CryptNDrive Administration Guide

Version 4.0.5-246-g427649bc

Lybero developement team

2019-11-12

Version : 4.0.5-246-g427649bc

Summary

1 Global view

1.1 Introduction

CryptnDrive is a secured web file sharing system providing end to end encryption through native web browser encryption of contents. This manual is split in 3 parts : the architecture of the system, the installation of the web server and finally the configuration of the system.

If you already have an instance and just want to change the configuration parameters, please jump directly to the Configuration chapter.

1.2 Architecture

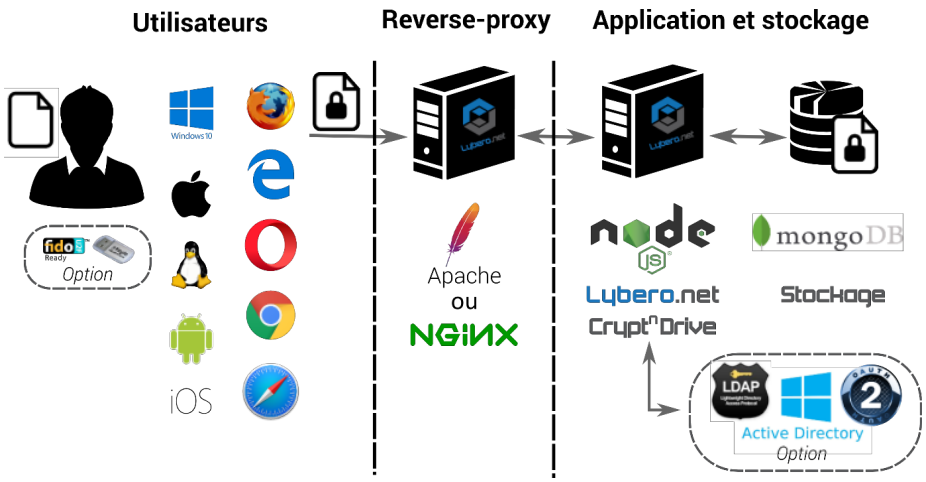

Figure 1: Technical architecture

The different components of the Transfer system are the following:

- A web access server in charge of communications between other components,
- A database storing all encrypted and unencrypted data (the metadata).
- Users web browsers.

1.3 Constraints

- Server or VM

- A linux distribution (redhat, centos, debian, ubuntu, suse, slackware, . . .). Certified on Debian 8 and Centos 7.
  - No specific CPU requirements
  - At least 60MB of RAM per users connected simultaneously
  - Disk according to the expected size of the storage (typically 2GB / user with an account)
  - The server must be able to send mails either through a local smtp server or through a remote one (we detail the configuration after)
  - ssh root access for software installation, the www-data (for debian based distribution) or www (for rpm distributions) account will be used
  - Apache 2.2 or later installed and functional, used in reverse-proxy for access to the nodejs server. Reverse proxy ensures ssl encryption of communications with browsers.
  - A valid SSL certificate for clients' web browsers
  - Nodejs 6.9.X or 6.10.X installed or installable
- Network constraints:
  - Port 80 (http) and 443 (https) must be accessible from the outside. A redirect is done from the port 80 to the port 443 (in the apache configuration).
  - Port 22 (ssh) must be accessible from the outside
  - If the mongodb database is on a third-party server, we recommend a dedicated and reserved network interface for this purpose, ssh access and monitoring.

1.4 Multiple instance installation on a single server

It is possible to have multiple instance running on a single server. Typically an instance on https://drive.thecompany.com and other ones such as https://transfer.thecompany.com and https://transfer.company-drive.com. In this case, separated database will be used in the mongodb server as well as separated nodejs servers for each subdomain.

2 Installation

2.1 Debian and Centos prerequisites

You must be root on the server to set up the application. If you are not, please, ask the person that is root to make the installation instead of you. In case you are not root, we will not provide to you any installation support.

2.2 Getting CryptnDrive

Lybero.net has its own server to provide installation archives. We need to have your username and password. Please type the following command and send us (contact@lybero.net) the result (remembering your password).

```
htpasswd -n <votre login>
```

A password will be asked, you will then send us (contact@lybero.net) the password hash that we will enter in our system.

You will then be able to recover the following file:

```
https://npmjs.lybero.net/repository/lynvictus-latest.tgz
```

This tar contains an already prepared version of the application with the modules. The full sequence to download the file is the following :

```
cd /root
mkdir Source
wget --no-check-certificate --user yourusername --password yourpassword \
https://npmjs.lybero.net/repository/lynvictus-latest.tgz
cd Source
tar -xvf ../lynvictus-latest.tgz
```

You then have in /root/Source a lynvictus-2.1.3 directory (from the cryptndrive version number). We will now copy (or move) this directory where it is needed by simply renaming it by the version number, then create an instance.

The content of lynvictus-2.1.3 is the following:

```
lynvictus-version
                |-- client
                |-- Documentation
                |-- Libs
                |-- node_modules
                |-- scripts
                        |-- common.sh
                        |-- configureInstance.sh
                        |-- configureServer.sh
                        |-- crypt.js
                        |-- install.sh
                        |-- migrate.js
                        |-- template
                                |-- apache_conf
                                        |-- instance.conf
                                        |-- letsencrypt.conf
                                |-- instance_conf
                                        |-- default_yml.template
```


```
                                |-- systemd.service
                |-- server.js
```

To install the app, please use shell scripts in scripts/. All of these need to be executed with root user and you can use –dry-run to see what the script will do without execution

2.3 Automatic installation

Use configureServer.sh to check your system

```
cd lynvictus-<version>
./configureServer.sh
./configureServer.sh --help # for help
./configureServer.sh --dryrun # just display and do nothing
```

It will check and install all required packages.

- It checks first if there is a lynvictus unix account, if no, the script create it
- It then checks if apache2, mongodb-server, curl, nodejs and certbot are installed, if not, it installs them with apt or yum

2.4 Manual installation

You need to have:

- Apache2
- MongoDB Community Edition 3.x
- certbot for ssl certificates
- curl
- Node.js 9.X

2.4.1 Getting MongoDB Community Edition 3.x on debian

To get MongoDB Community Edition 3.x :

```
apt remove -y mongodb"
apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 0C49F3730359A14518585931BC711F
echo "deb http://repo.mongodb.org/apt/debian jessie/mongodb-org/3.4 main" | tee /etc/apt/so
apt-get update
apt-get install -y mongodb-org
```

2.4.2 Getting MongoDB Community Edition 3.x on centos 7

To get MongoDB Community Edition 3.x :

```
 yum -y remove mongodb-server"
 cat >/etc/yum.repos.d/mongodb-org-3.4.repo <<EOF
[mongodb-org-3.4]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/\$releasever/mongodb-org/3.4/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc
EOF
 yum install -y mongodb-org
 /usr/bin/chown mongod:mongod /var/log/mongodb/mongod.log
 systemctl restart mongod.service
```

2.4.3 Getting Node.js 9.X on debian

To get node.js 9.x :

```
curl -sL https://deb.nodesource.com/setup_9.x | sudo -E bash - \
&& apt install -y nodejs
```

2.4.4 Getting Node.js 9.X on centos 7

To get node.js 9.x :

```
curl -sL https://rpm.nodesource.com/setup_9.x | bash - && yum install -y nodejs
```

2.4.5 Getting certbot on centos 7

```
sudo yum install epel-release
sudo yum install certbot
```

2.4.6 Creating a “lynvictus” user account

You also have to create a lynvictus UNIX account :

```
useradd -s /bin/bash -m -d /home/lynvictus -c "lynvictus user"
```


2.5 Configuration

You just have completed a manual or automatic installation of the software with all dependencies. Now it is time to configure the different elements. That is what is described hereunder.

Use install.sh scripts (in the scripts directory) to install a version of the lynvictus application. You can use it in different ways. The “normal” command is “./install.sh”. However, you can try first the –dryrun option to see what will be done, without doing anything.

```
./install.sh
./install.sh --help # for help
./install.sh --dryrun # just display and do nothing
./install.sh /var/html/drive # To install in the directory /var/html/drive
```

The “./install.sh” script will write to the to the destination folder.

It will create a version folder (named like the version number of the app) and copy all the folder’s files into the version folder and change owner to www-data:

```
/var/www/html/Lynvictus/<version>
    |-- client
    |-- Documentation
    |-- server.js
    |-- defaultConfig
```

2.6 Instance creation

Use configureInstance.sh to install a new instance or update an existing one, please, use the same destination as used in the install.sh scripts :

```
./configureInstance.sh <name>
./configureInstance.sh --help # for help and options
./configureInstance.sh --dryrun # Do nothing, just display actions
```

- name : instance’s name (without space)

It will creates a new instance folder into the destination folder or update it if exist and create a backup.

Then it will ask you for informations to fill the config file `configs/default.yml`. When you are upgrading an running installation, you can avoid this part with the option `--noconfig`

Then the script install the apache2 reverse proxy, configure its certificate for https using cerbot.

Warning At this step the host must be accessible in http (80/tcp) for host certificate validation.

Finally, the script installs a startup file in systemd for this instance. At this step you can start the server with

```
sudo systemctl restart <name>.service
```

cat /destination/folder/name/configs/default.yml

```
  # The instance Name
  # -
  instance: name
  # URL in case of
  # ---
  url : https://name.lybero.net
  # The server port(s)
  # -
  daemons:
 standard:
   port: 3000
  # ssl:
  #   port: 3000
  #   key : ssl/server.key
  #   crt : ssl/server.crt

  initial:
 # --- Want to create initial users (alice / bob / charlie)
 # --- and theres filesets
 # -
 withInitialDB: true
 # --- Want to fill the db with a lot of users
 # and filesets (the number)
 withInitialDBNum: 0

  # Database information
  #
  database:
 url: 'mongodb://localhost:27017/name'

  # The instance maximal storage size
  # Should be formatted like the following example :
  # 2o, 4Mo, 8Go or 16To
  # NB : No variable means unlimited storage
  # --
  # instanceMaximalStorageSize :
```

```
cat /etc/apache2/site-available/name.lybero.net.conf

  <VirtualHost name.lybero.net:80>
  ServerName name.lybero.net
```


```
ServerAdmin admin@mail.com
DocumentRoot /var/www/html

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

RewriteEngine on
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,QSA,R=permanent]
</VirtualHost>

<IfModule mod_ssl.c>
<VirtualHost name.lybero.net:443>
ServerName name.lybero.net
ServerAdmin admin@mail.com
DocumentRoot /var/www/html

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

SSLCertificateFile /etc/letsencrypt/live/name.lybero.net/fullchain.pem
SSLCertificateKeyFile /etc/letsencrypt/live/name.lybero.net/privkey.pem
Include /etc/letsencrypt/options-ssl-apache.conf

# --
# --- redirection for lynvictus demo ---
# --
ProxyPreserveHost On
ProxyRequests off

RewriteEngine On
RewriteCond %{HTTP:Upgrade} =websocket [NC]
RewriteRule /(.*)     ws://localhost:3000/$1 [P,L]

ProxyPass / http://localhost:3000/ retry=1 acquire=port timeout=600 \
Keepalive=On
ProxyPassReverse / http://localhost:port/
ProxyPassReverseCookiePath http://localhost:port https://name.lybero.net
# ---
</VirtualHost>
</IfModule>
```

2.7 Configuring the firewall

```
firewall-cmd --zone=public --list-all
```

```
firewall-cmd --zone=public --add-service=http --permanent
firewall-cmd --zone=public --add-service=https --permanent
firewall-cmd --zone=public --add-service=http
firewall-cmd --zone=public --add-service=https
firewall-cmd --reload
firewall-cmd --zone=public --list-all
```

2.8 Installing the reference instance

We want to be able to install different versions of the CryptnDrive application. For this, for each version, we install the different files and modules in a directory. Then we make symbolic links between the sub-directories of a version and an instance. This makes it possible to have a different configuration file for the instance in a basic way.

```
/var/www/html/Lynvictus
       |-- 2.1.3
              |-- client
              |-- Libs
              |-- .....
              |-- configs
       |-- instance
              |-- client (lien symbolique vers ../../2.1.3/client)
              |-- Libs  (lien symbolique vers ../../2.1.3/Libs)
              |-- ....
              |-- configs (contient un vrai fichier)
```

Copy of the version.

```
mkdir -p /var/www/html/Lynvictus
mkdir -p /etc/httpd/conf.d
cp -R lynvictus-2.1.3 /var/www/html/Lynvictus/2.1.3
```

We will create the “drive” instance on the server.

```
mkdir "/var/www/html/Lynvictus/drive"

SUBDIR_LIST=(client Libs mail node_modules notifServer.js scripts \
server.js SharedLibs ssl)

for SUBDIR in ${SUBDIR_LIST[*]}; do
    ln -s /var/www/html/Lynvictus/2.1.3/${SUBDIR} \
  /var/www/html/Lynvictus/drive/${SUBDIR}"
done
chown -R apache:apache /var/www/html/Lynvictus
```

2.9 Drive instance configuration

You have to modify the file '/var/www/hml/Lynvictus/drive/configs/default.yml'. It must indicate: the name of the instance (drive), the port on which the node server will run, the port on which the Mongo server runs and the name of the database. If you want to authenticate via Google, you must enter the clientID, clientSecret information provided by Google through the third-party application configuration interface.

So you need to edit the following lines: * url : url of the drive * port : port on which the drive will listen * database: mongodb base url

```
#The instance Name
#-
instance: drive
# URL for creating mails url
# ---
url : https://drive.lybero.net
# The server port(s)
#-
daemons:
  standard:
 port: 3000
initial:
  #--- Want to create initial users (alice / bob / charlie)
  #--- and their filesets
  #-
  withInitialDB: true
  #--- Number of users and filesets to fill the db with initially
  #--- for test purpose
  withInitialDBNum: 0

#Database information
#
database:
  url: 'mongodb://localhost:27017/drive'
#auths:
#    local: {}
#    google:
#        clientID:  to_be_configured_by_yourself _to_allow_a_google_authentication
#        clientSecret: to_configure_by_yourself_to_have_a_google_authentication
#        callbackURL: 'https://drive.cryptndrive.fr/oauthcallback'
```


2.10 Apache configuration for drive

Here we consider that there is only one node.js server running on port 3000.
File /etc/httpd/conf.d/drive.conf

```
<VirtualHost drive.lybero.net:80>
  ServerName drive.lybero.net
  ServerAdmin your.mail@your.domain
  DocumentRoot /var/www/html

  ErrorLog logs/error.log
  CustomLog logs/access.log combined

  RewriteEngine on
  RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,QSA,R=permanent]
</VirtualHost>

<IfModule mod_ssl.c>
Listen 443 https
<VirtualHost drive.lybero.net:443>
  ServerName drive.lybero.net
  ServerAdmin your.mail@your.domain
  DocumentRoot /var/www/html

  ErrorLog logs/error.log
  CustomLog logs/access.log combined

  SSLEngine on
  SSLProtocol    all -SSLv2 -SSLv3
  SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:\
  ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:\
  DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:\
  ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:\
  ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:\
  ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:\
  DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:\
  DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-SHA:\
  AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:\
  AES256-SHA:AES:CAMELLIA:DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:\
  !PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA
  SSLHonorCipherOrder  on
  SSLCompression    off
  SSLOptions +StrictRequire
  # Add vhost name to log entries:
  LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" \
  vhost_combined
```


```
  LogFormat "%v %h %l %u %t \"%r\" %>s %b" vhost_common
  SSLCertificateFile /etc/letsencrypt/live/drive.lybero.net/fullchain.pem
  SSLCertificateKeyFile /etc/letsencrypt/live/drive.lybero.net/privkey.pem
  Include /etc/letsencrypt/options-ssl-apache.conf

  # -
  # --- redirection for lynvictus demo --
  # -
  ProxyPreserveHost On
  ProxyRequests off

  #RewriteEngine On
  #RewriteCond %{HTTP:Upgrade} =websocket [NC]
  #RewriteRule /(.*)     ws://localhost:3000/$1 [P,L]

  ProxyPass /ws  ws://localhost:3000/ws
  ProxyPassReverse /ws  ws://localhost:3000/ws

  ProxyPass / http://localhost:3000/ retry=1 acquire=3000 timeout=600 \
  Keepalive=On
  ProxyPassReverse / http://localhost:3000/
  ProxyPassReverseCookiePath http://localhost:3000 https://drive.lybero.net
  # -
</VirtualHost>
</IfModule>
```

If you also have a base domain with static files, you must not forget to create the corresponding apache configuration:

```
############## /etc/httpd/conf.d/cryptndrive.conf
<VirtualHost www.cryptndrive.fr:80>
 DocumentRoot /var/www/html
 ServerName www.cryptndrive.fr
</VirtualHost>
####################################
```

Restarting apache

```
service httpd restart
```


2.11 Nginx reverse proxy configuration

```
server {
  listen 443 ssl;
  server_name localhost
  ssl on;
  ssl_certificate /etc/apache2/certificates/localhost.crt;
  ssl_certificate_key /etc/apache2/certificates/localhost.key;
  location /ws {
    proxy_pass http://localhost:3000/ws;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "Upgrade";
  }
  location / {
    proxy_pass http://localhost:3000;
  }
}
```

2.12 Network configuration

2.12.1 SELinux

In order for apache to connect to nodejs, it must be allowed to make a network connection.

The command is :

```
/usr/sbin/setsebool -P httpd_can_network_connect 1
```

2.12.2 Special case of NAT

If your machine where your virtual machine is, is behind a NAT, then you have to do a special network configuration. In fact, the IP address corresponding to the domain of the drive does not correspond to the IP address provided by the DNS for this domain. In this case, we must add in the file / etc / hosts the line:

```
192.168.0.1 drive.lybero.net
```

Obviously replace 192.168.0.1 with the machine's local IP and drive.lybero.net with the domain name used for your drive.

2.12.3 mail configuration

The configuration of the mail can be done in 2 different ways:

- using only the parameters in the CryptNDrive application. By default, outgoing mail is considered to be sent on port 25 of localhost. However, the configuration settings for sending mail on the CryptNDrive server are editable. You have to go to the main menu, then Administration, then Configuration.
- or by directly configuring a local mail server as a relay to a third-party SMTP service.

We will examine this last configuration in the case of OVH. All you need to do is set up a default mail user for your domain in OVH, and authenticate with that user.

2.12.3.1 Configuring postfix to use OVH relay You need to have several postfix modules for this to work:

```
yum install libsasl2-modules
yum install cyrus-sasl cyrus-sasl-lib cyrus-sasl-plain
yum install cyrus-sasl-gssapi
yum install cyrus-sasl-ntlm
```

We will create a file / etc / postfix / sasl-passwords with the email address and the password of the person whose email was created in the OVH interface:

```
[pro1.mail.ovh.net]:587 cryptndrive@cryptndrive.fr:mettre_le_mot_de_passe_ici
```

You have to transform the file into a db:

```
postmap hash:sasl-passwords
```

Now you have to create the ssl key files for the connection:

```
cd /etc/ssl/certs
make genkey
```

In /etc/postfix/main.cf, the parameters to be modified are:

```
myhostname = cryptndrive.fr
inet_interfaces = $myhostname, localhost
mydestination =
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/localhost.crt
smtpd_tls_key_file=/etc/pki/tls/private/localhost.key
```


```
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
smtp_tls_security_level = may


relayhost = [pro1.mail.ovh.net]:587
smtp_sasl_auth_enable=yes
smtp_sasl_mechanism_filter = !gssapi, plain, login
smtp_sasl_password_maps=hash:/etc/postfix/sasl-passwords
smtp_sasl_security_options= noanonymous
```

You need a program to try:

```
yum install mutt
```

It only remains to try to send an email with mutt.

2.13 TroubleShooting

A script is available for that !

on the server

```
curl http://localhost:3001/ping
# must return {"pong":true}
```

1. node is running for this instance ?

On the server.

```
./check.sh <name> # Do some checks and return a report to see what goes wrong
```


3 Configuration

3.1 Startup configuration

At the first start of the application you can login with this accounts:

User	Password	Description
`root`	`root`	Master account, with all rights (right of *manage* on the object *application*). You must change his password ! This account can be deleted after giving rights to somebody else.
`alice`	`alice`	A normal user. usable for testing.
`bob`	`bob`	A normal user. usable for testing.
`charlie`	`charlie`	A normal user. usable for testing.

All those users can be deleted.

3.2 Application configuration

All the application can be access by the *Menu* (left up button), then *Administration*, then *Configuration*.

This menu and the ability to modify the configuration is attach to the right of *manage* on the object *application*. The first configured user with this right is `root`.

3.2.1 General

Field	Description	Comment
`Instance`	Instance name	This field cannot be changed. It is just for information on the name of the instance.

Field	Description	Comment
`The application Name`	Application's name	You can change here the application name in the main menu (*top bar*).
`A subtitle`	Subtitle	Not used.
`Tel for contact`	A phone number	In the signature of all emails, this is the contact phone number
`Email contact`	An email	In the signature of all emails, this is the email for contact
`Logo`	An image	You can upload an image (any format) for the logo. The image will be reduced and resize to fit the logo size.
`Favicon`	An image	You can upload an icon (with the icon format .ico) to customize the favicon in web browsers

Field	Description	Comment
`Pooling interval`	Delay	This is the cron pooling delay for emails and purge of DB, it can be a number followed by s (second) m(minutes)

3.2.2 Notifications

This is the notification (chat between users and information from server) configuration.

Field	Description	Comment
`Expiration`	a delay	How much time you will keep any notifications. `0` mean no expiration.

3.2.3 Server

This is the server configuration.

Field	Description	Comment
`websockets`	true or false	Say if you want websockets for speedup transfer. In some installations, websocket can be filtered
`Default IP restriction`	IP Address CIDR Format	Select “accept” or “Deny”. This system to filter connections by IP

3.2.4 Rights

In this section, you can affect rights to specific groups, elements are clear enough to not go into the details for each one. Just understand that these rights

are for the entire application.

3.2.5 Groups for validation

You can specify here a group which will validate any invitation.

3.2.6 Chat

Field	Description	Comment
`enabled`	enable the chat function	This is activated by default, but you can disabled the chat if you want to.

3.2.7 authentications

By default two authentication methods are configured : `local` and `google`. But two others can by added which are : `LDAP` and `AD`.

Obviously, if you decide to work with LDAP, AD or Google authentications, users we'll be prompt at sign-up for a key passphrase. This passphrase is different from the authentication password, if a user loose his passphrase, he'll be able to connect to the application but will not be able to access any encrypted data.

For each of the following authentication type, you'll find these parameters :

Field	Description	Comment
`Type`	Type of authentication	Is one of the available methods
`enabled`	enable or disable this authentication type	
`Groups by default`	fallback group of all new users	All new users will be part of this group

3.2.7.1 Local Configuration fields for local authentication are :

Field	Description	Comment
self signing users		
`enable`	enable / disable for self signing users	by default, it is set to *true*, you can disable it to forbid user to signing
`expiration`	set the expiration delay for self signing users	0 by default (no expiration)
`groups by default`	Destination group for self signing users	No groups by default
invited users		
`expiration`	set the expiration delay for invited users	0 by default (no expiration)
`groups by default`	Destination group for invited users	No groups by default

3.2.7.2 Google So, the application can use external authentication (**oauth2**). The google authentication can be set up in this place

- `google` authentication

first you must set up for your server a Identity provider. Go in google development interface :

https://console.developers.google.com/apis/

and fill a new identity with the following elements

1. Create an `Id client Oauth`
2. Select `Application web`
3. fill the following fields :

Field	Description
`Name`	A name for this Oauth2 provider
`javascript origin`	Your Server (example *https://drive.lybero.net*)
`callback redirect`	the callback URL , your server/oauthcallback (example *https://drive.lybero.net/oauthcallback*)

And Google provides you two fields `clientId` and a `client Secret`:

Then you can enter in the configuration :

Field	Description	Comment
`The clientID provided by Google`	clientId	provided by Google
`The client secret provided by Google`	client Secret	provided by Google

WARNING Do not change the `Url for connection` and `Callback url` parameters unless you really know what you're doing.

3.2.7.3 LDAP

Field	Description	Comment
Url for connection - endpoint	Endpoint in application	**Do not change this setting** this parameter will be removed in a future update
Url for connection	typically ldap://	Url for connection to your LDAP
bindDN	domain name	like `dc=example,dc=com`
bindCredentials	the credentials for the app to authenticate	

Field	Description	Comment
searchBase	the search base of your LDAP users	
searchFilter	the search filter	to limit access
searchAttributes		
tlsOptions		

3.2.7.4 AD

Field	Description	Comment
Url for connection		
baseDN		
username		
password		

3.2.8 mail

This is the email server configuration, for sending emails (SMTP).

- `connexion` configure the way to contact the email server (SMTP protocol)

Field	Description	Comment
`Host`	hostname or IP address	The SMTP server accepting connection from the app.
`Port`	port number	25, 587 or 465 in general.
`Secure`	if you are using TLS	if true the connection will use TLS when connecting to server. If false (then TLS is used if server supports the STARTTLS extension. In most cases set this value to true if you are connecting to port 465. For port 587 or 25 keep it false

Field	Description	Comment
`ignoreTLS`	refuse server TLS	if this is true and secure is false then TLS is not used even if the server supports STARTTLS extension
`auth User`	the username	In case of a authenticated SMTP communication, the userName
`auth Password`	the password	In case of a authenticated SMTP communication, the user password
`tls rejectUnauthorized`	reject unknown certificates	If not false the server will reject any connection which is not authorized with the list of supplied CAs. This option only has an effect if requestCert is true.

Field	Description	Comment
`Subject prefix`	Prefix in any subjects	You can add this string in the subject field (can be used for filtering by user)
`from`	from email field	emil from
`replyTo`	replyTo email field	replyTo email field
`Digest delay`	delay	delay for digest email

3.2.9 errors configuration

You have the hability to send reports on application bug directly... where you want. Actually, only a Mattermost is available.

if you enable this feature, if a javascript bug occured, on a client or on the server, a report is pushed into a slack channel.

Field	Description	Comment
`enable`	Toggle to yes for enabling it	

Field	Description	Comment
`slack webhookURL`	the URL provided by your slack configuration	it define the channel to post on your team slack configuration

For sending your bugs directly to lybero.net, you can use this mattermost webhookURL :

`https://mattermost.lybero.net/hooks/epp6ohionjgizpuqgog67xcrph`

3.2.10 Theme

This part grant you the possibility to change some colors of Crypt n Drive, in order to keep it readable for everyone, we have a limited amount of possibilities, but this may change in a future update, to let you customize the app a little bit more.

In this section, get on one theme name and click the arrow to unfold properties.

Field	Comment
Primary color	main accent of the app, in the top bar, the left menu
Secondary color	color for errors, suppression buttons
Other colors	
encrypted vaults	main accent in the “vaults” section (root of the app)
decrypted vaults	not used
invitations	color of the invitation badge
quorums	main accent in the quorum section
groups	main accent in the groups section
users	main accent in the users section

3.3 Server Logs

All logs are managed by `rsyslogd` on the server. You can found it at

`/var/log/<instance name>.log`

3.4 Error reporting

Error reporting is usefull to find bugs on exotic configurations (browsers, architectures), wich cannot be tested in lab before publishing.

You can setup error reporting to lybero.net directly and permanently, or to your own server, or just during a debbuging session. as you want.

No sensitive datas are in the report (no password, passphrases, keys, files, etc. . .). only informations on the browser, the architecture, and the code file / line in trouble.

The error report contain the following information and nothing more (this is an example) :

```
<hostname>/<instanceName>
blob is undefined{
 "navigator": {
   "appCodeName": "Mozilla",
   "appName": "Netscape",
   "appVersion": "5.0 (X11)",
   "cookieEnabled": true,
   "language": "fr",
   "oscpu": "Linux x86_64",
   "product": "Gecko",
   "userAgent": "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.
 },
 "builder": "wallrich",
 "commit": "e1f0fc3",
 "builderhost": "kryha",
 "server": {
   "server": "kryha",
   "instance": "developement",
   "builder": "wallrich",
   "commit": "e1f0fc3",
   "builderhost": "kryha",
   "login": "ANONYMOUS",
   "userId": "0"
 }
}
```