XigmaNAS - Secure Server to Server RSYNC Backup

Introduction

XigmaNAS is great for server to server backups or mirroring because pretty much everything you need is already installed at both ends. It is very easy only if you are not concerned about the security of your data. You could just set one end up as an RSYNC client and the other as a server and then schedule updates, all in the WebGUI. This is insecure and I would only do it on a private, secure network, not over the Internet.

This article provides an overview of how to use RSYNC to securely send data from one XigmaNAS Server to another securely, especially for use over public networks. To accomplish this you run rsync over SSH and SSH is configured so that no password is required to begin a session. This can be used on a private network or over the Internet.

This article does not cover networking basics you will need to know and configure so your servers can communicate over the Internet.

While the whole process may seem daunting or difficult at first, it will actually wind up being pretty simple once you read and understand the linked documentation and see a few examples.

Prerequisites

Before deploying secure server to server backup you should be familiar with:

- Linux file and directory permissions.
- Mounting and detaching (unmounting) filesystems.
- Changing File Permissions and File Ownership.
- Executing commands in a terminal, CLI or via SSH session.
- Basic *nix commands and syntax.

Open a Terminal (Console, CLI) on your client PC and make sure you understand these commands:

- man mount
- man umount
- man chown
- man chmod

You can also look them up on FreeBSD.org's Man Pages.

Your servers should be capable of insecurely communicating with each other over the Internet. Use PING and TRACEROUTE to verify that each Server can reach the other. You can use WebGUI Tab> **Diagnostics|Ping** and **Diagnostics|Traceroute** for this purpose.

Install XigmaNAS, if you already haven't

See SUG Section 2.3-Installing XigmaNAS on disk.

Or you may run it from the CDROM as required for testing.

SUG Section 2.2-Using XigmaNAS with the CDROM and a removable disk (LiveCD mode).

Configure Password-less / Key Authentication

This is necessary for security and encryption of SSH. Please see SUG Section 2.6.4-Password-less & Key Authentication. Once you have manually tested that a secure connection can be established you can begin composing your rsync command.

Note - This article is only concerned with backing up the data on one server to the other. It is not necessary to configure key authentication on both servers, you only need to create the keys/login on the server that will be receiving the data. We are only concerned with sending data in one direction. Of course, nothing stops you from doing it on both servers.

Note - If you are using Embedded or LiveCD, (as we did for this article) you should store the .ssh folder on a share, then use a PostInit script to copy it to the proper location, or you will need to assign the home directory of the user you will authenticate as, to a permanent location. This is because both those installs run in RAM and the .ssh directory (where the authorized_keys file is stored) will be gone after reboot unless you take steps to ensure it is saved. Details about this are found in SUG Section 2.6.4-Password-less & Key Authentication

Compose RSYNC Command

You should now compose a one line command that will copy the data from one server to the other. This is essentially an rsync command that is made to run over SSH. The rsync documentation is the best source for learning how to put together such a command. Please use the following links to learn about this:

- The RSYNC Home Page - Your portal to the best documentation.
- RSYNC Man Page - Where you will find information about the rsync portion of your command. Long, complicated and hard to read, but very rewarding and you don't really have to read much to get going.
- RSYNC Section of Knowledgebase - Nice work by Danmero, he gets right to the important points.
- OpenSSH Client Documentation - Where you will find information about the SSH portion of your command.

- RSYNC Section of Knowledgebase - Nice work by Danmero, he gets right to the important points.

What you will wind up with is a command that looks very much like this:

```
rsync -v -h -a --delete --stats --log-file="some_path/your_log_filename.txt"
-e "ssh -v -l your_username -p your_port"
"/path_of_stuff_you_want_backed_up"
192.168.1.2:"/mnt/path_where_you_want_data_stored/"
```

In the above command you invoke RSYNC and tell it to use SSH (which you have previously set up to use password-less authentication) to connect with the other machine and update/copy your data.

Tip - Watch out for quotes. Special characters or spaces in your path and file names may require they be enclosed in quotes. A missing quote can ruin your day.

You will test this command at the console or CLI or via SSH session to make sure it works exactly as you desire. It will take some work and trial and error, but you will soon figure out what works best for you. Once you verify the command is working properly and your data is being copied manually, you can then automate the process with CRON.

Create CRON Job to Backup Automatically

Configuring CRON is detailed in SUG Section 3.2.5-System Advanced Cron, please read it.