

Mot de passe sécurisé

Certaines applications ou sites Ouebs proposent un mot de passe du genre "y3dF+Nk7" qui est, à première vue, bien sous tous rapports avec des majuscules, des chiffres, un signe et au moins 8 caractères.

Attention au piège de certains pirates. Ce mot de passe fait peut être partie d'une liste de mots de passe prédéfinis. La liste est parfois étendue sur plus de 10 000 mots de passe (figure 1). Ceux proposés paraissent avoir été générés de façon aléatoire alors qu'ils ne sont tirés qu'aléatoirement de cette liste.

Petite différence mais qui fait toute la différence en matière de sécurité. Ainsi il est facile et rapide un pirate de rejouer la liste entière et de réussir à se connecter. La conduite à tenir est de systématiquement remplacer le mot de passe proposé par un propre à soi. Ce dernier doit répondre à des critères strict pour être suffisamment efficace en cas d'attaque.

C'est à dire, contenir des chiffres, des majuscules, des signes, avoir au moins 8 caractères, être différents les uns des autres sont autant de critères communément acceptés pour définir un mot de passe efficace mais contraignant lorsqu'il faut sortir un mot de passe de suite pour une inscription en ligne. La solution est de s'en remettre à un logiciel libre. Ces derniers ont l'énorme avantage de publier leur code source et donc par la même de faire la preuve de leur innocuité. Malheureusement, ils sont seulement disponibles en ligne de commande avec le Terminal.

Qu'à cela ne tienne, nous allons construire un générateur et profiter de nos acquis avec XCode et AppleScript pour lui construire une interface Mac.

1) Le générateur PWGEN

Mon choix s'est porté sur "pwgen" (<http://sourceforge.net/projects/pwgen>) en licence libre GPL (<http://www.gnu.org/licenses>)

Télécharger le fichier suivant (ou plus récent) sur le bureau du Mac :

<http://surfnet.dl.sourceforge.net/sourceforge/pwgen/pwgen-2.06.tar.gz>

Ouvrir le Terminal (Applications/Utilitaires) et taper les commandes suivantes : (le \$ n'est pas à saisir, il reflète l'invite de commande du Terminal)

```
$ tar xzvf ~/Desktop/pwgen-2.06.tar.gz
```

```
$ cd pwgen-2.06
```

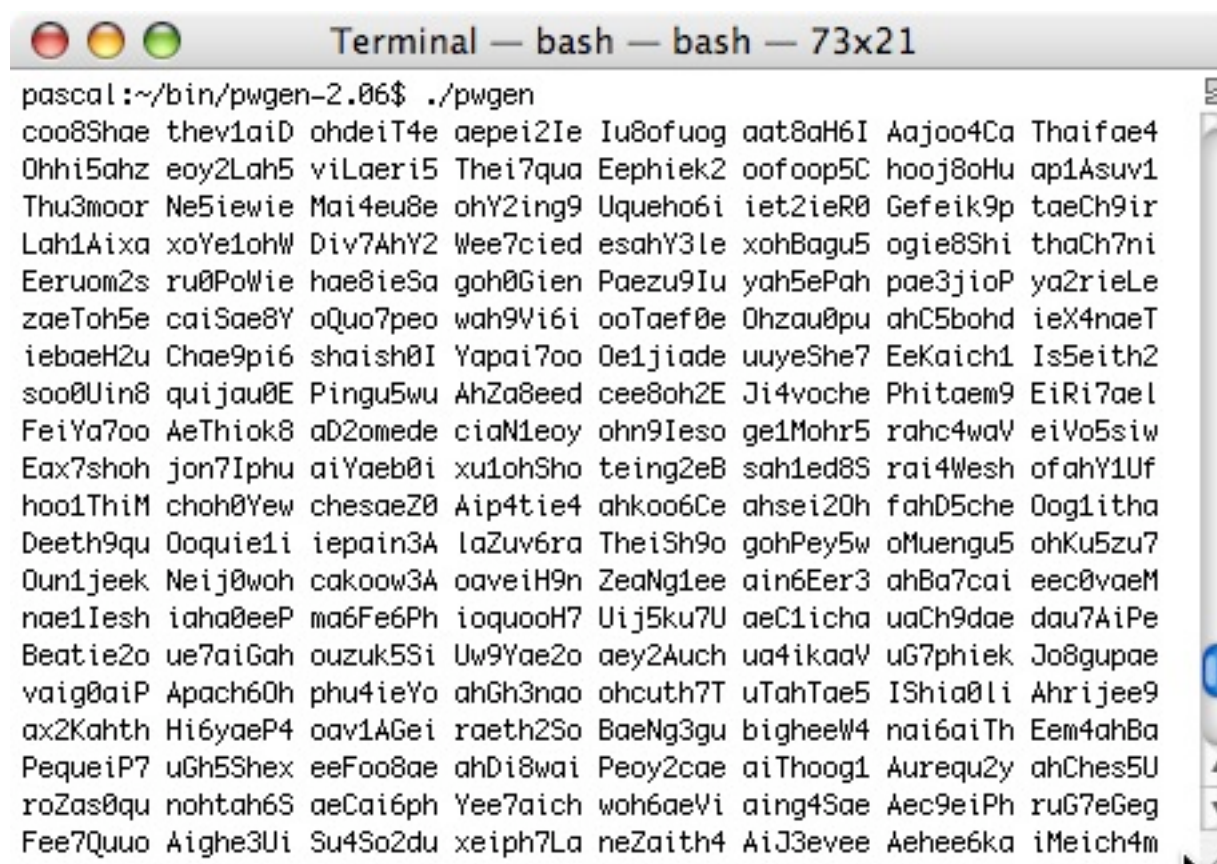
```
$ ./configure
```

```
$ make
```

```
$ ./pwgen
```

Ces manipulations sont en grandes parties automatiques. Elles ne doivent pas présenter de grandes difficultés.

Une liste de mots de passe s'affiche alors dans le terminal (figure 1).



```
pascal:~/bin/pwgen-2.06$ ./pwgen
coo8Shae thev1aiD ohdeiT4e aepei2Ie Iu8ofuog aat8aH6I Aajoo4Ca Thaifae4
0hhi5ahz eoy2Lah5 viLaeri5 Thei7qua Eephiiek2 oofoop5C hooj8oHu ap1Asuv1
Thu3moor Ne5iewie Mai4eu8e ohY2ing9 Uqueho6i iet2ieR0 Gefaik9p taeCh9ir
Lah1Aixa xoYe1ohW Div7AhY2 Wee7cied esahY3le xohBagu5 ogie8Shi thaCh7ni
Eeruom2s ru0PoWie hae8ieSa goh0Gien Paezu9Iu yah5ePah pae3jioP ya2rieLe
zaeToh5e caiSae8Y oQuo7peo wah9Vi6i ooTaeF0e Ohzau0pu ahC5bohD ieX4naeT
iebaeH2u Chae9pi6 shaish0I Yapai7oo 0e1jiade uuyeShe7 EeKaich1 Is5eith2
soo0Uin8 quijau0E Pingu5wu AhZa8eed cee8oh2E Ji4voche Phitaem9 EiRi7ael
FeiYa7oo AeThiok8 aD2omede ciaN1eoy ohn9Ieso ge1Mohr5 rahc4waV eiVo5siw
Eax7shoh jon7Iphu aiYaeb0i xu1ohSho teing2eB sah1ed8S rai4Wesh ofahY1Uf
hoo1ThiM choh0Yew chesaeZ0 Aip4tie4 ahkoo6Ce ahsei20h fahD5che Oog1itha
Deeth9qu 0oquie1i iepain3A laZuv6ra TheiSh9o gohPey5w oMuengu5 ohKu5zu7
Oun1jeek Neij0woh cakooW3A oaveiH9n ZeaNgi1ee ain6Eer3 ahBa7cai eec0vaeM
nae1Iesh iaha0eeP ma6Fe6Ph ioquooH7 Uij5ku7U aeC1icha uaCh9dae dau7AiPe
Beatie2o ue7aiGah ouzuk5Si Uw9Yae2o aey2Auch ua4ikaaV uG7phiek Jo8gupae
vaig0aiP Apach60h phu4ieYo ahGh3nao ohcuth7T uTahTae5 IShia0li Ahrijee9
ax2Kahth Hi6yaeP4 oav1AGei raeth2So BaeNg3gu bigheeW4 nai6aiTh Eem4ahBa
PequeiP7 uGh5Shex eeFoo8ae ahDi8wai Peoy2cae aiThoog1 Aurequ2y ahChes5U
roZas0qu nohtah6S aeCai6ph Yee7aich woh6aeVi aing4Sae Aec9eiPh ruG7eGeg
Fee7Quuo Aighe3Ui Su4So2du xeiph7La neZaith4 AiJ3evee Aehee6ka iMeich4m
```

- Figure 1 -

2) L'application XCODE

Ouvrir XCode. Un grand nombre de manipulations manuelles va suivre mais elles restent simples d'accès. Il faut bien toutes les respecter pour arriver au résultat.

Sélectionner le menu : File->New Project...

Sélectionner : AppleScript Application.

Puis cliquer sur Next.

Saisir Project Name : MDP

Sélectionner votre répertoire de session (la maison) pour Project Directory.

Cliquer sur Finish.

La fenêtre du projet nouvellement créé s'affiche.

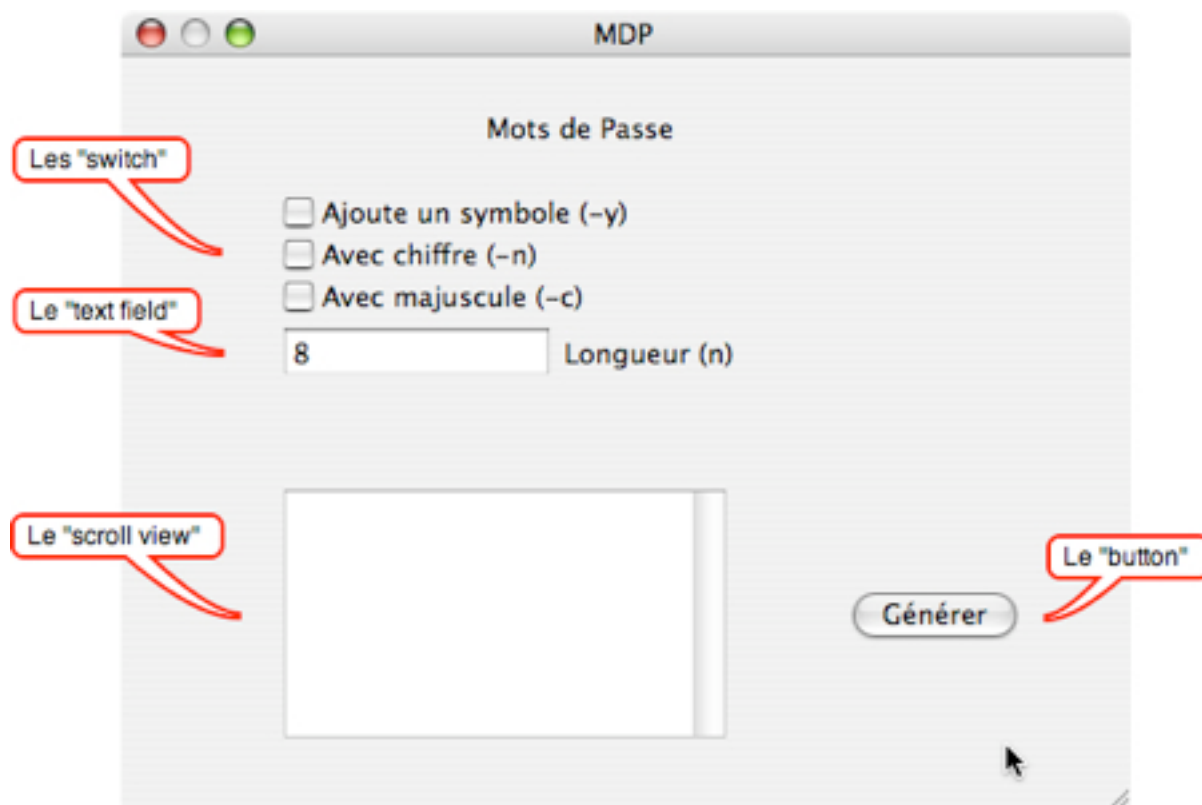
Double cliquer sur MainMenu.nib dans la fenêtre du projet.

Interface Builder se lance automatiquement.

Sur la fenêtre Window faites glisser des "switch" et un bouton depuis la palette Cocoa-Controls ainsi qu'un "textView" et un "textfield" depuis la palette Cocoa-Text.

Mettre les indications textuelles et une valeur par défaut dans longueur (figure 2).

Sauvegarder, si besoin cliquer sur "Save in 10.2 format".



- Figure 2 -

Ouvrir l'inspecteur avec le menu Tools->Show Inspector, sélectionner l'onglet AppleScript puis sélectionner chaque objet (Figure 2) pour effectuer les opérations qui suivent.

Mettre un nom sur chacune des options "option_y, option_n, option_c"

Mettre un nom "generer" sur le bouton et sélectionner "clicked" puis sélectionner "MDP.applescript" en bas de l'inspecteur

Mettre un nom "MDP" sur la fenêtre

Mettre un nom sur la zone de défilement "sortie" et sur son intérieur "sortie" avec un double clic puis supprimer "editable" dans Attributes.

Retourner dans XCode en cliquant sur Edit dans l'inspecteur AppleScript.
Copier le code suivant :

```

on clicked theObject
    (*Add your script here.*)
    tell window of theObject
        try
            if state of button "option_y" is 0 then
                set theOption_y to ""
            else
                set theOption_y to "y"
            end if
            if state of button "option_n" is 0 then
                set theOption_n to ""
            else
                set theOption_n to "n"
            end if
            if state of button "option_c" is 0 then
                set theOption_c to ""
            else
                set theOption_c to "c"
            end if
            set theResult to theOption_y & theOption_n & theOption_c
            if theResult is "" then
                set theOptions to ""
            else
                set theOptions to " -" & theResult
            end if
            set theOption_n to contents of text field "option_n"
            set chemin to POSIX path of the (path to me)
            set theResult to do shell script chemin & "Contents/MacOS/
pwgen -1 -B -N 6" & theOptions & " " & theOption_n
            set contents of text view "sortie" of scroll view "sortie" to
theResult
        on error
            set contents of text view "sortie" of scroll view "sortie" to
"Erreur de formatage !! "
        end try
    end tell
end clicked

```

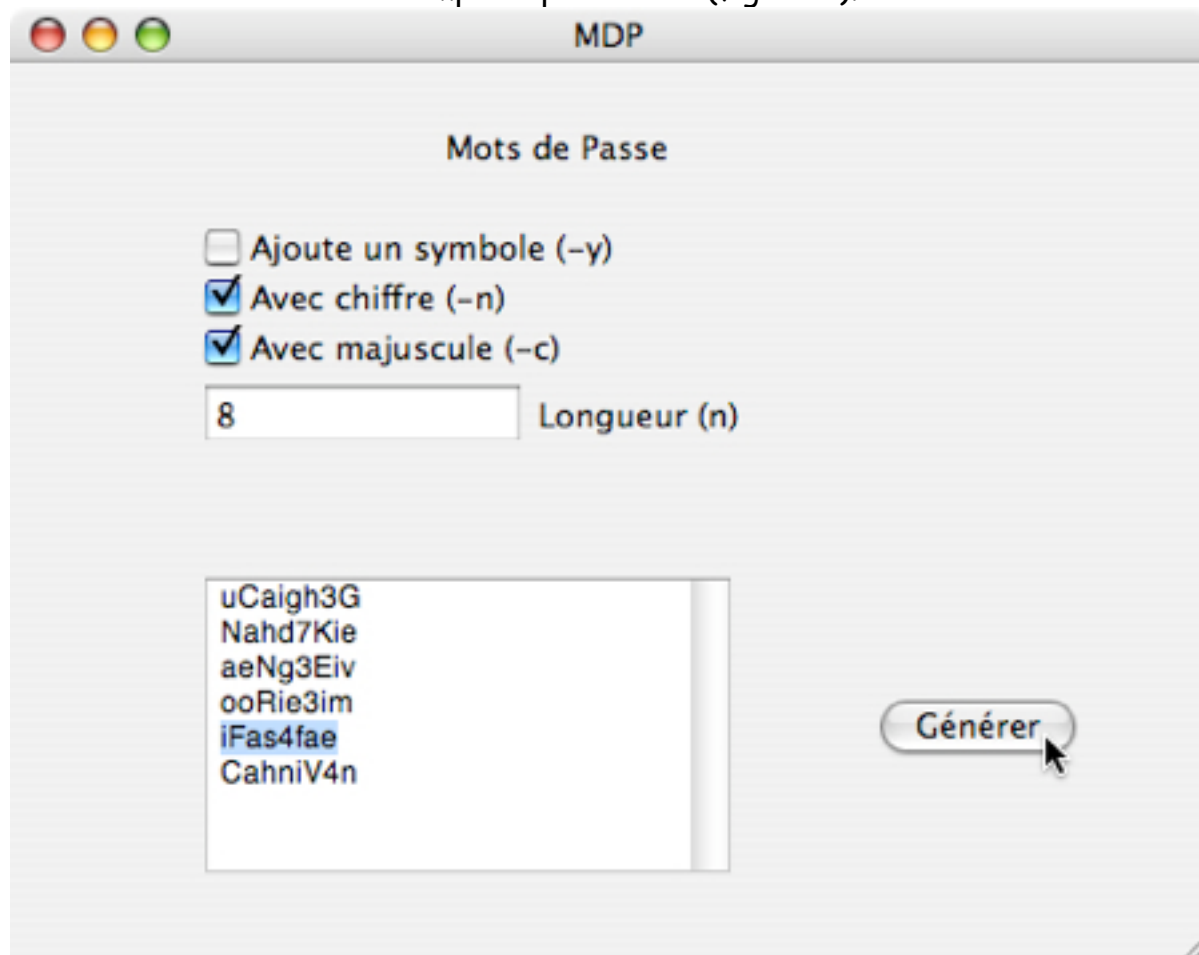
Ce code transforme les actions sur les cases à cocher en options pour la ligne de commande. Puis affiche le résultat de l'exécution du programme pwgen. Le plus gros du travail est fait. Sélectionner le menu Build->Build and Run.

L'application MDP s'ouvre, cliquer sur le bouton... Arg ! Erreur !?

Eh oui, il nous reste une dernière manipulation avec le Terminal pour insérer le programme pwgen dans l'application MDP :

```
$ cp -p ~/pwgen-2.06/pwgen ~/MDP/build/Debug/MDP.app/Contents/MacOS
```

Cliquer de nouveau sur le bouton "Générer". Ouf, les mots de passe s'affichent et sont utilisables avec un simple copier-coller (figure 3).



- Figure 3 -

Pour obtenir une application complètement autonome, sélectionner le menu Project->Set Active Build Configuration et l'option Release puis Project->Build. Le programme pwgen devra être également insérer dans l'application MDP :

```
$ cp -p ~/pwgen-2.06/pwgen ~/MDP/build/Release/MDP.app/Contents/MacOS
```

Désormais l'application MDP.app contenue dans le répertoire build/Release de notre projet XCode peut être transportée là où bon vous semble de façon à l'avoir toujours sous la main lorsque un formulaire d'inscription vous demandera un mot de passe.

Pascal Pignard, Octobre 2009.