ECED4406 – Lab #4

For Lab #4, you will analyze a password check program using power analysis.

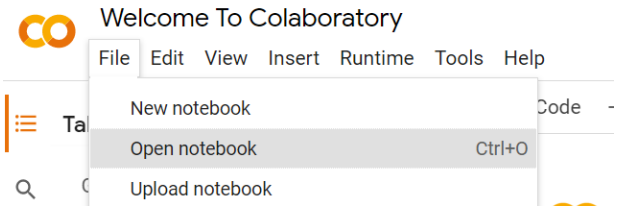
NOTE: This is called "lab2" in previous versions, you may see references to that.

Step 1: Jupyter Introduction

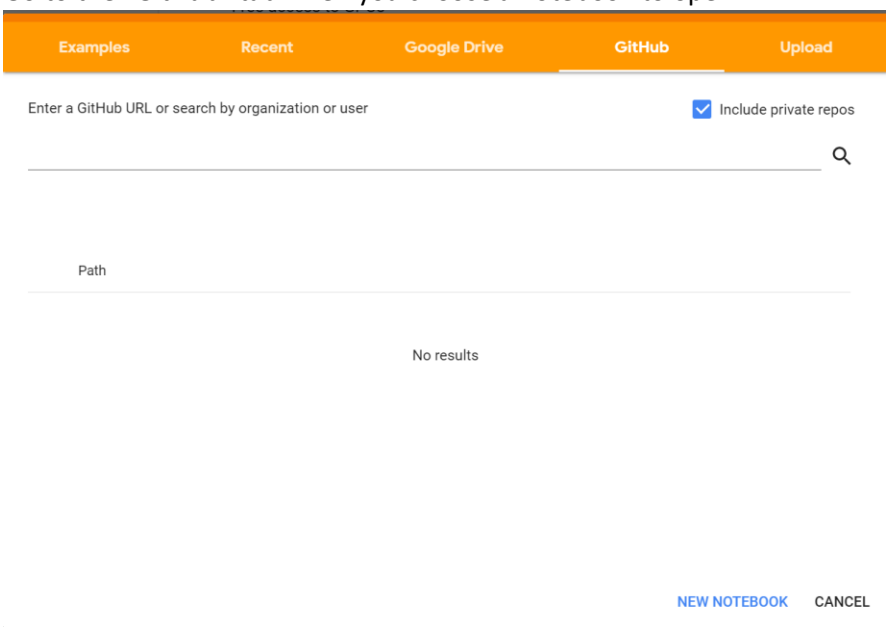
We introduced Jupyter in a previous lab – for reference, see the following video:
https://www.youtube.com/watch?v=v5W8Uff4x0Q

If you wish to use the Colab interface (requires a Google account), use the following instructions:

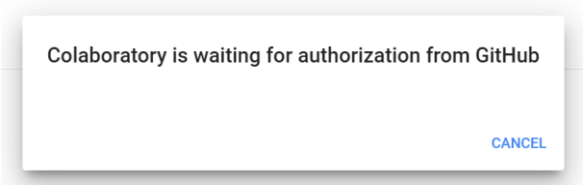
1. Select "Open Notebook"



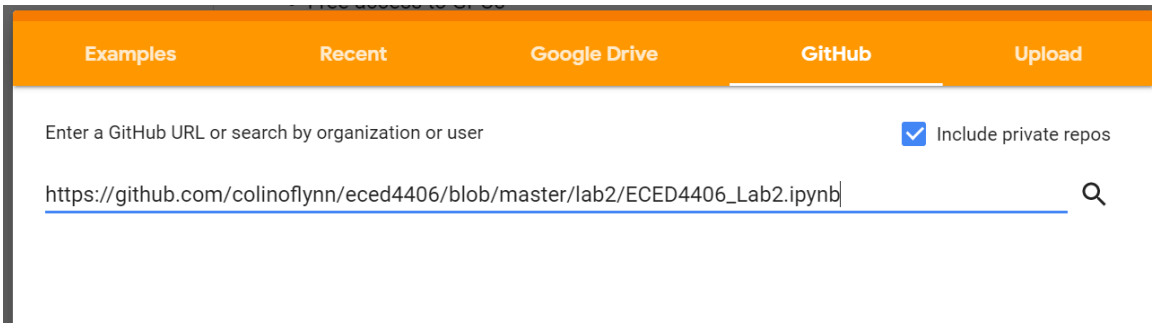
2. Go to the "GitHub" tab when you choose a notebook to open:



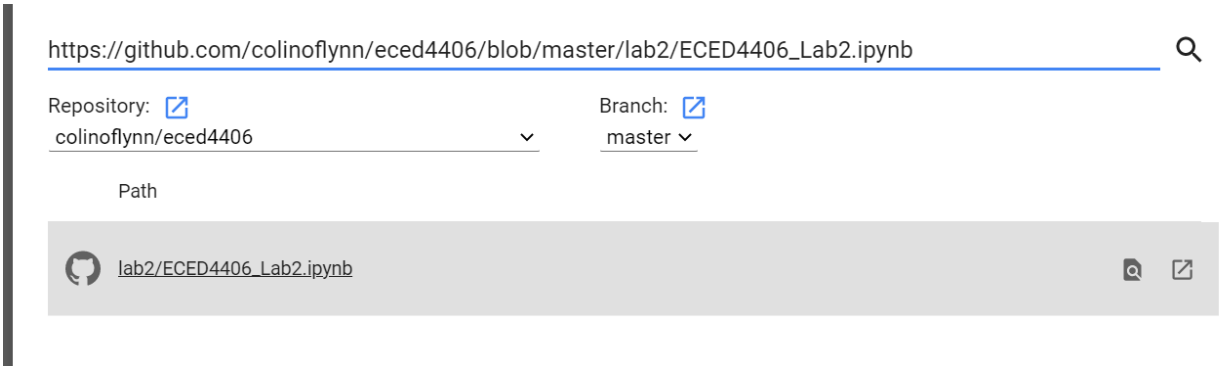
NOTE: If it asks to "Authorize with GitHub", you can skip that. We don't need to perform that step. Hit CANCEL **and close the window that pops up.**



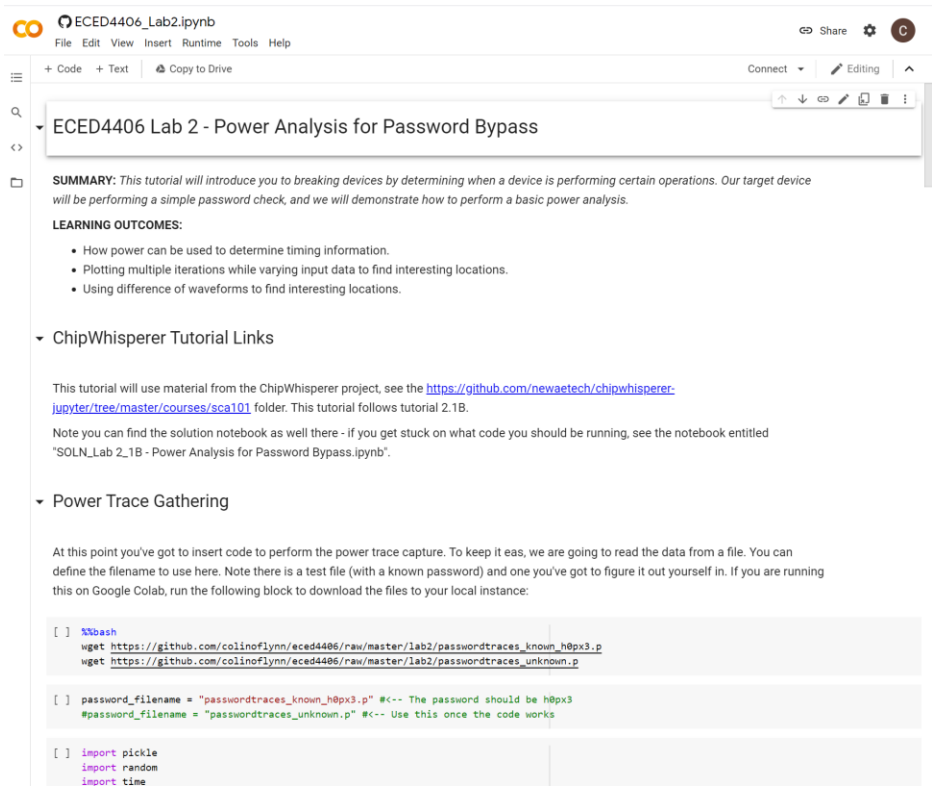
3. Copy the url **https://github.com/colinoflynn/eced4406/blob/master/lab4/ECED4406_Lab2.ipynb** into the search field, and press enter.



4. Click the link in the resulting search results



5. You should be rewarded with a notebook interface as shown:

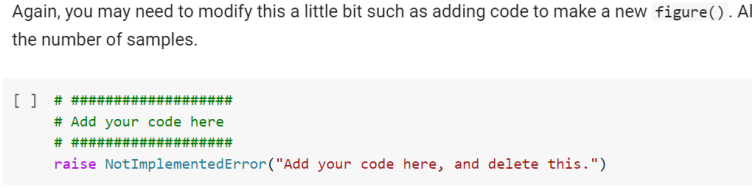


NOTE: You can also install Jupyter directly and run the **.ipynb** file. For many students it will be easier to run on the cloud platform.

Step 2: Lab Procedure

Building Attack

Within the lab, you'll find various "hints" about how to perform the attack. Working through these blocks, fill in the required functions:



Attack Hints

Where to find information if you can't get the right answer?

This lab is based on my ChipWhisperer labs – you can find a solution for a similar lab at https://github.com/newaetech/chipwhisperer-jupyter/blob/master/courses/sca101/SOLN_Lab%202_1B%20-

%20Power%20Analysis%20for%20Password%20Bypass.ipynb . This will be helpful in figuring out what is required at each block.

Performing Attack on Test vs Real Password Traces

There are two files in the folder - **passwordtraces_known_h0px3.p** and **passwordtraces_unknown.p**

You can use the "known" traces which have the password h0px3. The unknown password is also a 5-character password, but is a different password. Try to figure out the unknown password.

Questions to Answer [30 pts]

1. ***Freebie due to lab error [5 pts]***

2. Show a plot of a correct vs. uncorrect password guess. [5 pts]

3. Can you think of a way to perform the password check without the timing problem? [5 pts]

   NOTE: The code in question is here: https://github.com/newaetech/chipwhisperer/blob/develop/hardware/victims/firmware/basic-passwdcheck/basic-passwdcheck.c#L103 which looks like the following:

```
for(uint8_t i = 0; i < sizeof(correct_passwd); i++){
    if (correct_passwd[i] != passwd[i]){
        passbad = 1;
        break;
    }
}
```

   If you can modify the code to *avoid* the conditional you can succeed. There are a few ways to do this – architectural changes, C code changes, or viewing the assembly output. All of them are acceptable in this question.

4. Describe the program flow of your attack script. Include your software solution, making a flow-chart showing the flow, and/or bringing in some example results such as intermediate plots. [15 pts]

   NOTE – The Python notebook is provided as a learning tool. You can implement your solution as a "raw" python script if you wish, or you can simply use the provided Jupyter framework. Either way, you should include snippets of your source code to demonstrate how you are detecting the differences.