# How CERT/CC submits CVEs using Github, Sourcetree, and Bitbucket

*Author: Madison Oliver; July 2019*

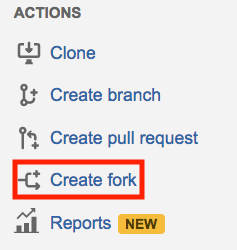
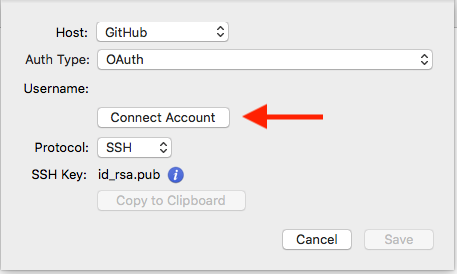
This is an example of how CERT/CC submits CVEs using Github (so it contains a lot of assumptions!). If you happen to work at CERT/CC, you can use this document pretty much as-is. If not, you should interpret these instructions through your own environment's lens. If you only use git and GitHub, you can skip all the BitBucket and SourceTree parts.

The guide uses SourceTree and Bitbucket for submission since a GUI is typically easier to use than the git command line interface (CLI). At CERT/CC, we have an internal Bitbucket server that is a clone of our public facing Github, and we have separate analyst and manager roles. The analyst writes the CVE .json file and submits it to our Bitbucket server. The manager then checks this file for accuracy, pushes it to CERT/CCs Github, and then pull requests to the CVEProject Github fork of cvelist. The purpose of using Bitbucket prior to Github is to improve accuracy of .json files prior to pushing them to Github - once they’re on Github, they’re public information. The purpose of splitting the analyst and manager roles are to reduce the amount of information/steps for analysts.

*All of the following can be completed using git from the command line instead of SourceTree. If you're comfortable enough using git to do this, please feel free to. The following is also completed on a Mac - everything should still remain similar (but not exactly the same) with Windows.*

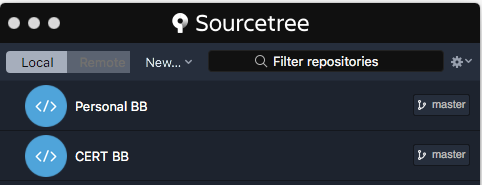
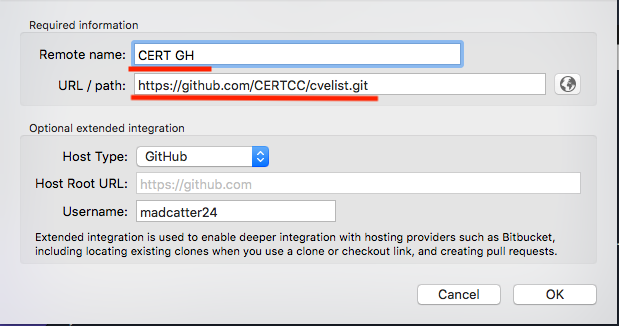
**Te following initial setup process needs to be completed only once and should be completed by both the analyst and manager. The manager should complete the the Manager Setup steps in addition to these.**

# Analyst Setup

1. Navigate to the CERT Bitbucket cvelist repository in Bitbucket
   1. Create a fork of this repository in your personal Bitbucket using the "Create fork" button in the left-hand menu bar
      1. 
2. Install Sourcetree (git GUI) for ease of use if you prefer not to use git: <https://www.sourcetreeapp.com/>
   1. Sign into this using your Github account when prompted while installing SourceTree
      1. 
3. Set up a local clone repository of your personal fork of the CERT Bitbucket (Personal BB)
   1. Two options:
      1. In SourceTree, File → New → Clone from URL
      2. In Bitbucket, use the "Clone" button in the left-hand menu bar. It will have the URL also there that you can enter (3.a.i.1), or you can choose "Clone in SourceTree" and it will open the SourceTree desktop application
      3. Pay attention to the destination path of where these local repositories are created on your host - you'll want to be able to navigate to these easily so consider having everything Git related in a single folder
         1. If you have trouble - with the repository open in SourceTree, go to Actions → Show In Finder (applicable on Mac, likely similar on Windows)
   2. Within Sourcetree, you should now have one repository - one local clone of your fork of the CERT Bitbucket cvelist
4. *For simplicity - a clone is a copy of a repository, and a fork (in the Github sense) is a copy of a repository on the same remote server.*

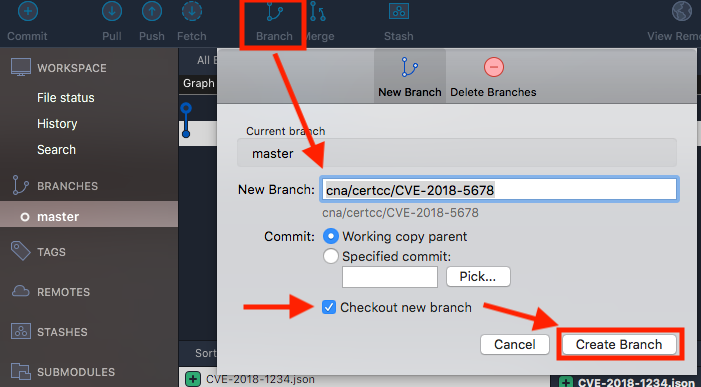
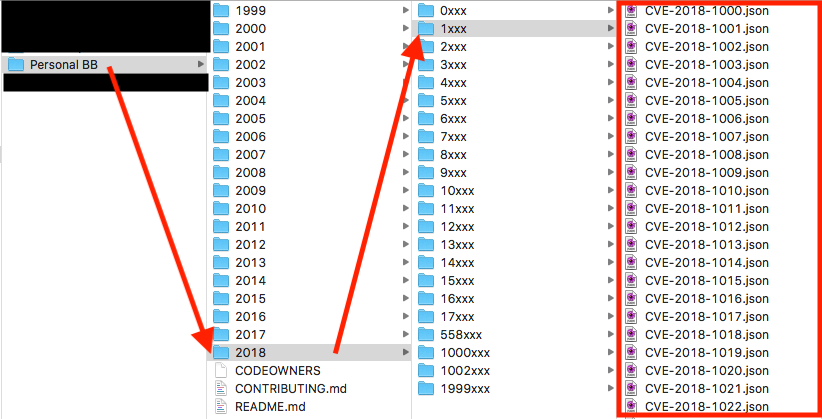
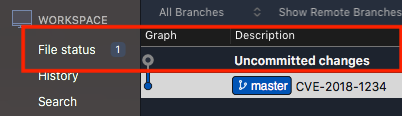
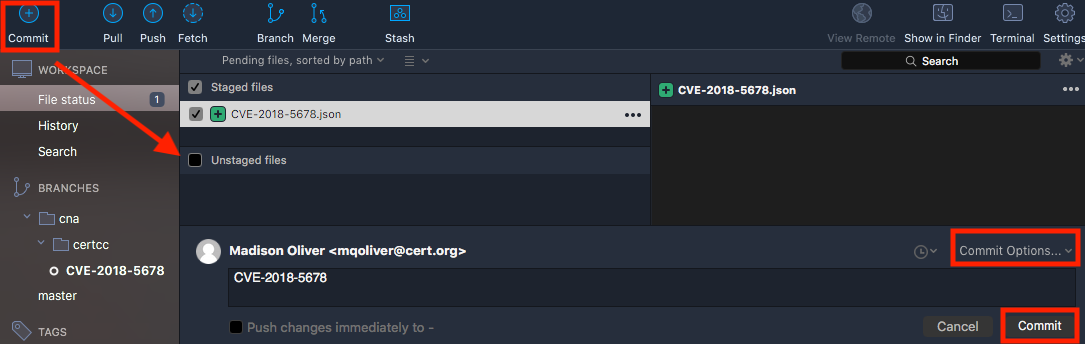
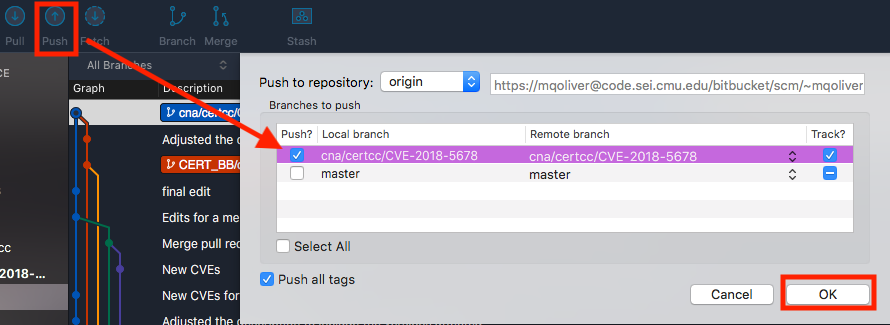
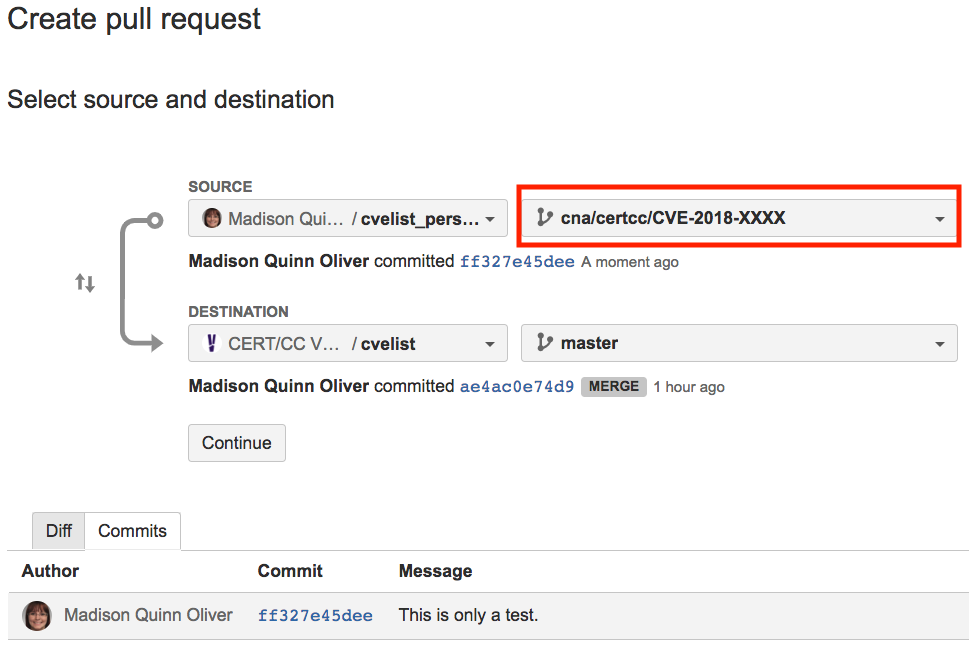
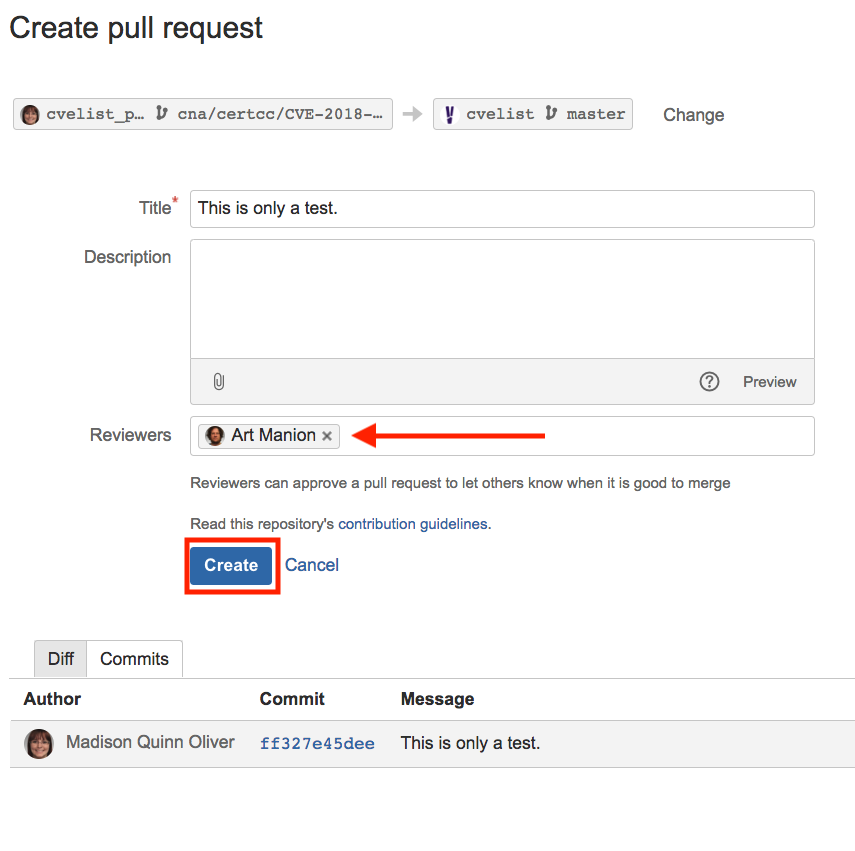
**The following is the process to setup local and remote repositories for the manager specifically.**

# Manager Setup

1. Within SourceTree, create a local clone of the CERT Bitbucket cvelist (CERT BB)
   1. The steps will be the same at Step 3 in the Analyst Setup and will result in you having two repositories - your personal local clone of your Bitbucket fork of the CERT cvelist and your personal local clone of the Bitbucket CERT CVE list
   2. 
   3. This repo is created to push the changes to CERT's Github. This can't happen in Bitbucket because there is no way to Push from Bitbucket to Github directly, which is why we use SourceTree (or git command line) to move things between Bitbucket and Gi
2. After the CERT BB repository is created, you will need to add the CERT Github cvelist as a remote repository in SourceTree
   1. Go to Repository → Repository Settings → Remotes tab
   2. Click "Add" - type the name you want to refer to this remote repository as ("CERT GH" in the screenshots below)
   3. Add the URL of this repository - <https://github.com/CERTCC/cvelist.git>
      1. Host type and Github user name should automatically change to Github and your Github username (this may not be the case if you did not log into SourceTree with Github)
      2. 

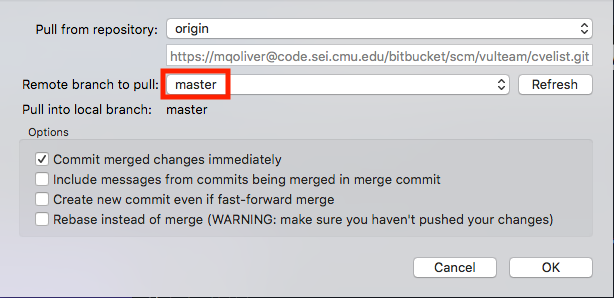
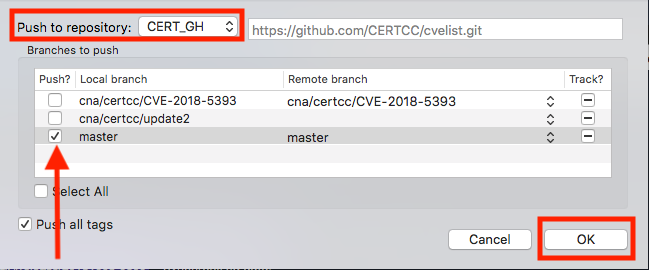
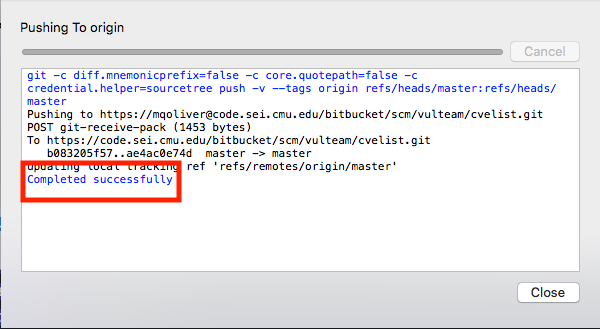
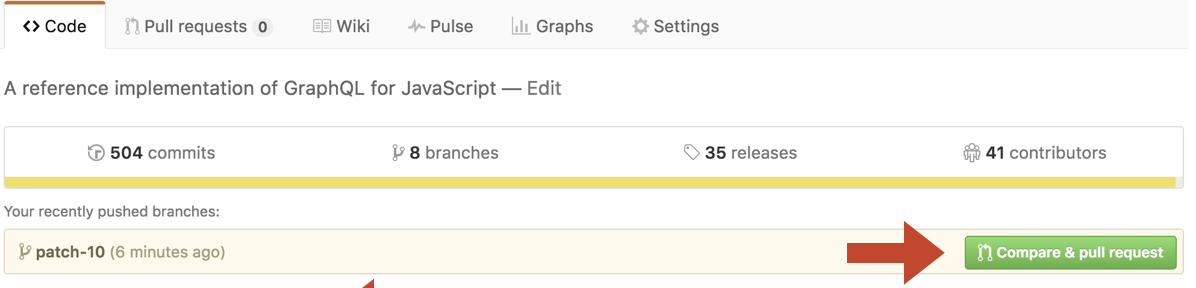
**The following is the process that the analyst should follow to submit their CVE's to the manager for approval.**

# Analyst Process

1. While working in your local clone of your fork of the CERT Bitbucket cvelist (Personal BB):
   1. Create a branch to work in named "cna/certcc/[SOME IDENTIFIER]", where "SOME IDENTIFIER" is either:
      1. the CVE ID if it is a single CVE
      2. The VU# number if it is multiple CVEs from one single vulnerability note
      3. A description of its purpose if it is from multiple VU#s (backlog, update, etc.)
      4. 
      5. Checkout the new branch by double clicking if you did not have the "Checkout new branch" checkbox checked when creating the branch (this is checked by default)
2. Outside of SourceTree, copy your CVE JSON files to the appropriate directory within the local clone of your fork of the CERT Bitbucket cvelist (Personal BB)
   1. Copy (or move) your JSON files to the appropriate directory - they are organized by year and then by leading numbers, i.e. CVE-2018-1234 would go in the 1xxx folder within the 2018 folder
   2. Replace the existing files (they only contain placeholder text) with your new JSON files
   3. 
   4. File Status in the top-left should show that there are uncommitted changes since your new files have been added to the appropriate directory but a commit has not been created yet
      1. 
   5. Commit the changes by clicking "Commit" in the top left while on the "File status" tab
      1. Check the box next to "Unstaged Files" to move them to "Staged Files"
      2. Add something to the commit message to show what this commit contains
      3. Under "Commit Options", be sure that "Sign commit" is selected
      4. Click "Commit" in the bottom right to commit it
      5. 
         1. The staging environment is where git stores files that will be included in the next commit ("git add" on the command line). In the way we'll be using it, all the files can likely go into the staging environment and into one single commit.
         2. If someone had multiple VU#s with multiple CVEs, they could add the CVEs from VU#1 to the staging environment, commit it, and then add the CVEs for VU#2 to the staging environment, and commit it. This would generate two commits that contain the CVE specific to each VU#. The benefit of this would be clarity but is often unnecessary in our case given the frequency we do this.
   6. Push this new commit to Bitbucket by selecting "Push" in the top bar
      1. Select the branch that you want to push (cna/certcc/CVE-2018-5678 in this example) and then click "Push"
      2. 
3. Move to Bitbucket web UI
   1. Create a Pull Request from your personal fork of the CERT CVE Bitbucket by clicking "Create pull request" in the left-hand menu
   2. Select the appropriate branch from your Bitbucket that contains your new JSON files
   3. Select the "master" branch of the CERT CVE Bitbucket as the destination and click continue
      1. 
   4. Adjust the title if needed - it is pulled directly from the commit message
   5. Choose a review for this PR - the reviewer should be the acting CVE manager
      1. 
   6. Click "Create" to generate the Pull Request and you're done!

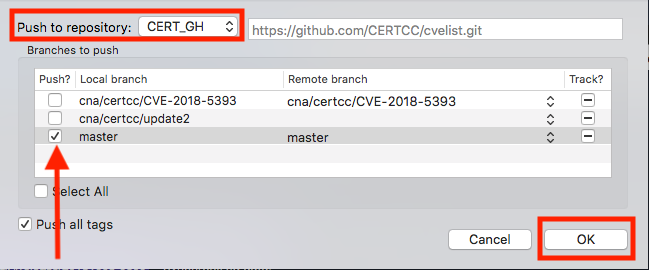
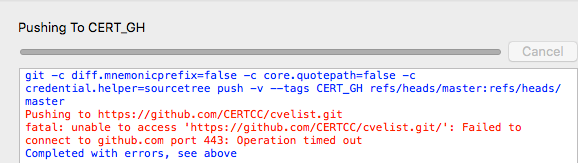
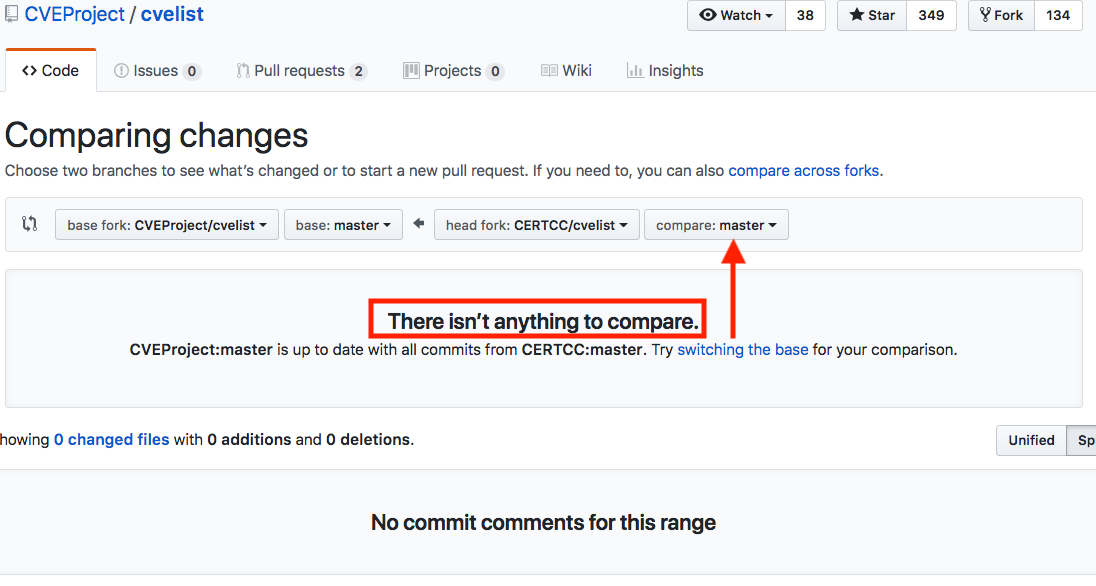
**The following process will be done by the manager after the analyst has initiated a Pull Request in Bitbucket.**

# Manager Process

1. Find the Pull Request in Bitbucket - find it in the Pull requests tab on the left-hand menu
   1. Open the pull request from the above screen
      1. Assuming it's all correct, click "Merge" in the top right to approve this request and merge the analyst's branch with the CERT Bitbucket CVE list master branch
2. Move to SourceTree
   1. Go to the master branch of your local clone of the CERT Bitbucket cvelist and pull the new changes using the "Pull" button in the top left menu bar
   2. Ensure that you're pulling the correct remote branch (it defaults to master, which is where we merged the analyst's changes, but there may be times it's a different branch)
      1. 
   3. After pulling the changes into your local clone, you're ready to push them to your organization’s GH
      1. Click push in the top menu bar
      2. 
      3. If this is successful, you will receive a "Completed successfully" message. If you do not receive that or an error message, it is likely that you did not choose which branch to push.
         1. 
3. Move to Github - <https://github.com/CERTCC/cvelist>
   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 our forked copy. Sometimes it isn't immediate 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 CERT Bitbucket repository, commit the new changes, and then push them to Github. They'll automatically show up as part of this pull request.

**The following section contains some help when troubleshooting this process.**

# Troubleshooting

1. I am trying to **push** from my local clone of the CERT Bitbucket to the remote CERT Github but am **not** receiving **any** message - no error or success message.
   1. It is likely that you did not select which branch to push to. In this case, nothing is pushed, so nothing is successful or fails. Ensure that you select the box next to the branch you want to push.
   2. 
2. I am receiving the following error when **pushing** or **pulling** from a **Github** repository.
   1. 
   2. This error is likely occurring because you're attached to a proxy. While on the proxy, SourceTree (git) is unable to communicate with Github. To successfully pull/push, turn off the proxy (but ensure that you still have an active Internet connection).
3. I am trying to create a pull request on Github from CERT's cvelist to MITRE's cvelist, but it is telling me that there is nothing to compare.
   1. 
   2. It is likely that you are using the wrong branch from CERT's Github and it does not include the changed files. Find where these changed files are - it is possible that they ended up in a different branch on your local clone of the CERT Bitbucket, and also possible that they were never pulled down from the CERT Bitbucket. There could have been a mistake when merging on Bitbucket that left the changed files in a different branch (as opposed to master) within Bitbucket and you are pulling down the wrong branch. Retrace your steps until you find where the changed files are and push/pull them correctly.
   3. Since these files "exist" in every repository (prior to replacing the old file with the changed file, the old file contains some basic filler text), you will need to cat/read the file to see if it is the changed one or look at its timestamp.