Passwords, keys, certificates

- KeePassXC
  - keepassxc-cli with keyfile and YubiKey hardware token

KeePassXC

Cross-Platform Password Manager

keepassxc-cli with keyfile and YubiKey hardware token

Install

```
https://keepassxc.org/download/#macos
```

Make alias (for Macintosh)

```
vi ~/.zshrc
```

```
alias kp='/Applications/KeePassXC.app/Contents/MacOS/keepassxc-cli'
```

If you have YubiKey Manager installed, you can see the serial number of connected token from CLI. Otherwise, observe from KeePassXC GUI.

```
/Applications/YubiKey\ Manager.app/Contents/MacOS/ykman info
```

Write a script to provide credentials

```
cd
vi kpopen
```

```
# Do not use backwards slashes in the path, as spaces are already in the double quotes!
export kdbx_path="/path/to/db.kdbx"
```

```
export kdbx_key="/path/to/db.kdbx.key"
# YubiKey slot:serial
export kdbx_token="2:22xxx984"
echo "Enter kdbx password:"
read -s kdbx_pass
```

Load credentials and give password

Set variables

```
source kpopen
```

Check variables are set:

```
set | grep kdbx
```

Putting all together:

Show entry named 'entry'

```
echo "${kdbx_pass}" | kp show -y ${kdbx_token} ${kdbx_path} --key-file ${kdbx_key} entry
```

Show password attribute of 'entry'

```
echo "${kdbx_pass}" | kp show -a password -y ${kdbx_token} ${kdbx_path} --key-file ${kdbx_key} entry
```

Making it more simple

```
vi ~/.zshrc
```

Add another alias:

```
alias kpsh='echo "${kdbx_pass}" | kp show -a password -y ${kdbx_token} ${kdbx_path} --key-file ${kdbx_key} $1'
```

Relaunch terminal, load credentials and request for pass of 'entry'

```
source kpopen
kpsh entry
```

-"It is easy to make things difficult. It is diffucult to make things easy."

Security considerations

- Pass remains within launched shell and is in the memory
- Unset pass when not in use or close terminal to kill the shell and its memory (variables)

```
unset kdbx_pass
set | grep kdbx
```