Project 2

COSC 2409 – Python Programming

Download the files for the project. This includes a .pdf with these instructions, as well as a couple of example scripts and the Scan.txt file you will need to use.

First, look over the text file: “scan.txt”

This is an actual scan of a few sub-nets on the network here at LCCC. Lots of data.

Your project will be to create another text file with condensed data from this file.

You should do the versions in order. Turn in each version as you get them done.

First, the basic version (you will get a C if done well)

Note that each station in the network is identified on a line that begins with “Nmap scan report for…” and then includes information that may or may not include a name, but does include the IP address. Then, there is more information about that station including open ports.

You are to create a second text file with ONLY the lines that begin with the item ‘Nmap’

Write a script with two functions that

1. Opens the scan.txt file for reading.
2. Opens IPLog.txt for writing.
3. Initializes a count of IP lines to 0
4. Reads all the lines from Scan.txt into a list, say lineList, using readLines().
5. Sends this list off to a function called logIPs(lineList). Note: lineList is a list sent as a parameter. You may name it anything you wish.
6. Closes the two files
7. Outputs to the console the count of IP lines

The function logIPs(lineList)

1. For each string (line) in the list,
    a. send it off to checkLine(line)

The function checkLine(line)

2. splits the string into a list of items using split
3. if the # of items is > 0
    a. if the first item is ‘Nmap’
        i. write the line to the output file
        ii. update the count

Intermediate version (you will get a B if done well)

You will note that in the Scan file some Nmap lines just have the IP address at the end, others have the computer name and the IP address in ( )'s.

First, study and review indexing a string and taking a 'slice' as explained in chapter 6, and demonstrated in "sampleStringSlice.py" provided in the project download.

Modify the Basic version by adding a third function that will take (parameter) a single string, which is either the actual IP address: '137.87.51.76' for example, or the IP address in ( )'s: '(137.87.51.76)' for example. Determine which it is, (check the first character) and if there are ( )'s strip them off using a slice. Return the IP address without ( )'s. If the string does not have ( )'s, simply return it as is.

Back in the checkLine function, when you find a line that starts with "Nmap" send the last item in the list created by split to this third function, and write the string that comes back to the IPLog.txt file instead of the entire line.

IPLog.txt should then have only IP addresses, nothing else, each on a line.

Advanced version (you will get an A if done well)

Note that for each node on the network, a list of open ports, if any, and the service associated with it is listed.

Open and read all lines from ‘Scan.txt’. Create a text file called “Ports.txt” that lists all the services associated with open ports, and the number of stations that have that service listed on an open port. Do not list duplicates, but count the number for each service. See the “parallelLists.py” script for an example of how to do this.

Just count open ports. Ignore filtered or closed ports.

I leave it up to you on how to determine if a line indicated an open port and what service it is being used for.

Use functions as appropriate.

Hint: There should be 25 different services, of those “ssh” should be counted 13 times