

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`.

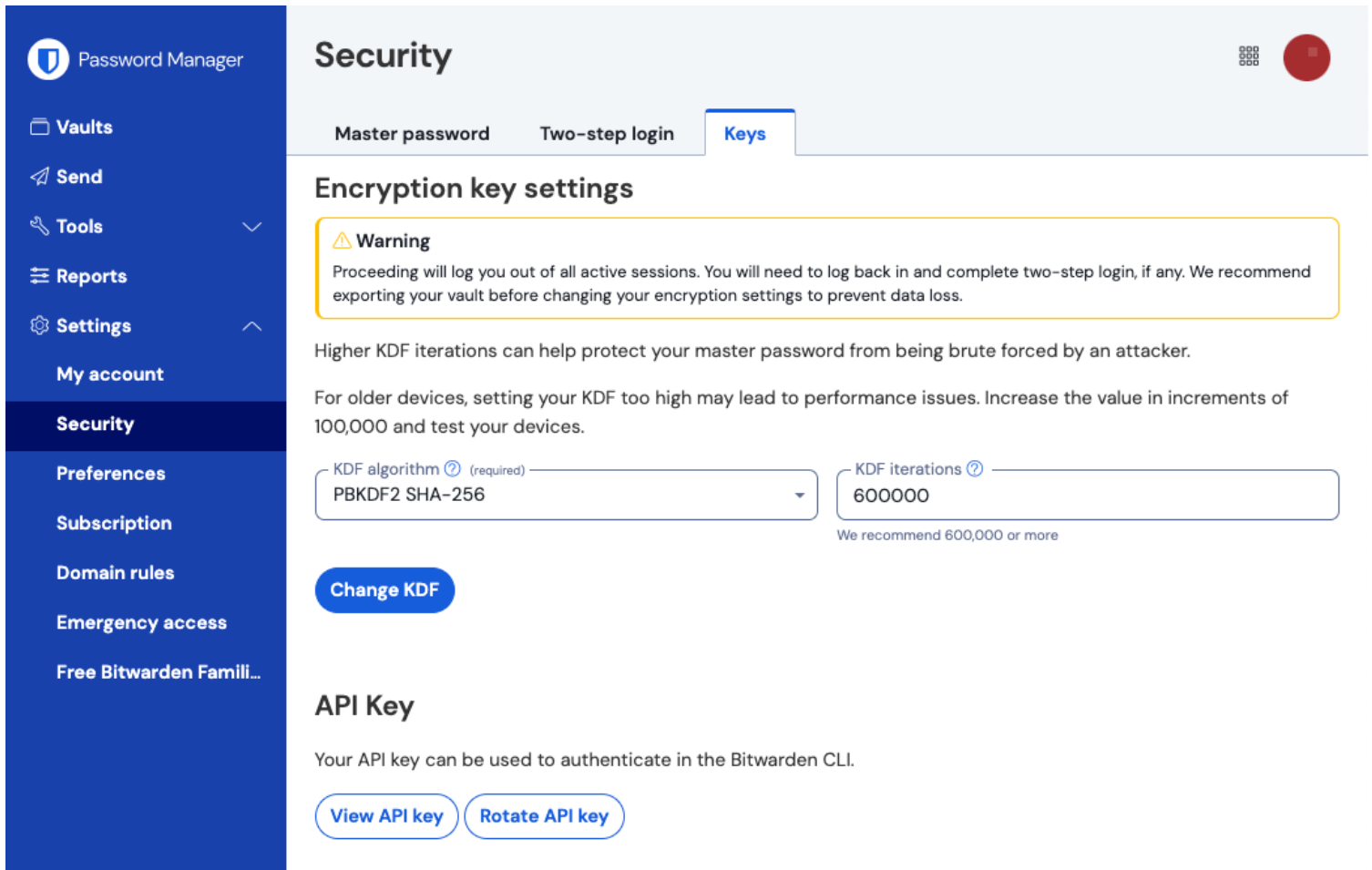
Tip

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 links: Password Manager, Vaults, Send, Tools, Reports, Settings (highlighted), My account, Security (highlighted), Preferences, Subscription, Domain rules, Emergency access, and Free Bitwarden Famili... The main content area has a light blue header with 'Security' and tabs for 'Master password', 'Two-step login', and 'Keys' (selected). Below the tabs is the 'Encryption key settings' section, which includes a yellow warning box about logging out of active sessions. It also contains text about KDF iterations and two input fields: 'KDF algorithm' (set to PBKDF2 SHA-256) and 'KDF iterations' (set to 600000). A 'Change KDF' button is below these fields. The 'API Key' section follows, with text explaining its use and two buttons: 'View API key' and 'Rotate API key'.

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: apikkey_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.