Лабораторная работа 2

Управление пользователями и паролями

/etc/passwd — файл, содержащий в текстовом формате список пользовательских учётных записей (аккаунтов).

Формат файла:

login : password : UID : GID : GECOS : home : shell

Каждая строка файла описывает одного пользователя и содержит семь полей, разделённых двоеточиями:

1. регистрационное имя или логин;
2. хеш пароля (см. ниже);
3. идентификатор пользователя;
4. идентификатор группы по умолчанию;
5. информационное поле GECOS (см. ниже);
6. начальный (он же домашний) каталог;
7. регистрационная оболочка, или shell.

Является первым и основным источником информации о правах пользователя операционной системы. Существует в большинстве версий и вариантов UNIX-систем. Обязан присутствовать в POSIX совместимой операционной системе.

Основным назначением /etc/passwd является сопоставление логина и идентификатора пользователя (UID). Изначально поле пароля содержало хеш пароля и использовалось для аутентификации. Однако, в связи с ростом вычислительных мощностей процессоров появилась серьёзная угроза применения простого перебора для взлома пароля. Поэтому все пароли были перенесены в специальные файлы, такие как /etc/shadow в GNU/Linux. Эти файлы недоступны для чтения обычным пользователям. Такой подход называется механизмом скрытых паролей.

Регистрационные имена должны быть уникальными и представлять собой строки не длиннее 32-х символов (любые, кроме двоеточия и символа новой строки). В некоторых старых версиях UNIX существует ограничение длины в 8 символов, оно же будет действовать при использовании административной базы данных NIS. По сути дела, имя пользователя — это его короткий и легко запоминаемый псевдоним, который используется при входе в систему и часто включается в адреса электронной почты.

Идентификатор пользователя — это число от 0 до 2^{32}-1. В старых системах оно может быть не более 32767. Пользователь с идентификатором 0 называется суперпользователем и имеет право на выполнение любых операций в системе. Принято соглашение о выделении «специальным» пользователям (bin, daemon), назначение которых — только запуск определённых программ, маленьких идентификаторов (меньше 100 или, в некоторых дистрибутивах Linux, меньше 500).

В UNIX пользователь может принадлежать к одной или нескольким группам, которые используются для задания прав более чем одного пользователя на тот или иной файл. Максимальное количество групп, в которых может состоять один пользователь, разное в разных вариантах системы.

Список групп с их участниками задаётся в файле /etc/group. В файле же /etc/passwd указывается идентификатор группы по умолчанию.

Всем файлам, созданным пользователем после регистрации в системе, будет автоматически присвоен этот номер группы (исключение — если для каталога, в котором создаётся файл, установлен в правах бит SGID, то будет присвоена такая же группа, как у самого каталога).

Файл /etc/group содержит записи обо всех группах в системе. Каждая его строка содержит:

1. символьное имя группы;
2. пароль группы — устаревшее поле, сейчас не используется. В нём обычно стоит «x»;
3. идентификатор группы, или GID;
4. список имён участников, разделённых запятыми.

Поле GECOS хранит вспомогательную информацию о пользователе (номер телефона, адрес, полное имя и так далее). Оно не имеет чётко определённого синтаксиса.

После входа в систему пользователь оказывается в своём домашнем каталоге. Исторически сложилось так, что домашний каталог пользователя root называется /root, а остальные имеют вид /home/имя_пользователя. Но могут применяться и другие схемы.

Если на момент входа в систему домашний каталог отсутствует, то система выдаёт сообщение об ошибке и отказывается допустить пользователя к командной строке. Такое поведение НЕ характерно для GNU/Linux. В большинстве дистрибутивов этой ОС просто выводится предупреждение, после чего пользователь попадает в каталог «/».

В поле регистрационной оболочки задаётся shell, то есть интерпретатор командной строки. Здесь может быть указана любая программа. Тем не менее, некоторые системы в целях безопасности требуют, чтобы суперпользователь явно разрешил использовать приложение в качестве интерпретатора командной строки. Для этого используется специальный файл /etc/shells, содержащий список «допустимых» оболочек.

Стандартные утилиты:

- ***vipw*** — запускает текстовый редактор, указанный в переменной среды EDITOR (или редактор по умолчанию, обычно vi), загружая в него копию файла /etc/passwd (в системах GNU/Linux /etc/shadow). После закрытия редактора переносит временную копию в сам файл. Не позволяет двум пользователям выполнять редактирование одновременно.
- ***useradd*** — создаёт новую учётную запись.
- ***usermod*** — изменяет данные учётной записи.
- ***userdel*** — удаляет существующую учётную запись.

- ***chfn*** — изменяет поле GECOS.
- ***chsh*** — устанавливает новый командный интерпретатор.
- ***passwd*** — задаёт новый пароль пользователя.

Механизм скрытых паролей

В файле /etc/shadow хранятся хеши паролей всех пользователей в системе. Процессы суперпользователя могут читать его напрямую, а для остальных создана специальная библиотека PAM. Она позволяет непривилегированным приложениям спрашивать у неё, правильный ли пароль ввёл пользователь, и получать ответ. Библиотека PAM как правило действует с привилегиями вызвавшего процесса. Таким образом, хеш не попадает «в чужие руки».

Файл /etc/shadow кроме имени (первое поле каждой строки) и хеша (второе поле) также хранят следующую информацию:

- дата последнего изменения пароля,
- через сколько дней можно будет поменять пароль,
- через сколько дней пароль устареет,
- за сколько дней до того, как пароль устареет, начать напоминать о необходимости смены пароля,
- через сколько дней после того, как пароль устареет, заблокировать учётную запись пользователя,
- дата, при достижении которой учётная запись блокируется,
- зарезервированное поле.

Даты обозначаются как число дней с 1 января 1970 года (начало эпохи UNIX).

Сигналы

Сигналы в UNIX, Unix-подобных и других POSIX-совместимых операционных системах являются одним из способов взаимодействия между процессами (англ. IPC, inter-process communication). Фактически, сигнал — это асинхронное уведомление процесса о каком-либо событии. Когда сигнал послан процессу, операционная система прерывает выполнение процесса. Если процесс установил собственный обработчик сигнала, операционная система запускает этот обработчик, передав ему информацию о сигнале. Если процесс не установил обработчик, то выполняется обработчик по умолчанию.

Названия сигналов «SIG…» являются числовыми константами (макроопределениями Си) со значениями, определяемыми в заголовочном файле signal.h. Числовые значения сигналов могут меняться от системы к системе, хотя основная их часть имеет в разных системах одни и те же значения. Утилита kill позволяет задавать сигнал как числом, так и символьным обозначением.

Управление отложенными заданиями

cron — демон-планировщик задач в UNIX-подобных операционных системах, использующийся для периодического выполнения заданий в определённое время.

Регулярные действия описываются инструкциями, помещенными в файлы crontab и в специальные директории.

Каждый пользователь системы имеет свой файл заданий crontab, в котором описано, в какое время и какие программы запускать от имени этого пользователя. Для редактирования файла crontab используется специальная одноименная программа crontab, позволяющая не прерывать процесс cron на время редактирования.

Для редактирования файла crontab вашего пользователя используется команда:

crontab -e

Основной файл конфигурации cron, /etc/crontab, выглядит примерно так:

```
* * * * * выполняемая команда
- - - - -
| | | | |
| | | | ----- День недели (0 - 7) (Воскресенье =0 или =7)
| | | ------- Месяц (1 - 12)
| | --------- День (1 - 31)
| ----------- Час (0 - 23)
------------- Минута (0 - 59)
```

Цель работы: изучить способы управления процессами.

Управление пользователями

1. Просмотрите файл /etc/shadow (с правами root). У всех ли пользователей содержимое второго поля выглядит приблизительно одинаково?
2. Какие символы могут содержаться в шифрованной строке пароля в /etc/shadow?
3. Зарегистрируйте пользователя test1, для которого запрещен вход в сеанс, имеющего домашний каталог /home/nouser и являющегося членом групп user и mail. Пользователь должен иметь UID равный 2000.
4. Создайте учетную запись для пользователя test2 с настройками по умолчанию. Проверьте, создался ли домашний каталог пользователя, наполнен ли он файлами и кому он принадлежит?
5. Измените имя пользователя test2 на test3.
6. Удалите пользователя test3.
7. Помимо файла /etc/default/useradd имеется еще один конфигурационный файл, влияющий на поведение команды useradd. Найдите его и изучите его содержание. Какая настройка позволяет изменять минимальный UID для новых пользователей?
8. Зарегистрируйте пользователя test4 с настройками по умолчанию и установите для него пароль. Изучите содержимое соответствующей записи в /etc/shadow.
9. Установите дату устаревания пароля для пользователя на 31 декабря текущего года. Проверьте, что изменилось в /etc/shadow.

10. Удалите пароль пользователя и проверьте изменения в /etc/shadow.
11. Заблокируйте учетную запись test4.
12. Создайте группу пользователей xusers с GID, равным 1010.
13. Зарегистрируйте себя в качестве участника группы xusers.
14. Как изменить имена и GID групп? Измените имя группы на yusers.
15. Сделайте так, чтобы при запуске оболочки из командной строки выдавалось приветствие.

Получение отчетов об активности пользователей

1. Определите, когда последний раз была загружена система.
2. Кто входил в сеанс за последние 2 недели?

Процессы, сигналы и приоритеты

1. Найдите пустые файлы в домашнем каталоге в фоновом режиме.
2. Запустите в фоновом режиме два задания: sleep 200 и sleep 2000, выведите информацию о состоянии заданий.
3. Снимите с выполнения второе задание, выведите информацию о заданиях.
4. Выполните команду exec ls - R /etc. Изучите её поведение.
5. Запустите порожденную оболочку bash. Исследуйте, посылая родительской оболочке сигналы TERM, INT, QUIT и HUP, что при этом происходит?
6. От имени обычного пользователя пошлите сигнал KILL любому процессу, запущенному от имени другого пользователя. Что произойдет?
7. Запустите в фоновом режиме команду sleep 1000. Проверьте, на какие сигналы из слудующих: TERM, INT, QUIT и HUP, реагирует эта команда.
8. Запрограммируйте оболочку так, чтобы при получении ей сигнала TERM создавался файл pwd.txt, содержащий информацию о текущем каталоге.
9. Запустите порожденную оболочку. Работает ли в ней созданный обработчик?
10. От имени обычного пользователя попытайтесь запустить оболочку bash со значением nice number, равным 1. Какое сообщение выводится?
11. От имени суперпользователя запустите команду индексирования базы данных поиска вследующем виде: time nice -n 19 updatedb. Затем выполните такую же команду, в которой значение nice number для updatedb будет -5. Сравните полученные результаты.

Отложенное и регулярное выполнение заданий (at, cron)

1. Проверьте, запущены ли какие-нибудь задания в cron или at для пользователя root.
2. Сделайте при помощи cron так, чтобы буферы ядра на диске очищались каждый час.
3. Сделайте при помощи cron запуск команды updatedb раз в сутки.
4. Сделайте сигнал окончания пары при помощи команды at – чтобы прозвенел звонок (воспроизвести звуковой файл или использовать ASCII символ beep).
5. Что произойдет, если в файл temp, используя at, одновременно записать разные данные?
6. Выясните приоритет выполнения одновременных заданий для at. Одновременный запуск организуйте разными способами, например, через now+1 minutes и HH:MM
7. С помощью cron сделайте так, чтобы каждые 10 минут убивался браузер firefox.
8. Очистите список заданий для cron и at.

9. Узнайте ip-адрес компьютера вашего соседа, зайдите на него по ssh и сделайте при помощи cron так, чтобы каждую минуту раздавался сигнал. Как от этого защититься?