Lab 4 – Versioning

Name: Paul Christiansen
Course/Section: IS-1003-001
Date: 3/18/2025

INTRODUCTION

In cybersecurity, professionals gotta be able to use and sync work across multiple platforms. In this lab we explore how to sync, create and connect a GitHub repository to our virtual machine via commands such as git and creating a local git account and then linking it to the main online GitHub account.

BREAKPOINT 1

This step was easy. While I was unable top link to an education account version, it thankfully was not needed for this lab. This was my first time using GitHub. While I have heard of it plenty of times from those in the cyber field like my uncle and father, I never actually personally used it. Navigating the forum was really easy and I was able to go down multiple rabbit holes exploring the cast amount of content and resources GitHub has to offer to its user. It was kinda mind-blowing to be honest.

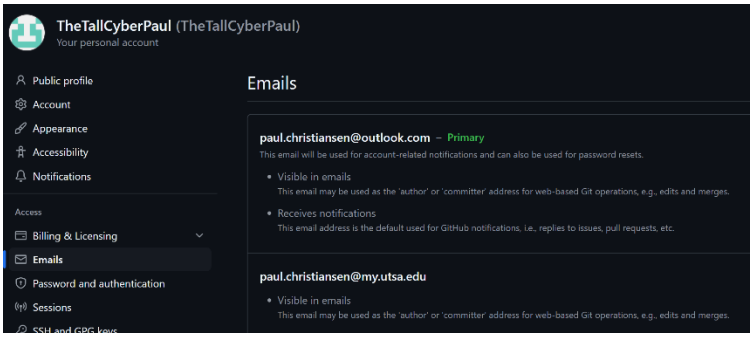

Figure 1 Git Hub Settings

BREAKPOINT 2

This was a very straight forward breakpoint. All that was required was setting up a connection that used a username and the SAME COREESPONDING email to my GitHub account. This is how we will be able to do the rest of the lab, so if someone were to mess up this part, they would have to do the entire lab from the beginning.

```
adarkpoet@pop-os:~$ git config --global user.name "adarkpoet"
adarkpoet@pop-os:~$ git config --global user.email "paul.christiansen@outlook
com"
adarkpoet@pop-os:~$ git config --list
user.name=adarkpoet
user.email=paul.christiansen@outlook.com
adarkpoet@pop-os:~$ cd documents
bash: cd: documents: No such file or directory
adarkpoet@pop-os:~$ cd Documents
adarkpoet@pop-os:~/Documents$ ls
adarkpoet@pop-os:~/Documents$ mkdir first-repo
adarkpoet@pop-os:~/Documents$ ls -l
total 4
drwxrwxr-x 2 adarkpoet adarkpoet 4096 Mar 18 19:25 first-repo
adarkpoet@pop-os:~/Documents$ cd first-repo
adarkpoet@pop-os:~/Documents/first-repo$
```

Figure 2 Creating Git Configuration

BREAKPOINT 3

So, this will provide a complete overview of the proceeding screenshots below. Figure 3 is simply creating the necessary directory to be able to do the lab. You also can see the process of adding the global user to be able to do the lab. During this step, it was important to make the directory the EXACT same name as the repository in GitHub, so that is why it is formatted the way it is.

```
adarkpoet@pop-os:~$ git config --global user.name "adarkpoet"
adarkpoet@pop-os:~$ git config --global user.email "paul.christiansen@outlook
com"
adarkpoet@pop-os:~$ git config --list
user.name=adarkpoet
user.email=paul.christiansen@outlook.com
adarkpoet@pop-os:~$ cd documents
bash: cd: documents: No such file or directory
adarkpoet@pop-os:~$ cd Documents
adarkpoet@pop-os:~/Documents$ ls
adarkpoet@pop-os:~/Documents$ mkdir first-repo
adarkpoet@pop-os:~/Documents$ ls -l
total 4
drwxrwxr-x 2 adarkpoet adarkpoet 4096 Mar 18 19:25 first-repo
adarkpoet@pop-os:~/Documents$ cd first-repo
adarkpoet@pop-os:~/Documents/first-repo$
```

Figure 3 Creating Directory

In Figure 4, I am creating the necessary README file for this lab. This is what we will be using to push (to send) to my GitHub account. What you see me doing is putting some text (echo) into the read me so I can call it and make sure the file actually exists, which I verify afterwards that it does exist.

```
adarkpoet@pop-os:~/Documents/first-repo$ echo "I exist" >> README.md
adarkpoet@pop-os:~/Documents/first-repo$ cat README.md
I exist
adarkpoet@pop-os:~/Documents/first-repo$ ls -l
total 4
-rw-rw-r-- 1 adarkpoet adarkpoet 8 Mar 18 19:26 README.md
adarkpoet@pop-os:~/Documents/first-repo$
```

Figure 4 Creating README

In figure 5 you see me use two new commands. Add and commit. What add does is add “x” change you just made to the staging area so it can be pushed or pulled to wherever you want that change to go. In this case, we added the README file to the LOCAL git repository we created in the virtual machine. This will eventually allow us to transfer it to our GitHub account at a later breakpoint. Commits are kinda like attempts to do a task. It’s our way of telling git to do “x” action. What it does is does the specified action and then records that action if completed successfully.

```
adarkpoet@pop-os:~/Documents/first-repo$ git init
hint: Using 'master' as the name for the initial branch. This default branch n
ame
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/adarkpoet/Documents/first-repo/.git/
adarkpoet@pop-os:~/Documents/first-repo$ git add README.md
adarkpoet@pop-os:~/Documents/first-repo$ git commit -m "first commit"
[master (root-commit) 1b0bc9e] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
adarkpoet@pop-os:~/Documents/first-repo$
```

Figure 5 Doing First Commit

Finally in figure 6, you see me use the branch command. What this does is tell the repository to move to "x" commit in the project. Its our way to not only move through different parts of a project but also how we group a new change made to a project. It creates a new branch from the commit we did when we did "x" task. In this case we just told git to bring us back to the main portion (the beginning) of the project so we could continue with whatever we needed to do in that specific snapshot.

```
adarkpoet@pop-os:~/Documents/first-repo$ git branch
* master
adarkpoet@pop-os:~/Documents/first-repo$ git branch -M main
adarkpoet@pop-os:~/Documents/first-repo$ git branch
* main
adarkpoet@pop-os:~/Documents/first-repo$
```

Figure 6 Using Branch command

BREAKPOINT 4

This breakpoint was specifically creating a secure connection before we remoted into the GitHub account. To do that, we created an SSH Key. An SSH Key, Secure Shell, is a type of key that provides an encrypted connection when we need to remote access into an environment that does not have a secure connection or are possible vulnerable areas in the network. More specifically, SSH is a network protocol that allows us to do this and what we do is generate an encryption key to allow us to have that secured connection into GitHub. With that said, figure 7 shows us generating the key (I had one previously) where I overrode an old key to generate a new one. What the generation provides is the following key information: It provides the key's fingerprint, allows us to track connection, and the image of the key. Its kinda like the code for the encryption but visually displayed.

```
/home/adarkpoet/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/adarkpoet/.ssh/id_ed25519
Your public key has been saved in /home/adarkpoet/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:+eqGY1A/MBfS1ftSbaBwXMWq/3v5sn8/myxfOME/GeE paul.christiansen@outlook.c
om
The key's randomart image is:
+--[ED25519 256]--+
|        . ..o ..o.|
|       . o . + . .|
|        . . o o = |
|       + ..  o.+ +|
|      . =S    +oE |
|     .   o.  o .+o|
|      . . ..  ooo+|
|       + ..   .+++|
|      . +o     +@&|
+----[SHA256]-----+
adarkpoet@pop-os:~/Documents/first-repo$
```

Figure 7 SSH Key Generation

Figure 8 below shows the fact that the key we generated earlier does exist but more importantly, shows the key that we need to give GitHub to generate the secure connection we need for this lab. We see this key by using the cat (concatenate) command which displays the contents in the files.

```
+----[SHA256]-----+
adarkpoet@pop-os:~/Documents/first-repo$ cd /home/adarkpoet/.ssh
adarkpoet@pop-os:~/.ssh$ ls -al
total 24
drwx------  2 adarkpoet adarkpoet 4096 Mar 18 19:16 .
drwxr-x--- 16 adarkpoet adarkpoet 4096 Mar 18 19:28 ..
-rw-------  1 adarkpoet adarkpoet  419 Mar 18 19:39 id_ed25519
-rw-r--r--  1 adarkpoet adarkpoet  111 Mar 18 19:39 id_ed25519.pub
-rw-------  1 adarkpoet adarkpoet  978 Mar 18 19:16 known_hosts
-rw-r--r--  1 adarkpoet adarkpoet  142 Mar 18 19:16 known_hosts.old
adarkpoet@pop-os:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBP7DrCbozA3hx9bGgSuB2idZQG0xircimxVEeX9Bm
wf paul.christiansen@outlook.com
adarkpoet@pop-os:~/.ssh$
```

Figure 8 Calling Key File

BREAKPOINT 5

The final breakpoint is putting all of the setup we have done in the last four breakpoints to allow us to push and pull commits done from either the virtual machine or from GitHub. The first thing I do, in figure 9, is establish where I am remoting into. The terminal confirms that I can either push (upload) from the git that I created in the VM to GitHub, or I could pull (download) from the GitHub to my VM. To start, I uploaded (push) the README file we created in the third breakpoint.

```
adarkpoet@pop-os:~/.ssh$ cd ~
adarkpoet@pop-os:~$ cd Documents/first-repo
adarkpoet@pop-os:~/Documents/first-repo$ ls
README.md
adarkpoet@pop-os:~/Documents/first-repo$ git remote -v
origin  git@github.com:TheTallCyberPaul/first-repo.git (fetch)
origin  git@github.com:TheTallCyberPaul/first-repo.git (push)
adarkpoet@pop-os:~/Documents/first-repo$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 223 bytes | 223.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:TheTallCyberPaul/first-repo.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
adarkpoet@pop-os:~/Documents/first-repo$
```

Figure 9 Pushing to GitHub

In the figure below, you can see that the push was successful as the README file was uploaded to the GitHub repository with the correct string we echoed earlier.

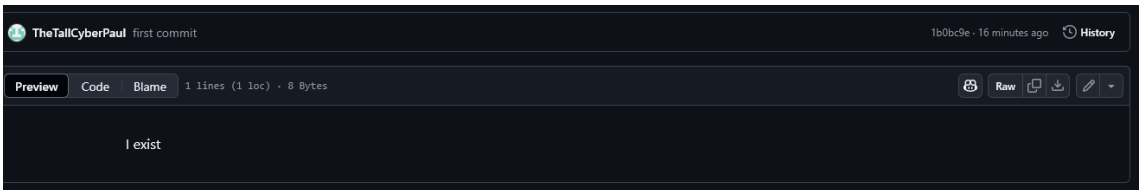

Figure 10 Results of Figure 9

Now, what happens if we make a change on the GitHub side but want the change on our VM? Well, we do the opposite of a push which is to pull. So, what I did in the figure below is (after changing the README file) is told the terminal to download (pull) the new updated file with the new information and update my files in the VM. It was successful!

```
adarkpoet@pop-os:~/Documents/first-repo$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 903 bytes | 903.00 KiB/s, done.
From github.com:TheTallCyberPaul/first-repo
 * branch            main       -> FETCH_HEAD
   1b0bc9e..f23736d  main       -> origin/main
Updating 1b0bc9e..f23736d
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
adarkpoet@pop-os:~/Documents/first-repo$
```

Figure 11 Results of the Pull Command

CONCLUSION

I remember seeing the numerous announcements about how this lab was going to be difficult and to expect to do a bunch of troubleshooting was fairly common. Except I didn't run into a single issue when going through the lab. I did it as I was reading it for the first time. It was unsettling to do as I was always expecting to hit some type of fatal error that would take hours to troubleshoot but I never ran into that. The hardest thing about the lab actually was trying to remember my account password since I hadn't touch the VM in so long. So, that ended up in me reinstalling stiff from lab 2 to make sure I have the correct environment funnily enough.

Outside of that, this was my first time using GitHub and I was pleasantly surprised at how easy it was to use. As I mentioned in my breakpoint, my uncle and my dad had used it before I had. Due to that, I thought it was something you had to have experience and extensive technical knowledge to use but boy was wrong. So, what I have learned and what I will be doing is exploring what GitHub has to off and to start using the platform more often!

REFERENCES

R. Mitra, "*Lab 4: Versioning*," The University of Texas at San Antonio (2024). Last accessed: *March 18, 2025*.

COLLABORATION

I worked by myself.