sysPass Documentation

Release 2.1.16

Rubén Domínguez

Jan 05, 2019

Contents

sysPass is a password management system written in PHP that allows a centralized passwords management in a multiuser environment.

CHAPTER 1

Features

- Interface based on Material Design Lite with HTML5 and Ajax
- Encrypted passwords within AES-256 CTR
- Multiuser with users, groups and profiles management
- Advanced profile management with 29 access levels
- MySQL/MariaDB, OpenLDAP and Active Directory authentication
- Activity notifications by email and in-app
- Public links to accounts without login
- Accounts changes history
- Accounts associated files management with images preview
- Multilanguage translated to Spanish, Catalan, German, Polish, Russian, French and Dutch.
- Link to external Wiki and DokuWiki's API integration
- Portable format backup and export to encrypted XML
- Actions and events log with the ability to send messages to a remote Syslog in CEF format
- Configurable and extensible using themes and plugins
- API for integrating with other applications
- Import from KeePass, KeePassX and CSV
- One step installation

CHAPTER 2

What sysPass does not do

- It does not store the master password in the server
- It does not send any data to an external service
- It does not encrypt the account's password individually, it uses a master password for all instead
- It does not perform password changes on the servers
- It does not encrypt the accounts' data, only the password and custom fields data, because you couldn't perform searches
- It isn't like APT: doesn't have Super Cow Powers!!

2.1 Installation

2.1.1 Debian 8 Installation

Prerequisites

- Web server (Apache/Nginx/Lighttpd) with SSL enabled.
- MariaDB or MySQL >= 5
- PHP >= 5.6 <= 7.0
- **PHP modules**
  - mysql
  - Curl
  - Json
  - GD
  - XML

  - mbstring
  - ldap (optional)
- The latest sysPass release https://github.com/nuxsmin/sysPass/releases

Installation

Packages installation in Debian GNU/Linux

```
apt-get install apache2 libapache2-mod-php5 php5 php5-curl php5-mysqlnd \
php5-curl php5-gd php5-json php5-mcrypt mysql-server
service apache2 restart
```

Optional to enable LDAP:

```
apt-get install php5-ldap
service apache2 restart
```

Directories and permissions configuration:

Create a directory for the application in the webserver root

```
mkdir /var/www/html/syspass
```

Copy and unzip the sysPass archive in the newly created directory

```
cp sysPass.tar.gz /var/www/html/syspass
cd /var/www/html/syspass
tar xzf syspass.tar.gz
```

Change the owner of 'syspass/config' directory. It should be the same user that the web server runs

```
chown www-data /var/www/html/syspass/config
chmod 750 /var/www/html/syspass/config
```

Create an change the backup directory owner

```
mkdir /var/www/html/syspass/backup
chown www-data /var/www/html/syspass/backup
```

Environment configuration

Open the web browser and point to the URL

https://IP_OR_SERVER_NAME/syspass/index.php

Note: Please, follow installer steps and after the successful finishing, you will be able to log into the application

To know how sysPass works, please see *Application*

Warning: It's advisable to read the security recommendations on *Security*

2.1.2 Debian 9 Installation

Warning: Work in progress

Prerequisites

- Web server (Apache/Nginx/Lighttpd) with SSL enabled.
- MariaDB >= 10.1
- PHP >= 5.6 <= 7.0
- **PHP modules**
  - mysql
  - Curl
  - Json
  - GD
  - XML
  - mbstring
  - ldap (optional)
  - mcrypt (if upgrading from <= 2.0)
- The latest sysPass release https://github.com/nuxsmin/sysPass/releases

Installation

Packages installation in Debian GNU/Linux

```
apt install apache2 libapache2-mod-php php php-curl php-mysqlnd php-curl \
php-gd php-json mariadb-server php-ldap php-mbstring

service apache2 restart
```

Optional to enable LDAP

```
apt-get install php5-ldap
service apache2 restart
```

Optional to enable SSL

In order to increase the access security to your sysPass installation, consider using SSL. See *Security* and the following resources for Debian:

- For LAN-only websites or to use a self-signed SSL certificate: https://doc.debian.org/configuration/Self-Signed_Certificate
- For Internet-facing websites consider using LetsEncrypt - see https://certbot.eff.org/

Directories and permissions configuration:

Create a directory for the application in the webserver root

```
mkdir /var/www/html/syspass
```

Copy and unzip the sysPass archive in the newly created directory

```
cp sysPass.tar.gz /var/www/html/syspass
cd /var/www/html/syspass
tar xzf syspass.tar.gz
```

Change the owner of 'syspass/config' directory. It should be the same user that the web server runs

```
chown www-data /var/www/html/syspass/config
chmod 750 /var/www/html/syspass/config
```

Create an change the backup directory owner

```
mkdir /var/www/html/syspass/backup
chown www-data /var/www/html/syspass/backup
```

Environment configuration

Open the web browser and point to the URL

https://IP_OR_SERVER_NAME/syspass/index.php

Note: Please, follow installer steps and after the successful finishing, you will be able to log into the application

To know how sysPass works, please see *Application*

Warning: It's advisable to read the security recommendations on *Security*

2.1.3 CentOS 7 Installation

Prerequisites

- Web server (Apache/Nginx/Lighttpd) with SSL enabled.
- MariaDB or MySQL >= 5
- PHP >= 5.6 <= 7.0
- **PHP modules**
  - Mysql
  - mcrypt
  - ldap (optional)
  - SimpleXML
  - XML

  - Curl
  - Json
  - GD
  - PDO
  - mbstring
- The latest sysPass release https://github.com/nuxsmin/sysPass/releases

Installation

Packages installation

```
yum install httpd php-mysql php-pdo php-ldap php-gd php-pdo php-xml php-mbstring␣
↪mariadb-server mariadb wget
```

In order to start and auto-start the Apache Web server:

```
systemctl enable httpd.service
systemctl start httpd.service
```

In order to start and auto-start the MariaDB server:

```
systemctl enable mariadb.service
systemctl start mariadb.service
```

We need to secure the MySQL installation:

```
/usr/bin/mysql_secure_installation
```

Installing EPEL repository for encryption module

Download and install the RPM for the EPEL repository:

```
wget http://dl.fedoraproject.org/pub/epel/beta/7/x86_64/epel-release-7-0.2.noarch.rpm
yum install epel-release-7-0.2.noarch.rpm
yum install php-mcrypt
systemctl restart httpd.service
```

Enable the firewall ports

Add firewall rules:

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

Directories and permissions configuration:

Create a directory for the application in the webserver root:

```
mkdir /var/www/html/syspass
```

Copy and unzip the sysPass archive in the newly created directory:

```
cp sysPass.tar.gz /var/www/html/syspass
cd /var/www/html/syspass
tar xzf syspass.tar.gz
```

Change the owner of 'syspass/config' directory. It should be the same user that the web server runs:

```
chown apache /var/www/html/syspass/config
chmod 750 /var/www/html/syspass/config
```

Create an change the backup directory owner:

```
mkdir var/www/html/syspass/backup
chown apache /var/www/html/syspass/backup
```

Modifying SELinux

In order to allow sysPass to write its own configuration file and backups, we have two choices:

Note: Choose one of the options

- Change the SELinux user and target context to make writable the config and backup directories:

```
chcon -R -t httpd_sys_rw_content_t /var/www/html/sysPass/config/
chcon -R -t httpd_sys_rw_content_t /var/www/html/sysPass/backup/
mkdir /var/www/html/sysPass/tmp && chcon -R -t httpd_sys_rw_content_t /var/www/html/
↪sysPass/tmp
```

- Disable SELinux by editing '/etc/sysconfig/selinux' file, change the "SELINUX" variable value to "permissive" and reboot your system.

Environment configuration

Open the web browser and point to the URL:

https://IP_OR_SERVER_NAME/syspass/index.php

Note: Please, follow installer steps and after the successful finishing, you will be able to log into the application

To know how sysPass works, please see *Application*

Warning: It's advisable to read the security recommendations on *Security*

2.1.4 Gentoo Installation

Prerequisites

- Web server (Apache/Nginx/Lighttpd) with SSL enabled.
- MariaDB or MySQL >= 5
- PHP >= 5.6 <= 7.0
- **PHP modules**
  - mysql
  - pdo
  - mcrypt
  - ldap (optional)
  - XML
  - SimpleXML
  - Curl
  - Json
  - GD
  - intl
- The latest sysPass release https://github.com/nuxsmin/sysPass/releases

Packages installation

```
emerge --ask dev-db/mysql
emerge --ask www-servers/apache

PHP_TARGETS="php5-6"
USE="apache2 pdo curl simplexml xml zlib crypt gd intl json opcache"
emerge --ask dev-lang/php
```

If you want to use LDAP backend, you would need to compile PHP to support it:

```
PHP_TARGETS="php5-6"
USE="apache2 pdo curl ldap minimal simplexml xml zlib crypt gd intl json opcache"
emerge --ask dev-lang/php
```

At the following links there are more detailed documentation about how to install and configure these packages:

https://wiki.gentoo.org/wiki/Apache

https://wiki.gentoo.org/wiki/PHP

https://wiki.gentoo.org/wiki/MySQL

Apache configuration

You need to enable PHP module in Apache by modifying the variable APACHE2_OPTS and adding "-D PHP5" in the file /etc/conf.d/apache2:

```
APACHE2_OPTS="... -D PHP5"
```

We start and set auto-start for the Apache service:

```
rc-update add apached default
rc-service apache2 start
```

MySQL Configuration

You need to set MySQL root password:

```
emerge --config dev-db/mysql
```

We start and set auto-start for the MySQL service:

```
rc-update add mysql default
rc-service mysql start
```

We need to secure MySQL installation:

```
mysql_secure_installation
```

Directories and permissions configuration:

Create a directory for the application in the webserver root:

```
mkdir /var/www/localhost/syspass
```

Copy and unzip the sysPass archive in the newly created directory:

```
cp sysPass.tar.gz /var/www/localhost/syspass
cd /var/www/localhost/syspass
tar xzf syspass.tar.gz
```

Change the owner of 'syspass/config' directory. It should be the same user that the web server runs:

```
chown apache /var/www/localhost/syspass/config
chmod 750 /var/www/localhost/syspass/config
```

Create an change the backup directory owner:

```
mkdir var/www/localhost/syspass/backup
chown apache /var/www/localhost/syspass/backup
```

Environment configuration

Open the web browser and point to the URL:

https://IP_OR_SERVER_NAME/syspass/index.php

Note: Please, follow installer steps and after the successful finishing, you will be able to log into the application

To know how sysPass works, please see *Application*

Warning: It's advisable to read the security recommendations on *Security*

2.1.5 Docker Installation

The installation with Docker_allows to keep the application isolated from the system and you could try other versions without installing any additional package.

sysPass could be ran in Docker containers which have been built using Debian 8 and verified to work fine, without **performing packages compilation**.

The Docker images could be downloaded from Docker Hub‘_ which have been built automatically from Docker files at https://github.com/nuxsmin/docker-syspass

There are two choices for the installation:

- With Docker Compose, which allows to perform a full deployment of sysPass and the database.
- With Docker, it would be needed to deploy sysPass and the database separately

Docker Compose

Create the "docker-compose.yml" file and run docker-compose:

```
version: '2'
services:
  app:
    container_name: syspass-app
    image: nuxsmin/docker-syspass:latest
    restart: always
    ports:
      - "80:80"
      - "443:443"
    links:
      - db
    volumes:
      - /var/www/html/sysPass/config
      - /var/www/html/sysPass/backup
  db:
    container_name: syspass-db
    restart: always
    image: nuxsmin/docker-syspass:mysql
    ports:
      - "3306"
    environment:
      - DB_REMOTE_HOST=syspass-app.syspass_default
      - DB_REMOTE_ROOT_PASS=syspass
    volumes:
      - /var/lib/mysql
```

```
docker-compose -p syspass -f docker-compose.yml up -d
```

This will download the sysPass stable image and the MySQL database.

Note: When using Docker Compose, a separated network is created automatically for the sysPass containers and it's possible to use the DNS resolution.

The sysPass container has 2 volumes: one for the "config" directory and another for "backup"

Warning: The sysPass container will publish the host's ports 80 and 443

Docker

To get the individual images you would need to run the following commands:

```
docker run -d --name syspass-db -h syspass-db nuxsmin/docker-syspass:mysql
docker run -d --name syspass-app -h syspass-app --link syspass-db nuxsmin/docker-
↪syspass:latest
```

Access

The database access parameters are:

- Host: syspass-db
- User: root
- Password: syspass

It's possible to install other sysPass images on Docker Hub

Note: Please, follow installer steps and after the successful finishing, you will be able to log into the application

To know how sysPass works, please see *Application*

Warning: It's advisable to read the security recommendations on *Security*

2.1.6 Hosting Mode

The hosting mode is for those installations that are running on a external hosting, where is not possible to create neither database nor connection user for it.

Note: It won't create neither database (except tables) nor connection user

The steps to perform the installation are the following:

- Create an user/password for sysPass connection at the hosting panel.
- Create the sysPass database (not tables) and give permissions to the previous user on it.
- Start the sysPass installation (delete 'config/config.php' file if exists) and use the user/password that was previously created for sysPass (the two first fields in the installation page).

- Provide a MySQL/MariaDB user with administration rights (it could be the same as previous if it has enough permissions), in order to create sysPass database tables. This user is used only for the installation process and it often would be the user/password for the hosting management.
- If database connection and permissions are right, the installation should terminate successfully.

Note: En caso de errores, verificar el archivo de registro del servidor web.

2.2 Configuration

2.2.1 LDAP Configuration

Active Directory

Tips

- Checks if connection user is member of group "Account Operators"

OpenLDAP

In order to setup an OpenLDAP server correctly, you can follow the article at https://wiki.debian.org/LDAP/OpenLDAPSetup which describes the steps to configure a fully operational server under a Debian like distribution.

In OpenLDAP, to the use the group membership feature you need to add an 'overlay' called 'membeof'. It's a module that adds an internal attribute to those users which belongs to a group.

These are the steps to configure that module:

- Create the file 'ldap_memberof_add.ldif' with this content:

```
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulePath: /usr/lib/ldap
olcModuleLoad: memberof
```

- Create the file 'ldap_memberof_config.ldif' with this content:

```
dn: olcOverlay=memberof,olcDatabase={1}hdb,cn=config
objectClass: olcMemberOf
objectClass: olcOverlayConfig
objectClass: olcConfig
objectClass: top
olcOverlay: memberof
olcMemberOfDangling: ignore
olcMemberOfRefInt: TRUE
olcMemberOfGroupOC: groupOfNames
olcMemberOfMemberAD: member
olcMemberOfMemberOfAD: memberOf
```

- Modify the LDAP configuration by running these commands:

```
ldapadd -D cn=admin,cn=config -w "password" -H ldapi:/// -f memberof_add.ldif
ldapadd -D cn=admin,cn=config -w "password" -H ldapi:/// -f memberof_config.ldif
```

Tips

- Check whether the sysPass 'admin' user is the same in OpenLDAP, you need to add this user to the LDAP group that have access permissions to sysPass.
- The username and email ofthe LDAP users are populated from 'displayname','fullname' and 'mail' attributes.
- You could use ldaps by setting a connection URI like 'ldaps:/ /my_ldap_server'.
- You could install phpLDAPadmin to create and manage the LDAP objects.

Links

- LDAP Debian Wiki: https://wiki.debian.org/LDAP/OpenLDAPSetup
- 'memberof' overlay config: http://www.cbjck.de/2012/05/enabling-the-memberof-overlay-for-openldap/

2.3 Application

sysPass is an application that uses a MySQL/MariaDB database to store the data of all its components except for the configuration, which is stored in an XML file within the "config" directory.

Warning: It's important that the "config" directory is not accessible from the web service, because it could reveal important information.

2.3.1 Encryption

Warning: If you already use a sysPass version <= 2.0, it's advisable to update to 2.1 version in order to use the new security improvements on the encryption mechanisms (CVE-2017-5999)

sysPass encryption is based on AES-256 in CTR mode by using the PHP OpenSSL module. It uses the Defuse/php-encryption library for the encryption modules and functions management.

The encrypted data (up to 2.0 version) are:

- Accounts' passwords
- Custom fields data
- sysPass XML format export

In order to use the application, for the first time, it will be needed to know either the master key or the temporary master key (see *Temporary Master Key*), because a Blowfish generated hash with a salt generated by using the MCRYPT_DEV_URANDOM random number generator, is the only saved. For the Blowfish hash generation a cost of 10 is used for the algorithm iterations

After log in with the master key, it's stored in the user's data. For its encrypted storage a password protected secure key is generated, by using the user's password, login and a salt generated using openssl_random_pseudo_bytes that is stored in the sysPass configuration within the tag "passwordSalt".

On the following logins the mater key is retrieved from the user's data and decrypted by using the user's password and login, besides the sysPass configuration salt. This key is stored in the user's session by encrypting with a Blowfish generated key from the PHP session and the session start time in UNIX format.

Note: Session key is regenerated every 120 seconds.

When the master key is changed it will be requested to every user the new master key or a temporary master key (see *Temporary Master Key*).

If an user changes its login password, in the next login he will be requested for the previous password in order to get the master key. If the master key couldn't be retrieved, it will be requested.

Temporary Master Key

A temporary master key could be generated to be used by the application users, so it won't be needed to tell the real master key.

For the temporary master key generation the real master key is used by encrypting it within a secure key generated key by using openssl_random_pseudo_bytes, which Blowfish hash is stored in the database "config" table.

Note: For the temporary master key checking a Blowfish generated hash is the **only** used

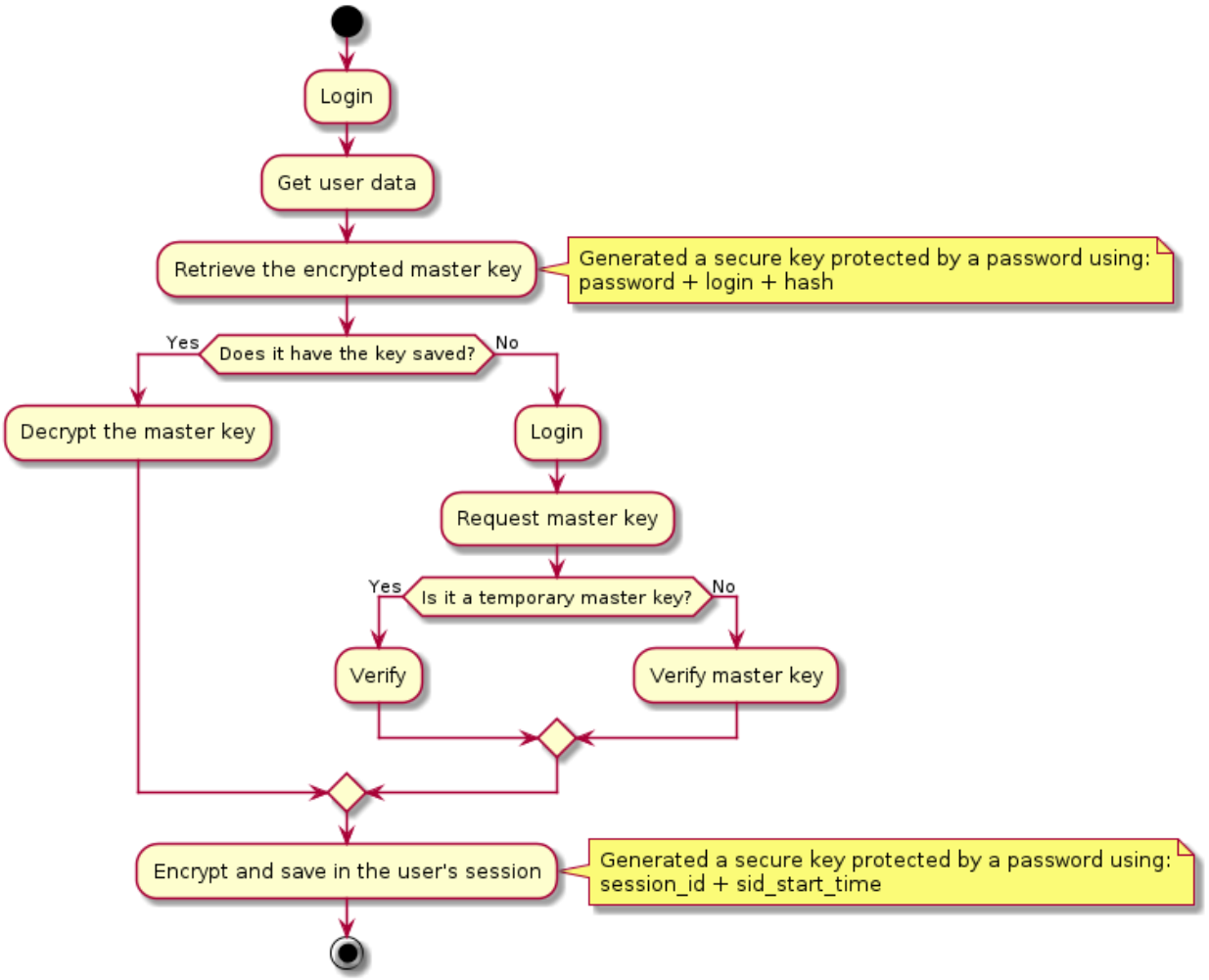
PKI

In order to improve the security of the sent data, PKI is being used to encrypt the passwords that are being sent from the application forms.

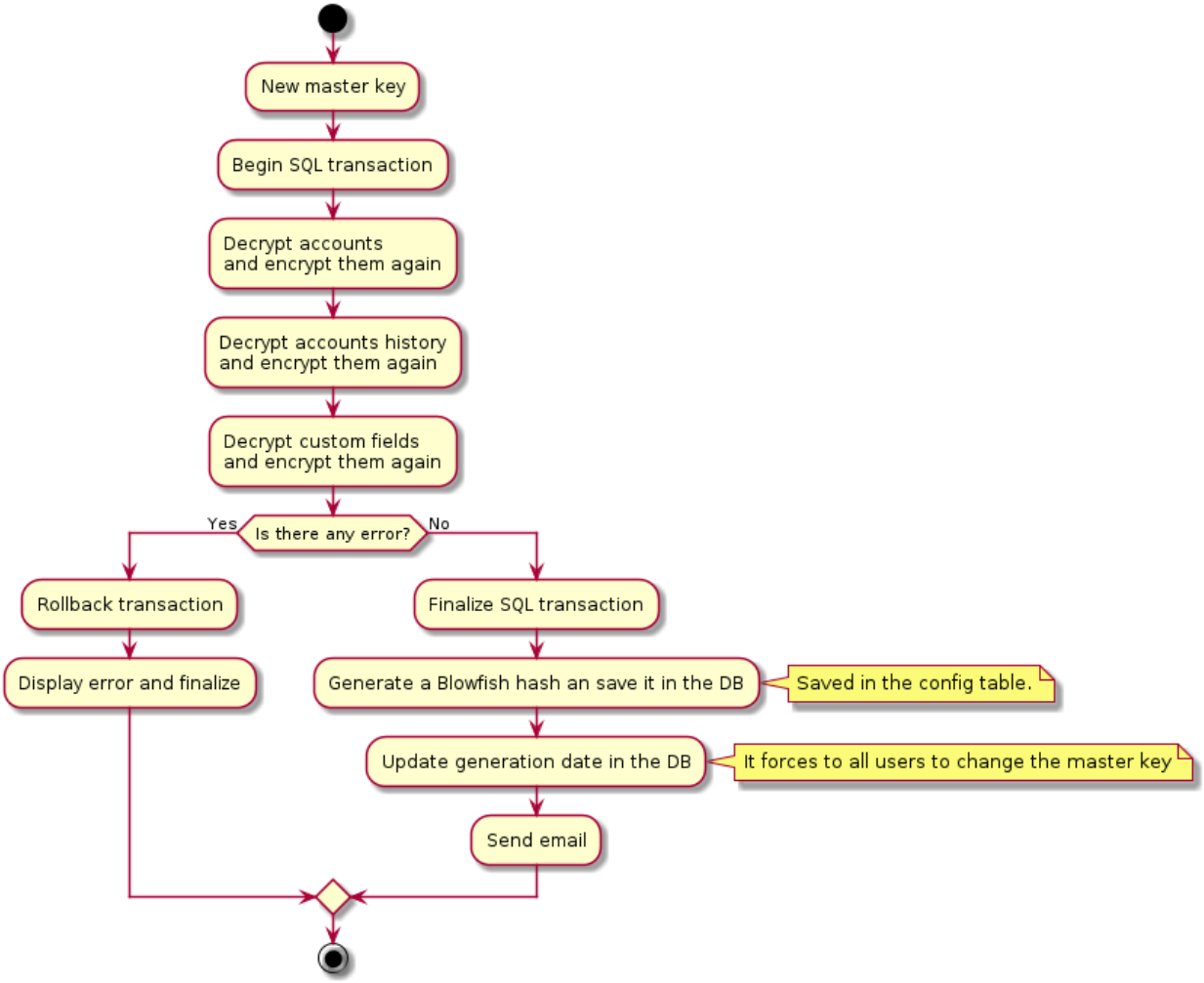
The public and private keys are generated within the application "config" directory.

2.3.2 Diagrams

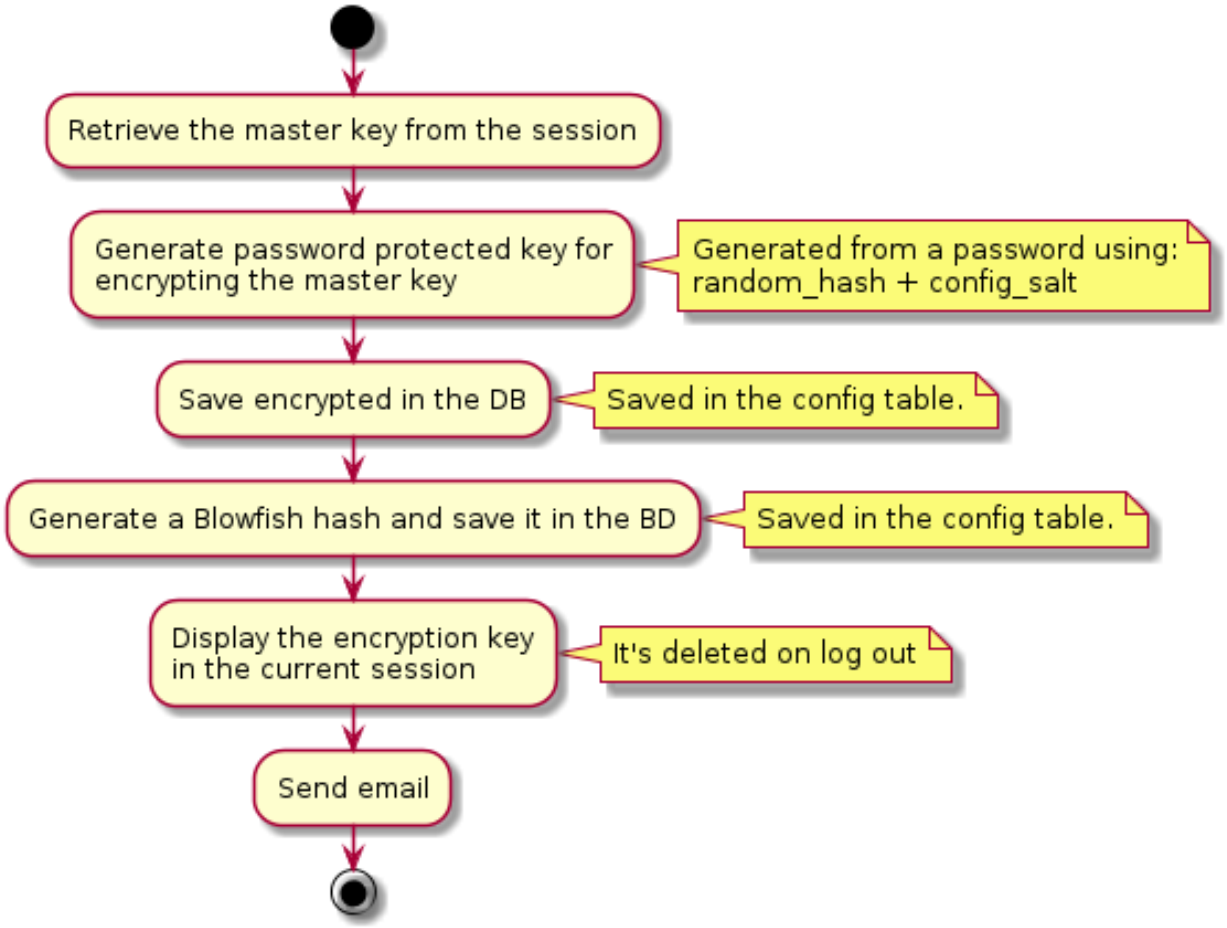
Login Process



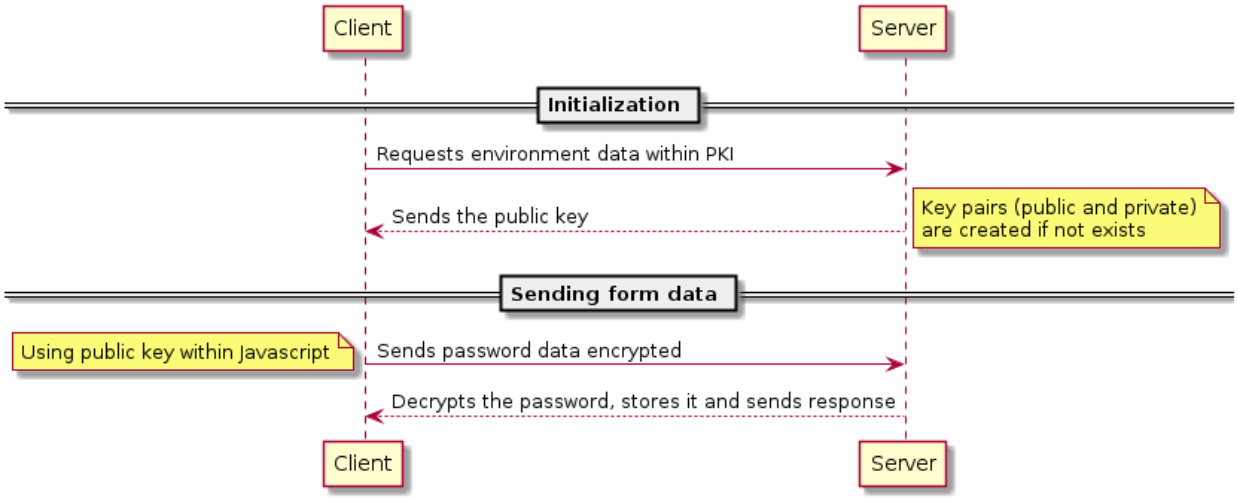
Master Key Process



Temporary Master Key Process



PKI Process



Warning: Be aware that the highest security risk is in the users themselves, because a compromised password could cause a security leak.

A sysPass comprised server could be dangerous if the database is placed with the webserver, because the network data could be sniffed so the passwords would be revealed.

2.3.3 Security

sysPass has some security mechanisms to mitigate some kind of events and actions that could compromise the application security. Among them are:

- Security token generation for sending forms
- Request type checking for every form
- Unwanted characters removing on received data
- Type casting on received data
- Hash generation for export and backup file names.
- PKI encryption is used for sending passwords within forms

Though these actions, it's needed to secure the webserver components and Though these actions, it's needed to secure the webserver components and communications:

- Use of HTTPS
- Limit access to the "config" and "backup" directories

In order to limit the access to the directories through Apache, ".htaccess" files could be used within the directories or by modifying the site configuration:

```
# Apache 2.2
<Directory ~ "/var/www/html/sysPass/(config|backup)">
  <Limit GET HEAD POST>
    Order Deny,Allow
    Deny from all
  </Limit>
</Directory>

# Apache 2.4
<Directory ~ "/var/www/html/sysPass/(config|backup)">
  Require all denied
</Directory>
```

Danger: It's important that the "config" directory is not accessible from the web service, because it could reveal important information.

2.3.4 Authentication

For the sysPass authentication it could be possible to use several methods:

- MySQL/MariaDB database (by default)
- LDAP directory (OpenLDAP, eDirectory, Active Directory, freeIPA, etc)

Note: If the LDAP option is enabled, the database authentication is used when the LDAP service is unavailable or the user doesn't exist.

For the database authentication, a generated Blowfish hash within the user's password is checked, so the password is **never** stored.

If LDAP is used:

- The user's Blowfish generated hash is stored in order to check it if the LDAP service is unavailable.
- Neither the user's login nor name nor email can be modified.

2.3.5 Authorization

For the sysPass authorization it could be possible to use several methods:

- Auth Basic (by default)
- Two Factor 2FA (Authenticator plugin)

The Auth Basic authorization is always being checked, so if the HTTP headers within the user data are being sent, it will be checked whether the sysPass user's login matches with the Auth Basic one.

The 2FA authorization through the Authenticator plugin is done by generating an OTP token from the Google Authenticator application. This authorization could be enabled from the users' preferences.

2.3.6 Permissions

The sysPass permissions are set in the users profiles. By default only accounts searching can be done.

There are 29 permission types:

- **Accounts**
  - Create - allows to add new accounts
  - View - allows to view the accounts' details[1]
  - View Password - allows to view the accounts' password[1]
  - Edit - allows to modify the accounts and its files[1]
  - Edit Password - allows to modify the accounts' password[1]
  - Delete - allows to delete accounts[1]
  - Files - allows to view account's files
  - Share Link - allows to create public links
  - Private - allows to create private accounts
  - Private for Group - allows to create private accounts only for the main group
  - Permissions - allows to view and modify the accounts' permissions[1]
  - Global Search - allows to perform a searching in all the accounts except in the private ones[2]
- **Management**

[1] Only the accounts that the user and its group are granted
[2] When the account access is not granted, it will only be able to perform a "Request for Account Modification"

  - Users - allows full access to the users management[3]
  - Groups - allows full access to the groups management
  - Profiles - allows full access to the profiles management
  - Categories - allows full access to categories management
  - Customers - allows full access to customers management
  - Custom Fields - allows full access to custom fields management
  - API Authorizations - allows full access to API authorizations management
  - Public Links - allows full access to the public links management
  - Accounts - allows full access to accounts management
  - Files- allows full access to files management
  - Tags - allows full access to the tags management
- **Configuration**
  - General - allows full access to the site, accounts, wiki, ldap and email configuration
  - Encryption - allows full access to the master key configuration
  - Backup - allows full access to perform backups[4]
  - Import - allows full access to import XML and CSV files
- **Others**
  - Event Log - allows full access to the event log

ACL

Users and Groups

- User profiles allow to set which actions could be done by the user
- An user can only display or modify accounts if:
  - Is the account's owner
  - Is member of account's primary group
  - Is member of account's secondary groups
  - His main group is listed as a secondary group of the account
  - Is included through a group and the "Secondary Groups Access" option is enabled
- An account can only be modified by either the users or secondary groups if the modification permission, on the account accesses, is enabled
- The private accounts can only be accessed by the owner
- The private accounts for groups can only be accessed by the users of the main group
- Application Admin: allows full access to all the application modules
- Accounts Admin: allows full access to all the accounts except private ones

[3] The "Application Admin" users cannot be modified by other users
[4] Only the "Application Admin" users can download the backup or XML files

API

The API access permissions are complementary to the accounts access permissions, so users and groups ACLs will be applied when an account is either listed or accessed.

Notes

2.3.7 Accounts Searching

The accounts searching performs a query for the entered text within the fields "name", "login", "url" and "notes".

Results filtering could be done by selecting category, customer or tags.

The tag filtering is cumulative ("OR"), so it will be included all the accounts with selected tags.

There are special filters that are entered in the text field:

Filter	Description
user:login	Get the accounts in which the user with login "login" has access
owner:login	Get the accounts in which the user with login "login" is the owner
maingroup:group_name	Get the accounts which have the main group with name "group_name"
group:group_name	Get the accounts in which the group with name "group_name" has access
file:file_name	Get the accounts which contains the file with name "file_name"
expired:	Get the accounts with expired password
private:	Get the private accounts for the current user

2.3.8 API

The sysPass API uses JSON-RPC v2 for messages sharing between the client an server.

The API access URL is "https://server/sysPass/api.php".

Methods

getAccountSearch

Performs an accounts searching

Parameter	Description
authToken	User's API token
text	Text to search for
count	Number of results to display
categoryId	Category Id for filtering
customerId	Customer Id for filtering

getAccountData

Gets an account's details

Parameter	Description
authToken	User's API token
id	Account Id
userPass	User's password that belongs the token

getAccountPassword

Gets an account's password

Parameter	Description
authToken	User's API token
id	Account Id
tokenPass	API token password
details	Get details in the response

addAccount

Adds a new account

Parameter	Description
authToken	User's API token
tokenPass	API token password
name	Account name
categoryId	Category Id
customerId	Customer Id
pass	Password
login	Access user
url	Access URL or IP
notes	Notes for the account

deleteAccount

Deletes an account

Parameter	Description
authToken	User's API token
id	Account Id

getCategories

Performs a categories searching

Parameter	Description
authToken	User's API token
name	Name to search for
count	Number of results to display

addCategory

Adds a new category

Parameter	Description
authToken	User's API token
name	Category name
description	Description

deleteCategory

Deletes a category

Parameter	Description
authToken	User's API token
id	Category Id

getCustomers

Performs a customers searching

Parameter	Description
authToken	User's API token
name	Name to search for
count	Number of results to display

addCustomer

Adds a new customer

Parameter	Description
authToken	User's API token
name	Customer name
description	Description

deleteCustomer

Deletes a customer

Parameter	Description
authToken	User's API token
id	Customer Id

backup

Performs an application backup

Parameter	Description
authToken	User's API token

2.3.9 Functionalities

sysPass implements the following functionalities:

- **Security**
  - Database authentication
  - LDAP directory authentication
  - Auth Basic authorization
  - Two Factor authorization
- **Permissions**
  - Module access control by profiles
  - Application administrator users
  - Accounts administrator users
  - Accounts user access control
  - Accounts group access control
  - Account modify control by users
  - Account modify control by groups
- **Items**
  - Encrypted custom fields for accounts, customers, categories and users
  - Accounts public links access without user/password
  - Accounts expiry date configuration
  - Accounts files management
  - Accounts tags management
  - Customers Management
  - Category management
  - Public links management
  - API authorizations management
  - Accounts management
  - Accounts history management
  - Plugins management
  - Users management
  - Groups management

  - Profiles management
  - In-App notifications management
- **Configuration**
  - Language Configuration
  - Visual theme configuration
  - Proxy Configuration
  - Accounts Configuration
  - Public links configuration
  - Wiki links configuration
  - Links against DokuWiki API configuration
  - LDAP configuration with user importing
  - Email notifications configuration
  - Master Key Change
  - Temporary Master Key Generation
  - Application and database backups
  - XML format exporting
  - Importing from XML, KeePass, KeePassX and CSV formats

2.3.10 Plugins

sysPass allows to use plugins through an architecture that implements observer pattern which is characterized by emitting a message to all subscribed observers.

The plugins are installed in 'sysPass/inc/Plugins' directory and they contain the following basic structure:

```
Plugins/
└── PluginName (1)
    ├── ajax
    ├── js
    │   ├── plugin.js
    │   └── plugin.min.js
    ├── locales
    │   ├── en_US
    │   │   └── LC_MESSAGES
    │   │       ├── NombrePlugin.mo (2)
    │   │       └── NombrePlugin.po (2)
    ├── themes
    │   └── material-blue
    │       └── views
    │           ├── main
    │           │   └── login.inc
    │           └── userpreferences
    │               └── preferences.inc
    │
    └── PluginNamePlugin.class.php (3)
```

The directory and file names need to be set in the following way:

1. Directory name within the plugin name: Example: **Authenticator**
2. Filename within the plugin name in lowercase: Example: **authenticator.po**
3. Filename within the plugin name followed by "Plugin.class.php". Example: **AuthenticatorPlugin.class.php**

The main class need to be named like the plugin and extends **PluginBase** class. This class must implement the following methods.

Methods

init

Method that is called every time the plugin is executed

```
/**
 * Inicialización
 */
public function init() {}
```

updateEvent

Method that is called when an event is emitted

```
/**
 * Evento de actualización
 *
 * @param string $event Nombre del evento
 * @param mixed  $object
 */
public function updateEvent($event, $object) {}
```

getEvents

Method that returns an strings array with the events that the plugin will be subscribed to

```
/**
 * Devuelve los eventos que implementa el observador
 *
 * @return array
 */
public function getEvents()
{
    return ['user.preferences', 'main.prelogin.2fa', 'login.preferences'];
}
```

getJsResources

Method that returns an strings array with the Javascript resources required by the plugin

```
/**
 * Devuelve los recursos JS y CSS necesarios para el plugin
 *
 * @return array
 */
public function getJsResources()
{
    return ['plugin.min.js'];
}
```

getAuthor

Method that returns the plugin author

```
/**
 * Devuelve el autor del plugin
 *
 * @return string
 */
public function getAuthor()
{
    return 'Rubén D.';
}
```

getVersion

Method that returns an integers array with the plugin version

```
/**
 * Devuelve la versión del plugin
 *
 * @return array
 */
public function getVersion()
{
    return [1, 0];
}
```

getCompatibleVersion

Method that returns an integers array with the minimum sysPass compatible version

```
/**
 * Devuelve la versión compatible de sysPass
 *
 * @return array
 */
public function getCompatibleVersion()
{
    return [2, 0];
}
```

getCssResources

Method that returns an strings array with the CSS resources required by the plugin

```
/**
 * Devuelve los recursos CSS necesarios para el plugin
 *
 * @return array
 */
public function getCssResources()
{
    return [];
}
```

getName

Method that returns the plugin name

```
/**
 * Devuelve el nombre del plugin
 *
 * @return string
 */
public function getName()
{
    return self::PLUGIN_NAME;
}
```

getData

Method that returns the plugin data

```
/**
 * @return array|AuthenticatorData[]
 */
public function getData()
{
    return (array)parent::getData();
}
```

Example

```
<?php

namespace Plugins\Authenticator;

use SP\Core\DiFactory;
use SP\Core\Plugin\PluginBase;
use SplSubject;

/**
* Class Plugin
```

(continues on next page)

(continued from previous page)

```
*
* @package Plugins\Authenticator
*/
class AuthenticatorPlugin extends PluginBase
{
  const PLUGIN_NAME = 'Authenticator';

  /**
   * Receive update from subject
   *
   * @link  http://php.net/manual/en/splobserver.update.php
   * @param SplSubject $subject <p>
   *                            The <b>SplSubject</b> notifying the observer of an␣
↪update.
   *                            </p>
   * @return void
   * @since 5.1.0
   */
  public function update(SplSubject $subject)
  {
  }

  /**
   * Inicialización del plugin
   */
  public function init()
  {
      if (!is_array($this->data)) {
          $this->data = [];
      }

      $this->base = __DIR__;
      $this->themeDir = __DIR__ . DIRECTORY_SEPARATOR . 'themes' . DIRECTORY_
↪SEPARATOR . DiFactory::getTheme()->getThemeName();

      $this->setLocales();
  }

  /**
   * Evento de actualización
   *
   * @param string $event Nombre del evento
   * @param mixed  $object
   * @throws \SP\Core\Exceptions\FileNotFoundException
   * @throws \SP\Core\Exceptions\SPException
   */
  public function updateEvent($event, $object)
  {
      switch ($event){
          case 'user.preferences':
              $Controller = new PreferencesController($object, $this);
              $Controller->getSecurityTab();
              break;
          case 'main.prelogin.2fa':
              $Controller = new LoginController($this);
              $Controller->get2FA($object);
              break;
```

(continues on next page)

(continued from previous page)

```
            case 'login.preferences':
                $Controller = new LoginController($this);
                $Controller->checkLogin();
                break;
        }
    }

    /**
     * Devuelve los eventos que implementa el observador
     *
     * @return array
     */
    public function getEvents()
    {
        return ['user.preferences', 'main.prelogin.2fa', 'login.preferences'];
    }

    /**
     * Devuelve los recursos JS y CSS necesarios para el plugin
     *
     * @return array
     */
    public function getJsResources()
    {
        return ['plugin.min.js'];
    }

    /**
     * Devuelve el autor del plugin
     *
     * @return string
     */
    public function getAuthor()
    {
        return 'Rubén D.';
    }

    /**
     * Devuelve la versión del plugin
     *
     * @return array
     */
    public function getVersion()
    {
        return [1, 0];
    }

    /**
     * Devuelve la versión compatible de sysPass
     *
     * @return array
     */
    public function getCompatibleVersion()
    {
        return [2, 0];
    }
```

(continues on next page)

(continued from previous page)

```
    /**
     * Devuelve los recursos CSS necesarios para el plugin
     *
     * @return array
     */
    public function getCssResources()
    {
        return [];
    }

    /**
     * Devuelve el nombre del plugin
     *
     * @return string
     */
    public function getName()
    {
        return self::PLUGIN_NAME;
    }

    /**
     * @return array|AuthenticatorData[]
     */
    public function getData()
    {
        return (array)parent::getData();
    }
}
```

Events

When an event is emitted the generating class instance is included as an argument, so it could be possible to access to the class events.

Currently, the generated events are the following:

Event	Class	Description
show.account.new	AccountController	Generated when new account view is displayed
show.account.copy	AccountController	Generated when copy account view is displayed
show.account.edit	AccountController	Generated when account edit view is displayed
show.account.editpass	AccountController	Generated when account's password edit view is displayed
show.account.view	AccountController	Generated when account details view is displayed
show.account.viewhistory	AccountController	Generated when account's history view is displayed
show.account.delete	AccountController	Generated when account delete view is displayed
show.account.request	AccountController	Generated when request for account modification view is displayed
show.account.search	AccountSearchController	Generated when accounts searching view is displayed
show.config	ConfigController	Generated when configuration view is displayed
show.eventlog	EventlogController	Generated when event log view is displayed
show.itemlist.accounts	ItemListController	Generated when items and customizations view is displayed
show.itemlist.accesses	ItemListController	Generated when accesses view is displayed
show.itemlist.notices	ItemListController	Generated when notifications view is displayed
login.preferences	LoginController	Generated when preferences are loaded on login
main.prelogin.*	MainController	Generated when an action before login is performed (from URL)
main.postlogin.*	MainController	Generated when an action after login is performed (from URL)

2.4 Updating

2.4.1 General

For the sysPass updating the following steps are needed:

1. Download the application from https://github.com/nuxsmin/sysPass/releases and uncompress the files
2. Set the sysPass directory owner and permissions
3. Copy the files ("config.xml", "key.pem" y "pubkey.pem") within the "config" directory from the current version to the new one
4. Open the application from a web browser

If the application requires a database upgrade:

1. **Perform a database backup**
2. Enter the updating code which could be found in the "config/config.xml" file within the tag "upgradeKey"

Note: After the updating, it will show a message and you could take a look to the updating details in the event log

2.4.2 2.1 Version

This version includes some improvements on the sysPass security by the following features:

- It uses Defuse/php-encryption library for the data encryption with OpenSSL by using AES-256 CTR (CVE-2017-5999)

- Improvements on the session keys security
- API authorizations password
- Improvements on the public links security
- Failed log in attempts detection. A delay is set after several attempts

This upgrade requires to re-encrypt all the accounts and encrypted data, so the master password and a valid user login (for registering changes) will be needed.

Though it's a safe process, it's advisable to make a full sysPass backup.

Important Changes

Because the encryption data changes, the following items need to be regenerated:

- Public links: the links are now an snapshot of the linked account, so if the account is updated, the link needs to be renewed.
- API authorizations: As of this version, a password is needed for those authorizations that require encrypted data.
- Temporary master password: it needs to be regenerated if it's being used.

Process

For the sysPass updating the following steps are needed:

1. Download the application from https://github.com/nuxsmin/sysPass/releases and uncompress the files
2. Set the sysPass directory owner and permissions
3. Copy the files ("config.xml", "key.pem" y "pubkey.pem") within the "config" directory from the current version to the new one
4. Open the application from a web browser

If the application requires a database upgrade:

1. **Perform a database backup**
2. Enter the updating code which could be found in the "config/config.xml" file within the tag "upgradeKey"
3. Please, enter the sysPass master password.
4. Please, enter a valid user login

Note: During the upgrade, it will display the encryption tasks processes.

Note: After the updating, it will show a message and you could take a look to the updating details in the event log

---

2.5 HOWTOs

2.5.1 How to test a sysPass update

Note: This procedure tells the steps to follow to try out a sysPass update without modifying the current installation

1. Make a database backup. It could be made either through the sysPass utility, MySQL workbench or mysqldump tool
2. Create a new database (eg. syspass21)
3. Create an user (eg. sp_admin21) and set the permissions over the newly created database
4. Import the backup in the newly created database. You could use the above tools
5. Create a new directory and unpack the new sysPass version package[1]
6. Copy all files within the "config" directory to the new path and check out the permissions[1]
7. Modify the "config/config.xml" file to set the correct database connection parameters ("dbname", "dbuser" and "dbpass"). Please check out that "dbHost" is correct
8. Point the browser to the application URL and follow the steps for upgrading

Notes

2.5.2 How to restore sysPass

Note: This procedure requires to have a database and application backup

1. Restore the database backup. It could be made either through the sysPass utility, MySQL workbench or mysqldump tool
2. Create the connection user (see "config.xml" file) and set the correct permissions over the restored database
3. Restore the application backup
4. Point the browser to the application URL

2.6 Frequently Asked Questions

2.6.1 What is sysPass?

sysPass is a password manager that allows to save passwords using bidirectional encryption with a master password to a database. Passwords are associated to accounts, and these have detailed information about it like: customer, category, notes, files, etc.

The initial idea was to make servers and services passwords accesible in a multiuser environment with security applied and make a portable bundle to store on a flash drive.

2.6.2 Where can I install sysPass?

The application can be installed on any system that has Apache, PHP and MySQL installed.

[1] See *Installation* for more details

2.6.3 How do I install sysPass

You can download the application from https://github.com/nuxsmin/sysPass/releases/latest and follow steps on *Installation*

2.6.4 Which authentication methods are used?

sysPass uses MySQL/MariaDB or LDAP as authentication backends.

If LDAP is used and it is for some reason not possible to connect to the configured LDAP server, it will use MySQL as backend. In this case, user login data will be the last used on user login by LDAP.

More information on: *Authentication*

2.6.5 What is the encryption for?

The database passwords encryption allows that in case of anyone get access to the database or a data exporting is performed, it won't be readable without the master key.

This solution is very convenient when you run the application from a flash drive, because if you lose it, the information is secured.

The encryption schema used is rijndael-256 in CBC mode.

More information on: *Encryption*

2.6.6 What is portable?

It means that you can run the application without really installing it.

This application can be portable by installing Apache, PHP and MySQL on a flash drive. You can use any available LAMP bundles like WAMP, XAMPP, etc.

The backup tool allows you to make a backup of whole the environment (application and database) for example to store it on a flash drive or put it somewhere safe as a backup.

2.6.7 Is there a master password for each account/user?

The master password is global for all accounts and users.

Each time a user is added, his personal password is changed or the master password is reset, the user needs to enter the master password on the next session login.

Each time the master password is changed, the users that are logged in, will only be able to view accounts details, until the new password is entered.

More information on: *Encryption*

2.6.8 What are Wiki links?

It allows you to link the accounts with a name pattern to an external Wiki that allow to pass the account name as a parameter in the URL.

There are two types of links, the one that links to a Wiki search page (and in which the account name is passed as a parameter), and the other that links to the account page in the Wiki.

2.6.9 What are categories?

Its goal is to classify the accounts to make more precise searches.

2.6.10 What are user groups?

These groups are used to give users access to accounts that have a certain group set as primary or secondary group

2.6.11 What is customer field?

Like categories, it is possible to do searches based on the customer. This field can be treated generically as department, company, division, etc..

En futuras versiones se podrán asociar usuarios a clientes.

2.6.12 Is there an account history?

Yes, each time an account is modified or deleted, the application saves a copy of the last state.

You can switch to a history point at account details page. If the master password that was used to save account history point differs from current, the password won't be shown.

2.6.13 What are profiles?

Profiles are used to define actions that the users can do.

There are 16 access levels that can be activated and it allows to define which modules can be accessed by the users in which are defined.

2.6.14 What is maintenance mode?

This mode is used to disable the users to log in to the application while you are doing operations on database, updating, etc.

The user that enables the maintenance mode, will be the only one that can use the application until a session log out. After that it will be needed to disable it in the "config/config.xml" file within the tag "maintenance"

2.6.15 Can I change Master Password?

Yes, you need to know the current one. It's advisable to make a database backup before this process.

2.6.16 I don't remember Master Password, can I decrypt the passwords?

No, it's not possible view the passwords without the Master Password.

2.6.17 Does backup runs on Windows?

Yes, it uses the PHP PHAR library to get it working.

2.6.18 The language doesn't change

Please take a look to the locales installed on your system (server), because sysPass uses the GNU gettext system for internationalization.

The installed locales should be on the UTF-8 variant.