Computer Security Concepts, COMP 435, Fall 2019
Instructor: Cynthia Sturton
Release Date: 8/29/2019
Due Date: 9/8/2019, 11:59 PM

Introduction to Python

Overview

The purpose of this assignment is to give you time to set up the environment you will use in subsequent labs and to become familiar with Python, which you will also need in the next assignment.

Requirements

You will be writing a program in Python and will need to have Python 2.7 installed. If you're working on a Mac, Python should be preinstalled; you can check that it's the correct version by entering `python --version` in a terminal.

It can also be downloaded at https://www.python.org/downloads/release/python-2716/

Be sure **not** to download a newer version of Python (>=3.0). The autograder will be running on 2.7, and there may be differences in the expected output.

A good starting point for learning Python is here: `https://docs.python.org/3/`. Some students may find an IDE for Python helpful. There are a number online, such as iPython and PyCharm, which you are encouraged to explore.

Detecting Duplicates

For this assignment you will be detecting duplicate hex values in a large text file. You will sort them by value and then display how many duplicates are present.

For example, let's say you have the following text file name "sample.txt":

```
9ec4c12949a4f31474f299058ce2b22a
9ec4c12949a4f31474f299058ce2b22a
43
aa
02
123
acd
aa
123
123
02
```

Your program should parse through the entire file, and for each value that is in the file more than once, it should print out the number of instances of that value. In this case the correct output will be:

```
2 2
aa 2
123 3
9ec4c12949a4f31474f299058ce2b22a 2
```

Note that the output is sorted from the smallest value to the largest value in hex. **Also note that leading zeros are omitted even if they were present in the input.** Your program must take in the file to read as a command line argument. For example:

```
python a1.py sample.txt
```

There are many ways to complete the assignment. You should use the built-in data structures that Python provides (e.g., sets or dictionaries). You should not be looping through the hex values multiple times to process the values or look for duplicates.

Grading Rubric

5/7 points for correctness
2/7 points for style

Submission Instructions

Name your Python program **a1.py** and submit it on gradescope. You may submit as many times as necessary. Make sure your program is named correctly, or the grader will not find it.