Лекция 3. Права доступа и пользователи

Цель лекции

- Узнать, какие бывают пользователи в Linux
- Научиться управлять пользователями
- Освоить модель управления доступом к файлам
- Изучить механизм команды sudo

Дополнительные материалы (ссылки, файлы)

Термины

Пользователь — ключевое понятие организации системы доступа в Linux. Когда пользователь регистрируется в системе, то есть проходит процедуру авторизации, например, вводя системное имя и пароль, он идентифицируется с учётной записью. В ней система хранит информацию о каждом пользователе: его системное имя и некоторые другие сведения, необходимые для работы с ним.
Именно с учётными записями, а не с самими пользователями, и работает система. Ниже приведён список этих сведений.
Учётная запись — хранимая в компьютерной системе совокупность данных о пользователе, необходимая для его опознавания (аутентификации) и предоставления доступа к его личным данным и настройкам.
Права доступа — совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы.
Группа пользователей — именованное множество пользователей с одинаковыми правами доступа к тем или иным объектам.

План

1. Стандартная классификация пользователей в Linux.
2. Обычные пользователи.
3. Системные пользователи.
4. Суперпользователь root.
5. Понятие UID, GID.
6. Хранение данных пользователей: /etc/passwd, /etc/group, /etc/shadow.
7. Создание, изменение пользователей: useradd, adduser, usermod, userdel.
8. Создание и изменение паролей: passwd.
9. Создание и удаление групп groupadd, groupdel.

10. Работа механизма sudo и его настройка.
11. Изменение владельца и группы владельца файлов.
12. Права доступа к файлам rwx.
13. Специальные биты доступа SUID, SGID, sticky bit.
14. Права по умолчанию и umask.
15. Различие обычных файлов, символических ссылок и директорий.

Текст лекции

Стандартная классификация пользователей в Linux

Для работы в операционной системе нам необходим пользователь. Именно пользователь в ОС идентифицирует нас для получения доступа к любым ресурсам. Также на пользователях основана система безопасности Linux – фиксация действий, ограничение полномочий и т.д.
Но не все пользователи в Linux равнозначны. Есть несколько видов пользователей для решения разных задач: обычные пользователи, системные пользователи и суперпользователь. Давайте посмотрим, зачем существует такое разделение и как его использовать на практике.

Обычные пользователи

Первый тип пользователей – **обычные пользователи**. Именно такого пользователя мы создавали, когда устанавливали систему. Обычные пользователи нужны для полноценной работы в системе. Их основные свойства: наличие домашней папки в каталоге /home, наличие пароля и рабочей оболочки (обычно bash). Именно существование пароля и оболочки даёт нам возможность заходить в систему через SSH и исполнять команды. Как правило, в своей домашней папке (/home/{user}), такой пользователь имеет полный доступ.

Системные пользователи

Второй тип пользователей – **системные пользователи** или пользователи-демоны. Такие пользователи нужны для запуска от их имени процессов, обычно демонов (процессов, работающих в фоне без взаимодействия с пользователями). Хотя мы их выделяем в отдельный тип, различия с обычными пользователями не велики. Системные пользователи как правило не имеют пароля, домашняя директория может не существовать или находиться за пределами /home/ и также они не имеют оболочки. Таким образом, мы не можем войти в систему и выполнять команды с использованием системных пользователей. Кроме того, права доступа к файлам настроены так, что системные пользователи имеют доступ только к тем файлам, с которыми должен работать процесс, связанный с этим пользователем. Все остальные права максимально ограничены. Сделано это для снижения рисков некорректной

работы приложений, система не позволит затронуть файлы других приложений и компонентов ОС.

Суперпользователь root

Суперпользователь или суперадминистратор – это отдельный тип пользователей, которые имеют неограниченный доступ к системе. Традиционно такой пользователь в системе один и имеет логин root (хотя возможны другие варианты).
Суперпользователь отличается идентификатором (UID = 0) и не ограничен правами доступа к файлам. Также root может полноценно управлять системой: устанавливать софт, менять настройки, запускать службы и т.д.
Раньше во многих системах этот пользователь использовался как основной, имел пароль, с ним можно было зайти в систему и работать. Сейчас большинство дистрибутивов закрывают прямой доступ к пользователю root, чтобы не допускать удалённый доступ к системе с полными правами. То есть, сейчас root существует, но не имеет пароля. Однако, административные задачи также должны решаться с использованием суперпользователя, поэтому мы получаем root-права через механизм sudo. О том, как работает этот механизм, мы поговорим немного позже.
Теперь настало время разобраться с идентификаторами пользователей (UID, GID).

Понятие UID, GID.

UID — идентификатор пользователя. Операционная система различает пользователей именно по UID, а не, например, по логину. Есть возможность создать двух пользователей с разными логинами, но одинаковым UID, что позволит обоим пользователям иметь одинаковые права доступа в системе. Это нарушение безопасности. UID у каждого пользователя должен быть уникальным, в ОС не должно быть двух пользователей с одинаковым UID.
UID — это число из диапазона от 0 до 65535, при этом UID 0 назначается суперпользователю. Во многих ОС диапазон от 1 до 999 используется под системные нужды, всё остальное — обычные пользователи.
GID — идентификатор группы пользователей. Каждый пользователь в ОС Linux принадлежит как минимум к одной группе — группе по умолчанию, которая создаётся одновременно с учётной записью пользователя и как правило совпадает с именем пользователя. У пользователя может быть несколько групп. Пользователь может входить в группу с GID 0 (группа суперпользователя root), и это не будет нарушением безопасности. Группы необходимы для регулирования доступа нескольких пользователей к различным ресурсам.
Увидеть все основные свойства пользователей (включая UID и GID) и групп можно в обычных текстовых файлах, которые расположены в каталоге /etc.

Хранение данных пользователей: /etc/passwd, /etc/group, /etc/shadow.

Первый файл это **/etc/passwd** – он содержит информацию о пользователях. Внутри он представляет собой простейшую базу данных, где каждая строчка – пользователь, а свойства пользователя разделены двоеточием.
Если перечислить все свойства слева направо, получим:

1. логин пользователя;
2. пароль (не используется);
3. UID;
4. GID (основная группа);
5. полное имя пользователя и дополнительные поля;
6. путь к оболочке.

То есть, просто посмотрев этот файл (он доступен на чтение всем пользователям), мы можем получить общую картину по всем пользователям в нашей системе.
Например, если у пользователя в качестве пути к оболочке стоит /usr/sbin/nologin или /bin/false, то мы понимаем, что зайти с этим пользователем по SSH или в консоли не получится. Можем найти и себя в списке, обычно это пользователь ближе к концу файла с UID 1000.
Второй файл – **/etc/group**. Здесь хранится информация о группах пользователей. Формат сходный с /etc/passwd. Поля:

1. имя группы;
2. пароль группы (не используется);
3. GID (идентификатор группы);
4. список членов группы (через запятую).

Из этого файла мы получаем информацию о полном списке групп и членстве пользователей в этих группах. Однако с последним моментом информация не полная. Дело в том, что основная группа у пользователя прописана в файле /etc/passwd. В файле /etc/group прописано членство пользователей только в дополнительных группах. Например, для нашего основного пользователя создаётся группа GID 1000, в списке которой нет записи о членах, хотя наш пользователь там есть (это его основная группа).
Третий файл – **/etc/shadow**. Здесь находятся пароли пользователей и их свойства. Права доступа к файлу сильно ограничены, так как содержит секретную информацию. Формат его тот же, что и у /etc/passwd.
Поля слева направо:

1. логин пользователя;
2. пароль (в защищённом виде);
3. дата последнего изменения пароля;
4. минимальное число дней между изменениями паролей;
5. максимальное время жизни пароля;
6. количество дней до истечения срока действия пароля;

7. количество дней после истечения срока действия пароля, когда учётная запись будет отключена;
8. срок действия учетной записи.

Просмотр этого файла в чистом виде даёт меньше информации, чем предыдущие файлы. В основном мы можем понять, у каких пользователей установлен пароль, а также какие пользователи заблокированы. Блокировка пользователя отображается как символ “!” перед паролем.
Несмотря на ограниченные права доступа, пароли в этом файле хранятся в виде хешей. Хеш-функция – функция, осуществляющая преобразование массива входных данных произвольной длины в выходную битовую строку установленной длины, выполняемое определённым алгоритмом. При проверке корректности ввода пароля система применяет хеш-функцию к введённому паролю. Если значение хеша совпадает с записанным в /etc/shadow, пароль верный. Если нет – значит пароль неправильный. Причем из хеша получить исходный пароль очень сложно, поэтому такой формат хранения паролей наиболее безопасный.
Итак, мы разобрали структуру основных файлов, хранящих информацию о пользователях и группах. Теперь приступим к управлению этими сущностями.

Создание, изменение пользователей: useradd, adduser, usermod, userdel

За создание пользователя отвечает команда **useradd**. Эта команда позволяет задать все нужные свойства пользователя в виде параметров и не требует дополнительного ввода от пользователя.
Например:
useradd -s /bin/bash -m -d /home/testuser testuser
Эта команда добавляет обычного пользователя с домашним каталогом (-m и -d) и стандартной оболочкой bash (-s), последний параметр – логин пользователя. Для полноценной работы останется назначить пароль командой passwd (об этом ниже). При создании пользователя в его домашней директории будут все файлы из каталога /etc/skel, а также создана одноимённая группа, которая станет для него основной. При желании можно не создавать домашний каталог (опция -M).
Другой вариант создания пользователя – скрипт **adduser**. Эта команда по умолчанию создаёт пользователя с теми параметрами, которые мы использовали для команды useradd. Все дополнительные данные (пароль, полное имя, дополнительные поля) скрипт спросит в процессе создания. Такой вариант подходит для интерактивного добавления пользователя.
Не забываем, что для модификаций свойств пользователей (или их создания) нам потребуются root-права. Поэтому команды запускаем либо от пользователя root, либо через sudo.
Для изменений свойств пользователя используется команда **usermod**. Её параметры практически полностью копируют параметры команды useradd. Смысл меняется с

“установить значение” на “изменить значение”. Например, опция -m означает перенос домашнего каталога. Самая частая конфигурация этой команды:
usermod -aG sudo testuser
Здесь мы добавляем (-a) пользователя testuser в группу (G) sudo. При этом его основная группа не меняется. Для смены основной группы необходим параметр -g:
usermod -g www-data testuser
За удаление пользователей отвечает утилита **userdel**. Для удаления пользователя и его домашней директории (-r) выполняем следующую команду:
userdel -r testuser
Теперь пора потренироваться в создании паролей для пользователей.

Создание и изменение паролей: passwd, chage

Для установки пароля используется утилита passwd. Если вызвать passwd без параметров, то будет меняться пароль текущего пользователя.
Для изменения пароля другого пользователя необходимо вызывать команду через sudo и указать логин:
sudo passwd testuser
Для настройки параметров паролей (срок действия, срок ротации и т.д.) необходима утилита chage. Она работает в интерактивном режиме и последовательно принимает значения всех параметров. Для просмотра текущих настроек пароля используем параметр -l:
sudo chage -l testuser
Мы научились проводить базовые настройки пользователей, самое время перейти к группам пользователей.

Создание и удаление групп groupadd, groupdel

Работа с с группами пользователей намного проще, чем с пользователями. У групп гораздо меньше свойств.
Для создания группы мы используем утилиту **groupadd**. Базовый вариант выглядит максимально просто:
groupadd testgroup
Здесь мы создаём группу testgroup с параметрами по умолчанию. В этом случае GID будет выставлен автоматически, если нужно вручную – используем параметр -g {GID}.
Для удаление группы используем **groupdel**:
groupdel testgroup
Добавление и удаление пользователей из группы относится к редактированию пользователей, что мы уже рассмотрели выше.
Вооружившись знаниями о пользователях и группах мы можем приступать к правам доступа, начнём с изменения владельца и группы файлов и директорий.

В нашей системе основной пользователь не является суперпользователем, тем не менее мы можем администрировать Linux через механизм sudo, давайте разберёмся как он работает.

Работа механизма sudo и его настройка

Как известно, в ОС Linux всегда есть один суперпользователь (администратор системы) root. У этого пользователя абсолютно неограниченные права на всю систему, начиная от установки пакетов, заканчивая удалением файлов и каталогов. Ограничить свободу действий в системе пользователя root практически невозможно. Во избежание ошибочных действий, которые могут привести к краху системы, работа под пользователем root не рекомендуется. А для выполнения административных действий обычным пользователем используют две утилиты: su и sudo.
su — команда, которая позволяет переключаться в пользователя (switch user, substitute user) или делает пользователя суперпользователем, при этом не завершая сеанс. Синтаксис: su - user_name — далее вводится пароль и меняется ID текущего пользователя. su - без параметров переключит текущего пользователя в суперпользователя. Данный метод работы под суперпользователем не очень хорош, так как нет никаких ограничений.
sudo — утилита, которая позволит выполнять административные действия в системе согласно настройкам в файле /etc/sudoers. Файл /etc/sudoers редактируется только пользователем, имеющим права администратора системы. В этом файле перечисляется набор административных команд, которые разрешено выполнять пользователю или группе пользователей. В Ubuntu пользователи, входящие в группу sudo, могут выполнять административные действия без каких-либо ограничений. Не рекомендуется злоупотреблять количеством участников данной группы.
Файл /etc/sudoers должен редактироваться или командой visudo, которая запускает текстовый редактор и позволяет избежать большинства синтаксических ошибок.
Синтаксис записи в файле /etc/sudoers:

1. User_name ALL= full_path_to_command. Например, запись user ALL= /usr/sbin/adduser позволит пользователю с именем user, используя sudo, добавлять учётные записи в системе.
2. User_name ALL=(ALL) ALL позволит пользователю, используя утилиту sudo, выполнять административные действия без ограничений.
3. %sudo ALL=(ALL) NOPASSWD:ALL позволит всем пользователям, входящим в группу sudo, выполнять любые административные действия в системе без подтверждения паролем. В целях безопасности не рекомендуется использовать в многопользовательских решениях.

Изменение владельца и группы владельца файлов

У каждого файла или директории есть владелец (owner) и группа владельца (group). Эти два параметра определяют то, как будут действовать права доступа.

Для изменения владельца нужно использовать утилиту **chown** (change owner). Мы можем передать только владельца (chown testuser) или владельца и группу (chown testuser:testgroup). Для изменения прав на вложенные элементы директорий (рекурсивно) нужно использовать опцию -R. Например:
chown -R testuser:testgroup testdir/
Если нужно изменить только группу владельца, то здесь полезна команда **chgrp**. Она имеет те же параметры, что и chown, только мы можем передавать исключительно группу. Пример:
chgrp -R testgroup testdir/
Не забываем, что в большинстве случаев нам потребуются административные права для изменения владельца или группы – используем root или sudo.
Теперь мы готовы перейти к главному: назначению прав доступа к файлам.

Права доступа к файлам rwx

Ограничение прав доступа – важнейший аспект системы безопасности в Linux. Права доступа также называют битами доступа, потому что каждый символ в правах (например, r, w или x) означает флаг, который имеет 2 значения – включен или выключен.
Всего в стандартной системе прав доступа три группы по три бита в каждой (rwx). Первая группа относится ко владельцу, вторая – к группе, третья – к остальным. Поиск совпадения производится слева направо. Если наш пользователь является владельцем файла, то он получает права владельца. Если он не владелец, но состоит в группе, которая назначена для файла, то он получит права для группы. И, наконец, если совпадения ни с владельцем, ни с группой не случилось, применяются права для остальных.
Биты прав доступа мы наблюдаем при вызове ls -l в первом столбце. Например:
-rw-rw-r–
Этот пример показывает обычный файл (-) с правами доступа rw (чтение и запись) для владельца и группы, и r (чтение) для остальных.
Символы в блоках — это права доступа. Они расшифровываются следующим образом:

- r (read) — возможность открытия и чтения файла или просмотр содержимого каталога.
- w (write) — возможность изменить содержимое файла или возможность создавать, удалять или переименовывать объекты в каталоге.
- x (execute) — возможность выполнить файл (запустить программу, скрипт) или возможность войти в каталог и получить атрибуты объектов.

Права доступа можно представить в численном виде, используя восьмеричную систему счисления:

- 0 – нет битов;
- 1 – x;
- 2 – w;
- 3 – wx;

- 4 – r;
- 5 – rx;
- 6 – rw;
- 7 – rwx.

За изменение прав доступа отвечает команда **chmod** (change mode).
Для назначения прав доступа мы можем использовать как текстовые обозначения, так и цифровые. Начнём с примера в символьном виде:
chmod u=rwx,g+w,o-r testfile
В этом примере мы назначили права rwx для владельца – u, добавили право на запись для группы и убрали право на чтение для остальных. Таким образом, у нас есть несколько типов субъектов прав:

- u (user) – владелец;
- g (group) – группа;
- o (others) – остальные;
- a (all) – все.

Далее мы для каждого раздела (ugoa) назначаем действие:

- = установить точные права;
- + добавить биты доступа;
- - убрать биты.

Такая форма удобна, не требует запоминания восьмеричных цифр битов доступа. Однако, если мы знаем точную формулу, гораздо быстрее назначить права в цифровом виде:
chmod -R 755 testdir/
Здесь мы назначили права (rwxr-xr-x) рекурсивно для всех вложенных элементов и самой директории. Получилось коротко и ёмко. Нужно помнить, что менять права доступа может либо владелец файла, либо суперпользователь root.
Для расширенного управления доступом существуют специальные биты доступа, которые мы рассмотрим ниже.

Специальные биты доступа SUID, SGID, sticky bit

Кроме вышеперечисленных атрибутов прав доступа к файлам и каталогам, существуют ещё специальные биты SUID, SGID, Sticky.
SUID (set user ID upon execution) — установка ID пользователя во время выполнения. Разрешает пользователям запускать файл на исполнение с правами того пользователя, которому принадлежит данный файл. В символьной записи присваивается следующим образом: chmod u+s file, где s — бит SUID. В численной записи имеет значение 4000, например chmod 4755 file присвоит права на чтение-запись-выполнение для владельца, права на чтение-выполнение — для группы владельца и всех остальных, 4 — выставит бит SUID. SUID работает с файлами.
SGID (set group ID upon execution) — установка ID группы во время выполнения, применяется преимущественно к каталогам. Данный атрибут устанавливает идентификатор группы каталога, а не группы владельца, который создал файл в этом

каталоге. В символьной записи присваивается следующим образом: chmod g+s dir1, где s — бит SGID. В численной записи имеет значение 2000, например chmod 2665 dir1 присвоит права на чтение-запись-просмотр для владельца и группы владельца и чтение-просмотр для всех остальных, а также установит на каталог бит SGID — 2.
Sticky bit — дополнительный атрибут, который устанавливается для каталогов. Файлы из каталога с таким битом может удалить только владелец (пользователь, создавший этот файл) или владелец каталога. В символьной записи присваивается следующим образом: chmod +t dir1 добавит к каталогу sticky bit. В численной записи имеет значение 1000, например chmod 1666 dir1 установит на каталог права на чтение-запись-просмотр для всех типов пользователей, но при этом удалять файлы могут только создатели благодаря установленному биту sticky.
Итак, мы рассмотрели базовые и специальные биты доступа. Но всё ещё остаётся вопрос: откуда берутся права на файлы и директории по умолчанию, когда мы их создаём? Пора ответить на этот вопрос.

Права по умолчанию и umask

Когда мы создаём файл или директорию они получают права по умолчанию. Настройками этих прав заведует команда **umask**. Для определения прав по умолчанию нужно знать значение umask, которое применяется к нашему сеансу в данный момент. Узнать значение можно просто выполнив команду umask. Мы получим значение из четырёх восьмеричных цифр, например 0022. Эти цифры относятся к битам доступа, которые будут вычитаться из набора полных прав по умолчанию: 666 для файлов и 777 для директорий. Первая цифра относится к специальным битам и не используется, вторая – для владельца, третья – для группы, четвёртая – для остальных.
Определить права по умолчанию можно вычитая значение umask для каждого разряда из полных прав. Например, если umask имеет значение 022, то для файлов права будут 644, а для директорий 755.
Изменить значение umask можно командой с набором битов. Например: umask 002.
Установить umask можно также в настройках оболочки пользователя (в файле ~/.bashrc) или глобально для всех пользователей (в /etc/bash.bashrc).

Различие обычных файлов, символических ссылок и директорий

Права доступа применяются по-разному для разных объектов: обычных файлов, директорий и символических ссылок. С обычными файлами мы уже разобрались, теперь нужно поговорить об остальных файлах.
Начнём с символических ссылок: здесь не имеет смысла смотреть на права доступа, он всегда полные (rwxrwxrwx), но реально будут применяться права файла, на который она ссылается.
Если мы говорим про жесткие ссылки, то права доступа хранятся в inode, значит у всех жестких ссылок всегда будут идентичные права доступа.

Для директорий особое значение имеют все обычные биты, как мы говорили выше. Не забывайте. что для обычного доступа на чтение директории нужны как минимум биты r и x.

Итоги занятия

- Изучили классификацию пользователей
- Рассмотрели основные атрибуты пользователей и групп
- Узнали, где хранятся сведения о пользователях
- Научились добавлять, изменять и удалять пользователей и группы
- Разобрались в механизме распределения прав доступа к файлам