Users and Accounts Management in BSD

Ivan "Rambius" Ivanov, rambiusparkisanius@gmail.com

November 5, 2008

1 User databases in BSD

User information is stored in a user database and BSDs maintain several databases for their local accounts:

/etc/passwd ASCII password file with passwords removed

/etc/master.passwd ASCII password file with passwords stored in it

/etc/pwd.db binary password file with passwords removed

/etc/spwd.db binary password file with passwords stored in it

Both `/etc/passwd` and `/etc/master.passwd` are text files with one line for each user account with fields separated with colons. The two binary files `/etc/pwd.db` and `/etc/spwd.db` are hash table-based lookup databases used for fast user lookups as searching though a text file line by line may be slow for systems with a large number of users.

Since `/etc/master.passwd` and `/etc/spwd.db` contain the passwords their permissions are 0600, so they are readable and writable only by the superuser. The other two files `/etc/password` and `/etc/pwd.db` contain no passwords and their permissions are 0644 - they are readable and writable by the superuser and readable by anyone. They are usually used for general user queries.

The file `/etc/master.passwd` is the master user database file - the other three files are generated from it by `pwd_mkdb(8)`. There are rare cases when `pwd_mkdb(8)` is invoked by the superuser directly - usually all the users managing utilities invoke it when they add, delete or modify a user.

Every user account is associated with the following information:

name	the login name of the account
password	its encrypted password
uid	the numeric id assigned to the account
gid	the account's primary group id
class	the account's login class
change	the time when the password should be changed
expiry	the time of the account expiration
gecos	may contains user's full name, office location and phone numbers
home dir	the user's home directory
shell	the user's login shell

Password fields stored only in `/etc/master.passwd` and in `/etc/spwd.db` are encrypted. Several encryption schemes are supported: MD5 hashes, Blowfish cypher and DES cypher.

2 Create user accounts

The utility `vipw(8)` is a general user managamement tool available on all BSDs. It is interactive and opens `/etc/master.passwd` in the default editor after locking it. The superuser can input an entry for a new user, delete an existing user or modify a user. After the editor is closed, the contents is verified and if no errors are found the other user database files are regenerated using `pwd_mkdb(8)`. The file is then unlocked.

If an user account is added using `vipw` the password should be encrypted in advance. One way to do on all BSDs if the passwords are in MD5 format is to use `openssl`:

```
$ openssl passwd -l
```

In addition on OpenBSD the password can be generated using `encrypt`:

```
$ encrypt -p -m
```

for md5 encryption or

```
$ encrypt -p -b 6
```

for blowfish encryption.

When a new account is added with `vipw` its home directory is not created automatically. Other utilities can be used in order to ease this task.

The interactive utilities `adduser` for FreeBSD and OpenBSD prompt for all the information needed to create a new user. There are also non-interactive tools: `pw` for FreeBSD, `user` and `useradd` for NetBSD and OpenBSD:

```
# pw user add testuser -c "Test User" -m -s /bin/sh
```

The option `-c` specifies the gecos field, `-s` - the user shell and `-m` means that the user directory will be created.

```
# user add -c "Test User" -m -s /bin/sh testuser
```

The command `user add` can be used on OpenBSD and NetBSD. It supports also `-p` for providing the encrypted password for the account.

If the user account is created with whatever method without supplying a password, `*` is used as a password and it disables the password authentication and the user cannot log in. A real password can be assigned using the command `passwd` after the account is added.

3 Create and delete groups

Groups can be added and deleted with similar utilities and commands as above: `pw` for FreeBSD, `group` and `groupadd` and `groupdel` for OpenBSD and NetBSD:

```
# pw group add testgroup
# pw group del testgroup

# group add testgroup
# group del testgroup
```

4 Modify user accounts

Some of the information associated with a user account can be changed, like the password, the shell and the groups membership.

To change a user on FreeBSD again `pw` can be used and on OpenBSD and NetBSD - `user` and `usermod`. The following commands add `testuser` to `testgroup1` and `testgroup2`:

```
# pw user mod testuser -G testgroup1,testgroup2
```

```
# user mod -G testgroup1,testgroup2 testuser
```

An important case is if a user is added to the group `wheel` since it gives the ability to `su` to the superuser.

5 Remove user accounts

An account can be deleted by starting `vipw` and deleting the line for the corresponding account. After the editor saves the changes, the other user database files are regenerated. The `vipw` utility does not remove the account's home directory. Again other utilities can be used to automatically do this. The command `rmuser` on FreeBSD and OpenBSD removes interactively a user and `pw`, `user` and `userdel` are non-interactive:

```
# pw user del testuser -r
```

```
# user del -r testuser
```

The option `-r` specifies that the home directory should be deleted.

6 Create a system account

System accounts are those used by other programs and service and are not used for logins. One way to prevent logins for an account is to assign to it a `*` password. This only disables the password authentication, for example using SSH with keys will still login the user. To fully prevent the logins `nologin` shell should be specified as the account's shell and optionally `/nonexistent` as home directory. A system account can be added with `pw` and `user` utilities in the following ways:

```
# pw user add testuser -c "Test System User" \
  -d /nonexistent -s /usr/sbin/nologin

# user add -c "Test System User" \
  -d /nonexistent -s /sbin/nologin testuser
```

7 Lock a user account or reset a locked user account

On FreeBSD if an encrypted password for an account in `/etc/master.passwd` is prefixed with `*LOCKED*` then the account is locked for all types of authentication.

The superuser can lock an account by starting `vipw` and putting the `*LOCKED*` prefix in front of a password. The account can be unlocked by removing the `*LOCKED*` prefix. The command line utility `pw` can also be used:

```
# pw lock testuser
# pw unlock testuser
```

On OpenBSD and NetBSD `user del` utility can be used:

```
# user del -p true testuser
```

The option `-p` with `true` value means that the user login information will the preserve. This options resets its password and sets the login shell to `nologin`.

8 Determine identity and group membership

The utility `id` the user and group names and numeric IDs of the current user; if a user name is supplied to the command, the same information for that user is returned:

```
$ id
uid=1005(testuser) gid=1005(testuser) \
  groups=1005(testuser),1006(testgroup1),1007(testgroup2)
```

or

```
$ id testuser
uid=1005(testuser) gid=1005(testuser) \
  groups=1005(testuser),1006(testgroup1),1007(testgroup2)
```

The command `id` supports the options `-u` that displays only the user id and `-G` that shows the id of the groups the user belongs to:

```
$ id -u testuser
1005
$ id -G testuser
1005 1006 1007
```

The option `-n` used in combination with `-G` or -u shows the names of the groups and the user and not the numeric ids:

```
$ id -u -n testuser
testuser
$ id -G -n testuser
testuser testgroup1 testgroup2
```

The two commands `groups` and `whoami` are the same as `id -Gn` and `id -un` correspondingly.

Another way to determine the identity of the current user is to use the command `who` that displays the users who are currently on the system - it can show their login names, ttys, times of the login and remote hostname if the login is not local. Combined `-m` option, it shows information about the terminal attached to the standard input:

```
$ who -m
testuser         ttyp1    Oct 29 08:58
```

9 Determine who is currently on the system or the last time a user was on the system

The BSD systems maintain a couple of records that store the users activities:

/var/run/utmp records the current users

/var/log/wtmp records the logins and logouts

/var/log/lastlog records the last logins

The command `users` is the simplest way to find out who is currently logged in the system. It reads this information from `/var/run/utmp`

```
$ users
root testuser1 testuser2
```

The command `w` shows not only the current users but also on which terminal they are logged in and what they are doing:

```
$ w
 1:06AM  up 22:21, 4 users, load averages: 0.01, 0.02, 0.00
USER             TTY      FROM              LOGIN@  IDLE WHAT
testuser1        v0       -                 12:58AM     1 less
testuser1        v1       -                 12:58AM     1 -sh (sh)
root             v2       -                  1:04AM     1 -csh (csh)
testuser2        p5       192.168.1.3        1:03AM     2 more
```

The first line prints a general statistics: the current time, the system uptime, the number of users currently logged in and the average load of the machine. The first line can be suppressed with `-h` option. Next, the output shows one line for each user, in this case `testuser1` is logged to the computer locally and is using terminals `ttyv0` and `ttyv1`, `root` is logged in on `ttyv2` and `testuser2` is logged in remotely. If the user is logged remotely, the `FROM` shows the machine they are logged from. The last column shows the process the user is currently running. The option `-d` shows all the processes running on the terminals:

```
$ w -d
```

If one is interested by the processes run by a specific user, an optional username can be provided:

```
$ w root
$ w -d root
```

The command `w` uses `/var/run/utmp` to retrieve its data. The command `who` uses this file as well and it outputs the user name, the terminal and the remote machine the user is logged from if not local:

```
$ who
testuser1        ttyv0    Nov  3 00:58
testuser1        ttyv1    Nov  3 00:58
root             ttyv2    Nov  3 01:04
testuser2        ttyp5    Nov  3 01:27 (192.168.1.3)
```

One of the difference between `w` and `who` is the `who` can use not only `/var/run/utmp` but also `/var/log/wtmp` to extract its statistics from. In this case, it shows records about every login:

```
$ who /var/log/wtmp
```

Similar information can be obtained using the command `last`. It shows the sessions of the specified users, ttys and hosts:

```
$ last
$ last testuser
$ last -t ttyv0
$ last -h 192.168.1.2
```

It can also show the reboots and the shutdowns of the system:

```
$ last reboot
$ last shutdown
```

Finally, just to view last logins one can use `lastlogin` command:

```
$ lastlogin
$ lastlogin testuser1
```

It derives its data from `/var/log/lastlog`.

10 Enable accounting and view system usage statistics

If there is a need for tracking all the processes on the system and the resources and the time they use, then the system accounting can be enabled. If enabled the information is stored in `/var/account/acct`. By default accounting is turned off and that file does not exist. In order to start it, one first creates the file and pass it to the command `accton` and optionally can add `accounting_enable` in `rc.conf` to start it at boot time:

```
# touch /var/account/acct
# accton /var/account/acct
# echo 'accounting_enable="YES"' >> /etc/rc.conf
```

If `accton` is invoked without the file argument, it stops the accounting:

```
# accton
```

After accounting is started it logs every process in `/var/account/acct` which may result in huge amount of log entries and huge usage of disk usage.

Once it is started information from the accounting files can be extracted with the commands `sa` and `lastcomm`.

11 Change a user's default shell

Users can change their own shell using the commands:

```
$ chpass -s /bin/csh
```

or

```
$ chsh -s /bin/csh
```

If `-s` option of `chsh` or `chpass` commands is not provided, the default editor is opened and the user can interactively type in the new shell.

The superuser can change the shell for a given user using the same two commands and supplying the user login name:

```
# chpass -s /bin/csh testuser
```

or

```
# chsh -s /bin/csh testuser
```

In addition the superuser can use `pw` command to change the shell of other users by providing :

```
# pw user mod testuser -s /bin/csh
```

12 Control which files are copied to a new user's home directory during account creation.

When a user account is created its home directory is populated with a number of "dot" files like `.cshrc`, `.shrc`, `.profile`, etc. The original versions of these files are located on FreeBSD in `/usr/share/skel` prefixed with `dot` and on OpenBSD and NetBSD in `/etc/skel`.

13 Change a password

A user can changes its password using `passwd` command:

```
$ passwd
```

The user will be prompted for its old password and then to type and retype the new password.

The superuser, of course, can change the passwords of other users as well by providing their login names:

```
$ passwd testuser
```