PyPPP

Release 1.0

February 09, 2016

Contents

CHAPTER 1

PyPPP

PyPPP is a python library which implements PPP (Perfect Paper Passwords) which is a single-use "passcode" system design to go into multi-factor authentication systems. It also contains a authentication system which works with the Django web framework.

1.1 First steps

- *Download*
- *Installation*
- *Using PyPPP*
- *Using PyPPP with Django*

1.2 Getting Help

If you have any problems with PyPPP, or have any suggestions you can contact kylef.

CHAPTER 2

Quick install guide

2.1 Directly from the git repository

This is the preferred way to install PyPPP, but it does require the git source code repository system to be installed on your computer, although this is the preferred way, if you do not already have git. It will be easier to install via the other options. Some of the commands used in this are only available on UNIX-alike systems, it may not work on windows.

It is very easy to install PyPPP via git, first you need to open a terminal. Then type the following after first using `cd` command to move into the directory you wish to install PyPPP in. I use the location: `/home/kylef/git/` to install PyPPP in.

```
$ git clone git://github.com/kylef/pyppp.git
$ ln -s pyppp/pyppp <PYTHONPATH>
```

2.2 Using a package-management tool

This is one of the easiest ways to install PyPPP. There are two different package-management tools which you could use:

2.2.1 pip

pip is one of the more popular package-management systems for python. You can find documentation, and how to install pip itself here. Once you have pip installed and running, simply type:

```
$ pip install pyppp
```

2.2.2 easy_install

Another option is to use easy_install, first you need to install easy_install. You can find documentation and how to install easy_install here. Once you have easy_install up and running, just type:

```
$ easy_install pyppp
```

2.3 Manually installing

To manually install PyPPP, you will first need to download PyPPP (zip, or tar.gz). Once you have a copy of PyPPP, open it and run:

```
$ python setup.py install
```

This will install pyppp and will require administrative privileges on your computer as it is a system-wide install.

CHAPTER 3

How to get PyPPP

PyPPP is available under the BSD license.

3.1 Get the latest version

3.1.1 Via tar.gz / zip archive

You can download the latest version in a archive created directly from the git repository via the following links:

- tar.gz: http://github.com/kylef/pyppp/tarball/master
- zip: http://github.com/kylef/pyppp/zipball/master

3.1.2 Via git repository

Git is one of the best ways to download pyppp, it allows you to easily pull the latest version of pypp, just by typing `git pull` within the pyppp directory. It will also allow you to download the complete history of pyppp's source code, to download git or how to use it. See: http://git-scm.com/

You can use the following command to clone the git repository:

```
$ git clone git://github.com/kylef/pyppp.git
```

CHAPTER 4

Using PyPPP

PyPPP is very easy to use, everything you need is wrapped in an object, which you either pass the AES key to at the start, or you can tell it to generate a random key for you.

```
from pyppp import PyPPP
p = PyPPP('8B2C8B6781D72852D8A4485425774794896B95A98526054EC79F39D106A6D82F')
passcode = p.retrieve_passcode(0) # To retrieve the first passcode
card = p.retrieve_card(1) # To retrieve the first card (returns a list)
```

4.1 pyppp.PyPPP Class

4.1.1 __init__

The init method optionally takes a key, if you do not pass the object a key here, you will have to use `generate_random_sequence_key` to generate a key.

This method expects the key to be in hex and 64 characters long.

```
from pyppp import PyPPP
p = PyPPP('8B2C8B6781D72852D8A4485425774794896B95A98526054EC79F39D106A6D82F')
```

4.1.2 generate_random_sequence_key

The `generate_random_sequence_key` is used to generate a sequence key, depending on your python version and operating system, this may be done differently.

If you are using python >2.4, the random_sequence key will be generated from the operating system, on UNIX-like system's this will use `/dev/urandom`, on Windows it will use `CryptGenRandom`. If a randomness source is not found, python's random.randint will be used, random.randint is completely deterministic, it is not recommended. If you do not have os.urandom, you should upgrade to a later python and/or operating system. Or change the source code to generate a random number.

```
>>> from pyppp import PyPPP
>>> p = PyPPP()
>>> p.generate_random_sequence_key()
>>> print p.key()
    '6c065d008eeaa624e529d83aa02a53df46fdb553870f5bec5ee9b081f69e6ded'
```

4.1.3 retrieve_passcode

This method allows you to get the passcode from a count, for example:

```
>>> print p.retrieve_passcode(0)
    B%UX
```

4.1.4 retrieve_passcodes

This method allows you to get a range passcodes, for example:

```
>>> print p.retrieve_passcodes(0, 4) # Retrieve 5 passcodes (0 to 4)
    ['B%UX', 'rUqx', 'v4d!', '#kAu', 'cx2u']
```

4.1.5 retrieve_card

This method allows you to get a range passcodes, for example:

```
>>> print p.retrieve_card(1)
    ['B%UX', 'rUqx', 'v4d!', '#kAu', ..., 'N5bX', 'Li4N', 'LLw#', 'GHA+']
```

4.1.6 get_sequence_info

The easiest way to get information on a sequence is using `get_sequence_info`

```
>>> info = p.get_sequence_info(7234)
>>> print info['card']
    104
>>> print info['column']
    D
>>> print info['row']
    4
```

CHAPTER 5

Using PyPPP with a Django Project

Once you've installed PyPPP and want to use it in your Django applications, do the following;

1. Add 'pyppp.django' to the 'INSTALLED_APPS' setting of your Django project.
2. Add 'pyppp.django.backends.PPPBackend' to AUTHENTICATION_BACKENDS in your project's settings.py:

```
AUTHENTICATION_BACKENDS = (
        'django.contrib.auth.backends.ModelBackend',
        'pyppp.django.backends.PPPBackend',
)
```

3. Change 'LOGIN_URL' to '/ppp/login/' to your settings.py.
4. Change 'LOGOUT_URL' to '/ppp/logout/' to your settings.py.
5. Run manage.py syncdb so that Django will create the post tables.
6. Add the URLs to your project's urls.py:

```
urlpatterns = patterns('',
        ...
        url(r'^ppp/', include('pyppp.django.urls')),
        ...
)
```

Once you have installed and added PyPPP to your Django project, you can use the url /ppp/card/ to grab the current card. Login using PPP at: /ppp/login/, and logout at /ppp/logout/.