

PASSWORD MANAGER > DEVELOPER TOOLS

CLI Authentication Challenges

View in the help center:

<https://bitwarden.com/help/cli-auth-challenges/>

CLI Authentication Challenges

The August 2021 release of Bitwarden (**2021-09-21**) introduced [Captcha](#) requirements to increase security against bot traffic. On the CLI, Captcha challenges are substituted with authentication challenges that can be validated using your account's [personal API key](#) `client_secret`.

Tip

For automated workflows or for providing access to an external application, we recommend using the `bw login --apikey` method. This method follows a more predictable authentication flow and revoking an application or machine's access can be achieved by rotating the [API key](#).

Get your personal API key

To get your personal API key:

1. In the Bitwarden web app, navigate to **Settings** → **Security** → **Keys**:

The screenshot shows the Bitwarden web app interface. On the left is a dark blue sidebar with navigation options: Password Manager, Vaults, Send, Tools, Reports, Settings, My account, Security (highlighted), Preferences, Subscription, Domain rules, Emergency access, and Free Bitwarden Famili... The main content area is titled 'Security' and has three tabs: Master password, Two-step login, and Keys (selected). Under the 'Keys' tab, there is a section for 'Encryption key settings'. A yellow warning box states: 'Warning: Proceeding will log you out of all active sessions. You will need to log back in and complete two-step login, if any. We recommend exporting your vault before changing your encryption settings to prevent data loss.' Below this, text explains that higher KDF iterations protect the master password but may cause performance issues on older devices. Two input fields are shown: 'KDF algorithm (required)' set to 'PBKDF2 SHA-256' and 'KDF iterations' set to '600000'. A 'Change KDF' button is below. The 'API Key' section follows, stating the key is used for CLI authentication, with 'View API key' and 'Rotate API key' buttons.

Keys

2. Select the **View API key** button and enter your master password to validate access.

3. From the **API key** dialog box, copy the **client_secret**: value, which is a random string like `efrbgT9C6BogEfxi5pZc48XyJjfpR`.

Answering challenges

Depending on your preferences, you can [save an environment variable](#) to automatically pass authentication challenges or [manually enter](#) your `client_secret` whenever a challenge is made:

Answer challenges with an environment variable

Authentication challenges will look for a non-empty environment variable `BW_CLIENTSECRET` before prompting you to enter one manually. Saving this variable with the [retrieved client_secret](#) value will allow you to automatically pass authentication challenges. To save this environment variable:

Bash

Bash

```
export BW_CLIENTSECRET="client_secret"
```

PowerShell

Bash

```
env: BW_CLIENTSECRET="client_secret"
```

Warning

If your `client_secret` is incorrect, you will receive an error. In most cases, this is because you have [rotated your API key](#) since saving the variable. [Use the above steps](#) to retrieve the correct value.

Answer challenges manually

When an authentication challenge is made and no `BW_CLIENTSECRET` value is found, you will be prompted to manually enter your `client_secret` value:

```
[ user@users-MacBook-Pro / % bw login
[? Email address: myemailaddress@me.com
[? Master password: [hidden]
[? Additional authentication required.
[API key client_secret: apiskey_clientsecret ←
[? Two-step login method: Authenticator App
[? Two-step login code: 340450
You are logged in!

To unlock your vault, set your session key to the `BW_SESSION` environment variable. ex:
$ export BW_SESSION="..."
> $env:BW_SESSION="..."

You can also pass the session key to any command with the `--session` option. ex:
$ bw list items --session ...
```

Login Prompt with Auth Challenge

 **Warning**

If your `client_secret` is incorrect, you will receive an error. In most cases, this is because you have [rotated your API key](#) since saving the variable. [Use the above steps](#) to retrieve the correct value.