за недостатка прав (иллюстр. 0.5). Соответственно, и сам файл создан не был (иллюстр. 0.6).

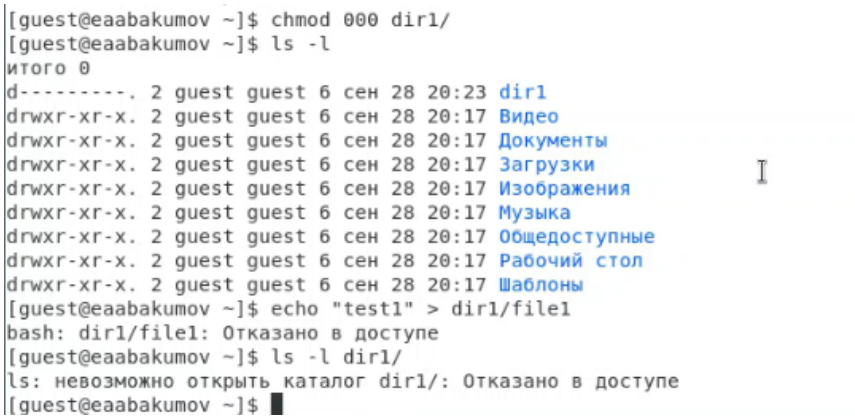

Figure 0.5: Обнуление прав и попытка создания файла

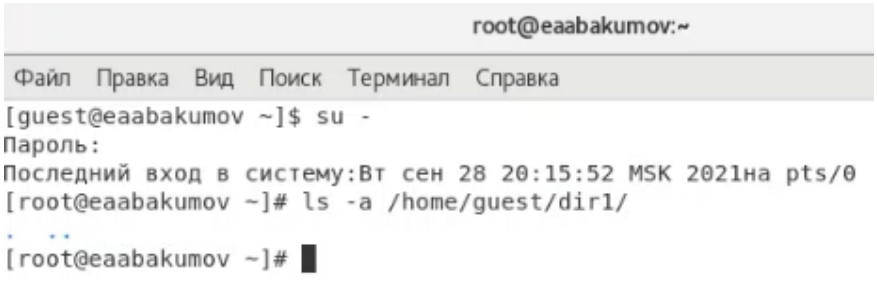

Figure 0.6: Пустая папка dir1 после попытки создать файл

5. Следующим шагом проведем эксперимент по выявлению минимально необходимых прав для действий над файловой структурой. Для этого используем нашу папку dir1, файлы внутри неё и функционал прав доступа ОС Linux. Для каждой комбинации атрибутов доступа (r, w, x) на папку и на файл попробуем осуществить ряд действий и таким образом выявим минимально необходимые права для каждого действия. Атрибуты используем только для владельца, поэтому комбинаций будет $2^3 \cdot 2^3 = 2^6 = 64$. В каждой строчке будет по 8 действий. Проверять осуществимость функции будем следующими командами:

- touch для создания файла в директории;
- rm для удаления файла в директории;
- echo для записи в файл;
- cat для чтения из файла;
- mv для переименования файла;
- chattr для изменения атрибутов файла;
- cd для смены директории;
- ls для просмотра файлов в директории.

На иллюстрации можно увидеть вывод приведенных выше команд для первой строки таблицы (права на директорию - 000, права на файл - 000, иллюстр. 0.7).

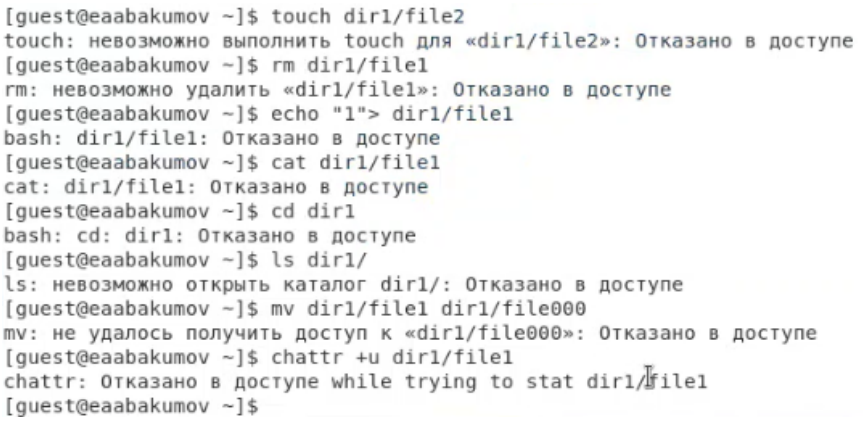

Figure 0.7: Пример ввода команд для проверки прав

Полученные результаты представлены в виде таблицы (иллюстр. 0.8).

Права директории	Права файла	Смена директории	Просмотр файлов в директории	Создание файла	Удаление файла	Запись в файл	Чтение файла	Переименование файла	Смена атрибутов файла
000	000	—	—	—	—	—	—	—	—
100	000	+	—	—	—	—	—	—	—
200	000	—	—	—	—	—	—	—	—
300	000	+	—	+	+	—	—	+	—
400	000	—	+	—	—	—	—	—	—
500	000	+	+	—	—	—	—	—	—
600	000	—	+	—	—	—	—	—	—
700	000	+	+	+	+	—	—	+	—
000	100	—	—	—	—	—	—	—	—
100	100	+	—	—	—	—	—	—	—
200	100	—	—	—	—	—	—	—	—
300	100	+	—	+	+	—	—	+	—
400	100	—	+	—	—	—	—	—	—
500	100	+	+	—	—	—	—	—	—
600	100	—	+	—	—	—	—	—	—
700	100	+	+	+	+	—	—	+	—
000	200	—	—	—	—	—	—	—	—
100	200	+	—	—	—	+	—	—	—
200	200	—	—	—	—	—	—	—	—
300	200	+	—	+	+	+	—	+	—
400	200	—	+	—	—	—	—	—	—
500	200	+	+	—	—	+	—	—	—
600	200	—	+	—	—	—	—	—	—
700	200	+	+	+	+	+		+	—
000	300	—	—	—	—	—	—	—	—
100	300	+	—	—	—	+	—	—	—
200	300	—	—	—	—	—	—	—	—
300	300	+	—	+	+	+	—	+	—
400	300	—	+	—	—	—	—	—	—
500	300	+	+	—	—	+	—	—	—
600	300	—	+	—	—	—	—	—	—
700	300	+	+	+	+	+	—	+	—
000	400	—	—	—	—	—	—	—	—
100	400	+	—	—	—	—	+	—	+
200	400	—	—	—	—	—	—	—	—
300	400	+	—	+	+	—	+	+	+
400	400	—	+	—	—	—	—	—	—
500	400	+	+	—	—	—	+	—	+
600	400	—	+	—	—	—	—	—	—
700	400	+	+	+	+	—	+	+	+
000	500	—	—	—	—	—	—	—	—
100	500	+	—	—	—	—	+	—	+
200	500	—	—	—	—	—	—	—	—
300	500	+	—	+	+	—	+	+	+
400	500	—	+	—	—	—	—	—	—
500	500	+	+	—	—	—	+	—	+
600	500	—	+	—	—	—	—	—	—
700	500	+	+	+	+	—	+	+	+
000	600	—	—	—	—	—	—	—	—
100	600	+	—	—	—	+	+	—	+
200	600	—	—	—	—	—	—	—	—
300	600	+	—	+	+	+	+	+	+
400	600	—	+	—	—	—	—	—	—
500	600	+	+	—	—	+	+	—	+
600	600	—	+	—	—	—	—	—	—
700	600	+	+	+	+	+	+	+	+
000	700	—	—	—	—	—	—	—	—
100	700	+	—	—	—	+	+	—	+
200	700	—	—	—	—	—	—	—	—
300	700	+	—	+	+	+	+	+	+
400	700	—	+	—	—	—	—	—	—
500	700	+	+	—	—	+	+	—	+
600	700	—	+	—	—	—	—	—	—
700	700	+	+	+	+	+	+	+	+
Права директории	Права файла	Смена директории	Просмотр файлов в директории	Создание файла	Удаление файла	Запись в файл	Чтение файла	Переименование файла	Смена атрибутов файла

Figure 0.8: Установленные права и разрешённые действия

На основе данных полученной выше таблицы построим вторую таблицу, иллюстрирующую минимально необходимые права для совершения определенных операций.

Операция	Мин. права на директорию	Мин. права на файл
Создание файла	300	000
Удаление файла	300	000
Чтение файла	100	400
Запись в файл	100	200
Переименование файла	300	000
Создание поддиректории	300	-
Удаление поддиректории	300	-

Выводы

В ходе работы мы успешно провели эксперимент по выявлению минимально необходимых прав для действий над файловой структурой, получили ряд практических навыков работы в консоли с атрибутами файлов, закрепили теоретические основы дискреционного разграничения доступа в ОС Linux.

Список литературы

1. Права доступа к файлам в linux. // Losst. 2020. URL: https://losst.ru/prava-dostupa-k-fajlam-v-linux (дата обращения 01.10.2021).

2. Права в Linux (chown, chmod, SUID, GUID, sticky bit, ACL, umask). // habr.com. 2019. URL: https://habr.com/ru/post/469667/ (дата обращения 01.10.2021).

3. Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян. Информационная безопасность компьютерных сетей: лабораторные работы. // Факультет физико-математических и естественных наук. М.: РУДН, 2015. 64 с..