ManPKI Documentation

Release 1.0.dev3

Gaëtan FEREZ

Sep 27, 2017

Contents

X.509 PKI Manager Daemon

Github

CHAPTER 1

Quickstart

First, *Install manpki*.

ManPKI Daemon deliver an api to manage X.509 PKI. ManPKI implement root authority, subauthority, extension and profiles from X.509 Standard. Modules can be integrated in this daemon like ldap or ocsp to extend functions (example : ldap integration, ocsp responder).

By default, manpkid run only on local using unix socket. Next, it can be configured to be network accessible. The package manpki-cli provide a shell to connect to manpkid daemon using local socket or remote connection.

CHAPTER 2

Functionnality

X.509 Implementation

ManPKI API implement function to manage a X.509 PKI. Root CA and Sub CA can be managed by deploying multiple daemon (one daemon by certificate authority).

Module extension

The API can be extended with the installation of new python module. For example, a module can installed to map certificate to ldap directory or to implement OCSP responder

PAM Authentication

All users must be authenticated to the daemon before interaction with it. The API authentication are based on PAM authentication.

TLS Web Server

The daemon implement directly his TLS Web Server. Another web server or proxy are not necessary.

JOSE Signature

All request are based on JSON and all json message are signed by JOSE. The key are unique for each session.

Access rights

The API access are based on some basic groups.

Base Group	Description
`user`	Can only request certificate and show information
`ra`	Can validate request certificate
`ca`	Can manage all the Certificate Authority
`admin`	Can manage the CA and the API Web Server

Events

Some event can be fire by the daemon for other module registered. For example, the daemon fire an event when the ca are created. This event can be listen by a manpki module to insert the certificate in ldap directory

CHAPTER 3

Installation and Configuration

From git

To install manpki from git, download master.zip from github/GaetanF/manpki.git or clone the repository :

```
$ git clone https://github.com/GaetanF/manpki.git
```

You need to install all dependencies needed by the program present in requirements.txt :

```
$ make deps
```

And install ManPKI :

```
$ make install
```

Configuration

ManPKI need some folder before running. If you have make tool can directly use it to correctly configure the structure.

```
$ manpkid --init
```

File structure is define below :

Directory	Description
VARDIR/cert	Contain all files related to the PKI (cert, ca, crl, privatekey)
VARDIR/cert/public	Contain all certificates (cert and ca)
VARDIR/cert/private	Contain all private key (cert and ca)
VARDIR/db	Contain the manpki database formated in JSON
CFGDIR	Contain manpki.conf
LOGDIR	All logs created by ManPKI

You need to configure your personal account to have admin role in the application.

```
$ tools/manageUser.py -a -u $USER -g admin
```

ManPKI daemon can be started directly using manpkid executable or by init scripts

```
$ manpkid -d
```

The main executable have some arguments :

```
$ manpkid -h
usage: manpkid [-h] [-v] [-D] [-l LOGFILE] [-d] [-i]

ManPKI daemon.

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show version
  -D, --debug           debug mode
  -l LOGFILE, --logfile LOGFILE
                        log file
  -d, --daemon          daemon
  -i, --init            initialize manpki
```

CHAPTER 4

CLI

ManPKI API have his own client named manpki-cli

Installation

From git

To install manpki from git, download master.zip from github/GaetanF/manpki.git or clone the repository :

```
$ git clone https://github.com/GaetanF/manpki-cli.git
```

You need to install all dependencies needed by the program present in requirements.txt :

```
$ make deps
```

And install ManPKI :

```
$ make install
```

Usage

To launch the shell :

```
$ manpki shell
```

The main executable have some arguments :

```
$ manpki -h
usage: manpki [COMMAND]
```

```
available commands:
  service (not available)
  check   (not available)
  queue   (not available)
  shell

Try manpki help [COMMAND]
```

When you launch the shell utility, it’s start in disconnected mode. You need to connect to your ManPKI daemon locally or remotely if daemon is configured to allow the remote access.

```
$ manpki shell
Welcome to the ManPKI shell !
[disconnected manpki-cli]$ connect
[ferezgaetan@local manpki-cli]$ help
```

CHAPTER 5

API

Definition

ManPKI Daemon are only accessible by the API. The daemon directly implement a secured and authenticated web API.

Only user who have a local account on the server running the daemon can authenticated on the daemon.

API method accessible are describre bellow.

URL	Description
`/v1.0/ca`	CA management
`/v1.0/cert`	Certificate management
`/v1.0/extension`	SSL Extension management
`/v1.0/profile`	Profile based on SSL Extension management
`/v1.0/server`	API Server management
`/ping`	Ping/Pong daemon
`/discovery`	Discover all available method for current user
`/info`	Get some information of current session
`/login`	Log in the application
`/logout`	Disconnect the user
`/locale/<lang>`	Get all locale for the specified language
`/render`	Get render system for the CLI

Global

Resource	Operation	Description
	GET /v1.0/server/restart	
	POST /v1.0/server/set	
	POST /v1.0/cert/set	
	POST /v1.0/ca/param	
	GET /v1.0/server	
	PUT /v1.0/cert	
	GET /v1.0/ca	
	PUT /v1.0/ca	
	PUT /v1.0/ca	
	GET /v1.0/cert/param/	
	GET /v1.0/cert/param/(param)	
	GET /v1.0/ca/param/	
	GET /v1.0/ca/param/(param)	
	GET /v1.0/extension/	
	GET /v1.0/extension/(oid)	
	POST /v1.0/extension/(oid)	
	PUT /v1.0/extension/(oid)	
	DELETE /v1.0/extension/(oid)	
	GET /v1.0/profile/	
	GET /v1.0/profile/(profileid)	
	POST /v1.0/profile/(profileid)	
	PUT /v1.0/profile/(profileid)	
	DELETE /v1.0/profile/(profileid)	
	GET /v1.0/cert/	
	GET /v1.0/cert/(certid)	

CA

POST /v1.0/ca/param

Set parameter to the CA

Return boolean if parameter are correctly set

GET /v1.0/ca

Show CA Information

Shell show ca

Context None

Return ca information

PUT /v1.0/ca

PUT /v1.0/ca

Create a CA

Param force if present force the creation of the ca even if already exist

Shell create

Context ca

Return json info about the new ca

GET /v1.0/ca/param/

GET /v1.0/ca/param/(*param*)
Get specifed or all parameter of the CA

Return json info about parameters of the ca

Cert

POST /v1.0/cert/set
Set cert element

Param basecn Base CN of the next certificate

Param email Email for the next certificate

Shell set cert

Context cert

Return information if element are correctly set

PUT /v1.0/cert
Create new certificate

Param cn CN of the certificate

Param mail Email for the certificate

Param profile SSL Profile

Shell create

Context cert

Return information of the new certificate

GET /v1.0/cert/param/

GET /v1.0/cert/param/(*param*)
Get certificate parameter

Param param Specific parameter

Shell show cert param

Context None

Return information of the certificate parameter

GET /v1.0/cert/

GET /v1.0/cert/(*certid*)
Show all cert or specific cert information

Param certid Certificate Identifier

Shell show cert

Context none

Return ca information

Profile

GET /v1.0/profile/

GET /v1.0/profile/(*profileid*)

Show all or specific SSL Profile

Param profileid ID of the profile

Shell show profile

Context None

Return information of the profile

POST /v1.0/profile/(*profileid*)

Set profile

Param profileid ID of the profile

Shell set profile

Context profile

Return information of the profile

PUT /v1.0/profile/(*profileid*)

Add a new profile

Param profileid ID of the profile

Shell add profile

Context profile

Return information of the profile

DELETE /v1.0/profile/(*profileid*)

Delete a profile

Param profileid ID of the profile

Shell delete profile

Context profile

Return message about the profile deletion

Extension

GET /v1.0/extension/

GET /v1.0/extension/(*oid*)

Show all or specific SSL Extension

Param oid OID of the extension

Shell show extension

Context None

Return information of the extension

POST /v1.0/extension/(*oid*)

Set an extension

Param oid OID of the extension

Shell set extension

Context extension

Return information of the extension

`PUT /v1.0/extension/`(*oid*)

Add a new extension

Param oid OID of the extension

Shell add extension

Context extension

Return information of the extension

`DELETE /v1.0/extension/`(*oid*)

Delete an extension

Param oid OID of the extension

Shell delete extension

Context extension

Return message about the deletion

Server

`GET /v1.0/server/restart`

Restart Web API Server

Shell reload server

Context server

`POST /v1.0/server/set`

Set server parameter

Param host Host to listen for Web API Server (socket: to listen on unix socket, ip: to listen on ip address)

Param port Port to listen

Param cert Path or CertID of the certificate

Param key Path or CertID of the certificate

Shell set

Context server

Return message if parameter are correctly set.

`GET /v1.0/server`

Set information about the server

Shell show server

Context None

Return all information about the server.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

HTTP Routing Table

/v1.0