ZFS encryption

Features
Content:

1.  Napp–it encryption features
   1.1  Setup encryption on a Client
   1.2  Setup a webbased keyserver
   1.3  Create encrypted filesystems
   1.4  Modify properties

2.  ZFS encryption
   2.1  ZFS Locking/ unlocking
   2.2  Keysource
   2.3  User locking/ unlocking via SMB
   2.4  Setup user locking
   2.5  Auto locking based on timetables
   2.6  Timetables
   2.7  Setup Timetables

3.0  more manuals

Timetable functions partly under development
1. Napp–it encryption features

Data security with protection against theft, ransomware, sabotage and accidental data lost is mandatory. Data privacy rules like the EU General Data Protection Regulation (DSGVO) demand state of the art storage features in any organisation size, best and most attractive offered with ZFS, snaps and individually encrypted storage and backup.

Basic napp–it Features

- ZFS Encryption with the option of a different key per filesystem
- ZFS encrypted backups
- Encrypted/ raw replication of a locked filesystem (Illumos, OmniOS/OI)
- Keysource prompt, local files or files on a webbased keyserver even with a file/webbased method you can always unlock via prompt.
- Keylocations are easily switchable (just copy keyfolders)
- webbased method does not store a key locally
- SMB shares are removed automatically on a lock
- SMB shares can be restored on an unlock

Napp–it Pro Features

- Automount of encrypted filesystems (local or webbased keys needed) during server reboot
- User–Lock/Unlock of a filesystem via SMB and optionally a special smbkey
- Auto–Lock/Unlock based on a timetable (under construction)
- Keysplit where the two parts of a key can be stored on different locations either (L)ocally or on a (W)ebserver (L1:L1,L1:L2,L1;W1,W1;W1,W1;W2)
- Keyserver (http or https, Pro complete) centralised keymanagement without local keystorage
- Keysplit to store key parts on two independent locations.
- HA keyserver to allow a key request from a second redundant webservice W1, W2, W1’ and W2’
- Restrict access based on timetable or IP
- Easy keymanagement
  just copy keyfolders between the different locations to copy/backup keys
1.1 ZFS Encryption setup (clientside)

1.1.1 General settings
Setup defaults in menu About > Settings

<table>
<thead>
<tr>
<th>Base encryption settings</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>First local keydata folder or filesystem 1 ex pool-1/keydata (L1) for keysplit and keypart-2 on a second filesystem.</td>
<td>av/keydata</td>
</tr>
<tr>
<td>Access id for all web keyrequests from this server use value from keyserver host settings</td>
<td>W1SPxICM42J</td>
</tr>
<tr>
<td>Allow web/filebased keys save or split the keys either L=locally or on W=webserver</td>
<td>all ▼</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Extended encryption settings (Pro Complete)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Secondary local keydata folder or filesystem 2 ex pool-2/keydata (L2) for keysplit and keypart-2 on a second filesystem.</td>
<td>av/keydata</td>
</tr>
<tr>
<td>Keyserver settings</td>
<td>av/keydata</td>
</tr>
<tr>
<td>Keyserver data/keyex pool-1/keydata can be same or different to local keydata folder or filesystem</td>
<td></td>
</tr>
<tr>
<td>Keyservers access allowed from ip starting with</td>
<td>ex 172.</td>
</tr>
<tr>
<td>enable Keyservers access from remote clients Allow web-based http/https keyrequests for encrypted filesystems Keysplit and -ms) keyservers require a Pro complete license</td>
<td>yes ▼</td>
</tr>
</tbody>
</table>

Local keydata folder.
Below this folder, local keys are stored.

Webserver url
When you create a new encrypted filesystem, keys are stored here when you select W1

Access-ID
On a keyserver you must create a folder for any client. Copy/paste the hostid for this server from the keyservers

This folder should be encrypted (manually unlocked) or on a removable pool ex USB or on an iSCSI target

ex https://172.16.1.12:82

ex Aoi7c5TR12hZ
1.2 ZFS Encryption server setup (Keyserver, napp-it Pro Complete)

1.2.1 General settings
Setup defaults in menu About > Settings

- **Keyserver data**
  - When this server is a keyserver, place webbased keys below this folder
  - local1, local2 and keyserver folder can be the same as napp-it create a folder local or keyserver below.

- **Keyserver access allowed**
  - A client has access when the ip starts with this number
  - ex. 172.16

- **Keyserver enabled**
  - Enable/Disable web key requests globally
  - yes/no

1.2.2 Client settings
Create clients in menu Services > Key Server > Clients

Create an entry for each client for whom you want to store keys. The keys are stored below the keyserver filesystem. If you want a second keyserver W2 for the second keyoart, copy over the keyfolder to the second keyserver. In a keyfolder ex keydata/keyserver/clients/d4xclxitQoyo you will find a folder for keypart-1 and keypart-2 what makes it easy to copy/move/backup keys or keyparts.

Folder structure under keyserverfs:

```
/keyserver
/keyserver/clients
/keyserver/clients/CXmJ8IHGcHst
/keyserver/clients/CXmJ8IHGcHst/defaults.cfg
/keyserver/clients/CXmJ8IHGcHst/keys-part1
/keyserver/clients/CXmJ8IHGcHst/keys-part2/
/keyserver/logs
/keyserver/timetables
```

You can share this via SMB (root only) for a client with this id client settings filesystem keys/ keypart-1 filesystem keys keypart2

Folder structure for local keys below keyfs

```
/local/keys-part1
/local/keys-part2
```

filesystem keys/ keypart-1 filesystem keys/ keypart-2
1.3 ZFS Create encrypted filesystems

1.3.1 ZFS Filesystem > Create (requires Oracle Solaris or newest OmniOS/OI with newest pool version)

If you select a web-based location, the key is uploaded during creation (requires you have created a client entry on the keyserver(s) and copied the client id over to About > Settings)

If you want to use two webserver, you must create a client folder on the first webserver and copy it over to the second webserver via sftp or a SMB share.

In the passphrase line you can click show/hide to view the passphrase. You can copy backup now (or backup the files later). If you loose keys you will never be able to access data.

1.4 Details of encrypted filesystems

1.4.1 Open menu ZFS Filesystems > Encryption
Settings:
Infos about your encryption related settings. (read only)

Filesystems:
This is an overview of encrypted filesystems and their setting. You can lock/unlock and modify settings like Automount (during server boot), Auto_lock (timer based lock/unlock) and the according Timetable and SMB Userlock.

You can check availability of local or webbased keylocations like the two webserver W1 and W2 and their HA/ failover alternatives W1’ and W2’.

autolock service: running state

This service is needed for SMB user lock/unlock and timer based lock/unlock. You can start/stop this service in menu Services.

1.4.2 Unlock a filesystem

When you clock on „locked“ in the row of a filesystem you can unlock the filesystem

You can accept the default keylocation or you can enter a different like L1:L1 or L1:L2. You can also copy in the real key to unlock directly. As an option you can restore the SMB share or unlock without a share enabled.

In any case, the key is not stored locally as the unlock functionality is based on prompt where the key is entered directly in the background via an „Expect“ functionality.

If you want to reorganize your keylocations ex from L1 to L1 or W1 to L1, just copy over the keys to the new location and modify the keylocation in menu ZFS Filesystems > Encryption.
2. More on ZFS Encryption

ZFS encryption as a ZFS filesystem property is in Oracle Solaris 11.x and with Open-ZFS in OmniOS 151032+ and OpenIndiana 2019.10. To enable encryption update OS and Pool version.

Encryption as a ZFS property has huge advantages over device or disk based encryption that are available on the different Open-ZFS plattforms (BSD, Illumos or Linux) in a non-compatible way as they work below ZFS on OS level and not for ZFS filesystems but for the whole pool with a single key for all filesystems.

ZFS embedded encryption allows a key per filesystem for different levels of security and access time. For backups you can access unlocked data or you can backup encrypted data with encrypted ZFS replication. You can lock access to very sensitive data strictly to working hours with extended access to other areas while they remain protected in case of a theft.

Especially the last aspect becomes more and more important in the light of new rules regarding sensitive personal data like the EU DGSVO that will be a common EU law from may 2018. It demands among others a state of the art data security at technical level. You must define not only the exact amount of data, who has access, why is this access granted and how long even for backup is data kept but you must also ensure the security of the data itself and minimising the risk. In every case this can only guaranteed with encryption to avoid a hacking outside working hours and to secure data in case of a hardware theft, and with a backup mechanism that includes encryption. The EU fines if you have not filled the regulatory demands are hefty with up to 4% of the worldwide revenue in severe cases.

Currently Oracle Solaris and newest Open-ZFS has ZFS-Encryption at filesystem level with replication to encrypted destinations. Illumos with OmniOS/O adds a raw replication mode where transfer and destination is encrypted with the source key or a replication of a single unencrypted filesystem to an encrypted target (already created or created during zfs receive).

The encryption property must be set when you create a new filesystem. You can define a security level like aes-256-ccm, a keyformat like raw, hex or passphrase and a keysource like prompt, file, https or a keyserver.

see https://docs.oracle.com/cd/E53394_01/html/E54801/gkkih.html#scrolltoc

2.1 ZFS locking/ unlocking

As a basic napp-it free feature, you can lock/ unlock a filesystem in menu ZFS Filesystems. Handling is similar to other ZFS properties like sharing or compress (simply click on the property you want to modify). This requires full access to Storage Management. After reboot all encrypted filesystems remain locked until they are unlocked manually with a key that can be different for every filesystem.
2.2 Keysource

ZFS encryption requires a key per filesystem. On a smaller setup you can use prompt as keysource. This means that you must enter the key manually to unlock a filesystem. For locking you do not need a key.

With many filesystems or when you do not want to give a key to storage operators, you can use file, https (web) or a keyserver as source. The keysource is a ZFS property. Once defined and accessible you can lock/ unlock a filesystem without knowing the key as long as the keysource is available.

For a private/soho setup you may use an USB stick where you place the filebased keys to allow a simple locking/ unlocking. If you remove the stick nobody can unlock the filesystem. Handling is simple. On the evening you lock your filesystems and in the morning you plug into your stick and unlock the filesystems (keep a copy of the keys elsewhere as backup). If you forget to unplug the Stick on a theft or someone has copied the stick, this method is unsecure.

More secure is a remote keyserver with restricted access with splitted keys. You can also use one or two local filesystems for one or bot keyparts. You should use local keys only in a Soho environment ex with keys on an USB stick or a remote iSCSI Lun.

2.3 User locking/unlocking

If you use a ZFS filer for different user groups with different security needs or access times, you may want to allow users ex a head of a department to lock/ unlock their data themselves without help of a storage admin and without access to the storage management software.

This requires an auth based mechanism that must be integrated into the storage system. While one can think of a lot of sophisticated methods, I follow the Keep it Simple approach. There is already a secure method to restrict access to server resources based on local or centralized AD users and this is SMB where you can restrict access to shares, folders and files or track access via file auditing (Solaris 11.4, see https://docs.oracle.com/cd/E37838_01/html/E61027/osmaa-whatsnew.html#OSMAAosmaa-whatsnew).

SMB does not offer a method to control service states. To allow locking/ unlocking of filesystems, napp-it can create a SMB share with a watched folder per encrypted filesystem. Only authorized users have access to these folders. In these folders a user will find a switching file and a file with the service and locking state as filename.

A simple move of the switching file (=switcher.move) into a subfolder „lock“ or „unlock“ will initiate a locking or unlocking (requires that Solaris has access to the keys with a file, web or keyserver based method). You can extend this with an SMB key. Copy the switching file then to your desktop, insert the SMB key und use this file to unlock.

Napp–it watched SMB folders

SMB share UserAuth with a subfolder per encrypted filesystem

Watched folders =lock and =unlock with a switcher file that locks/ unlocks a filesystem by moving the file into a subfolder and a status file.
2.4 Setup user locking/unlocking

To enable userlocking via watched folders you must

- create an encrypted filesystem with a (L) or (W) as keysource
- set userlock to on or smbkey in menu ZFS Filesystems > Encryption
- enable autolock service (Menu services, this is a Pro Feature)
- enable smb share of of the UserEncrypt filesystem, set permissions

- now set SMB Userlock to yes

In a next step, you must start the autolock service in menu Services > autolock.
This will check for and optionally create a „UserEncrypt“ ZFS filesystem with a watched subfolders per encrypted filesystem.

At this point you can enable/ disable the whole userlock functionality be enabling or disabling the SMB share UserEncrypt and setup ACL permissions for this share or more specific for the subfolders.
Tipp: You can set permissions for any user when you SMB connect as root.

A user with permissions can now lock/unlock a filesystem by simply moving the switcher file into the subfolders /=lock or /=unlock and check service and lockstate with a service control file that is updated automatically.

- or set SMB Userlock to yes

With smbkey as method, you must copy the control file =switcher.move to your desktop. Open the file with an editor and insert the smbkey (from menu ZFS Filesystems > Encryption). To unlock a filesystem with userlock=smbkey, you must copy this file into the watched folder =unlock

watched folder for filesystem tank/eins
2.5 Enable auto locking/unlocking (under development)

If user-locking is working you can add auto-locking. This means that a filesystem can be automatically unlocked and/or locked example always in the evening or automatically unlocked at working hours. This will limit the risk of a data loss due to a hack.

To enable Autolock, select a timetable in menu ZFS filesystems and set Autolock to yes.

Test a timetable

Use menu ZFS filesystems > Encryption > Check to parse a timetable for a given date. The timetable for this check is a file in /var/web-gui/_log/autolock/timetables/my_unlock_working_hours.cfg

```plaintext
# defaults
unlock_hour=7:00
lock_hour=20:00
# can be overwritten by rules, interval 15min

# overwrite lock/unlock per holidays.cfg (day of week) with exit
alias=holidays_quick.cfg

# overwrite lock/unlock per workday (day of week)
alias=workdays.cfg
```
2.6 Timetables (under development)

You can print a timetable to control the autolock function (unlock time and lock time) for next weeks, months or year in menu „ZFS Filesystems > Encryption > Timetable“ for a selected timetable.

---

2.7 Setup Timetables (under development)

Timetables are textfiles in /var/web-gui/_log/autolock/timetables/. If you want to setup your own timetable, create a new timetable file and start the name with my. Do not modify default timetables as they are overwritten on updates. In a timetable you can set unlock_hour and lock_hour as defaults and add aliases where the time is set.

```
# Timetables

Timetables are used to define lock_hour and/or unlock_hour time for current date.
Day/ Date settings always based on aliases.

Timetable work like a firewall that is processing lock or unlock from top to bottom:
The first matching value does the job (cancel or trigger).
```

```
# Timetable example

# working_time.cfg with entries like

# defaults
unlock_hour=7:00
lock_hour=20:00    # can be overwritten by rules, interval 15min

# priority setting with exit
alias=my_priority_quick.cfg

# overwrite lock/unlock per holidays.cfg (day of week) with exit
alias=holidays.quick.cfg

# overwrite lock/unlock per workday (day of week)
alias=workday.cfg
```
2.7.1 Aliases (under development)

Aliases are textfiles in /var/web-gui/_log/autolock/aliases/. If you want to setup your own alias or modify one, create a new alias file and start the name with my. Do not modify default aliases as they are overwritten on updates. In an alias you can set unlock_hour and lock_hour as defaults and define and/or rules to modify unlock_hour or lock_hour. If you use the same aliasname like a default alias, every reference to the default will be automatically replaced by the my one. Example my_holiday.cfg overwrites holiday.cfg automatically.

******
Aliases
******

Aliases can be used as references in timetables to specify a date, a day of week or an hour either as a single value or a date range like 18.2-25.2 (hour must be a single value, dow can be a list like mo,di,fr). (you should not use a concrete time or date in a timetable)

If you want to modify aliases, duplicate an alias file with some name that starts with my_. It will be then used automaticalls instead the default one.

You can also create other my_aliases.cfg for other rules

##
Alias files can contain a if_dow, if_day, if_month, if_year, if_date, if_between (yyyy.mm.dd-yyyy.mm.dd)
If you define more than one if, the are AND related. If you use a commalist, they are OR related

# example dow
if_dow=mon,tue,wed,thu,fri; unlock_hour=8:00; lock_hour=21:00; # without exit -> continue parsing
if_dow=sat; unlock_hour=8:00; lock_hour=13:00; # without exit -> continue parsing
if_dow=sun; unlock_hour=; lock_hour=; # no autolock/unlock

# example quick (holiday or priority rules)
if_between=2018.12.23-2019.01.06; unlockhour=; lockhour=21:00; exit; # exit at the end means a quick rule

# example multiple ifs
if_between=2018.12.23-2019.01.06; if_dow=sun; unlockhour=; lockhour=13:00; exit; # exit at the end means a quick rule,

In general
Rules are based on keywords if_day, if_dow (mon to sun), if_month, if_year, if_between and exit
Rules are separated with „ ; “ in an and manner and within the same key with a commo for or relations ex if_dow=sat,sun

At the moment, edit the timetables and aliases with WinSCP.

3.0 more manuals

https://www.napp-it.org/manuals/index_en.html