***Лабораторная работа №9***

**Администрирование пользователей в операционной системе Linux.**

**Цель работы:** получить навыки администрирования пользователей в операционной системе

**Методический материал**

**Пользователи и группы**

Операционная система Linux является многопользовательской. Контроль доступа к ресурсам в ней основан на идентификации пользователя. Linux позволяет группировать пользователей так, что контроль доступа упрощается и не приходится для внесения каждого изменения затрагивать много пользователей.

В Linux каждый пользователь является членом одной основной группы и, возможно, одной или нескольких дополнительных групп. Кроме того, можно войти в систему под одной учетной записью обычного пользователя и стать суперпользователем, выполнив команду «su» или «sudo –s». Принадлежность файлов в Linux, а также права доступа к ним тесно связаны с идентификаторами пользователей и групп.

Если вы не сменили ваш идентификатор пользователя, то он остается тем же, что и при входе в систему. Если вы стали другим пользователем, идентификатор пользователя может отображаться в приглашении. Если идентификатор пользователя не отображается в приглашении, то текущий идентификатор можно узнать с помощью команды «whoami».

Пример определение идентификатора текущего пользователя:

/home/ian$ whoami

tom

/home/ian$ exit

exit

$ whoami

ian

Аналогичным образом можно узнать, в какие группы вы включены, выполнив команду groups. Команда id выводит информацию как о пользователях, так и о группах. Если к командам groups и id добавить в качестве параметра идентификатор пользователя, то вы получите информацию о пользователе с этим идентификатором, а не о текущем пользователе. В листинге 2 приведено несколько примеров. Обратите внимание на то, что если для команды id не указывать идентификатор пользователя, то помимо основной информации также выводится информация о контексте SELinux.

Определение членства в группах:

[ian@echidna ~]$ id

uid=1000(ian)

gid=1000(ian)

groups=1000(ian),505(development),8093(editor)

context=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023

[ian@echidna ~]$ id ian

uid=1000(ian) gid=1000(ian) groups=1000(ian),8093(editor),505(development)

[ian@echidna ~]$ groups

ian development editor

[ian@echidna ~]$ id tom

uid=1012(tom) gid=1012(tom) groups=1012(tom),505(development)

[ian@echidna ~]$ groups tom

tom : tom development

[ian@echidna ~]$ su tom

Password:

[tom@echidna ian]$ groups

tom development

[tom@echidna ian]$ groups ian

ian : ian editor development

Каждый пользователь имеет свой идентификатор и является членом одной основной группы, а каждый файл в Linux имеет одного владельца и одну группу, связанные с ним. Чтобы определить пользователя и группу, которым принадлежит файл, используйте команду:

ls -l

Определение владельца файла:

[ian@echidna ~]$ ls -l /bin/bash .bashrc helloworld.C

-rw-r--r--. 1 ian ian 124 Mar 31 2010 .bashrc

-rwxr-xr-x. 1 root root 943360 May 21 2010 /bin/bash

-rw-rw-r--. 1 ian development 116 Nov 30 10:21 helloworld.C

В этом примере владельцем файла .bashrc является пользователь ian, и этот файл принадлежит группе ian, которая является основной группой этого пользователя.

Аналогично, владельцем файла «/bin/bash» является пользователь root, и файл принадлежит группе root. Однако владельцем файла helloworld.C является пользователь ian, но этот файл принадлежит группе development. Имена групп и пользователей принадлежат различным пространствам имен, поэтому указанное имя может являться как именем пользователя, так и именем группы. В действительности многие дистрибутивы по умолчанию создают для каждого нового пользователя назначаемую ему группу с таким же названием.

В модели безопасности Linux для каждого объекта файловой системы существует три типа разрешений: чтение (r), запись (w) и выполнение (x). Разрешение на запись также позволяет изменять или удалять объект. Кроме того, эти разрешения указываются отдельно для владельца файла, членов группы файла и для всех остальных.

Обратите внимание на то, что в столбце листинга содержится строка из 11 символов. Первый символ указывает на тип объекта (в этом примере символ "-" означает обычный файл), а остальные девять символов образуют три группы по три символа в каждой. Первая группа содержит разрешения на чтение, запись и выполнение для владельца файла. Символ "-" означает, что соответствующее разрешение не предоставлено. Таким образом, пользователь ian может читать и записывать в файл .bashrc, но не выполнять его, тогда как пользователь root может читать, записывать и выполнять файл /bin/bash. Вторая группа содержит разрешения на чтение, запись и выполнение для группы файла. Члены группы development могут читать и записывать в файл helloworld.C (который принадлежит пользователю ian), а все остальные могут только читать его. Аналогично, члены группы root и все остальные могут читать или выполнять файл /bin/bash.

Одиннадцатый символ указывает на использование альтернативного метода доступа к файлу. Если символ, следующий за битами режима файла, является пробелом, альтернативный метод доступа не используется. Если же этот символ является печатным знаком, то используется альтернативный метод доступа. Таким методом может являться, например, список контроля доступов. Символ '.' (точка) команды GNU ls означает, что для файла используется только контекст безопасности SELinux. Файлы, для которых используются любые другие комбинации альтернативных методов доступа, отмечены знаком '+' (плюс).

Для директорий используются те же самые флаги разрешений, что и для обычных файлов, но интерпретируются они по-другому. Пользователи, имеющие разрешение на чтение директории, могут просматривать ее содержимое. Пользователи, имеющие разрешение на запись в директорию, могут создавать и удалять из нее файлы. Пользователи, имеющие разрешение на выполнение, могут входить в эту директорию и во все ее поддиректории. При отсутствии разрешений на выполнение объекты файловой системы, находящиеся в директории, недоступны. При отсутствии разрешений на чтение объекты файловой системы, находящиеся внутри директории, недоступны для просмотра при выводе содержимого директории, но к ним можно получить доступ, зная полный путь.

Пример разрешений к директории:

[ian@echidna ~]$ ls -l /home

total 32

drwxr-x---. 38 editor editor 12288 Nov 30 10:49 editor

drwxr-x---. 4 greg development 4096 Nov 30 12:44 greg

drwx------. 21 gretchen gretchen 4096 Nov 30 11:26 gretchen

drwxr-xr-x. 41 ian ian 4096 Nov 30 10:51 ian

drwx------. 21 ianadmin ianadmin 4096 May 28 2010 ianadmin

d-wx--x--x. 21 tom tom 4096 Nov 30 11:30 tom

[ian@echidna ~]$ ls -a ~greg/.ba\*

/home/greg/.bash\_history /home/greg/.bash\_profile

/home/greg/.bash\_logout /home/greg/.bashrc

[ian@echidna ~]$ ls -a ~gretchen

ls: cannot open directory /home/gretchen: Permission denied

[ian@echidna ~]$ ls -a ~tom

ls: cannot open directory /home/tom: Permission denied

[ian@echidna ~]$ head -n 3 ~tom/.bashrc

# .bashrc

# Source global definitions

Первый символ в подробном выводе содержимого директории описывает тип объекта (d означает директорию). Пользователям группы development предоставлены разрешения на чтение и выполнение домашней директории пользователя greg, поэтому пользователи tom и ian могут просматривать ее содержимое. Пользователи группы gretchen и все остальные пользователи не имеют разрешений на чтение или выполнение домашней директории gretchen, поэтому пользователь ian не может получить доступ к ней. Для домашней директории пользователя tom предоставлены разрешения на выполнение, но не на чтение, поэтому пользователь ian не может просмотреть ее содержимое, но может получить доступ к объектам внутри этой директории, точно зная о том, что они существуют.

Вывод команды ls -l может содержать объекты файловой системы, отличные от файлов и директорий, что можно определить по первому символу листинга. Типы объектов файловой системы:



**Суперпользователь**

В Linux Суперпользователь (Super User) называется root. Пользователь root может контролировать каждый процесс, имеет доступ к каждому файлу и может выполнить любую функцию в системе. Ничего не может быть спрятано от root'а. Говоря административным языком, root - это высшее существо. Поэтому очень важно, чтобы учетная запись root была защищена секретным паролем. Не стоит использовать root'а для выполнения обычных задач.

Другим пользователям можно дать привилегии суперпользователя, но делать это надо с осторожностью. Обычно предпочтительнее настраивать отдельные программы, чтобы определенные пользователи могли выполнять их как root, вместо того, чтобы предоставлять всем суперпользовательский доступ.

**Создание новых пользователей**

Пользователь добавляется командой useradd. Из консоли это делается, например, так:

useradd -c "normal user" -d /home/userid -g users\-G webadm,helpdesk -s\ /bin/bash userid

Эта команда создает нового пользователя по имени "userid" (последний параметр в команде). Дается комментарий, который говорит, что "userid" -- "normal user" (обычный пользователь). Для него будет создан домашний каталог "/home/userid". Его основной группой будет users, но он также будет входить в группы "webadm" и "helpdesk". В качестве обычного консольного окружения новый пользователь из примера будет использовать оболочку "/bin/bash".

**Изменение паролей**

Из консоли пользовательский пароль меняется командой passwd:

passwd userid

Изменить пароль другого пользователя таким способом может только root. После ввода команды будет предложено ввести и подтвердить устанавливаемый пароль. Если они совпадут, то данные пользователя будут обновлены, а пароль изменен. Пользователь также может изменить свой собственный пароль, написав в командной строке консоли passwd - в этом случае прежде чем вводить новый пароль, необходимо будет ввести старый.

**Удаление пользователей**

Из консоли пользователи удаляются командой userdel:

userdel -r userid

Необязательный ключ -r удалит, кроме пользователя, его домашний каталог совсем содержимым. Если каталог необходимо оставить, не пишите -r. Этот ключ не будет автоматически удалять все файлы в системе, которые принадлежат пользователю, а удалит только его домашний каталог.

**Хранение паролей в Linux**

Конфигурация Linux имеет в своей основе текстовые документы. Все пользователи в Linux описаны в файле под названием «/etc/passwd». Командой «more» вы можете постранично просмотреть этот файл:

more /etc/passwd

Структура этого файла вполне доходчива. Каждая строка содержит нового пользователя с параметрами, разделенными двоеточием:

userid:x:75000:75000::/home/userid:/bin/bash

* В первой колонке находится имя пользователя.
* Во второй - его пароль.
* В третьей - пользовательский числовой id.
* В четвертой - id основной группы пользователя.
* В пятой - полное имя пользователя.
* В шестой - расположение пользовательского каталога. Обычно этот каталог находится в каталоге «/home» и называется по имени пользователя.
* Седьмая колонка содержит путь к оболочке интерпретатора команд (shell), используемого по умолчанию.

В более наглядном виде структура файла паролей представляется следующим образом:



Заметьте, что в приведенном выше примере, в колонке с паролем стоит "x". Это вовсе не означает, что у пользователя такой пароль.

Одна из отличительных особенностей управления пользователями в Linux – наследие стиля UNIX в файле паролей. Пользователь, который регистрируется, должен быть в состоянии прочитать файл «/etc/password», чтобы посмотреть, что его имя существует в системе.

Хранение паролей в том же файле даст возможность потенциальным взломщикам узнать их - они могли бы запустить файл «/etc/passwd» и получить имена и зашифрованные пароли, чтобы дальше работать с отдельной программой прямого подбора. Одно время пароли хранились внутри файла в виде обычного текста. Такая конфигурация возможна и сейчас, но встречается редко из-за последствий. Решено было создать нечто под названием теневой пароль. Файл с теневым паролем не обязательно должен быть читаемым, поэтому взломщики не получат пароли в какой бы то ни было форме для работы с ними.

На месте пароля в файле «/etc/passwd» записывается "х", а закодированная версия пароля идет в файл «/etc/shadow». Такая технология улучшает безопасность посредством разделения информации о пользователе и пароля. Алгоритм кодирования паролей MD5 еще улучшил безопасность, позволяя только надежные пароли. Ниже приведен пример записи теневого пароля:

userid:$1$z2NXZR19$PZpyL84DmPKBXMeURaXXM.:12138:0:186:7:::

Вся функция теневого пароля остается за кадром, а вам редко потребуется делать что-нибудь большее, чем включать ее.

Этот подход, тем не менее, тоже не оптимален, поскольку предоставляет некоторую информацию о пользователе потенциальным взломщикам. Лучший выбор – хранить пользователей в отдельном репозитории, таком как LDAP.

**Группы**

В Linux можно создавать группу и добавляете в ее участников. Ресурсы могут иметь права, приписанные группе. Члены группы имеют доступ к ресурсу, ассоциированному с этой группой. Создание группы выполняется консольной командой groupadd:

groupadd mygroup

Эта команда создаст группу без участников с названием "mygroup". Группы прописаны в файле с названием «/etc/group». Каждой группе отводится отдельная строка:

mygroup:x:527:

* Первая колонка показывает имя группы.
* Вторая -- пароль. Опять-таки "x" означает, что настоящий пароль хранится в теневом файле «/etc/gshadow».
* Третья колонка содержит разделенные запятыми идентификаторы участников группы.

Чтобы добавить члена группы, используется команда gpasswd с ключом -a и id пользователя, которого вы хотите добавить:

gpasswd -a userid mygroup

Удалить пользователей из группы можно той же командой, но с ключом -d вместо -a:

gpasswd -d userid mygroup

Вносить изменения в группы можно также прямым редактированием файла «/etc/group». Однако учтите, что при этом реальную опасность представляет случайное дублирование идентификационного номера. Все ресурсы предпочитают использовать не имя пользователя или группы, а именно id номер. Если вы случайно продублируете его, то дадите доступ туда, куда не собирались. Например, если изменить id номер пользователя на 0, который соответствует пользователю root, то войдя в систему он окажется суперпользователем! Также если удалить в файле строку с пользователем или группой, этот пользователь или группа удалятся.

Это все ошибки, которые может совершить человек. Программа же все делает правильно. Однако, иногда быстрое редактирование файла «/etc/group» - это самый быстрый способ решить простую проблему. Только представьте, что вы обладаете некоторой реальной властью, когда редактируете эти файлы. Будьте осторожны.

Покуда здесь мы не говорим подробно о контроле доступа, вам потребуется некое понимание того, как пользователи и группы соотносятся с файлами. Если посмотреть на расширенный вывод списка файлов в каталоге, можно увидеть что-то вроде следующего:

-rw-r--r-- 1 userid mygroup 703 Jun 23 22:12 myfile

Не обращая пока что внимания на остальные колонки, посмотрите на третью, четвертую и последнюю. В третьей колонке находится имя владельца файла, userid. Четвертая колонка содержит группу, ассоциированную с файлом, mygroup. Последняя колонка - это имя файла. Каждый файл имеет только одного владельца и одну группу. Можно дать права Other (Остальным), пользователям, которые не попадают ни в какую категорию.

Единственный владелец файла - это обычно для операционных систем, но единственная группа-владелец кажется ограничением администраторам, не знакомым с технологией. Это не так. Поскольку пользователи могут быть членами любого количества групп, можно легко создавать новые и новые группы для сохранения безопасности ресурса. В системе Linux определения группы скорее основываются на необходимом доступе к ресурсам, нежели на подразделениях компании. Если ресурсы организованы в системе логично, то создайте побольше групп, чтобы потом настраивать доступ к ресурсам.

**Изменение разрешений**

Предположим, вы создали сценарий командной оболочки "Hello world". Как правило, вновь создаваемые сценарии не являются исполняемыми. Чтобы добавить для сценария разрешение на выполнение, используйте команду chmod с опцией +x, как показано в листинге:

[ian@echidna ~]$ echo 'echo "Hello world!"'>hello.sh

[ian@echidna ~]$ ls -l hello.sh

-rw-rw-r--. 1 ian ian 20 Nov 30 13:05 hello.sh

[ian@echidna ~]$ ./hello.sh

bash: ./hello.sh: Permission denied

[ian@echidna ~]$ chmod +x hello.sh

[ian@echidna ~]$ ./hello.sh

Hello world!

[ian@echidna ~]$ ls -l hello.sh

-rwxrwxr-x. 1 ian ian 20 Nov 30 13:05 hello.sh

Подобным образом можно использовать опцию +r для установки разрешений на чтение и опцию +w – для установки разрешений на запись. Фактически, можно использовать любые комбинации опций r, w и x. Например, команда chmod +rwx устанавливает для файла разрешения на чтение, запись и выполнение. Запуск chmod в такой форме добавляет разрешения, которые еще не были установлены.

**Самостоятельная работа**

1. Ознакомьтесь с методическим материалом. Реализуйте приведенные примеры применительно к своей системе.

2. Создайте пять пользователей из числа студентов вашей группы (включая себя).

3. Создайте две группы: девушки и юноши (называйте на свой вкус)

4. Распределите студентов по соответствующим группам.

5. Создайте две директории.

6. Задайте права доступа к директориям таким образом, чтобы члены каждой созданной группы получили доступ лишь к одной из директорий.

7. Предоставьте себе доступ к обеим директориям.

8. Проанализируйте записи о созданных пользователях и группах в файлах:

/etc/passwd– пользователи,

/etc/group– группы,

/etc/shadow– зашифрованные пароли.

9. При помощи команды userdel удалите одну из учетных записей (не себя) и проверьте результат удаления учетной записи пользователя и его домашнего каталога.