

Лабораторная работа

Управление пользователями. Контроль доступа к файлам

Цель работы:

- знакомство с форматами файлов для хранения информации учетных записей
- изучение команд для управления учетными записями

Теоретические сведения

Все операционные системы семейства GNU/Linux - многопользовательские, поэтому каждый физический пользователь должен быть зарегистрирован в системе, т. е. иметь собственное имя пользователя, которое используется для идентификации, аутентификации и разграничения прав доступа в системе, а также мониторинга и аудита пользовательской активности .

При присвоении имени новому пользователю следует соблюдать следующие правила:

- имя может содержать символы латинского алфавита (a-z, A-Z), цифры (0-9), символы подчеркивания, тире и точку
- может оканчиваться символом «\$»
- нельзя использовать: пробел, :, @, # и специальные символы (табуляция, конец строки)
- не может начинаться с тире, содержать только цифры или состоять из «.» или «..»
- не рекомендуется использовать точку в начале имени.

1. Учётные записи в Linux

В операционных системах GNU/Linux, учётная запись пользователя представляет собой структурированный набор данных, необходимых для идентификации субъекта в системе и определения его прав доступа. Фундаментальной особенностью Linux является разделение привилегий, реализуемое через механизм учётных записей.

Система хранения информации о пользователях организована в виде трёх основных файлов:

- /etc/passwd - содержит основные атрибуты учётных записей
- /etc/shadow - хранит зашифрованные пароли с дополнительными параметрами безопасности
- /etc/group - определяет групповую принадлежность пользователей.

2. Структура файла /etc/passwd

Файл /etc/passwd представляет собой текстовую базу данных, где каждая строка соответствует одной учётной записи и состоит из семи полей, разделённых двоеточиями:

```
username:password:UID:GID:gecos:homedir:shell
```

Чтобы просмотреть содержимое файла, используйте текстовый редактор или, например, команду cat: `cat /etc/passwd`.

Username - уникальный идентификатор пользователя в системе. В Linux имена пользователей чувствительны к регистру и могут содержать до 32 символов.

Password - в современных системах это поле всегда содержит значение 'x', указывающее на использование файла `/etc/shadow` для хранения паролей. Исторически здесь хранился хэш пароля в формате DES.

UID - числовой идентификатор пользователя. Диапазон значений: 0: суперпользователь (root), 1-999: системные пользователи, 1000+: обычные пользователи.

GID - идентификатор основной группы пользователя. При создании файлов именно этот GID будет назначаться по умолчанию.

GECOS - поле для дополнительной информации, часто содержащее: полное имя пользователя, номер кабинета/офиса, контактные телефоны, должность или академическую группу.

Home - рабочая директория пользователя, куда устанавливаются персональные настройки и сохраняются пользовательские файлы.

Shell - интерпретатор команд по умолчанию. В большинстве дистрибутивов Linux оболочкой входа по умолчанию является Bash. Для служебных учётных записей часто указывают `/sbin/nologin` или `/bin/false`.

Обычно первая строка описывает пользователя root, за которым следуют системные и обычные учётные записи пользователей. Новые записи добавляются в конец файла.

3. Привилегированная учётная запись root

Суперпользователь (root) обладает уникальными характеристиками: он обладает неограниченными правами в системе, возможностью изменения любых системных параметров, его UID и GID равны 0, у него специальный домашний каталог (`/root`).

4. Механизмы повышения привилегий

В операционных системах GNU/Linux существует два основных способа получения административных прав: `su` и `sudo`.

Команда `su` позволяет переключиться на другого пользователя, включая root, для этого требуется ввести пароль целевого пользователя. По умолчанию, если не указано имя пользователя, `su` переключается на root. Эта команда запускает новую оболочку с правами выбранного пользователя, предоставляя полный доступ к системе без дополнительного контроля. Команда `sudo` выполняет только одну указанную команду с повышенными привилегиями, обычно от имени root, но запрашивает пароль текущего пользователя, а не root. Sudo более безопасен, потому что позволяет гибко настраивать права через файл `/etc/sudoers` и ведёт логи всех выполненных команд, что упрощает аудит.

5. Основные команды для управления учётными записями в Linux

Adduser - это скрипт, который запрашивает пароль и создаёт домашнюю директорию (`/home/имя_пользователя`), `useradd` - низкоуровневая команда,

требует ручной настройки (пароль, shell, домашняя папка) с помощью опций. Оба варианта требуют `sudo` для выполнения.

Примеры:

Интерактивное создание (с паролем и домашней директорией)
`adduser имя_пользователя`

Базовое создание (без пароля и дополнительных настроек)
`useradd имя_пользователя`

Просмотр информации о пользователе:

`id имя_пользователя` # UID, GID и группы

`finger имя_пользователя` # Контактные данные

`getent passwd имя_пользователя` # Основные данные из
/etc/passwd

Удаление пользователя:

Удаляет пользователя, но оставляет его файлы
`sudo userdel [опции] имя_пользователя`

С опцией `-r` также удаляется домашняя директория пользователя.

Изменение настроек и параметров уже существующей учетной записи:

`usermod [опции] имя_пользователя`.

Группы пользователей

Группы являются удобным инструментом для управления правами доступа и упрощения администрирования пользователей. В файле `/etc/group` хранится информация о существующих группах пользователей. Каждая группа представлена отдельной строкой, содержащей четыре поля, разделённых двоеточиями: `groupname:password:GID:members`.

`Groupname` - уникальное название группы, имя группы должно содержать только строчные буквы, цифры и дефисы.

`Password` - в современных системах обычно содержит `x`, это означает, что пароль (если он задан) хранится в `/etc/gshadow`.

`GID` - уникальный числовой идентификатор группы (диапазон: 0-999 для системных групп, 1000+ для пользовательских).

`Members` - перечень имён пользователей, входящих в группу через запятую.

Файл `/etc/group` доступен для чтения всем пользователям, редактировать его могут только администраторы.

Управление паролями

Блокировка и разблокировка учётной записи:

`sudo usermod -L имя_пользователя` # Заблокировать (Lock)

`sudo usermod -U имя_пользователя` # Разблокировать (Unlock)

Изменение пароля

Обычный пользователь может изменить пароль только для себя, `root`-пользователь может изменить пароль любого пользователя. Синтаксис команды: `passwd имя_пользователя`.

Опции:

`-l` - заблокировать учётную запись

`-u` - разблокировать

- e - принудительно сбросить пароль (пользователь сменит его при следующем входе)
- d - удалить пароль (оставить учётную запись без пароля)
- S - показать статус пароля (дата изменения, срок действия).

Политики паролей

Настройка срока действия пароля осуществляется командой `chage`.

Пример:

```
sudo chage -l username    # просмотр текущих настроек
sudo chage -M 90 username # установить срок действия 90 дн
ей
```

Настройка пользовательского окружения и логирование

Файлы профиля - это специальные конфигурационные скрипты, которые автоматически выполняются оболочкой (обычно `bash`) в определенных ситуациях. Эти файлы содержат три основных типа настроек:

- переменные окружения:
 - `PATH` - список каталогов для поиска программ
 - `USER` и `HOME` - информация о пользователе
- `UMASK` - базовые права для новых файлов
- алиасы - удобные сокращения для длинных команд
- параметры оболочки.

Системные файлы, применяемые для всех пользователей:

`/etc/profile` - главный системный конфигурационный файл. Содержит настройки, которые должны применяться ко всем пользователям системы. Выполняется только при входе в систему.

`/etc/bashrc` - системные настройки для `bash`. Выполняется при каждом запуске интерактивной оболочки (терминала). В нем обычно размещают общесистемные алиасы.

Пользовательские файлы:

`~/.bash_profile` - персональные настройки пользователя при входе в систему. Если этого файла нет, используется `~/.profile`.

`~/.bashrc` - индивидуальные настройки для каждой консоли. Содержит персональные алиасы, цветовые схемы, кастомные функции, настройки приглашения командной строки.

Основные сценарии выполнения:

- при входе в систему настройки применяются один раз за сессию, сначала загружается `/etc/profile`, затем `~/.profile`, `~/.bashrc` и `/etc/bash.bashrc`
- при запуске терминала настройки применяются для каждой новой консоли, выполняются только `~/.bashrc` и `/etc/bash.bashrc`.

Логирование пользовательской активности

В Linux ведется автоматическая запись информации о действиях пользователей. Основные файлы логов:

- `/var/run/utmp` - содержит данные о текущих активных сеансах; для просмотра используется команды `who` или `w`

- /var/log/wtmp - хранит историю всех входов и выходов, позволяет узнать, когда и кто работал в системе; анализируется командой last
- /var/log/lastlog - показывает, когда пользователь входил в систему в последний раз; для просмотра используется команда lastlog.

Права доступа к файлам в Linux

В операционных системах GNU/Linux права доступа к файлам и каталогам определяются для трех категорий пользователей тремя типами разрешений.

Категории пользователей:

- владелец (owner) - пользователь, который создал файл или каталог
- группа (group) - пользователи, входящие в группу, которой принадлежит файл или каталог
- остальные (others) - все остальные пользователи, не являющиеся владельцем и не входящие в группу.

Типы разрешений:

- чтение (r) - разрешает просмотр содержимого файла или списка файлов в каталоге;
- запись (w) - разрешает изменение файла или добавление/удаление файлов в каталоге;
- исполнение (x) - разрешает выполнение файла (для программ или скриптов) или вход в каталог.

Права доступа отображаются в виде строки из 10 символов:

-, d, l, c, b, p, s	r	w	x	r	w	x	r	w	x
тип объекта	права владельца			права группы			все остальные		

Символы в поле тип объекта означают: "-" - файл, d - каталог, l - символическая ссылка, c - символьное устройство, b - блочное устройство, p - именованный канал, s - сокет.

Каждому типу разрешения соответствует числовое значение ($r=4=100_2$, $w=2=010_2$, $x=1=001_2$), которые суммируются для каждой категории пользователей, например: $r-x = 100_2 + 001_2 = 101_2 = 5$. Таким образом, права $gwxr-xr--$ в числовом формате будут записаны как 754.

Специальные биты в правах доступа:

SUID (Set User ID) (S, s) - при установке (`chmod u+s`) файл выполняется с правами владельца, а не запустившего его пользователя. Записывается в x для владельца.

SGID (Set Group ID) (S, s) - при установке (`chmod g+s`) файл выполняется с правами группы владельца, а для каталогов новые файлы наследуют группу директории, а не создателя. Записывается в x для группы.

Sticky Bit (T, t) - при установке (`chmod +t`) в каталоге только владелец файла может его удалить/переименовать. Заменяет x для всех остальных.

Специальный бит записывается заглавной буквой при отсутствии права на выполнение.

При монтировании файловой системы можно указать параметр `nosuid` для отключения флагов `suid` и `setgid`.

Команды для управления правами доступа

В Linux управление правами доступа осуществляется с помощью нескольких утилит.

chmod (change mode) – изменяет права доступа (rwx, SUID, SGID, Sticky Bit) для файлов и каталогов.

Синтаксис команды:

Символьный формат: `chmod [ugoa][+|=][rwxXst] файл`

Пример:

```
chmod u+x script.sh #Дать владельцу право на выполнение
chmod 644 file.txt #rw-r--r-(только владелец может писать)
```

chown – изменяет владельца и/или группу файла.

Синтаксис команды:

`chown [опции] пользователь:группа файл`

Если не указать группу, изменяется только владелец.

Опции: R - рекурсивно (для каталогов); -reference=файл - скопировать права другого файла.

chgrp - смена группы, альтернатива chown для изменения только группы.

Синтаксис команды:

`chgrp [опции] группа файл`

umask – задает биты доступа, устанавливаемые по умолчанию для всех новых файлов. Бит **x** никогда не устанавливается для созданных файлов. Umask обладает «инверсной» логикой, т.е. единицы в маске задают нули в правах доступа, поэтому для обычного пользователя `umask=0002` или `-rw-rw-r--`.

Порядок выполнения работы

Задание 1. Изучение основных способов хранения информации о пользователях

1. Ознакомьтесь с содержимым файлов: `/etc/passwd`; `/etc/shadow`; `/etc/group`.
2. Для своей учетной записи выполните разбор информации, хранимой в указанных файлах.

Задание 2. Создание пользователей и работа с терминалами

1. Создайте двух пользователей: `user1` = ваша_фамилия_1 (латиницей) и `user2` = ваша_фамилия_2, задайте для них пароли.
2. Зарегистрируйтесь в трех разных терминалах: первый терминал - `user1`, второй терминал - `user2`, третий терминал - суперпользователь (`root`).
3. Проверьте права доступа: с правами `user1` попробуйте войти в каталог `/root`, объясните результат.
4. Используя команду `ls -la /`, просмотрите список основных каталогов, укажите, каких прав доступа недостает для входа в каждый из них.

Задание 3. Установка прав на каталоги и файлы

1. С правами root создайте два временных каталога:
`mkdir -m 777 /home/temp1`
`mkdir -m 1777 /home/temp2`
2. Проверьте права доступа к каталогам `/home/user1` и `/home/user2` (должны быть 755)
3. Вернитесь в консоль `user1` и создайте в домашнем каталоге четыре подкаталога: `qu1` (777), `qu2` (404), `qu3` (1333), `qu4` (505). Объясните, какие из этих прав лишены смысла и почему.
4. Установите `umask 022`. Поясните, какие права будут присваиваться новым файлам и каталогам.
5. В каждом из каталогов (`qu1`, `qu2`, `qu3`, `qu4`) создайте по три текстовых файла с именами месяцев (например, `jan`, `feb`, `mar`). Запишите в них календарь на соответствующий месяц текущего года (используйте команду `cal`).
6. В каких каталогах создание файлов не удалось? Почему?
7. Измените права доступа для "недоступных" каталогов (`qu2`, `qu4`), создайте файлы, а затем верните прежние права.

Задание 4. Дополнительные атрибуты файлов

1. С правами root заблокируйте файл `feb` от любых изменений (используйте `chattr`). Установите для файла `mar` запрет на любые операции, кроме добавления данных.
2. С правами `user1` попробуйте добавить строку `finish` в конец файлов `feb` и `mar`. Объясните результат.

В отчете по работе приведите: использованные команды, скриншоты по пунктам заданий, ответы на все вопросы, указанные в заданиях.

Дополнительное задание

Задание 1. SGID (Set Group ID) для каталогов

- Создайте каталог `/home/shared_dir` и установите SGID-бит:
`chmod g+s /home/shared_dir`
- Создайте файл в этом каталоге от имени `user1` и проверьте его группу.
- Создайте файл от имени `user2`. Объясните, почему оба файла имеют одну группу-владельца.

Задание 2. Sticky Bit

- Установите sticky bit на `/home/temp1`: `chmod +t /home/temp1`
- Попробуйте удалить файл в `/home/temp1`, созданный другим пользователем (например, `user2` удаляет файл `user1`). Объясните результат.

Задание 3. Найдите все файлы в системе с SUID/SGID:

```
find / -type f -perm /6000 2>/dev/null
```

- Выберите один из них (например, `/usr/bin/passwd`) и объясните, зачем нужны эти биты.
- Почему SUID-бит на `/bin/bash` — это угроза безопасности?

- Как администратор может найти и обезвредить опасные SUID-файлы?

Контрольные вопросы

1. В каком файле хранятся пароли учетных записей пользователей? В каком виде хранятся пароли?
2. Какая команда позволяет добавить пользователя с именем user_1 и установить в интерактивном режиме пароль?
3. Как можно изменить пароль учетной записи?
4. Как можно удалить пароль учетной записи? Приведите примеры
5. Как записываются права доступа к файлу в числовом формате? Приведите пример записи для файла с правом на чтение и исполнение для владельца.
6. Какая команда используется для изменения прав на выполнение скрипта script.sh для владельца файла?
7. Какие права нужны для входа в каталог?
8. Какой пользователь может удалить файл file с правами 750, размещенный в каталоге /home/user_1?
9. Как изменить владельца файла?
10. Приведите пример команды, создающей каталог dir_1 с правами на чтение и запись только для владельца и установленным Sticky bit, задав эти параметры в числовом формате