# Submitting New CVEs Using CLI

*Author: Madison Oliver; July 2019*

The purpose of this documentation is to show a user how they can submit **new** JSON files to MITRE’s CVE Github repository using the git command line interface as opposed to a GUI-interface. The following steps assume that your git environment is already set up (if it isn’t, please review the Setup document <Add link to “Submitting New CVEs Using GitHub - Initial Tools Setup” once in public location>).

[Submission](#_gjdgxs)

[Updating](#_1t3h5sf)

# Submission

Let’s say that your local clone is stored in user/Documents/YOUR-REPOSITORY/… and you have want to submit CVE-2019-1234.json to MITRE's CVEProject fork of cvelist (<https://github.com/CVEProject/cvelist>)

1. While working in your repository, create a new branch (-b) and check it out
	1. Create a branch to work in named "cna/[ORG]/[SOME IDENTIFIER]", where "SOME IDENTIFIER" could be:
		1. the CVE ID if it is a single CVE
		2. The organization’s publication tracking number if it is multiple CVEs from one single vulnerability publication
		3. A description of its purpose if it is from multiple publications
		4. The date of the submission in YYYYMMDD format
	2. $ git checkout -b NEW-BRANCH-NAME
2. Move your completed .json files to the appropriate directory, e.g. CVE-2019-1234.json goes into YOUR-REPOSITORY/2019/1xxx/
	1. This can be accomplished on the command line by using cp or mv or in your file explorer with copy/paste or drag and drop
	2. You will have to replace the file, as one already exists with filler text
3. Add these files to the staging environment. Any files added with “add” will be included in the next commit. You can add files individually (a) or add all of the unstaged files at once (b)
	1. $ git add CVE-2019.1234.json
	2. $ git add --all
	3. To check what files are added, use “git status”. This will show you the files that have already been added, “Changes to be committed”, and the files that have not yet been added, “Untracked files”
		1. $ git status
4. Commit the files to your repo
	1. $ git commit -S -m "*message text goes here*"
	2. -S signs the commit with your PGP key — if you haven't yet set up your git environment with a signing key, check out GitHub's documentation at <https://help.github.com/en/articles/signing-commits>
	3. -m denotes the message that is part of the commit
5. Once all the files have been committed, push them to your organization’s Github. Since/if it’s a new branch, you will have to set the remote branch as the upstream branch
	1. $ git push origin NEW-BRANCH-NAME
	2. You can also use “HEAD” instead of the NEW-BRANCH-NAME to push the current branch you are on if it is the branch that you want to push
6. On GitHub (<https://github.com>), you should be able to initiate a pull request to MITRE’s repository
	1. There should be a green box that says "Compare and pull request" that you can click to generate the pull request
		1. 
			1. If that box is not there, just click "New pull request". The box is generated by GitHub acknowledging differences between MITRE's CVE list and your forked copy. Sometimes it isn't immediate recognize it depending on how quickly you do this, so if that box is not there it doesn't necessarily mean the changes aren't there.
	2. Initial the pull request, put an appropriate message in there (it can be the same as the commit message) and submit it to MITRE
		1. They will come back with any requested changes if needed. To make these changes, update the files on your host machine in the same repo and branch that you initially put it in, commit the new changes, and then push them to Github. They'll automatically show up as part of this pull request as a new commit.

# Updating

If you want to update an **existing** CVE (that is, one that exists in MITRE’s Github already and is completed), you should **first** pull the latest updates from MITRE’s repo (<https://github.com/CVEProject/cvelist>) to avoid any merge conflicts (<https://www.atlassian.com/git/tutorials/using-branches/merge-conflicts>), then update the file and submit it using the aforementioned steps.

1. Add MITRE’s CVE list as a remote repo that you can pull from, named “mitre” below
	1. $ git remote add mitre <https://github.com/CVEProject/cvelist.git>
2. Pull the MITRE branch that you want to receive updates from, almost always the master branch
	1. $ git pull mitre master
3. Then finally push these changes to your organization’s Github
	1. $ git push