Instructions

Things You'll Need:

- Access to the console
- Solaris operating system

1. 1

   Halt the operating system by pressing "Stop+A." This is only necessary if you already have the operating system running. Use a break command when using an ASCII system. This interrupts the operating system in the same way "Stop+A" would.

2. 2

   Note the root partition. This is important since this is the partition in which you plan on changing the password. By default, Solaris sets the root directory to /dev/dsk/c0t0d0s0 for systems using Ultra5/10 and Blade 100. Blade 1000 uses /dev/dsk/c0t1d0s0.

3. 3

   Type "boot cdrom -s" once you have reached the "OK" prompt. This instructs the computer to load from the CD-ROM drive when you reboot.

4. 4

   Mount the partition using the #mount /dev/dsk/c0t0d0s0 /a" command line. If there is any complication at this stage, it means the root partition is actually located in a different spot than the default. You'll need to find the true root partition using the ls /tmp/dev/dsk command line.

5. 5

   Ensure that your terminal is set to accept commands through a full-screen editor. You won't be able to enter all of the necessary commands to recover the password otherwise.

6. 6

   Use the fsck -y /dev/dsk/c0t0d0s0 command once you receive any error messages stating that partitions failed to unload properly. This checks the hard drive's integrity.

7. 7

Open the editor for the password file by typing /a/etc/shadow. This will open the root password file.

8. 8

   Remove the encrypted password as it appears in the password file.

9. 9

   Type cd/ followed by unmount a/ to prepare for the reboot. Rebooting is required to fully reset the password.

10. 10

    Reboot your system using the boot -s command line. This immediately reboots your system from the hard drive.

11. 11

    Enter a new password for the root partition once your system reloads.

Understanding /etc/shadow file

by nixcraft on February 23, 2006 · 57 comments

Q. Can you explain /etc/shadow file used under Linux or UNIX?

A. **/etc/shadow** file stores actual password in encrypted format for user's account with additional properties related to user password i.e. it stores secure user account information. All fields are separated by a colon (:) symbol. It contains one entry per line for each user listed in /etc/passwd file Generally, shadow file entry looks as follows (click to enlarge image):

/etc/shadow file fields

(Fig.01: /etc/shadow file fields)

1. User name : It is your login name
2. Password: It your encrypted password. The password should be minimum 6-8 characters long including special characters/digits

3. Last password change (lastchanged): Days since Jan 1, 1970 that password was last changed
4. Minimum: The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change his/her password
5. Maximum: The maximum number of days the password is valid (after that user is forced to change his/her password)
6. Warn : The number of days before password is to expire that user is warned that his/her password must be changed
7. Inactive : The number of days after password expires that account is disabled
8. Expire : days since Jan 1, 1970 that account is disabled i.e. an absolute date specifying when the login may no longer be used

The last 6 fields provides password aging and account lockout features (you need to use chage command to setup password aging). According to man page of shadow - the password field must be filled. The encrypted password consists of 13 to 24 characters from the 64 character alphabet a through z, A through Z, 0 through 9, \. and /. Optionally it can start with a "$" character. This means the encrypted password was generated using another (not DES) algorithm. For example if it starts with "1" it means the MD5-based algorithm was used.

vi editor Quick Reference

Introduction
vi pronounced as ” vee eye ” is a unix editor available on almost all the unix operating systems , solaris , bsd ,aix , hpux etc.
This document is a quick reference to vi editor and will be of help if your are new to unix , learning unix or just refreshing your vi knowledge after a few years.

Requirements:
In order to work correctly the vi need correct terminal type (TERM) setting .The TERM setting depends on the type of terminal you have . Commonly used TERM types are vt100 , vt220 and ansi . In most cases vt100 will work fine . In case vi is not able to understand the TERM you have given, it starts in open mode giving you a line by line display .
Generally TERM is taken from .profile or /etc/profile but can be set at the command line as :
$TERM=vt100
$export TERM

echo $TERM will display the current TERM set.

Create new file or Open existing file in vi
vi without any file name will open a new file where you can enter the text and edit but while coming out you will be asked to enter a valid file name to save the text. vi with a file name as

argument will open that file for editing if the file already exists it opens it otherwise it creates a new file by the argument.
Example : $vi testfile
Creates or opens the existing file called testfile

Modes in vi

vi operates in following two modes :
i.) **Command Mode** : After a file is opened it is opened in command mode ,that is , input from the keyboard will be treated as vi commands and you will not see the words you are typing on the screen .

ii.) Insert Mode: To enter the text you have to put vi in insert by pressing 'i' or 'a' after which you can add the text and whatever is being type will be seen on the screen. . To switch between these mode Esc key is used . Esc i
(text mode) Esc (command mode)

Saving & Exiting vi editor
You can exit vi in different ways :

1.) Quit without saving : If you don't want to save the work :q will take you out without saving your editing in vi.
2.) Write & quit : . Simple :w saves the current file but don't exit. For save and quit :wq is used in vi.
3.) Forced Quite : An ! (Exclamation sign at the end of exit commands (:q! , :wq!) causes a forced quit from vi after ignoring editing (for :q!) or writing (for :wq!) all the change

Vi Commands –
Reference

Moving Cursor in File	
Left	h
Right	l
Up	k
Down	j

Line

Beginning	^ or B
end	$

Sentence

Next sentance	)
Previous sentance	(
Paragraph	
Next	}
Previous	{
file	
Go to end of file	:$
on chacter forword	:w
One word forword	:W
go to a line number	:line_number
display file info .	^ g
Inserting and appending text	
inserts text to the left of cursor	i
nserts in the beginning of line	I
appends text to right of cursor	a
appends to the end of line	A
Adding new line	
add a new line below the current line	o
adds a new line above the current line.	O
Deleting the text :	
deletes text above the text	x

deletes text character on right of cursor	X
deletes line 20	20d
deletes current line	dd
delete till end of current line.	D
Replacing a character & word	
replace the character above the cursor.	r
replces characters until Esc is pressed.	R
replaces the word from cursor to the end indicated by $ sign .	cw
replaces till end of line.	C
Substitute	
subistutes current charcater.	s
substitutes entire line.	S
Undo last changes	
undo last change.	u
undo changes to the current line.	U
Copy and pasting lines	
copys the current line into buffer.	yy

copies 5 lines from the current line.	5yy
pastes the current buffer.	p
Searching	
Searches for the word name in the file	:/name
n continues search forward.	n
N searches backwards	N
Saving	
saves the text does not quit.	:w
saves & quit the editor .	:wq!
save	ZZ
Quit without saving	q!
Search & Replace	**s/<search>/<replace>/g .**
Repeating last command	.
Recovering a unsaved vi file.	**vi -r filename**

Learn – Solaris 10 OS

Introducing User Administration

- Introduction

An important system administration task is setting up user accounts for each user who requires system access. Each user needs a unique account name, a user identification (UID) number, a home directory, and a login shell. You also have to determine which groups a user may access.

- Main Components of a User Account

The following is a list of the main components of a user account:

- User name — A unique name that a user enters to log in to a system. The user name is also called the login name.
- Password — A combination of up to 256 letters, numbers, or special characters that a user enters with the login name to gain access to a system.
- UID number — A user account's unique numerical identification within the system.
- Group identification (GID) number — A unique numerical identification of the group to which the user belongs.

Note: You can add a user to predefined groups listed in the `/etc/group` file.

- Comment — Information that identifies the user. A comment generally contains the full name of the user and optional information, such as a phone number or a location.
- User's home directory — A directory into which the user is placed after login. The directory is provided to the user to store and create files.
- User's login shell — The user's work environment is set up by the initialization files that are defined by the user's login shell.

- System Files That Store User Account Information

The Solaris 10 OS stores user account and group entry information in the following system files:

- `/etc/passwd`
- `/etc/shadow`
- `/etc/group`

Authorized system users have login account entries in the `/etc/passwd` file.

The `/etc/shadow` file is a separate file that contains the encrypted passwords. To further control user passwords, you can enforce password aging. This information is also maintained in the `/etc/shadow` file.

The `/etc/group` file defines the default system group entries. You use this file to create new group entries or modify existing group entries on the system.

- System Files That Store User Account Information (continued)

The `/etc/passwd` File

Due to the critical nature of the `/etc/passwd` file, you should refrain from editing this file directly. Instead, you should use the Solaris™ Management Console or command-line tools to maintain the file.

The following is an example of an `/etc/passwd` file that contains the default system account entries.

```
../../../x:0:0:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
gdm:x:50:50:GDM Reserved UID:/:
webservd:x:80:80:WebServer Reserved UID:/:
nobody:x:60001:60001:NFS Anonymous Access User:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
```

Each entry in the `/etc/passwd` file contains seven fields. A colon separates each field. The following is the format for an entry:

`loginID:x:UID:GID:comment:home_directory:login_shell`

The table defines the requirements for each of the seven fields.

Fields in the `/etc/passwd` File

Field	Description
`loginID`	Represents the user's login name. It should be unique to each user. The field should contain a string of no more than eight letters (A-Z, a-z) and numbers (0-9). The first character should be a letter, and at least one character should be lowercase. **Note:** Even though some programs allow a maximum of 32

	characters, as well as user names that contain periods (.), underscores (_), and hyphens (-), this practice is not recommended and might cause problems with other programs.
x	Represents a placeholder for the user's encrypted password, which is kept in the `/etc/shadow` file.
UID	Contains the UID number used by the system to identify the user. UID numbers for users range from 100 to 60000. Values 0 through 99 are reserved for system accounts. UID number 60001 is reserved for the `nobody` account. UID number 60002 is reserved for the `noaccess` account. While duplicate UID numbers are allowed, they should be avoided unless absolutely required by a program. **Note:** The maximum value for a UID is 2147483647. However, the UIDs over 60000 do not have full utility and are incompatible with some Solaris OS features. Avoid using UIDs over 60000 so as to be compatible with earlier versions of the operating system.
GID	Contains the GID number used by the system to identify the user's primary group. GID numbers for users range from 100 to 60000. (Those between 0 and 99 are reserved for system accounts.)
comment	Typically contains the user's full name.
home_directory	Contains the full path name to the user's home directory.
login_shell	Defines the user's login shell. There are six possible login shells in the Solaris OS: the Bourne shell, the Korn shell, the C shell, the Z shell, the BASH shell, and the TC shell.

The table shows the default system account data for entries in the `/etc/passwd` file.

Default System Account Entries

User Name	User ID	Description

root	0	The root account that has access to the entire system. It has almost no restrictions and overrides all other logins, protections, and permissions.
daemon	1	The system daemon account that is associated with routine system tasks.
bin	2	The administrative daemon account that is associated with running system binary files.
sys	3	The administrative daemon account that is associated with system logging or updating files in temporary directories.
adm	4	The administrative daemon account that is associated with system logging.
lp	71	The line printer (lp) daemon account.
uucp	5	The daemon account associated with UNIX®-to-UNIX Copy Protocol (UUCP) functions.
nuucp	6	The account that is used by remote systems to log in to the host and start file transfers using uucp.
smmsp	25	The sendmail message submission daemon account.
listen	37	The network listener daemon account.
gdm	50	Gnome Display Manager daemon.
webservd	80	Account reserved for WebServer access.
nobody	60001	The anonymous user account that is assigned by a Network File System (NFS) server when an unauthorized root user makes a request. The nobody user account is assigned to software processes that do not need any special permissions.
noaccess	60002	The account assigned to a user or a process that needs access to a system through some application instead of through a system login procedure.
nobody4	65534	The anonymous user account that is the SunOS™ 4.*x* software version of the nobody account

Note: The nobody account secures NFS resources. When a user is logged in as root on an NFS client and attempts to access a remote file resource, the UID number changes from 0 to the UID of nobody (60001)

- System Files That Store User Account Information (continued)

The `/etc/shadow` File

Due to the critical nature of the `/etc/shadow` file, you should refrain from editing it directly. Instead, maintain the fields of the file by using the Solaris Management Console or command-line tools. Only the `root` user can read the `/etc/shadow` file.

The following is an example `/etc/shadow` file that contains initial system account entries.

```
../../../rJrdhjNWQQHoY:6445::::::
daemon:NP:6445::::::
bin:NP:6445::::::
sys:NP:6445::::::
adm:NP:6445::::::
lp:NP:6445::::::
uucp:NP:6445::::::
nuucp:NP:6445::::::
smmsp:NP:6445::::::
listen:*LK*:::::::
gdm:*LK*:::::::
webservd:*LK*:::::::
nobody:*LK*:6445::::::
noaccess:*LK*:6445::::::
nobody4:*LK*:6445::::::
```

Each entry in the `/etc/shadow` file contains nine fields. A colon separates each field.

Following is the format of an entry:

`loginID``:`*`password`*`:`*`lastchg`*`:`*`min`*`:`*`max`*`:`*`warn`*`:`*`inactive`*`:`*`expire:`*

The table defines the requirements for each of the eight fields.

Fields in the `/etc/shadow` File

Field	Description
`loginID`	The user's login name.
`password`	A 13-character encrypted password. The string `*LK*` indicates a locked account, and the string `NP` indicates no valid password. Passwords must be constructed to meet the following requirements: Each password must be at least six characters and contain at least two alphabetic characters and at least one numeric or special character. It cannot be the same as the login ID or the reverse of the login ID.
`lastchg`	The number of days between January 1, 1970, and the last

	password modification date.
min	The minimum number of days required between password changes.
max	The maximum number of days the password is valid before the user is prompted to enter a new password at login.
warn	The number of days the user is warned before the password expires.
inactive	The number of inactive days allowed for the user before the user's account is locked.
expire	The date (given as number of days since January 1, 1970) when the user account expires. After the date is exceeded, the user can no longer log in.
flag	To track failed logins. The count is in low order four bits; the remainder is reserved for future use, set to zero.

- System Files That Store User Account Information (continued)

The `/etc/group` File

Each user belongs to a group that is referred to as the user's primary group. The GID number, located in the user's account entry within the `/etc/passwd` file, specifies the user's primary group.

Each user can also belong to up to 15 additional groups, known as secondary groups. In the `/etc/group` file, you can add users to group entries, thus establishing the user's secondary group affiliations.

The following is an example of the default entries in an `/etc/group` file:

```
../../../:0:
other::1:root
bin::2:root,daemon
sys::3:root,bin,adm
adm::4:root,daemon
uucp::5:root
mail::6:root
tty::7:root,adm
lp::8:root,adm
nuucp::9:root
staff::10:
daemon::12:root
sysadmin::14:
smmsp::25:
gdm::50:
```

```
webservd::80:
nobody::60001:
noaccess::60002:
nogroup::65534::
```

Each line entry in the `/etc/group` file contains four fields. A colon character separates each field. The following is the format for an entry:

`groupname:group-password:GID:username-list`

The table defines the requirements for each of the four fields.

Fields in the `/etc/group` File

Field	Description
`groupname`	Contains the name assigned to the group. Group names contain up to a maximum of eight characters. Usually contains an empty field or an asterisk. This is a relic of earlier versions of UNIX.
`group-password`	**Caution:** A group-password is a security hole because it might allow an unauthorized user who is not a member of the group but who knows the group password, to enter the group. **Note:** The `newgrp` command changes a user's primary group association within the shell environment from which it is executed. If this new, active group has a password and the user is not a listed member in that group, the user must enter the password before the `newgrp` command can continue.
`GID`	Contains the group's GID number. It is unique on the local system and should be unique across the organization. Numbers 0 to 99, 60001, 60002 and 65534 are reserved for system group entries. User-defined groups range from 100 to 60000.
`username-list`	Contains a comma-separated list of user names that represent the user's secondary group memberships. By default, each user can belong to a maximum of 15 secondary groups. **Note:** The maximum number of groups is set by the `kernel` parameter called `ngroups_max`. You can set this parameter in the `/etc/system` file to allow for a maximum of 32 groups. Not all applications will be able to reference group memberships

greater than 16. NFS is a notable example.

- System Files That Store User Account Information (continued)

The `/etc/default/passwd` File

Set values for the following parameters in the `/etc/default/passwd` file to control properties for all users' passwords on the system:

- `MAXWEEKS` — Sets the maximum time period (in weeks) that the password is valid.
- `MINWEEKS` — Sets the minimum time period before the password can be changed.
- `PASSLENGTH` — Sets the minimum number of characters for a password. Valid entries are 6, 7, and 8.
- `WARNWEEKS` — Sets the time period prior to a password's expiration to warn the user that the password will expire.

Note: The `WARNWEEKS` value does not exist by default in the `/etc/default/passwd` file, but it can be added.

The password aging parameters `MAXWEEKS`, `MINWEEKS`, and `WARNWEEKS` are default values. If set in the `/etc/shadow` file, the parameters in that file override those in the `/etc/default/passwd` file for individual users.

The Solaris 10 OS release introduces a number of new controls for password management. These controls are configured by setting values in the `/etc/default/passwd` file.

- `NAMECHECK=NO` — Sets the password controls to verify that the user is not using their login name as a component of the password.
- `HISTORY=26` — Forces the `passwd` program to log up to 26 changes to the user's password. This prevents the user from reusing the same password for 26 changes. Setting the `HISTORY` value to zero (0) will case the password log for a user to be removed on the next password change.
- `DICTIONLIST=` — Causes the `passwd` program to perform dictionary word lookups.
- `DICTIONDBDIR=/var/passwd` — Causes the `passwd` program to perform dictionary word lookups.

Complexity of the password can be controlled using the following parameters:

```
#MINDIFF=3
#MINALPHA=2
#MINNONALPHA=1
#MINUPPER=0
#MINLOWER=0
#MAXREPEATS=0
#MINSPECIAL=0
#MINDIGIT=0
#WHITESPACE=YES
```

By default, all of the above parameters are commented out.

Note: By forcing greater complexity of password structure, you may inadvertently cause the users to write down their passwords as they may be too difficult for the user to remember. When setting a password change policy, you must not underestimate the problems that too much complexity may cause.

How to recover/reset root password in Sun solaris

Dated : June 29, 2009

It is recommeded that the security for the physical access to the server is restricted so as to ensure that there is no unauthorized access and anyone who follows this routine is an authorized personnel.

Boot the server with a Sun Solaris Operating System CD (I'm using a Solaris 10 CD but doesn't matter really) or a network boot with a JumpStart server. Change the Boot order accordingly in your x86 system and start the server and launch a single user mode (No Password).

In Solaris 10, you have the default Fail Safe boot option in the Boot Loader. With this you do not need a CD or JumpStart server as selecting it will launch the Single-User shell. I haven't used the early Solaris versions on an x86 system so if anyone knows it is availabe then please post your comments.

When you boot from the CD, select the option for the Sigle-user mode (Option 6 on Solaris 10 CD)

This will look for the currently installed Solaris OS on your system and prompts a Yes/No question to mount the Root filesystem as a Read/Write file system onto /a.

Select yes to mount the root file system (/dev/dsk/c0t0d0s0 here) on /a. If you select No, no harm as you still can mount it manually using:

```
solaris# mount /dev/dsk/c0t0d0s0 /a
```

NOTE: /a is a temporary mount point that is available when you boot from CD or a JumpStart server

Now, with the root file system mounted on /a. All you need to do is to edit the shadow file and remove the encrypted password for root.

```
solaris# vi /a/etc/shadow
```

Now, exit the mounted filesysten, unmount the root filesystem and reboot the system to single-user mode booting of the disk.

```
solaris# cd /

solaris# umount /a

solaris# init s
```

This should boot of the disk and take you to the single-user mode. Press enter at the prompt to enter a password for root.

This should allow you to login to the system. Once in, set the password and change to multi-user mode.

NOTE: Single-User mode is only to ensure that the root user without password is not exposed to others if started in multi-user mode before being set with a new password.

```
solaris# passwd root

solaris# reboot
```

This should do. You should now be able to logon as root with the new password.