Encrypted partitions/folders with auto-mount

Encrypted Shared Folders - Synology DSM

This guide covers a concept to allow a Synology NAS to automatically mount encrypted shared folders on boot or reboot after a power failure. The keystore should be made unavailable, which prevents the NAS from mounting the encrypted shared folder(s). This guide assumes you are familiar with the encryption and keystore concepts of a Synology NAS running DSM 6.2 or above. In my setup, all shared folders which contain sensitive data are protected through a volume key. The key is served from a key server on a different location, which serves the key on condition that the correct machine key (stored on the machine to protect) and the correct IP address, from which the request originates, is provided.

1. A Synology NAS with encrypted shared folder(s) (“data server”)
2. A linux machine which can be accessed through HTTPS to obtain the volume strings (encrypted volume keys). This machine can be a Synology NAS, or any linux server, for example a VPS. (“key server”)
3. Upload a php file to the key server which provides the encrypted volume strings
4. Upload a bash file to the data server which gets the encrypted volume strings from the key server, decrypts the volume strings to volume keys by means of the machine key, and mounts the encrypted shared folders
5. Add a php script activating and deactivating “synology.php” to deactivate the key server in case of a compromised NAS
6. Clean up the Synology keystore at /usr/syno/etc/.encrypt/

Encrypted Partitions - Debian 9

This guide covers a concept to allow a Debian machine to automatically mount encrypted volumes on boot or reboot after a power failure. This guide assumes you are familiar with LUKS encryption and concepts of Debian. In my setup, I put the entire /home directory on a 4TB RAID1 partition, which is protected through a volume key. The key is served from a key server on a different location, which serves the key on condition that the correct machine key (stored on the machine to protect) and the correct IP address, from which the request originates, is provided.

1. A Debian machine with encrypted volume(s) (“data server”)
2. A linux machine which can be accessed through HTTPS to obtain the volume strings (encrypted volume keys). This machine can be a Synology NAS, or any linux server, for example a VPS. (“key server”)
3. Upload a php file to the key server which provides the encrypted volume strings
4. Upload a bash file to the data server which gets the encrypted volume strings from the key server, decrypts the volume strings to volume keys by means of the machine key, and mounts the encrypted partitions
5. Add a php script activating and deactivating “synology.php” to deactivate the key server in case of a compromised NAS

Key setup

Terminology

- machine key: 24 character password (key) with which the volume key is encrypted
- volume key: 24 character password (key) to unlock encrypted volumes or shared folders on that particular machine
- volume string: encrypted volume key which is stored on the key server

The key server sends the volume string upon demand from a trusted IP address. The volume string can be decripted with the machine key to obtain the volume key. To obtain a valid volume key, all 3 conditions must be met: valid requester IP address, matching volume string, and matching machine key.

Synology NAS

1. Generate a machine key with the Secure Password Generator.
2. Upload cryptmount.sh
3. On the data server, goto **Control Panel → Shared Folder** and Create a new encrypted shared folder. Click **Create → Create**:
   - Name: <folder name>
   - Check “Hide sub-folders and files from users without permissions”
   - Uncheck “Enable Recycle Bin”
4. Click “Next”, then:
   - Check “Encrypt this shared folder”, then enter a strong volume key. Generate the key with the Secure Password Generator, select password length 24, and unselect “Include Symbols”.
   - Uncheck “Add encryption key to Key Manager”
5. Click “Next”, then assign options and permissions.
6. Add task to **Control Panel → Task Scheduler → Create → Triggered Task** and run script

```
bash /volume1/homes/bco/batch/cryptmount.sh
https://your.server.tld/synology.php encrypted <machine key>
pingable.server.tld 60 10
```

7. Reboot the machine for the final test.

Note

1. On a DS213+ wget is compiled without https support. This requires to connect to the Key Server through SSH and request the encryption key on http.
2. To facilitate this I created a folder outside of the server root, which is accessible only by dedicated ip addresses. Add localhost and/or 127.0.0.1 to that list.

Data Server

1. Generate a machine key with the Secure Password Generator.

2. Generate an encrypted partition with a strong volume key. Generate the key with the Secure Password Generator, select password length 24, and
3. Open an elevated command line prompt and generate the volume string with the following command. You must do this on the machine you intend to unlock later.

```
 echo -n "<encryption key>" | openssl enc -aes-256-cbc -a -salt -pass
pass:<machine key>
```

4. After having modified file “synology.php” on the key server (see below), run “cryptmount.sh” on the data server to check correct encryption and decryption of the keys.
5. Comment line after “# testing” and uncomment line after “# production”.
6. Setup rc.local following Debian 9 Setup and add a batch command which runs one or several cryptmount.sh on boot as root
7. Reboot the machine for the final test.

Key Server

1. Upload synology.php to the HTTPS key server document root. Modify IP addresses and encryption strings in password array.
2. Add the output of the openssl command (the encryption string) to the password array of file “synology.php” on the key server, using the name of the share as array index.

Auto-logon

Make sure to allow sufficient time on boot of the machine to mount the encrypted volume and auto-logon a user, if the user home directory resides on the encrypted volume. 30 seconds should be sufficient, depending on the maximum time your machine needs to get access to the network, but might be longer. The following steps must be completed in sequence during machine boot:

1. Network must be up, and key server must be reachable. You can check this with ping, but for a Synology NAS the executing user must have sudo rights without password for ping.
2. Mount encrypted volume as root
3. logon <user>. In XFCE, modify autologin-user and autologin-user-timeout in /etc/lightdm/lightdm.conf.

Links

Synology

- Synology encrypted shares auto mounter
- Is this a (relatively) safe way to auto-mount encrypted folders?
- How can I mount encrypted folder via SSH in Synology NAS?

Debian

- Mount encrypted volumes from command line?

- auto login on xfce in jessie

From:
https://wiki.condrau.com/ - **Bernard's Wiki**

Permanent link:
https://wiki.condrau.com/deb9:encrypted

Last update: **2020/04/28 04:21**