The rsnapshot backup solution

Rsnapshot is a filesystem snapshot utility for making backups of local and remote systems. Using rsync and hard links, it is possible to keep multiple, full backups instantly available. The disk space required is just a little more than the space of one full backup, plus incrementals. See the rsnapshot web site for more information.
A good read is the Rsnapshot HOWTO.

Another article worth checking is my companion article about using a Linksys NAS as rsnapshot server.

Getting the software

I have created a Slackware package for rsnapshot that you can download from my SlackBuilds repository.

Installing rsnapshot

These are my notes on setting up “rsnapshot” so that it can start backing up (creating multiple snapshots) remote servers on the LAN or even on the Internet(at least those that have sshd running and ssh as well as rsync installed). In this document, is will call the machine that is running rsnapshot the *backupserver*. The machines that are going to be backed up are called the *target servers* or the *remote servers*.
Basically, using rsync over an ssh connection means that you have to do two things:

1. Make sure the backupserver can login to the target servers using ssh, without requiring a password. We will achieve this by configuring the targets for “*public-key authentication*”.
2. On the target servers, do **not** run as the root account, but use a dedicated non-privileged account (called for instance “rbackup”) that is allowed to do only one thing after logging in using *PubKeyAuthentication*: to run the `rsync` command.

Good example docs about how to set this up are:
http://blog.innerewut.de/articles/2005/05/25/remote-filesystem-snapshots-with-rsnapshot

and it's follow-up article:
http://blog.innerewut.de/articles/2005/06/03/follow-up-on-remote-filesystem-snapshots-with-rsnapshot

as well as some improvements to this:
http://sourceforge.net/mailarchive/forum.php?thread_id=8991705&forum_id=41320

Now, some detailed instructions compiled from the above sources:

Configuring the backup server

- Determine what user account will run rsnapshot. I suggest using “root” since that user can set all the rsync-ed file access and ownership bits exactly like the originals.
- Create a private/public key pair for the rsnapshot account (being “root” in our case) and copy the **public** key over to any and all of the servers that we are going to backup using rsnapshot:

  ```
      local# ssh-keygen -t rsa
      local# scp id_rsa.pub rbackup@remote-server:id_rsa_rsnapshot.pub
      local# ssh remote-server
      ; you might have to create the directory ~/.ssh if it doesnt exist:
      ; "chmod 700 ~/.ssh ; chown rbackup ~/.ssh"
      ; if you don't get the permissions on ~/.ssh and
  ~/.ssh/authorized_keys right,
      ; passwordless login will FAIL!
      remote# cat id_rsa_rsnapshot.pub >> ~/.ssh/authorized_keys
      remote# chmod 600 ~/.ssh/authorized_keys
      remote# chown rbackup ~/.ssh/authorized_keys
  ```

  where *local#* and *remote#* represent the root prompts of your local backupserver and remote (aka target) server.

- Make sure that the `/etc/rsnapshot.conf` file has this argument for the rsync:

  ```
  --rsync-path=rsync_wrapper.sh
  ```

  The documentation mentioned above states that supplying a full path to the `rsync_wrapper.sh` script does not work (on OpenBSD anyway) so the wrapper should be available in the /usr/bin/ or /usr/local/bin directory of the REMOTE server. Would be nice to have the script in rbackups's homedir, so that we can contain the necessary tooling. But using a full path didn't work for me either.

This is a sample last line in our `/etc/rsnapshot.conf` on the backup server:

```
backup<TAB>rbackup@fileserver.my.lan:/home/<TAB>fileserver.my.lan/<TAB>rsync_long_args=--rsync-path=rsync_wrapper.sh --delete --numeric-ids --relative --delete-excluded
```

And this is what should go in the crontab for root on the backup server:

```
0 */4 * * *       /usr/bin/rsnapshot hourly
30 23 * * *       /usr/bin/rsnapshot daily
15 22 * * *       /usr/bin/rsnapshot monthly
```

Configuring the remote server(s)

- Create a user “*rbackup*”, which should be un-privileged, i.e. should not be part of the “*wheel*” group.
- Append the *BACKUPSERVER*'s rbackup user's public key to the *REMOTE* rbackup's `~/.ssh/authorized_keys` and modify it to allow access from one specific machine (the backupserver), only allowed to run one specific command. This is for security reasons of course.

```
from="192.168.200.34",command="/home/rbackup/validate-rsync.sh" ssh-rsa
AAAAB3NzaC1yc2EAAAAB.......0i9yTN7QTrcqKU9ugIesi3+EZnw5ES5wbpo8=
rbackup@TheVault
```

Make sure your version of **`from="192.168.200.34"`** contains the IP address of your rsnapshot server!

- Create the `/home/rbackup/validate-rsync.sh` validation script with these contents:

```
#!/bin/sh
case "$SSH_ORIGINAL_COMMAND" in
  *\&*)
    echo "Rejected 1"
    ;;
  *\;*)
    echo "Rejected 2"
    ;;
    rsync*)
    $SSH_ORIGINAL_COMMAND
    ;;
  *true*)
    echo $SSH_ORIGINAL_COMMAND
    ;;
  *)
    echo "Rejected 3"
    ;;
esac
```

and run the following commands to make it executable for user `rbackup`:

```
chown rbackup /home/rbackup/validate-rsync.sh
chmod 754 /home/rbackup/validate-rsync.sh
```

- Create a wrapper script for rsync (which uses sudo) in `/usr/local/bin/rsync_wrapper.sh` which contains:

```
#!/bin/sh
/usr/bin/sudo /usr/bin/rsync "$@";
```

- Add this line to `/etc/sudoers` so that user `rbackup` can run the wrapper's `rsync` command with *root* privileges:

```
rbackup ALL = NOPASSWD: /usr/bin/rsync
```

Testing remote login

When all configuration is complete, and before your scheduled cron jobs start, you should test whether the non-interactive passwordless login from the rsnapshot server to the remote *rbackup* account is functional.

You need to make a ssh connection at least once, to add the public key of the remote machine to your root account's *known_hosts* file.

```
# ssh rbackup@client.my.lan
Rejected 3
Connection to client.my.lan closed.
```

The “`Rejected 3`” message actually means that your configuration was successful!

From:
https://wiki.alienbase.nl/ - **Alien's Wiki**

Permanent link:
https://wiki.alienbase.nl/doku.php?id=linux:rsnapshot

Last update: **2010/04/24 19:25**