GitHub

Tools like check50 and submit50 rely on `git`, a popular tool for saving different versions of code, and GitHub, a popular website for saving those versions in the cloud. To push (i.e., save) your code to GitHub using `git`, it used to be possible to log into GitHub via a command line (as in a terminal window) using a GitHub username and password. As of August 13, 2021, that's no longer possible, which means you can no longer use `check50` or `submit50` using your GitHub username and password either.

But you can still use `check50` and `submit50`! You just need to log in a bit differently, either using SSH or a personal access token. Odds are you'll find SSH more convenient for Visual Studio Code and CS50 IDE, and personal access tokens more convenient for CS50 Sandbox and CS50 Lab.

SSH

1. Open a terminal window, if not open already, within Visual Studio Code, CS50 IDE, CS50 Sandbox, or CS50 Lab.
2. Execute `ssh-keygen`. When prompted to "save the key," just hit Enter, without typing anything.
3. You'll then be prompted for a "passphrase" (i.e., password). If you only use your GitHub account for CS50, no need to input a passphrase; just hit Enter. Otherwise, input a passphrase (that you won't forget!), then hit Enter, then input it again, then hit Enter again. For security's sake, you won't see what you type. You'll then see a "randomart image" that you can ignore.
4. Execute `cat ~/.ssh/id_rsa.pub`. You'll then see your "public key," multiple lines of seemingly random text. Highlight and copy all of those lines, from `ssh-rsa` to the end. **But don't highlight your terminal window's prompts (which contain `$`) before or after those lines.**
5. Visit https://github.com/settings/keys, logging in with your GitHub username and password as usual. Don't use the passphrase you just created, if any.
6. Click **New SSH Key**.
7. Paste your public key into the text box under **Key**. Optionally input a title under **Title** (e.g., `CS50`).
8. Click **Add SSH Key**.

You should now be able to use `check50` and `submit50` (and `git`) without GitHub username and password. But if you created a passphrase, you might still be prompted for that.

If you created a passphrase but forgot it

1. Visit https://github.com/settings/keys, click **Delete** next to your old SSH key, then click **I understand, please delete this SSH key**.
2. Follow all of the same SSH steps, above, again. When prompted to "overwrite" (your old key), input `y`, then hit Enter.

Personal Access Token

1. Visit https://github.com/settings/security, logging in with your GitHub username and password as usual, and configure two-factor authentication.
2. Visit https://github.com/settings/tokens.
3. Click **Generate new token**.
4. Input a note under **Note** (e.g., `CS50 IDE` if using CS50 IDE).
5. Select **No expiration** (or something shorter) via the drop-down menu under **Note**.
6. Check **repo** under **Select scopes**.
7. Click **Generate token**.
8. Highlight and copy the "personal access token" that appears. Odds are it will start with `ghp_`.
9. Paste that personal access token somewhere safe (e.g., in a password manager).

You should now be able to use `check50` and `submit50` (and `git`) without GitHub username and password. When prompted to log in, use your GitHub username and that personal access token instead of your password.

If you created a personal access token but forgot it (or it expired)

1. Visit https://github.com/settings/tokens, click **Delete** next to your old personal access token, then click **I understand, delete this token**.
2. Follow all of the same Personal Access Token steps, above, again.