**Министерство науки и высшего образования Российской Федерации Российский государственный университет нефти и газа (национальный исследовательский университет) имени И.М. Губкина**

**Кафедра безопасности информационных технологий Дисциплина:** Сети и системы передачи информации

**Реферат**

На тему: «Защита файлов и директорий с помощью утилиты chmod в Linux»

**Выполнили:**

Студент группы КТ-21-05, Огородников Филипп Александрович \_(подпись)

Студент группы КИ-21-02, Гелич Кирилл Андреевич (подпись)

**Проверил:**

Старший преподаватель Уймин Антон Григорьевич \_(подпись)

**Москва, 2023**

**Оглавление**

1. [Введение 3](#_bookmark0)
2. [Основные принципы защиты файлов и папок в Linux 4](#_bookmark1)
3. [Права доступа в Linux 5](#_bookmark2)
4. [Синтаксис утилиты chmod 7](#_bookmark3)

[Буквенная нотация 7](#_bookmark4)

[Числовая нотация 8](#_bookmark5)

1. [Специальные типы разрешений 11](#_bookmark6)

[Setuid 11](#_bookmark7)

[Setgid 12](#_bookmark8)

[Sticky Bit 13](#_bookmark9)

1. [Новые файлы и директории 14](#_bookmark10)
2. [Типы файлов в Linux 16](#_bookmark11)

[Заключение 18](#_bookmark12)

[Список использованных источников 19](#_bookmark13)

# Введение

Безопасность данных играет важную роль в IT-сфере и особенно актуальна при работе с файловой системой Linux. Компьютерные данные могут содержать конфиденциальную информацию, личные данные пользователей, а также важные для бизнеса исходные коды, их утеря или утечка могут привести к серьезным последствиям.

Защита файлов и директорий в операционной системе Linux осуществляется с помощью различных механизмов, давая возможность пользователям контролировать доступ к своим данным. Основные методы защиты включают в себя управление правами доступа.

Каждый файл и директория в Linux имеет набор прав доступа для владельца, группы и других пользователей. Права доступа могут быть установлены с помощью команды “chmod”, позволяя разрешить или запретить чтение, запись и выполнение файлов в зависимости от пользовательской принадлежности. Они могут быть установлены отдельно для владельца файла, группы и других пользователей. Это позволяет ограничить доступ к данным и предотвратить случайное или злонамеренное удаление, или изменение файлов.

Дополнительно к основным правам доступа, в Linux существует возможность использовать ACL для установки более детальных прав доступа к файлам и каталогам для различных пользователей и групп. Это обеспечивает более гибкий контроль над доступом к файлам и каталогам. Установка suid, sgid и sticky битов предоставляет дополнительные возможности управления правами доступа.

В контексте защиты файлов и директорий в Linux, обеспечение безопасности данных при помощи утилиты chmod становится фундаментальной частью стратегии информационной безопасности. Надлежащая конфигурация прав доступа к файлам и директориям обеспечивает защиту конфиденциальной информации, целостность и доступность данных.

# Основные принципы защиты файлов и папок в Linux

* + 1. Разделение прав доступа - Linux использует систему прав доступа, основанную на пользователях и группах. Каждый файл и папка имеет набор разрешений, которые указывают, какие операции можно выполнять с файлом, и кто может выполнять эти операции.
		2. Владение и групповое владение - Каждый файл и папка в Linux имеет владельца и группу, которые определяют, кто может изменять файл, и кто может просматривать его содержимое. Владелец может установить разрешения для себя, группы и остальных пользователей.
		3. Режимы доступа - Linux имеет три основных режима доступа: чтение (r), запись (w) и выполнение (x). Каждый режим может быть задан для владельца, группы и остальных пользователей.
		4. Аудит доступа - Linux поддерживает механизм аудита, который записывает все действия пользователей системы, связанные с изменениями в файловой системе. Это позволяет администраторам отслеживать и анализировать возможные нарушения безопасности.
		5. Использование шифрования - Linux предлагает различные методы шифрования файлов и папок для защиты информации от несанкционированного доступа. Это позволяет сохранить данные в безопасности даже в случае компрометации файловой системы.

В данной работе мы рассмотрим основные принципы защиты файлов и папок с помощью утилиты chmod, ознакомимся с владельцами, их группами и остальными пользователями, а также с режимами доступа.

# Права доступа в Linux

В Linux работающие с файлами и директориями пользователи разделяются на три группы:

* Владелец файла или директории – тот пользователь который его создал.
* Группа владельцев файла или директории - другие пользователи, которые имеют такие же права, как и создавший файл или директорию пользователь.
* Все остальные пользователи.



Рис.1 Группы

Для обозначения этих групп используются буквы u, g и o соответственно.

В Linux системе файл может быть:

* Читаемым (readable) — содержимое файла может быть просмотрено.
* Записываемым (writeable) — содержимое файла может быть изменено.
* Исполняемым (executable) — бинарный файл может быть исполнен. В Linux системе директория может быть:
* Читаемая (readable) - можно посмотреть содержимое директории.
* Изменяемая (w) - можно создавать и удалять объекты в этой директории, причем если вы имеете право на запись, то удалять вы сможете даже те файлы, которые вам не принадлежат.
* Входимая (x) — право, которое позволяет войти в директорию (это право всегда проверяется в первую очередь, и даже если вы имеете все нужные права на объект, который лежит дальше в цепочке директорий, но не имеете права

“X” для доступа хотя бы к одной директории на пути к этому файлу, то до него будет невозможно добраться )

Утилита chmod может использоваться для установки прав доступа к файлам и директориям. На рис. 2 показано как это выглядит в операционной системе Linux:



Рис. 2 Права доступа

**Тип файла – Владелец – Группа владельца – Остальные пользователи**

# Синтаксис утилиты chmod

Утилита chmod принимает описания прав доступа в двух нотациях: буквенной нотации и числовой нотации, использующей восьмеричные числа. При этом у пользователя имеется возможность выполнения всех возможных операций с использованием каждой из нотаций.

## Буквенная нотация

Синтаксис буквенной нотации достаточно прост, сначала мы выбираем одну или сразу все группы из трех, описанных выше, которым будем выдавать, забирать или устанавливать права далее определяемся с какими конкретно правами мы будем работать, выглядит это так:

chmod ugo − / = /+ rwx “путь к файлу или директории” “- “– забрать права;

“=” – установить права; “+” – добавить права;

Приведем пример:

Допустим, мы создали файл и для всех остальных пользователей хотим выдать права на редактирование этого файла, чтобы это сделать нам нужно написать команду:

chmod o(others) + (выдать) w(writeable) “путь к файлу”



Рис.3 Выдача прав на чтение группе “остальные пользователи” Теперь, чтобы забрать эти права нам нужно написать:

chmod o – w “путь к файлу”



Рис.4 Выдача прав на чтение группе “остальные пользователи”

Если у нас есть файл script.sh (скрипт, который мы хотим выполнить), для того чтобы мы могли его запустить, нам нужно выдать соответствующие права, к примеру мы хотим, чтобы этот скрипт мог запустить любой пользователь:

chmod ugo + x “путь к файлу”

Также мы можем написать более правильную версию этой команды: chmod a + x “путь к файлу”, в данном случае a (all) – все три выделенные группы.



Рис. 5 Выдача прав на исполнение всем пользователям

Синтаксис для выдачи прав пользователя для определенной директории никак не меняется от выдачи прав для файла. К примеру, мы хотим, чтобы никто кроме нас и нашей группы в которой мы состоим не мог просматривать определенную директорию:

chmod o – r “путь к директории”



Рис. 6 Забираем права у “остальных пользователей” на просмотр директории

## Числовая нотация

В операционной системе Linux существует 7 типов файлов, которым соответствуют следующие аббревиатуры: (-), (d), (p), (l), (c или b), (s), (D). Что они значат будет описано позже, этот первый символ всегда держим в уме отдельно от прав, т.е. от следующих девяти символов, чтобы не запутаться.

Следующие аббревиатуры - биты, соответствуют правам на файл, они представлены в символьном виде. Каждый из этих битов соответствует определённому числу (в восьмеричной системе счислений) (r)(w)(x) = (4)(2)(1)

= (read)(write)(execute). Поэтому данную запись прав можно изменять двумя

способами. Представление с помощью символов называют относительным, а с помощью чисел называют абсолютным.

Когда мы видим вначале строки, например: −𝑟 − 𝑥𝑟 − 𝑥𝑟 − 𝑥, то представляем в уме таблицу, в том же порядке, слева направо. Причём первый символ — это всегда тип данного файла, а остальные 9 это права на него. Для понимания, преобразуем в уме обсуждаемую запись −𝑟 − 𝑥𝑟 − 𝑥𝑟 − 𝑥 в двоичный вид. Там, где есть символ на месте трёх бит, то это 1, если нет (-), то

1. В итоге получаем в двоичном виде 101 101 101. Что соответствует 555.



Рис. 7 Таблица абсолютного предоставления прав доступа

Пример:

Ранее мы выдавали права доступа всем пользователям для исполнения файла и делали мы это так: chmod a + x “путь к файлу” для того, чтобы это сделать абсолютным способом, с помощью чисел, нам нужно будет написать: chmod 111 “путь к файлу” – здесь первая единица это для владельца, вторая для группы владельца и третья для остальных пользователей.

Но если мы выдадим такие права, то выглядеть они будут следующим образом: − − −𝑥 − −𝑥 − −𝑥 , это означает, что единственное, что можно сделать с файлом – это выполнить его.



Рис. 8 Выдача прав на выполнение всем пользователям

Для того, чтобы не потерять права, которые были настроены ранее, нужно более продуманно использовать этот способ выдачи прав, пример:

Мы хотим оставить чтение и выполнение файла для остальных пользователей, но не хотим, чтобы они могли его редактировать:

chmod 775 “путь к файлу”, теперь эти права будут выглядеть так:

−𝑟𝑤𝑥𝑟𝑤𝑥𝑟 − 𝑥



Рис. 9 Выдача прав −𝑟𝑤𝑥𝑟𝑤𝑥𝑟 − 𝑥

Теперь, для владельца и для его группы установлены все права, а для остальных пользователей только просмотр и выполнение файла.

# Специальные типы разрешений

Специальные типы разрешений позволяют задавать расширенные права доступа на файлы или каталоги. Всего таких разрешений три: Setuid, Setgid, Sticky Bit.

## Setuid

Setuid – это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо. Классический пример использования этого бита в операционной системе — это команда sudo. Напишем следующие команды: which sudo /usr/bin/sudo

ls -l /usr/bin/sudo

Как мы видим на месте, где обычно установлен классический бит x (на исполнение), у нас выставлен специальный бит s. Это позволяет обычному пользователю системы выполнять команды с повышенными привилегиями без необходимости входа в систему как root, разумеется, зная пароль пользователя root.

Теперь создадим файл и как мы уже знаем владелец, его группа и остальные пользователи не имеют прав на его выполнение (x) для того, чтобы добавить бит setuid нужно написать:

chmod u + s “путь к файлу”, после того, как мы выполним эту команду мы увидим следующее: −𝑟𝑤𝑆𝑟𝑤 − 𝑟 − − , вместо ожидаемой s, мы видим “S”, это происходит, когда setuid установлен, но сам владелец файла не имеет прав на его выполнение.



Рис. 10 Бит setuid

Добавим это разрешение с помощью уже известной нам команды: chmod u + x “путь к файлу”



Рис. 11 Бит setuid

Если мы хотим установить setuid бит с помощью чисел, нам нужно написать следующим образом:



Рис. 12 Бит setuid с помощью чисел

## Setgid

Принцип работы Setgid очень похож на setuid с отличием, что файл будет запускаться пользователем от имени группы, которая владеет файлом.

Аналогично setuid, бит setgid выставляется с помощью команды chmod

g + s.



Рис. 13 Бит setgid

После выполнения этой команды мы опять вместо “s” увидим “S” для того, чтобы это исправить выдадим группе права на исполнение файла: chmod g+x “Путь к файлу”



Рис. 14 Бит setgid

Если мы хотим установить setgid бит с помощью чисел, нам нужно написать следующим образом:



Рис. 15 Бит setgid с помощью чисел

## Sticky Bit

Последний специальный бит разрешения – это Sticky Bit. В случае, если этот бит установлен для папки, то файлы в этой папке могут быть удалены только их владельцем. Пример использования этого бита в операционной системе — это системная папка /tmp. Эта папка разрешена на запись любому пользователю, но удалять файлы в ней могут только пользователи, являющиеся владельцами этих файлов.

Для того, чтобы его установить, нам нужно написать следующую команду:

chmod +t “Путь к директории”, после этого, права на директорию будут выглядеть следующим образом:

𝑑𝑟𝑤𝑥𝑟 − 𝑥𝑟 − 𝑡



Рис. 16 Бит Sticky bit

Если мы хотим задать Sticky Bit при помощи чисел (абсолютного способа), то следует написать:

chmod 1644 “Путь к директории”, тогда директория получит Sticky Bit и стандартный набор прав 644.



Рис. 17 Бит Sticky bit с помощью чисел

# Новые файлы и директории

При создании файла пользователем, он как создатель получает права r (readable) и w (writeable) – это значит, что он может читать и изменять содержимое этого файла. Группа, в которой он состоит получает права r (readable), то есть все остальные пользователи его группы могут только прочитать содержимое файла. Остальные пользователи также получают права только на чтение. Выглядит это так:

−𝑟𝑤 − 𝑟 − −𝑟 − −

При создании файла права определяются по определенному алгоритму (формуле). Исходными правами доступа для файла является набор восьмеричных цифр 0666, как было рассмотрено ранее, эти права должны означать −𝑟𝑤 − 𝑟𝑤 − 𝑟𝑤 − , но это не так.

Все связано с тем, что у каждого пользователя в Linux есть “маска”, узнать которую можно написав команду umask, стандартной маской считается значение 0022. Именно из-за того, что указана такая маска, права на файл рассчитываются следующим образом: 0666 - 0022 = 0644, что и означает:

−𝑟𝑤 − 𝑟 − −𝑟 − −



Рис. 18 Права на созданный файла

При создании директории пользователем, он как создатель получает права на просмотр, изменение и вход в директорию:

𝑑𝑟𝑤𝑥𝑟 − 𝑥𝑟 − 𝑥

Его группа и все остальные пользователи, как мы видим получают все права, кроме прав на изменении директории, то есть они не могут создавать там свои файлы/директории, либо удалять их.

Как описано ранее для новых файлов есть специальная формула получения прав, для директорий она тоже есть и выглядит она так: 0777 – 0022 = 0755, стандартная маска = 0022. Отсюда и получается, что 0755 – это:

𝑑𝑟𝑤𝑥𝑟 − 𝑥𝑟 − 𝑥



Рис. 18 Права на созданную директорию

В данном случае первый бит занимает буква d (directory) – директория, что нам явно показывает на тип обьекта.

# Типы файлов в Linux

Ранее мы упомянули о том, что первый бит занимает тип файла, теперь разберемся какие типы файлов вообще существуют в Linux, с которыми работает команда chmod:

1. “-“ – Обычный файл (file) - Именованная область данных на носителе информации. Абстракция, используемая как базовый объект для удобства взаимодействия с данными в операционных системах.
2. “d” – Каталог (directory) - Объект в файловой системе, упрощающий организацию файлов. Каталог реализован как специальный файл, где регистрируется информация о других файлах и каталогах на носителе информации.
3. “p” – Именованный канал (pipe) — Это канал, у которого есть имя в файловой системе. В программировании именованный канал или именованный конвейер

— один из методов меж-процессного взаимодействия, расширение понятия конвейера в Unix и подобных ОС. Именованный канал позволяет различным процессам обмениваться данными, даже если программы, выполняющиеся в этих процессах, изначально не были написаны для взаимодействия с другими программами.

1. “l” – Символическая ссылка (link) — Специальный файл в файловой системе, в котором вместо пользовательских данных содержится путь к файлу, открываемому при обращении к данной ссылке (файлу).
2. “c” – Файл символьного устройства (device file) - Вид файла устройства в UNIX/Linux-системах, обеспечивающий интерфейс к устройству, реальному или виртуальному, с возможностью посимвольного обмена информацией
3. “b” – Файл блочного устройства – Специальные файлы устройств содержат данные, необходимые операционной системе для взаимодействия с физическими устройствами.
4. “s” – Сокет (socket) - название программного интерфейса для обеспечения обмена данными между процессами. Сокет — абстрактный объект, представляющий конечную точку соединения.
5. “D” – Дверь (door) - механизм межпроцессного взаимодействия, используемый в ряде операционных систем семейства Unix, разновидность функционального вызова.

Утилита chmod редактирует права доступа для каждого из перечисленного типов файлов.

# Заключение

В заключении хотим отметить, что утилита chmod в Linux предоставляет широкие возможности для защиты файлов и директорий. Правильное управление правами доступа позволяет обеспечить безопасность данных и предотвратить несанкционированный доступ к ним.

Понимание синтаксиса утилиты и специальных типов разрешений является важным для эффективной защиты файловой системы.

Также важно учитывать типы файлов, так как каждый из них имеет свои особенности в контексте управления правами доступа. Использование правильных методов защиты файлов и директорий с помощью утилиты chmod поможет обеспечить надежность и безопасность в работе с данными в Linux.

# Список использованных источников

1. <https://linux-faq.ru/page/komanda-chmod>- команда chmod, выдача прав доступа.
2. <https://habr.com/ru/companies/ruvds/articles/523872/> - Системы защиты Linux.
3. [https://unisonhome.ru/nastroika-sistemy-zashhity-failov-i-papok-v-operacionnoi-](https://unisonhome.ru/nastroika-sistemy-zashhity-failov-i-papok-v-operacionnoi-sisteme-linux/) [sisteme-linux/](https://unisonhome.ru/nastroika-sistemy-zashhity-failov-i-papok-v-operacionnoi-sisteme-linux/) - Основные принципы защиты файлов и папок в Linux.
4. <https://zinvapel.github.io/it/tools/2018/01/10/linux-users/> - Управление доступом в Linux.
5. <https://uraganltd.blogspot.com/2018/02/unixchmod.html> - Права на папки и файлы (unix/chmod).
6. <https://itdid.ru/file_permissions.html>- Unix|Linux всё о правах доступа к файлам.