pyPKI Installation & Configuration

Introduction

Follow instructions in Establish multi-level OpenSSL Public Key Infrastructure Assumptions:
- pki user exists
- pki homedirectory is in /home/pki
- pki root is in /home/pki/pki-root

Install pyPKI

Clone GIT into directory:

```
cd /home/pki
git clone https://dverslegers@bitbucket.org/dverslegers/pypki.git
```

> Make sure the git utility is available on the system before attempting to use this command.

Install pyPKI and it's dependencies:

```
pip install pypki
```

> Make sure pip is available on your system. If this is not yet the case you can add it by issuing sudo apt-get install python-pip

Configure pyPKI

Basic configuration

pyPKI only requires very little configuration parameters to work:
- root directory for the pki (in our example pki-root)
- location of openssl.cnf configuration file (in our example pki-root/openssl.cnf)
- Names of the Certificate Authorities which need to be supported by the tool (in our example RootCA and IntermCA).

> Make sure the names correspond to the names of the CA sections in the openssl.cnf file.

These configuration parameters should be set in pypki/config/pki.cfg
In addition pyPKI requires the addition of three parameters for each CA in the openssl.cnf file. These options determine wether or not to use smartcards, if so which slot to use on the smartcard and the location of the certificate authority chain file (see Establish multi-level OpenSSL Public Key Infrastructure doc). Therefore make the following changes to each CA section in the openssl.cnf file:

```
# Section for pyPKI
use_smartcard = False
smartcard_slot = 0:2
chain_file = $dir/ca_chain.crt
```

Ensure you generated the ca chain file before using pyPKI.

As of version 1.1 pypki will copy the commom name value in the SAN field of the generated certificates to avoid issues like "NET::ERR_CERT_COMMON_NAME_INVALID" (missing_subjectAltName)". In order to enable this please add the following line to your interim CA section inside the openssl.cnf file:

```
copy_extensions = copy
```

Register users

Users with access to pyPKI are stored in pypki/core/users.py. You can edit this file to reflect the users and passwords which you would like to grant access. The default user and password is admin:admin.

Generate new server certificate

pyPKI uses SSL to secure communications between the browser and the application. A self- signed certificate is included in package so that pyPKI would be able to run out of the box. It is however recommended to follow the steps mentioned below to generate a new private key and renew the certificate used by pyPKI:

```
openssl genrsa -nodes -out pkiweb.key 2048
openssl x509 -req -days 365 -in pkiweb.csr -signkey pkiweb.key -out pkiweb.crt
```

> These commands should be executed in the pypki base directory.

Running pyPKI

Starting pyPKI is as simple as typing:

```
python ./pki_web.py ip:port
```

pyPKI will listen on port 8080 by default in case ip:port is ommitted
Go ahead and browse to https://localhost:8080 to access pyPKI, the default login is admin:admin.

Enabling smartcard fo CA private key storage

Follow the instructions available at my blog on how to enable a yubikey with PIV applet to store your Certificate Authority private key. Once the required modifications have been performed it suffices to change the user smartcard and smartcard slot options in openssl.cnf for the relevant CA('s):

```
# Section for pyPKI
use_smartcard = True
smartcard_slot = 0:2
```

Make sure to refer to the correct smartcard and slot when specifying the smartcard slot parameter.