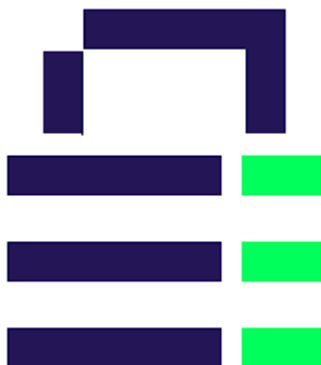


# Custom Web Login Launcher for Secret Server



A powershell script to automate the launch of a webpage and automatically log into it.

License **GPL v3**

## Features

- Automatically download the latest chrome driver
- Install prerequisites like Selenium
- Automatically log into a webpage using secret credentials from Secret Server
- Kill any outstanding or existing chromedriver processes

## Dependencies

- [Selenium](#) - Browser automation
- [Chrome Driver](#) - Web driver for automation and testing

## Usage

To run web\_login.ps1 for the first time: From command line:

```
powershell ./web_login.ps1
```

From powershell:

```
./web_login.ps1
```

This will configure install Selenium, and download the respective chrome driver.

Arguments can be passed to the script for automation (required for use as a launcher within Secret Server).

(**-username**) - the username used to log into the given webpage.

- Examples: powershell ./web\_login.ps1 -username or ./web\_login.ps1 -username

(**-password**) - the password used to log into the given webpage.

- Examples: `powershell ./web_login.ps1 -password` or `./web_login.ps1 -password`

(**-webpage\_url**) - the url for the login webpage.

- Examples: `powershell ./web_login.ps1 -webpage_url` or `./web_login.ps1 -webpage_url`

(**-username\_id**) - the id of the username field on the given login webpage.

- Examples: `powershell ./web_login.ps1 -username_id` or `./web_login.ps1 -username_id`

(**-password\_id**) - the id of the password field on the given login webpage.

- Examples: `powershell ./web_login.ps1 -password_id` or `./web_login.ps1 -password_id`

(**-submit\_id**) - the id of the submit, sign-in, or login button on the given webpage.

- Examples: `powershell ./web_login.ps1 -submit_id` or `./web_login.ps1 -submit_id`

Example run using arguments:

```
powershell C:\path\to\script\.\web_login.ps1 -username myusername -password mypassword  
-webpage_url https://mypage.com/login -username_id signin-username -password_id  
signin-password -submit_id signin-button
```

or

```
C:\path\to\script\.\web_login.ps1 -username myusername -password mypassword -  
webpage_url https://mypage.com/login -username_id signin-username -password_id signin-  
password -submit_id signin-button
```

## Configuration as a Custom Launcher

To configure this script as a launcher in Secret Server, navigate to "**Administration -> Secret Templates -> Launchers**". Create a new launcher by clicking the "**Create**" button in the top right.

### General Settings

- Launcher Type - **Process**
- Launcher Name - **Your Web Launcher Name**
- Active - **Yes**
- Launcher Image - **Optional**
- Wrap Custom Parameters with Quotation Marks - **No**
- Track Multiple Windows - **Yes**
- Record Additional Processes - **< None >**
- Use SSH Tunneling with SSH Proxy - **No**

### Windows Settings

- Process Name - **powershell.exe**
- Process Arguments - **-NoExit "C:\path\to\script.\web\_login.ps1" -username \$USERNAME -password \$PASSWORD -webpage\_url "<https://your-page.com/login>" -username\_id "username-element-id" -password\_id "password-element-id" -submit\_id "submit-button-element-id"**

- Run Process as Secret Credentials - **No**
- Load User Profile - **No**
- Use Operating System Shell - **No**

#### Advanced

- Escape Character - `
- Characters to Escape - `$&(){}<>|;'`,

NOTE - The backtick (`) character is the escape character for powershell. The hyphen character (-) cannot be used or escaped as it is used to indicate an argument in powershell.

## Adding the launcher to a Secret Template

Once the launcher has been created, it can be added to a Secret Template by navigating to **"Administration -> Secret Templates -> Your Template Name -> Mapping"**. Add the launcher to the template by clicking the **"Add Mapping"** button in the top right. Select the launcher that was just created and map **"Username"** and **"Password"** to their respective fields. The **"Domain"** field can be left mapped to **""**.

## Finding Element IDs for a webpage

Below is an example of how to find the IDs of the username, password, and sign-in elements on a webpage. Navigate to the login url of your webpage. In this example, the Rapid7 Insight url will be used (["https://insight.rapid7.com/login#/"](https://insight.rapid7.com/login#/)).

From there, right click on the username field and select **"Inspect"**. This will show the properties of the username field. Look for the **"ID"** of the element. In this example, the ID of the username/email field is **"okta-signin-username"**. Repeat this process for the password field and the submit/login button.

```
Q Search HTML
<div class="o-form-fieldset-container" data-se="o-form-fieldset-container">
  <div class="o-form-fieldset o-form-label-top margin-btm-5" data-se="o-form-fieldset">
    <div class="okta-form-label o-form-label" data-se="o-form-label"> </div>
    <div class="o-form-input" data-se="o-form-input-container">
      <span class="o-form-input-name-username o-form-control okta-form-input-field input-fix" data-se="o-form-input-username"> event
        <input id="okta-signin-username" type="text" placeholder="" name="username" value="" aria-label="" autocomplete="username" aria-required="true" required="">
      </span>
    </div>
    ::after
  </div>
  <div class="o-form-fieldset o-form-label-top margin-btm-30" data-se="o-form-fieldset"> </div>
```

Using the above method, inspecting the password field gives us the ID, which is **"okta-signin-password"**. And again, inspecting the **"Sign In"** button gives us the ID **"okta-signin-submit"**.