Лабораторная работа №3. Управление учетными записями. Управление правами доступа к файлам и каталогам.

1. Linux, как и любая unix-подобная система, является не только многозадачной, но и многопользовательской, т.е. эта *операционная система* позволяет одновременно нескольким пользователям работать с ней. Но система должна как-то узнавать, какой или какие из пользователей работают в данный момент. Именно для этих целей в Linux существует два понятия – **учетные записи** и **аутентификация**, которые являются частями одного механизма.

Учетная запись пользователя – это необходимая для системы *информация* о пользователе, хранящаяся в специальных файлах. *Информация* используется Linux для аутентификации пользователя и назначения ему прав доступа.

Аутентификация – системная процедура, позволяющая Linux определить, какой именно *пользователь* осуществляет вход.

Вся *информация* о пользователе обычно хранится в файлах **/etc/passwd** и **/etc/group**.

/etc/passwd – этот *файл* содержит информацию о пользователях. *Запись* для каждого пользователя занимает одну строку:

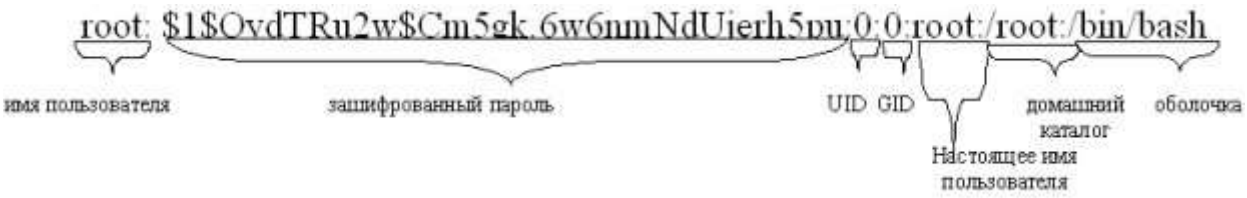

Рис. 3.1.

имя пользователя – имя, используемое пользователем на все приглашения типа login при аутентификации в системе.

зашифрованный пароль – обычно хешированный по необратимому алгоритму *MD5 пароль* пользователя или символ '!', в случаях, когда интерактивный вход пользователя в систему запрещен.

UID – числовой *идентификатор* пользователя. Система использует его для распределения прав файлам и процессам.

GID – числовой *идентификатор группы*. Имена групп расположены в файле /etc/*group*. Система использует его для распределения прав файлам и процессам.

Настоящее имя пользователя – используется в административных целях, а также командами типа finger (получение информации о пользователе через *сеть*).

Домашний каталог – *полный путь* к *домашнему каталогу пользователя*.

Оболочка – *командная оболочка*, которую использует *пользователь* при сеансе. Для нормальной работы она должна быть указана в файле регистрации оболочек **/etc/shells** .

/etc/group – этот *файл* содержит информацию о группах, к которым принадлежат пользователи:

Рис. 3.2.

Имя группы – имя, применяемое для удобства использования таких программ, как newgrp .

Шифрованный пароль – используется при смене группы командой newgrp. *Пароль* для групп может отсутствовать.

GID – числовой *идентификатор группы*. Система использует его для распределения прав файлам и процессам.

Пользователи, включенные в несколько групп – В этом *поле* через запятую отображаются те пользователи, у которых по умолчанию (в файле **/etc/passwd**) назначена другая *группа*.

На сегодняшний день хранение паролей в файлах **passwd** и **group** считается ненадежным. В новых версиях Linux применяются так называемые теневые файлы паролей – **shadow** и **gshadow**. *Права* на них назначены таким образом, что даже чтение этих файлов без прав суперпользователя невозможно. Нужно учесть, что нормальное функционирование системы при использовании теневых файлов подразумевает одновременно и наличие файлов **passwd** и **group**. При использовании теневых паролей в /etc/passwd и /etc/*group* вместо самого пароля устанавливается символ 'x', что и является указанием на хранение пароля в /etc/shadow или /etc/gshadow.

Файл **shadow** хранит защищенную информацию о пользователях, а также обеспечивает *механизмы* устаревания паролей и учетных записей. Вот структура файла **shadow :**

Рис. 3.3.

- а - **имя пользователя** ;
- б - **шифрованный пароль** – применяются алгоритмы хеширования, как правило MD5 или символ '!', в случаях, когда интерактивный вход пользователя в систему запрещен;
- в - число дней с последнего изменения пароля, начиная с 1 января 1970 года;
- г - число дней, перед тем как пароль может быть изменен;
- д - число дней, после которых пароль должен быть изменен;
- е - число дней, за сколько пользователя начнут предупреждать, что пароль устаревает;
- ж - число дней, после устаревания пароля для блокировки учетной записи;
- з -дней, отсчитывая с 1 января 1970 года, когда учетная запись будет заблокирована;
- и - зарезервированное поле;

Файл **gshadow** так же накладывает дополнительную функциональность, вкупе с защищенным хранением *паролей групп*. Он имеет следующую структуру:

Рис. 3.4.

Имя группы – имя, используемое для удобства использования таких программ, как newgrp.

Шифрованный пароль – используется при смене группы командой newgrp . *Пароль* для групп может отсутствовать.

Администратор группы – *пользователь*, имеющий право изменять *пароль* с помощью gpasswd.

Список пользователей – В этом *поле* через запятую отображаются те пользователи, у которых по умолчанию (в файле**/etc/passwd**) назначена другая *группа*.

2. В Linux, кроме обычных пользователей, существует один (и только один) *пользователь* с неограниченными правами. Идентификаторы UID и *GID* такого пользователя всегда 0 . Его имя, как правило, root , однако оно может быть легко изменено (или создано несколько символьных имен с одинаковым *GID* и *UID*), так как *значение* для применения неограниченных прав доступа имеет только *GID* 0 . Для пользователя root *права доступа* к файлам и процессам не проверяются системой. При работе с использованием учетной записи root необходимо быть предельно осторожным, т.к. всегда существует возможность уничтожить систему.

3. В Linux используется развитая система распределения прав пользователям. Но для точного опознания пользователя одного имени недостаточно с точки зрения безопасности. Именно поэтому используется и *пароль – произвольный набор символов произвольной длины, обычно ограниченной лишь используемыми методами шифрования*.

Сегодня в большинстве версий Linux пароли шифруются по алгоритмам *3DES* и *MD5* (устарело, теперь SHA512). Когда *алгоритм 3DES* является обратимым, то есть такой *пароль* можно расшифровать, *MD5* – это необратимое преобразование. Пароли, зашифрованные по алгоритму *3DES* не применяются при использовании теневых файлов для хранения паролей.

При аутентификации, *пароль*, введенный пользователем, шифруется тем же методом, что и исходный, а потом сравниваются уже зашифрованные копии. Если они одинаковые, то *аутентификация* считается успешной.

Учитывая ежедневно увеличивающиеся требования к безопасности, в Linux есть возможность использовать скрытые пароли. Файлы**/etc/passwd** и **/etc/group** доступны для чтения всем пользователям, что является довольно большой брешью в безопасности системы. Именно поэтому в современных версиях Linux предпочтительнее использовать скрытые пароли. Такие пароли располагаются в файлах **/etc/shadow** и **/etc/gshadow**, для паролей пользователей и групп соответственно.

4. *Команда* login запускает *сеанс* интерактивной работы в системе. Она проверяет правильность ввода имени и пароля пользователя, меняет каталог на домашний, выстраивает окружение и запускает командный *интерпретатор. Команду* login как правило не запускают из командной строки — это обычно за пользователя делают менеджеры консоли — например getty или mgetty.

Команда su (*switch user*) позволяет сменить *идентификатор* пользователя уже в процессе сеанса. *Синтаксис* ее прост: su username, где username – *имя пользователя*, которое будет

использоваться. После этого *программа* запросит *пароль*. При правильно введенном пароле, su запустит новый *командный интерпретатор* с правами пользователя, указанного su и присвоит сеансу его идентификаторы. Если *имя пользователя* опущено, то *команда* su использует имя root.

```
[student@ns student]$ su root
Password:
[root@ns student]#_
```

При использовании команды su пользователем root она, как правило, не запрашивает *пароль*.

Команда newgrp аналогична по своим возможностям su с той разницей, что происходит смена группы. *Пользователь* должен быть включен в группу, которая указывается в командной строке newgrp . При использовании команды newgrp пользователем root она никогда не запрашивает *пароль*. *Синтаксис* команды аналогичен синтаксису команды su: newgrp groupname, где groupname – *имя группы*, на которую *пользователь* меняет текущую.

Команда passwd является инструментом для смены пароля в Linux. Для смены своего пароля достаточно набрать в командной строке passwd:

```
[student@ns student]$ passwd
Changing password for student
(current) UNIX password:
New password:
Retype new password:
passwd: all authentication tokens updated successfully
[student@ns student]$_
```

Для смены пароля группы и управления группой используется *команда* gpasswd Для смены пароля достаточно набрать в командной строке gpasswd GROUPNAME . Сменить *пароль* вам удастся только если Вы являетесь администратором группы. Если *пароль* не пустой, то для членов группы вызов newgrp пароля не требует, а не члены группы должны ввести *пароль*. *Администратор* группы может добавлять и удалять пользователей с помощью параметров -a и -d соответственно. Администраторы могут

использовать *параметр* -r для удаления пароля группы. Если *пароль* не задан, то только члены группы с помощью команды newgrp могут войти в группу. Указав *параметр* -R можно запретить *доступ* в группу по паролю с помощью команды newgrp (однако на членов группы это не распространяется). Системный *администратор* (root) может использовать *параметр* -A, чтобы назначить группе администратора.

Команда chage управляет информацией об устаревании пароля и учетной записи. Обычный *пользователь* (не root) может использовать команду только для просмотра своих параметров устаревания пароля:

```
gserg@ADM:/$ chage -l gserg
Last password change                                    : Май 03, 2007
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

Суперпользователь же может использовать также иные параметры, такие как:

-d дата (в формате системной даты, например ДД.ММ.ГГГГ) – устанавливает дату последней смены пароля пользователем.

-E дата – установить дату устаревания учетной записи пользователя

-I N – установить количество дней неактивности N с момента устаревания пароля перед тем как учетная *запись* будет заблокирована

-m N – задает минимальное количество дней (N) между сменами пароля

-M N – задает максимальное количество дней (N) между сменами пароля

-W N – задает количество дней, за которые будет выдаваться предупреждение об устаревании пароля.

1. Для каждого объекта в файловой системе Linux существует набор прав доступа, определяющий взаимодействие пользователя с этим объектом. Такими объектами могут быть файлы, каталоги, а также специальные файлы (например, устройства) — то есть по сути любой *объект* файловой системы. Так как у каждого объекта в Linux имеется владелец, то *права* доступа применяются относительно владельца файла. Они состоят из набора 3 групп по три атрибута:

- чтение(r), запись(w), выполнение(x) для владельца;

- чтение, запись, выполнение для группы владельца;
- чтение, запись, выполнение для всех остальных.

Такие *права* можно представить краткой записью:

rwxrwxrwx – разрешено чтение, *запись* и выполнение для всех

rwxr-xr-x – *запись* разрешена только для владельца файла, а чтение и выполнение для всех.

rw-rw-r-- – *запись* разрешена для владельца файла и группы владельца файла, а чтение – для всех.

Такое *распределение прав* позволяет гибко управлять ресурсами, доступными пользователям.

2. *Права* доступа распространяются и на каталоги. Они означают:

r – если установлено *право на чтение* из каталога, то можно увидеть его содержимое командой ls .

w – если установлено право записи в каталог, то *пользователь* может создавать и удалять файлы из текущего каталога. Причем удалить *файл* из каталога *пользователь* может даже если у него нет прав на *запись* в *файл*. Есть возможность исправить эту ситуацию. Об этом я скажу позже.

x – если установлено право исполнения на каталог, то *пользователь* имеет право перейти в такой каталог командами наподобие cd .

Таким образом появляется возможность создания так называемых "скрытых" каталогов, когда невозможно получить *список* файлов, но *пользователь* точно знающий *имя файла* может скопировать его из "скрытого" каталога.

3. Для распределения прав доступа в Linux существует множество команд. Основные из них – это chmod, chown и chgrp.

Команда chmod (*Change MODe* – сменить режим) – изменяет *права* доступа к файлу. Для использования этой команды также необходимо иметь *права* владельца файла или *права* root . *Синтаксис* команды таков:

chmod mode filename , где

filename – *имя файла*, у которого изменяются *права* доступа;

mode – *права* доступа, устанавливаемые на *файл*. *Права* доступа можно записать в 2 вариантах – символьном и абсолютном.

В символьном виде использование команды chmod будет выглядеть следующим образом:

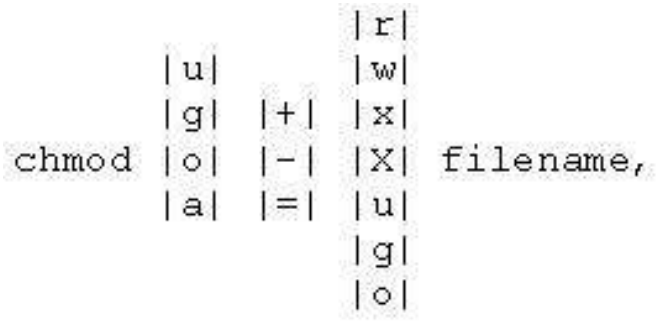

Рис. 3.5

где:

u,g,o,a – установка прав для пользователя, группы, остальных пользователей, всех групп прав доступа соответственно.

+,-,= – добавить, удалить, установить разрешение соответственно.

r,w,x,X,u,g,o – право чтения, записи, выполнения, выполнения если есть такое право еще у какой либо из групп доступа, такие же как у владельца, такие же как у группы, такие же как у остальных пользователей.

filename - *Имя файла*, у которого изменяются *права*.

Просмотр разрешений, установленных на *файл* осуществляется командой ls с ключом -l:

```
[student@ns student]$ ls -l lesson5.txt
-rw-------  1  student student  39 Nov 19 15:17  lesson5.txt
[student@ns student]$ chmod g+rw lesson5.txt
[student@ns student]$ ls -l lesson5.txt
-rw-rw----  1  student student  39 Nov 19 15:18  lesson5.txt
[student@ns student]$ chmod o=u lesson5.txt
[student@ns student]$ ls -l lesson5.txt
-rw-rw-rw-  1  student student  39 Nov 19 15:18  lesson5.txt
[student@ns student]$ chmod o-w lesson5.txt
[student@ns student]$ ls -l lesson5.txt
-rw-rw-r--  1  student student  39 Nov 19 15:19  lesson5.txt
[student@ns student]$_
```

Для использования абсолютного режима необходимо представить *права* доступа к файлу в виде 3-х двоичных групп. Так например:

rwx r-x r-- будет выглядеть как: 111 101 100

Теперь каждую двоичную группу перевести в 8-ричное число: 111 – 7, 101 – 5, 100 – 4 .

Чтобы задать файлу такие *права* необходимо выполнить команду:

[student@ns student]$ ls -l lesson5.txt

-rw-rw-r-- 1 student student 39 Nov 19 15:19 lesson5.txt

[student@ns student]$chmod 754 lesson5.txt

[student@ns student]$ ls -l lesson5.txt

-rwxr-xr-- 1 student student 39 Nov 19 15:19 lesson5.txt

[student@ns student]$_

Задание для обучаемых: попробовать изменить *права* файлу *lesson5.txt* и задать следующие: r w x r- - r - - (744), r - - - w- - - x(421), - - x - w - r - -(124).

Также предложить им проделать то же самое в символьном виде.

Команда chown (*CHange* OWNer – сменить владельца) – позволяет сменить владельца файла. Для использования этой команды необходимо либо иметь *права* владельца текущего файла или *права* root . *Синтаксис* команды прост:

chown username:groupname filename , где

username – *имя пользователя* – нового владельца файла;

groupname – *имя группы* – нового владельца файла;

filename – *имя файла*, у которого сменяется владелец.

Имя группы в синтаксисе команды можно не указывать, тогда будет изменен только *владелец файла*.

Команда chgrp используется для изменения владельца-группы файла. *Синтаксис* ее таков:

chgrp groupname filename,

где:

groupname – *имя группы*, которой будет принадлежать *файл*

filename – имя изменяемого файла

Имейте в виду, что использовать команды chown и chmod может только *пользователь-владелец файла* и root , а команду chgrp - *пользователь-владелец файла*, *группа-владелец файла* и root .

4. Существуют еще несколько особых прав, которые могут устанавливаться на файлы и каталоги. О некоторых из них мы поговорим при изучении темы "процессы". Но один рассмотрим сейчас. Это так называемый *sticky* bit (*бит* прикрепления).

В первых версиях Юникс этот *бит* использовался для того, чтобы заставить систему при работе программы оставлять образ ее кода в памяти. Тогда при следующем обращении к программе на ее *запуск* тратилось намного меньше времени так как чтение кода с

устройства более не требовалось. Для файлов и сегодня в Linux осталось прежнее *значение* этого бита. А вот для каталогов этот*атрибут* приобрел новое *значение*. Если *sticky* bit установлен на каталог, то удалить файлы из такого каталога может только*пользователь-владелец файла*, и то только если у него есть право на *запись* в *файл*. *Группа*-владелец и остальные пользователи даже при наличии прав на *запись* в *файл* не смогут удалить его при установленном на каталог *sticky* bit.

Бит прикрепления устанавливается командой chmod в символьном виде:

chmod +t filename

Задания на лабораторную работу.

В ходе работы необходимо изучить теоретические сведенья, связанные с администрированием пользователей, а также проделать практические задания и ответить на контрольные вопросы, описанные ниже.

1. Ознакомиться с содержимым файлов:

- /etc/passwd,
- /etc/shadow,
- /etc/group.

2. Создать следующие группы:

- workers,
- teachers,
- students.

3. Создать пользователей user_[номер варианта]_ N, где N =1, 2, .., 5, uid учетной записи должен быть равен 1000+N.

Пользователей с N равным 1 и 2 добавить в группу workers вручную внеся изменения в конфигурационный файл.

После добавления пользователей осуществить проверку файла /etc/group на ошибки.

Пользователей с N равным 3, 4 и 5 добавить в группу students при помощи команд администрирования*.

Если у Вас возникли вопросы по поводу использования тех или иных ключей воспользуйтесь командой man для

получения справки: man [имя команды].

Проверьте результат, выполнив действия п.1.

4. Создать пользователя teacher_[номер варианта].

В комментарии к учетной записи должны быть Ваше имя и фамилия.

uid учетной записи должен быть равен 3000. Пользователя добавить в группу teachers.

5. Для всех пользователей задайте пароли, используя команду passwd.

6. Создать директорию labs в корневом каталоге. В нем создать каталоги library и tests

7. Создать файлы book_[фамилия студента]_N и поместить их в library

8. Создать текстовый файл test_[имя студента], и поместить в tests. Файлы должны содержать скрипт на создание пользователя user[номер варианта] и задание ему пароля pass[номер варианта]. Сделайте эти файлы исполняемыми для пользователей группы students.

9. В директории labs создать файл list, который должен содержать список файлов директории /etc.

10. Дать право на изменение файла только пользователю teacher_[номер варианта], а на чтение пользователям группы workers.

11. Настроить права доступа к каталогу library и tests, таким образом, чтобы пользователи группы teachers могли изменять и создавать там файлы, а пользователи группы students имели доступ на чтение

Контрольные вопросы:

1. Почему в конфигурационных файлах пароли не хранятся в явном виде?

2. Почему не рекомендуется выполнять повседневные операции, используя учетную запись root?

3. В чем отличие механизмов получения особых привилегий su и sudo?

4. При выполнении команды ls –la получаем результат:

-rw-r-x-r-- 1 den factory 4464 30 May 2008 text.txt. Что это значит?

5. Как задать права на каталог и все объекты в нем содержащиеся?