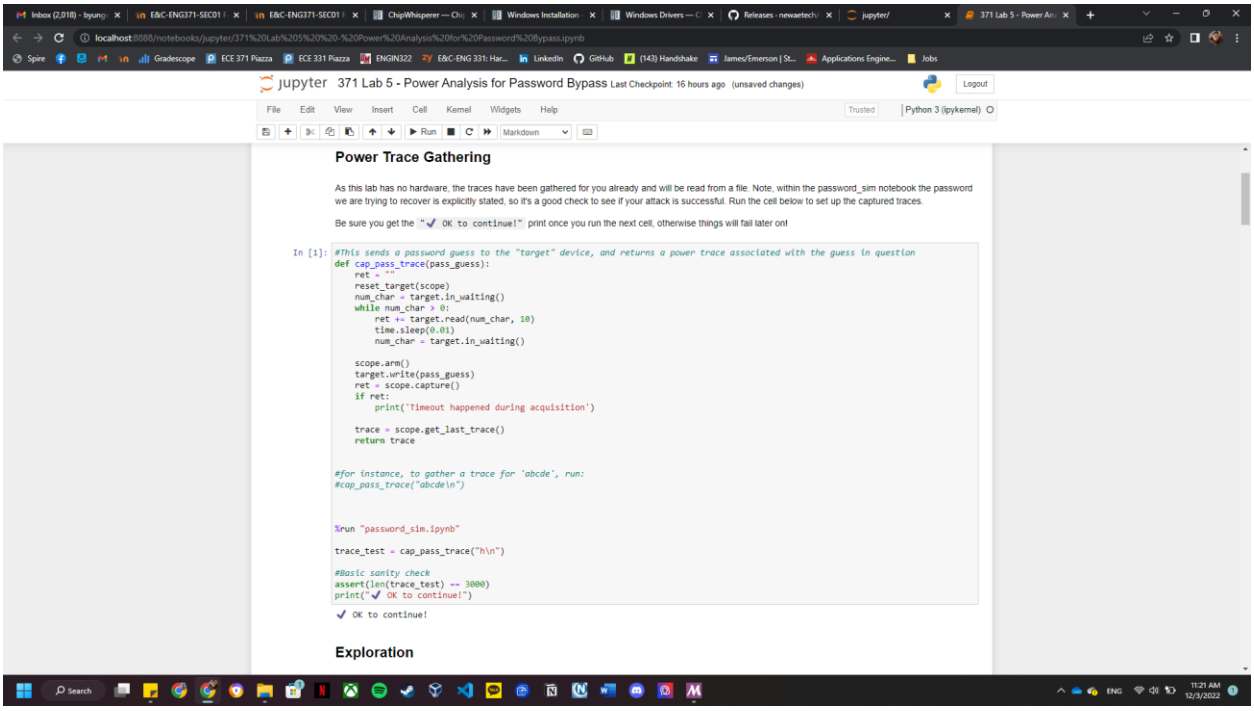
Byung Woong Ko

Jeffrey Song

Lab5 Report

In this lab, we installed chipwhisperer and implemented side channeling attack which utilized DPA to find the “correct” password. Chipwhisperer was utilized to collect traces and help us utilize and analyze the attack. The password was stored in passwords_sim.ipynb and was provided in passwords_full.p Tutorial helped us get familiar with the Jupyter Notebooks environment. The lab helped us step by step on how DPA can be utilized to perform side channeling attack. Details of each implementations are described below:

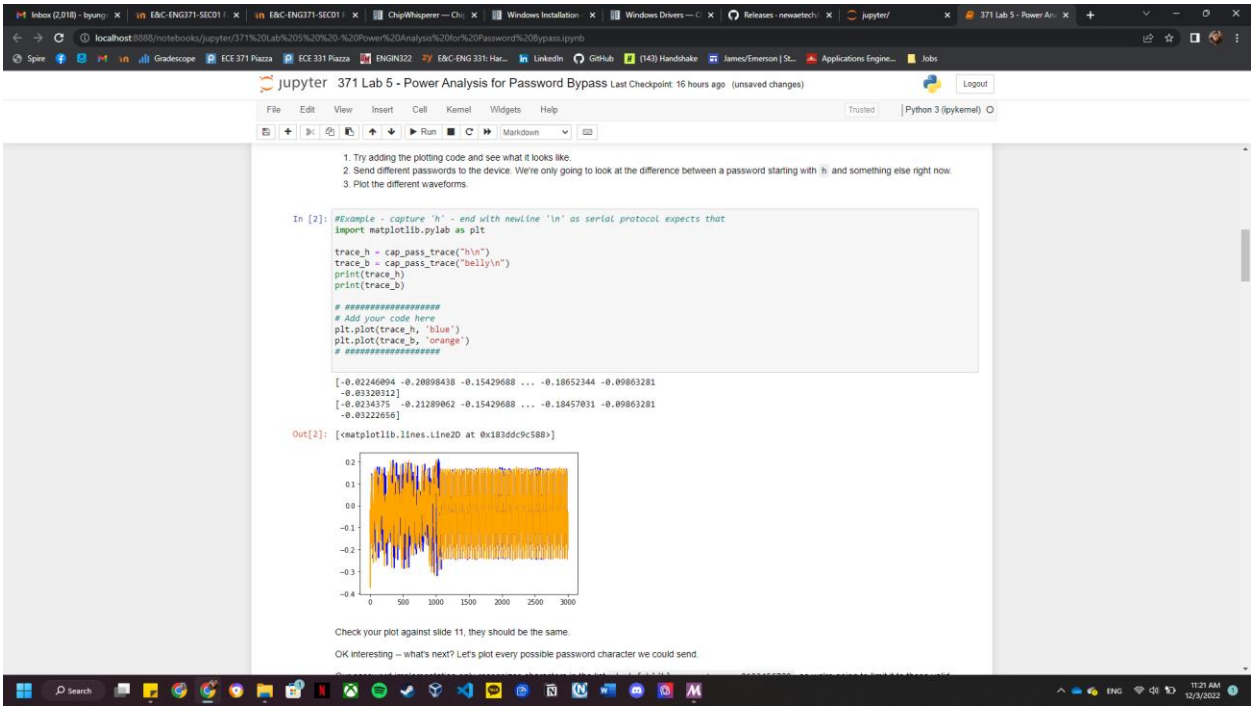


X axis: None

Y axis: None

Attack: not made

Description: Loads “secret code” from password_sim.ipynb

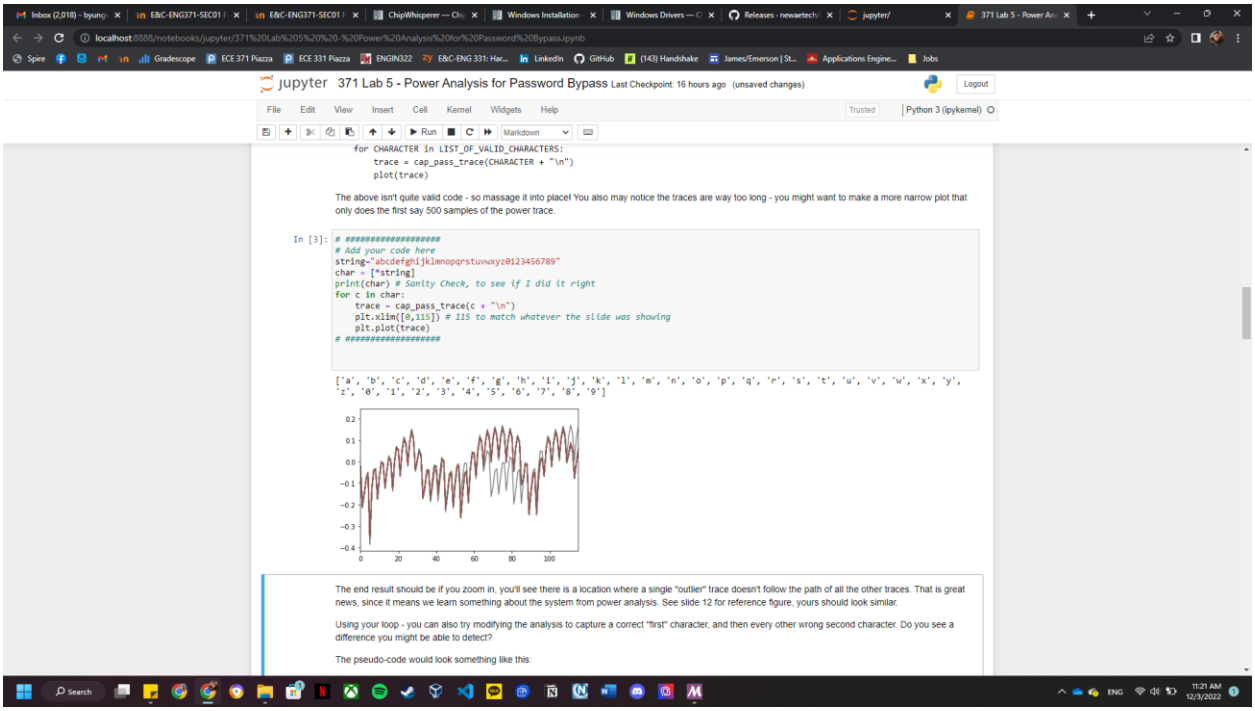


X axis: samples of power trace

Y axis: power of “h” trace in orange and power of “belly” trace in blue

Attack: unsuccessful

Description: I added another trace (print as sanity check) and plot the trace in different colors.

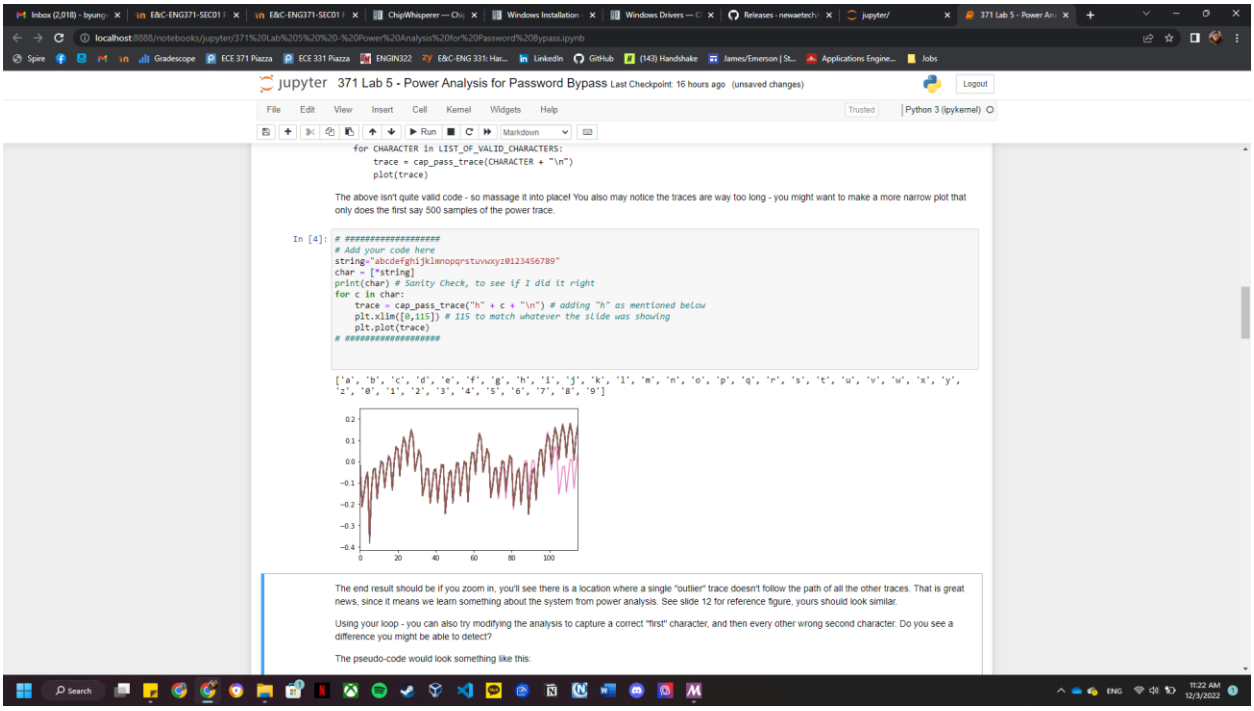


X axis: samples of power trace

Y axis: power of traces made with each variables in list char

Attack: somewhat successful (was not successful because it we did not hack the entire password).

Description: Added a string of all possible characters (alphabet and numbers), traced them, and plotted the samples. Found 1 trace “h” that had different power value than the rest (denoted in grey). Made the plot to be 115 samples so that it resembles what was shown in the example slide.

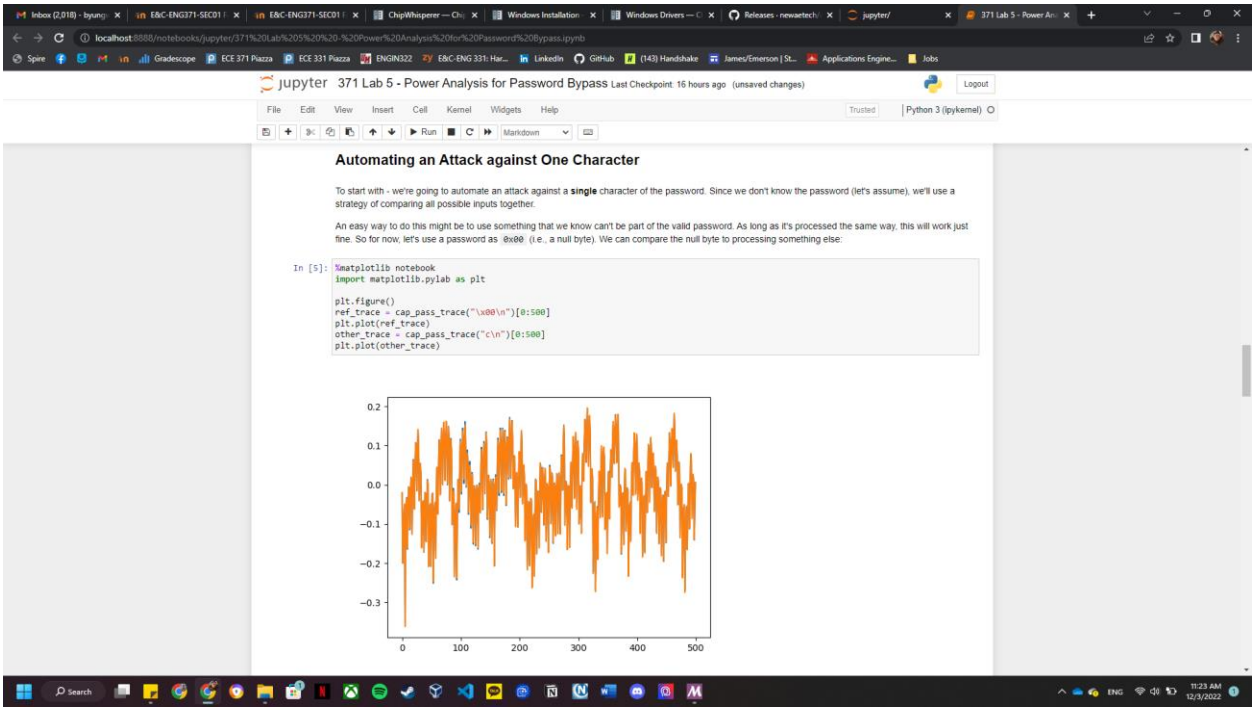


X axis: samples of power trace

Y axis: power of traces made with “h” + each variables in list char

Attack: somewhat successful (was not successful because it we did not hack the entire password).

Description: Added a string of all possible characters (alphabet and numbers) and plotted them. Little difference from the above code was that I added “h” before the next character. This is because “h” was a given first correct letter in the code. Found 1 trace that had different power value than the rest (denoted in pink). Made the plot to be 115 samples so that it resembles what was shown in the example slide.

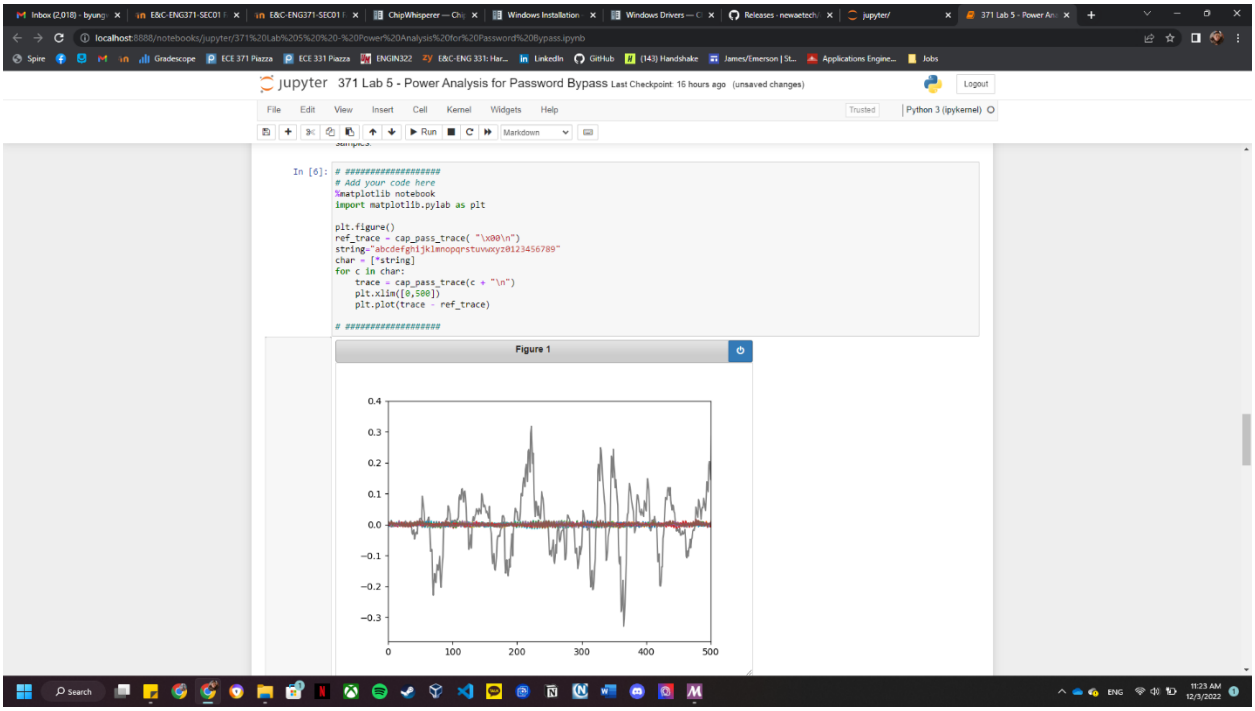


X axis: samples of power trace

Y axis: power of traces made with null byte and “c”

Attack: unsuccessful

Description: given code which creates a figure of trace samples of null and trace samples of “c” which are both not the correct password. The results show that there are very few differences between null and c character. Limited to 500 samples.

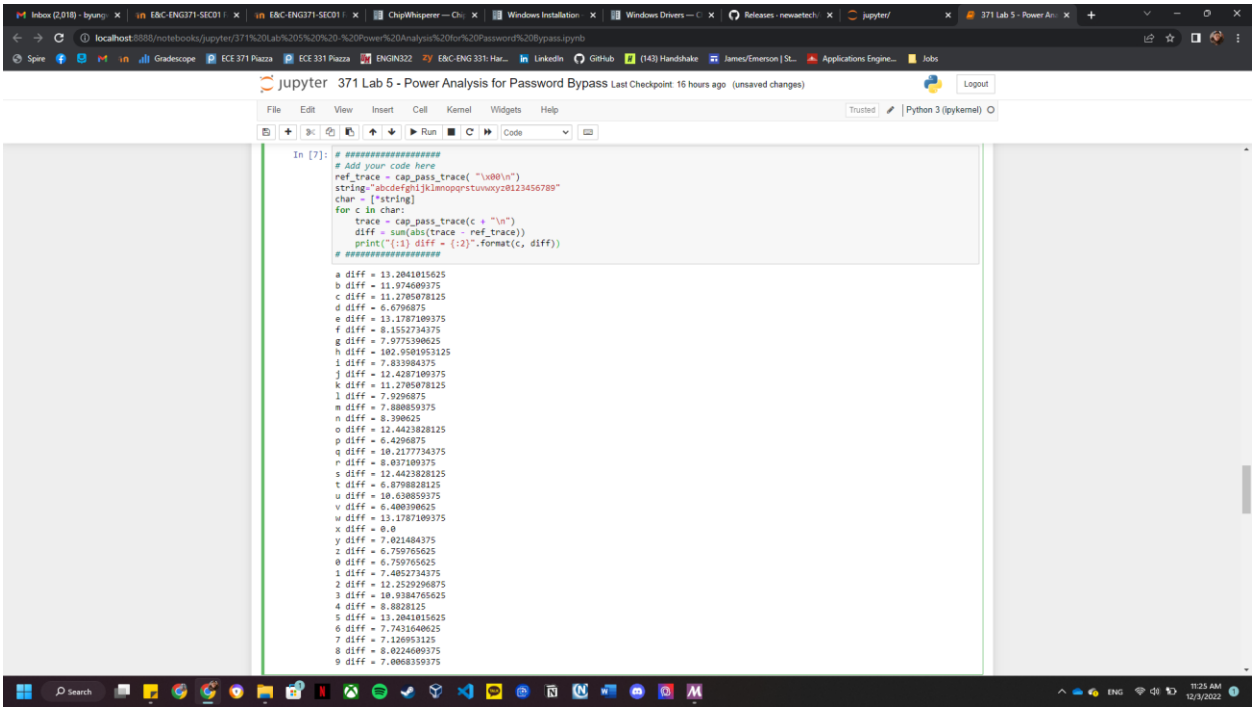


X axis: samples of power trace

Y axis: difference in power of traces made by each character traces - null byte traces

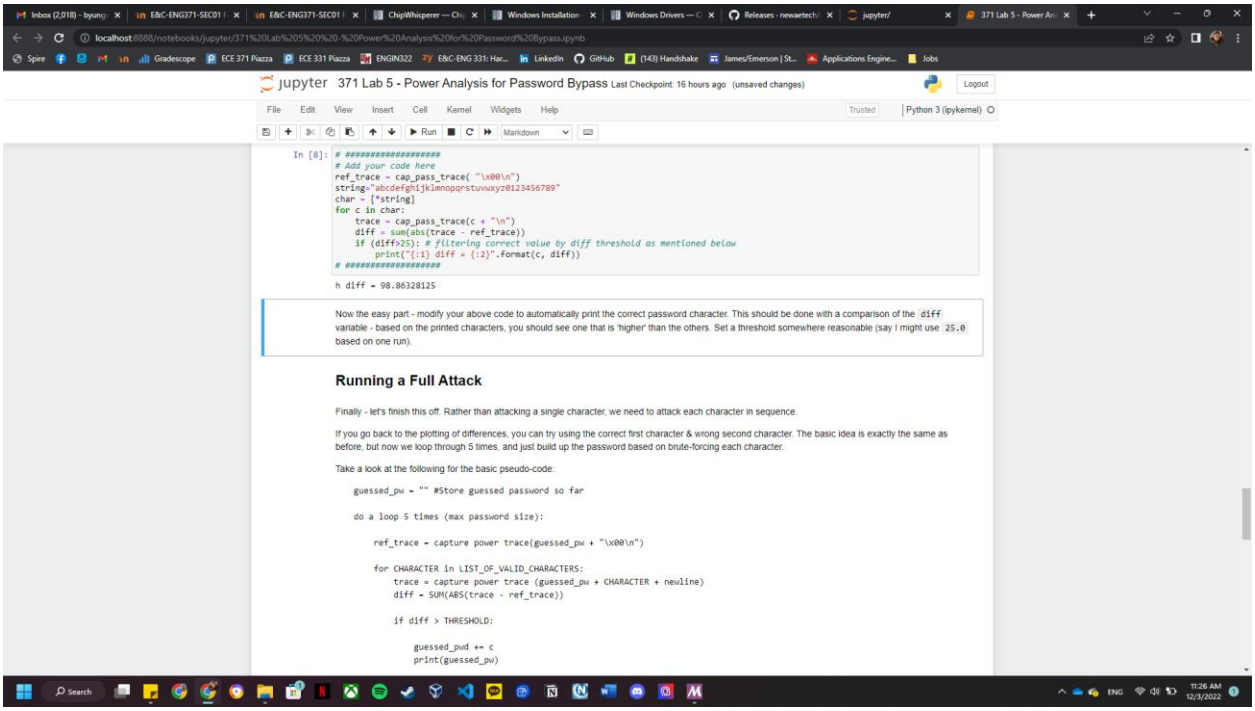
Attack: somewhat successful (was not successful because it we did not hack the entire password).

Description: Added a string of all possible characters (alphabet and numbers) and traced them. We also traced null byte then plotted the found difference between all character traces vs null byte trace. If the trace was “wrong” then the result should be somewhat similar to the null byte trace, which is why if we subtract them, it will result to something very close to zero. The correct trace however, will stand out. Limited to 500 traces.



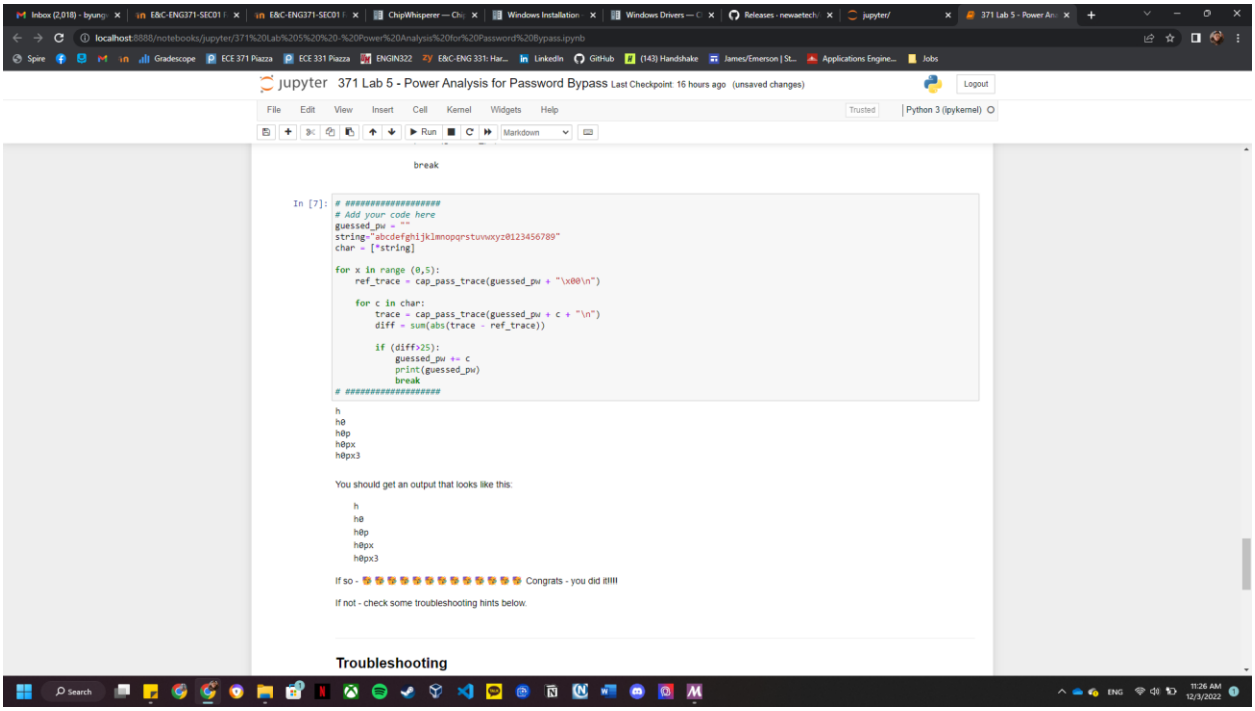
Attack: somewhat successful (was not successful because it we did not hack the entire password).

Description: Added a string of all possible characters (alphabet and numbers) and traced them. We also traced null byte then found the difference between all character traces vs null byte trace. If the trace was “wrong” then the result should be somewhat similar to the null byte trace, which is why if we subtract them, it will result to something very close to zero. The correct trace however, will stand out. So we print out the all the character, then the variable diff (diff is a list of each character in char list vs null byte. We sum the absolute value of all sample traces made from the single character value – null byte samples and store it in diff. Most characters will result to something close to zero while correct one will reach somewhere around 100). Limited to 500 traces.



Attack: somewhat successful

Description: Same thing as above, but we modify the code to show only the value where the diff is greater than a certain threshold, we used 25 as mentioned in the jupyter lab note. The result showed only 1 trace so the attack was somewhat successful (was not successful because it we did not hack the entire password).



Attack: successful

Description: using knowledge acquired above, and knowing there is only 5 characters to the password, we created a loop that goes over 5 times, where each loop will iterate through all the characters in char, find the difference between the character trace and null trace, sum the absolute value of the sample list in diff, and find the “correct” char value. The char value will then be added to a “answer string” called guessed_pw. If new character is added, it goes to the back of the char list (which is a string). In the next loop, the guess_pw will be added to both null and the next iteration of char trace(i.e. guessed_pw + null and guessed_pw + each char value). We also add guessed_pw to null bit as well so that we can average out the power difference of the first correct character. We loop this 5 times which where guessed_pw will eventually have more and more “correct” characters until the 5th loop, where it will have all the correct characters in order.