1.4 Creating User Accounts

Introduction

Outlined here are the minimal security steps the Bonsa Framework uses in server builds. Given that these account names are on the Internet you may want to change them. However, this may be greatly mitigated with RSA SSH key based authentication.

Naming Convention

You may want to understand the naming convention used here if you want to build your own. Otherwise, the examples are self-explanatory and have not encountered any issues.

Create Catch-All serveradmin user

The purpose of serveradmin is the catch-all place to setup things like scripts. It may also, depending on requirements for your organization be used to manually setup software like application servers.

Our user convention is firstName.lastName, however, we chose to user serveradmin rather than server.admin for fast typing as this and similar accounts are often used via sudo.

> In a more security sensitive environment consider distinct accounts ie for running a manual setup of applications (ie tomcatadmin).

Also, the serveradmin account is limited in that it can not use sudo. If an attacker compromises the application, sudo is still out of reach.

Finally, in order to easily use Zero Footprint, create serveradmin consistently (same GID's and name) across all your systems.

Add the user and assign a password to that user,

```
sudo addgroup --gid 3000 serveradmin
sudo useradd -d /home/serveradmin -m -g serveradmin -u 3000 -c "Admin
catch-all" -s /bin/bash serveradmin
sudo passwd serveradmin
```

Create Staff Users

We will also create **staff** users associated with the built in staff group so we know who is working on the machine. As a policy, our team requires that unless absolutely necessary, staff log in as their own account and then su to serveradmin or use sudo for maintenance work. That way we can have a trail of who does what.

```
sudo useradd -d /home/brian.hitch -m -g staff -u 2000 -c "Support Bryan
Hitch" -s /bin/bash brian.hitch
sudo useradd -d /home/john.cassaday -m -g staff -u 2001 -c "Support John
Cassaday" -s /bin/bash john.cassaday
sudo useradd -d /home/warren.ellis -m -g staff -u 2002 -c "Support Warren
Ellis" -s /bin/bash warren.ellis
```

Options:

-b, --base-dir BASE_DIR base directory for the home directory of the new account
-c, --comment COMMENT GECOS field of the new account
-d, --home-dir HOME_DIR home directory of the new account
-D, --defaults print or change default useradd configuration
-e, --expiredate EXPIRE_DATE expiration date of the new account
-f, --inactive INACTIVE password inactivity period of the new account
-g, --gid GROUP name or ID of the primary group of the new account
-G, --groups GROUPS list of supplementary groups of the new account
-h, --help display this help message and exit
-k, --skel SKEL_DIR use this alternative skeleton directory
-K, --key KEY=VALUE override /etc/login.defs defaults
-l, --no-log-init do not add the user to the lastlog and faillog databases
-m, --create-home create the user's home directory
-M, --no-create-home do not create the user's home directory
-N, --no-user-group do not create a group with the same name as the user
-o, --non-unique allow to create users with duplicate (non-unique) UID
-p, --password PASSWORD encrypted password of the new account
-r, --system create a system account
-R, --root CHROOT_DIR directory to chroot into
-s, --shell SHELL login shell of the new account
-u, --uid UID user ID of the new account
-U, --user-group create a group with the same name as the user
-Z, --selinux-user SEUSER use a specific SEUSER for the SELinux user mapping
--extrausers Use the extra users database

Notice the -u which set's the user's GUIDs. We found it essential to standardize on the GUID of the accounts across all our systems consistently. Not doing so causes problems when it comes to cloning systems or moving programs across different environments. As a practice, we use the following GUID's ranges,

- Staff 2000-2499
- Guest Staff Users 2500-2999
- Custom services 3000 - 3999

Additionally, we use the GUID range 4000-4999 for clients who would send in staff to work on the servers. Since the number of users with this kind of access should not be too large we can make the group blocks match the user blocks,

Group	Users
4000	RedClient1 = 4000 RedClient2 = 4001 RedClient3 = 4002 RedClient5 = 4003
4010	BlueClient1 = 4010 BlueClient2 = 4011
4020	GreenClient1 = 4020 GreenClient2 = 4021 GreenClient3 = 4022

Next, we add to the Staff users the following groups,

- adm - so staff can view logs in apps setup without having to use the sudo command

Here is the command,

```
sudo usermod -a -G adm brian.hitch
sudo usermod -a -G adm john.cassaday
sudo usermod -a -G adm warren.ellis
```

When adding an existing user to an existing group the user must log out and log back in for changes to take effect.

The above step could have been done on user create. However, this illustrates user modification as part of the tutorial.

Do not forget to set a passwords for the new accounts,

```
sudo passwd brian.hitch
Enter new Unix password:
Reenter new Unix password:
passwd: password updated successfully
sudo passwd john.cassaday
sudo passwd warren.ellis
```

Allow staff Group to sudo

Rather then editing the /etc/sudoers using visudo, this approach ensures that system upgrades will not overwrite your changes.

Download File Using tscripts

Download tscripts,

```
sudo su - root
cd /etc/sudoers.d/
sudo wget www.bonsaiframework.com/tscripts/01_enable_sudo_for_staff
sudo chmod o-r /etc/sudoers.d/01_enable_sudo_for_staff
exit
```

Manual Method

If you want to create the file manually,

```
# Locks the file for single user access (important in a multi-user system)
and validates for syntax errors.
sudo visudo -f /etc/sudoers.d/01_bonsai_disable_password_auth
```

visudo launches your default editor to a special file. Add the following to the file,

```
# Members of the staff group may gain root privileges.
%staff ALL=(ALL) ALL
```

Going forward, make sure to use **visudo** to edit the 01_bonsai_disable_password_auth file to ensure proper permissions and locking,

> At this point it is important to log out and log in with your staff account to continue any new work. This will allow for a proper audit trail of the system from this point forward.

Create Auxiliary Users

If you want to make this into a truly enterprise system we will also need a few more users.

remotebackup - User to create remote backups. The assigned UID will be 3001.

```
sudo useradd -d /home/remotebackup -m -g backup -u 3001 -c "Remote Backup"
-s /bin/bash remotebackup
```

Create regular Users

If you would like to add regular users without giving them sudo access then follow below instructions

Create regular group

```
sudo addgroup support
```

Create users and add them to group "support"

```
sudo useradd -d /home/tom.hitch -m -g support -u 2500 -c "Support Tom
Hitch" -s /bin/bash tom.hitch
sudo useradd -d /home/rohan.cassaday -m -g support -u 2501 -c "Support
Rohan Cassaday" -s /bin/bash rohan.cassaday
sudo useradd -d /home/dennis.ellis -m -g support -u 2502 -c "Support Dennis
Ellis" -s /bin/bash dennis.ellis
```

Granting Non-staff User to use sudo with Certain Commands

In some cases you might want a non-staff user (Roderick can we do group too, it would be better) to certain commands. Usual scenarios are to restart services that require root such as Apache (that would be a better example here)

```
sudo visudo
```

Scroll to the bottom and enter the username in this case we use the name bob and enter the commands you would like bob to be able to sudo with, in this case we want bob to be able to create directories

```
bob ALL=(root) /bin/mkdir
```

Now test the command

```
sudo mkdir bob
```

Lets use the find command if you do not know what to add the error message tells you the path that needs to be added to the file as an example lets display the find command error

```
Sorry, user bob is not allowed to execute '/usr/bin/find something' as root
on prodserver
```

Now that we have the command path just add that to bob in the visudo file and test. For multiple commands separate with a comma

```
bob ALL=(root) /bin/mkdir, /usr/bin/find
```

To use sudo without being prompted for a password add NOPASSWD:

```
bob ALL=(root) NOPASSWD: /bin/mkdir, /usr/bin/find
```