



FINAL REPORT

INTERNET AND WEB SYSTEMS-1

Chandrasekaran, Navaneeth

Dec 4th 2018



Internet and Web systems

I wanted to solve the problem of password management and security. I did the following for my project for Internet and Websystems.

- Chrome extension to manage passwords
- A webpage to compute the passwords

Quick Links

- [Chrome Extension](#):
- [Webpage](#)
- Docker Image: `docker pull navaneethcsiva/iws`
- [Source code and reports](#)

Chrome Extension - Mystiko

Mystiko is a Chrome extension designed to make passwords management easier. It will automatically generate a unique and secure password for every website and it is going to be different for each website that makes your passwords more secure.

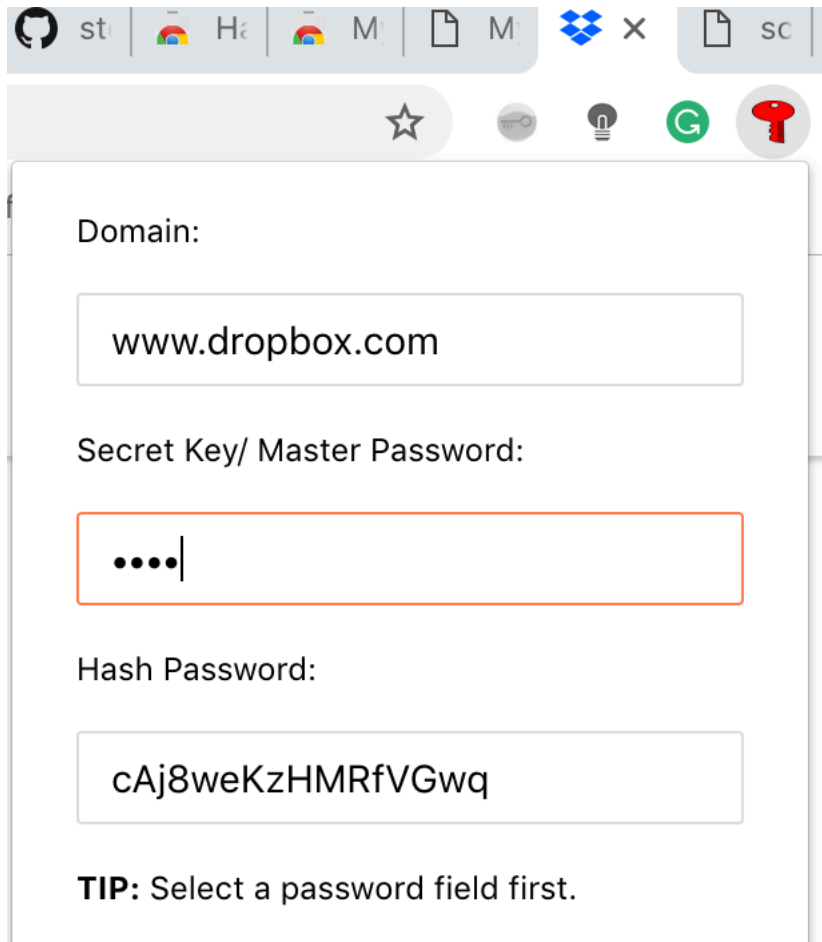
The main advantage of this tool is, it is stateless. That means, there is no server behind this application which makes sure that our passwords are never stored anywhere. It is calculated on the go.

Installation

Install Mystiko from the Chrome App Store ([link](#)). You will then see the Mystiko icon next to your address bar.

A quick tour

Click the key icon and this will pop up:



Domain:

www.dropbox.com

Secret Key/ Master Password:

...

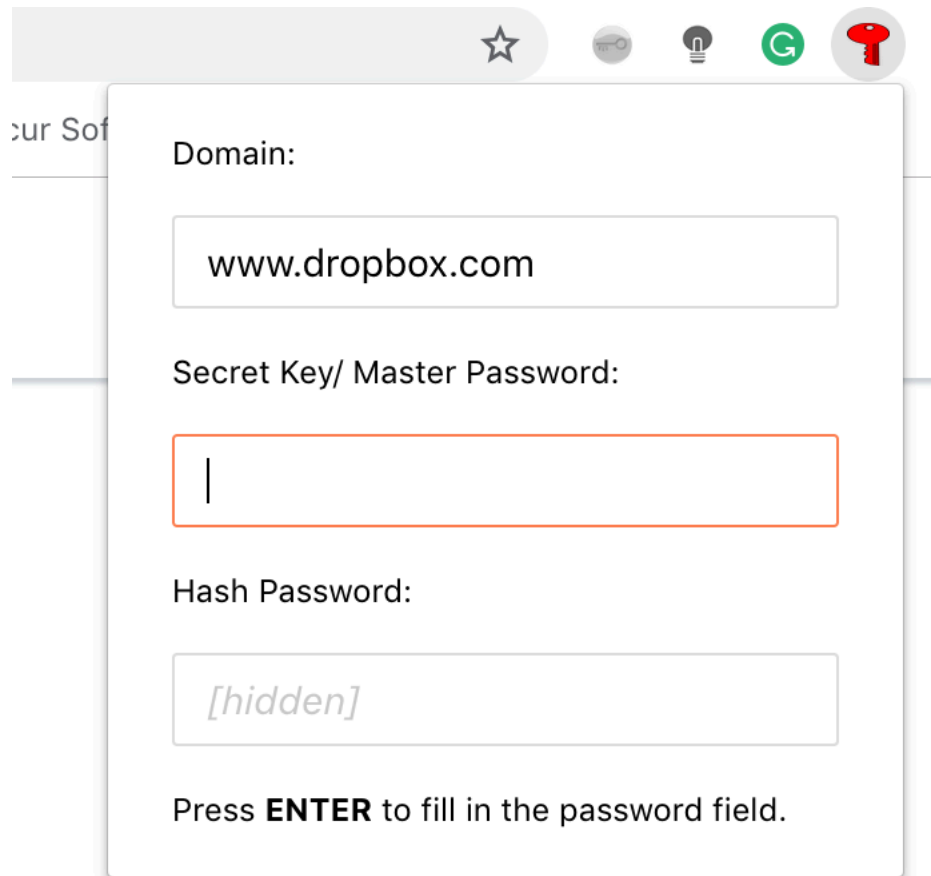
Hash Password:

cAj8weKzHMRfVGwq

TIP: Select a password field first.

You will be able to see the hashed password value here. This is because you didn't select any password field in particular.

Mystiko generates a password based on your master password and the current domain. Usually you will want to select a password field first. Then Mystiko doesn't show the generated password, giving you the option to fill in the field instead:



The image shows a web browser window with a Mystiko password generation dialog box open. The browser's address bar and tabs are visible at the top. The dialog box contains three input fields: 'Domain:', 'Secret Key/ Master Password:', and 'Hash Password:'. The 'Domain:' field contains 'www.dropbox.com'. The 'Secret Key/ Master Password:' field is empty and has a red border. The 'Hash Password:' field contains '[hidden]'. Below the fields, there is a prompt: 'Press **ENTER** to fill in the password field.'

Domain:

www.dropbox.com

Secret Key/ Master Password:

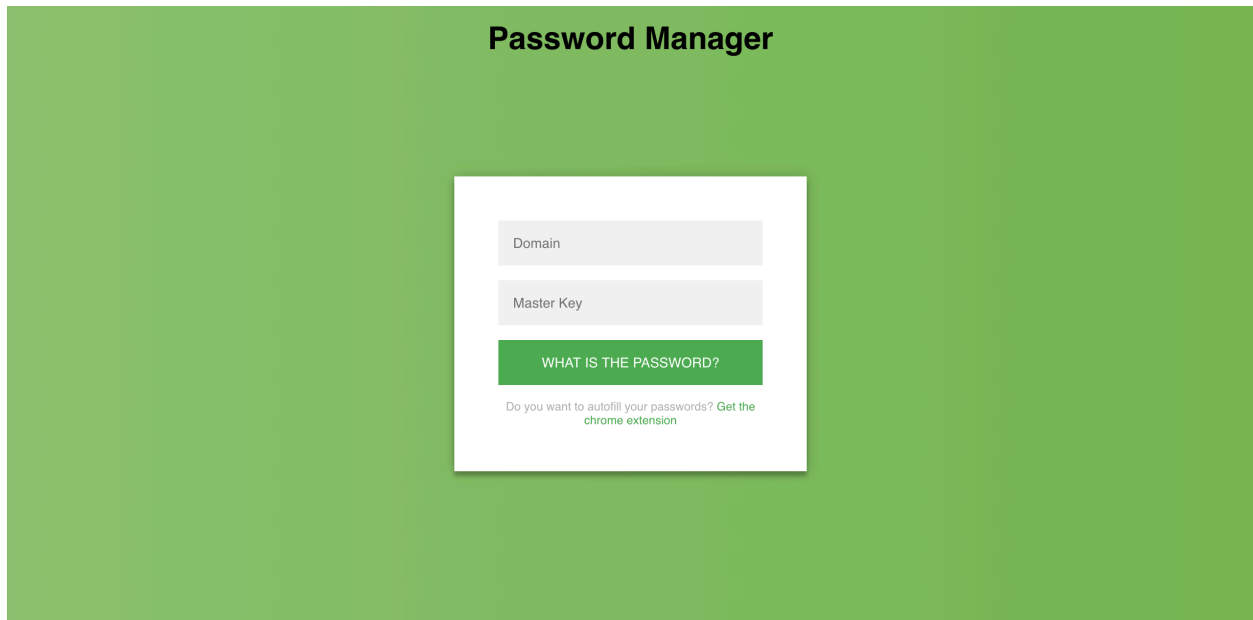
Hash Password:

[hidden]

Press **ENTER** to fill in the password field.

Website - Mystiko

In addition to the chrome extension, there is also a website developed for this tool which makes your password available even when you dont have access to your chrome extension. [http:// weblab.cs.uml.edu/~nchandra/513_f2018/](http://weblab.cs.uml.edu/~nchandra/513_f2018/)

The image shows a web browser window with a green background. At the top, the text "Password Manager" is displayed in bold black font. Below this, there is a white rectangular form. Inside the form, there are two input fields: the first is labeled "Domain" and the second is labeled "Master Key". Below these fields is a green button with the text "WHAT IS THE PASSWORD?". At the bottom of the form, there is a small text link that says "Do you want to autofill your passwords? Get the chrome extension".

How passwords are generated

Let's say your master key password is bananas and you are registering for a website let's say drop box. Mystiko combines the domain name and the master key and puts a / between them and creates a key as follows: `www.dropbox.com/test`. This string is hashed again and again for 2^{16} and then the first 16 characters of the result value will be used as the password. So, for the above case the final password will be `cAj8weKzHMRfVGwq`.

Security

If someone gets your password, it is impossible for them to backtrack to your master key because of the hashing algorithm.

One strategy for cracking your secret key is to try hashing all English words, for example. This is called a dictionary attack. An attacker might even try to pre-compute the hashes

of all English words and other common passwords. Then they could simply look up hashes in this hash table to crack them. The table in this attack is called a rainbow table.

- A brute force attack is practically impossible with the current computational power available.

Reference

- <https://www.linkedin.com/pulse/serve-static-files-from-docker-via-nginx-basic-example-arun-kumar/>
- <https://hub.docker.com/r/navaneethcsiva/iws/>
- <https://crypto.stanford.edu/sjcl/>
- <https://github.com/bitwiseshiftleft/sjcl>
- <https://docker-curriculum.com>
- <https://medium.freecodecamp.org/how-to-create-a-chrome-extension-part-1-ad2a3a77541>
- <https://www.sitepoint.com/create-chrome-extension-10-minutes-flat/>
- <https://usersnap.com/blog/develop-chrome-extension/>
- <https://www.producthunt.com/posts/hashpass>
- <http://dennisspan.com/deploying-google-chrome-extensions-using-group-policy/>
- <https://www.itninja.com/question/how-to-package-and-deploy-chrome-extension>
- http://www.adambarth.com/experimental/crx/docs/external_extensions.html
- <https://circleci.com/blog/continuously-deploy-a-chrome-extension/>