## APPENDIX

### ROTHKO SOURCE CODE

\# Rothko takes to two files as input and writes the

\# output to result.txt in the current directory

import sys

firstarg = str(sys.argv[1])

secondarg = str(sys.argv[2])

\# Read first file into an list called firstfilelist

firstfilelist = []

with open(firstarg, "r") as f:

for line in f:

firstfilelist.append(line)

\# Read first file into an list called secondfilelist

secondfilelist = []

with open(secondarg, "r") as f:

for line in f:

secondfilelist.append(line.splitlines())

\# Open file for writing

output = open('result.txt', 'w')

\# Loop through the first list

x = 0

for y in firstfilelist:

currentrow = firstfilelist[x]

\# Create a sting and strip the list chars

currentrow = str(currentrow)

currentrow = currentrow.strip("'")

currentrow = currentrow.strip('[')

currentrow = currentrow.strip(']')

currentrow = currentrow.strip("'")

currentrow = currentrow.rstrip('\r\n')

\# Loop trhough the second list

a = 0

for b in secondfilelist:

secondcurrentrow = secondfilelist[a]

secondcurrentrow = str(secondcurrentrow)

secondcurrentrow = secondcurrentrow.strip("'")

secondcurrentrow = secondcurrentrow.strip('[')

secondcurrentrow = secondcurrentrow.strip(']')

secondcurrentrow = secondcurrentrow.strip("'")

result = (str(currentrow)+str(secondcurrentrow))

print result

a = a+1

\# And write the combined result to result.txt in the current directory

output.write(result+'\n')

x = x+1

### FOLDERBOULDER SOURCE CODE

#!/usr/bin/python

#FolderBoulder is a tool designed to enumerate folders and files

#on web servers.

import sys

import requests

# List of valid URLs found to exist

uncoveredURLs = []

# List for number of tried URLs

requestCounter = []

def displayHelp():

print 'Usage: folderboulder.py <http://hostname> <foldernames> <filenames>'

print ''

print 'Example folderboulder.py foldernames.txt filenames.txt'

def openFolderFiles(foldernames,filenames,hostname):

#Read filenames into list

listOfFiles = []

with open(filenames) as fn:

for filenames in fn:

listOfFiles.append(filenames.rstrip())

#Open file of folder names

with open(foldernames) as f:

for foldername in f:

foldername = foldername.rstrip()

#Loop thru the listOfFiles list and append it to the URL

for currentFileName in listOfFiles:

currentFileName = currentFileName.rstrip()

createHTTPrequest(hostname,foldername,currentFileName)

# Create the HTTP request and print the result

def createHTTPrequest(hostname,foldername,currentFileName):

r = requests.get(hostname+'/'+foldername+'/'+currentFileName)

print('Trying: '+r.url+' HTTP Response:'+str(r.status\_code))

requestCounter.append('x')

if str(r.status\_code) == '200':

uncoveredURL(r.url)

# Add any URL that gets a 200 response to the uncoveredURLs list

def uncoveredURL(uncoveredURL):

uncoveredURLs.append(uncoveredURL)

# Print the final result

def printResults():

if len(uncoveredURLs) > 0:

print ''

print 'RESULT'

print 'Number of requests made: '+str(len(requestCounter))

print 'Number of Valid URLs found: '+str(len(uncoveredURLs))

print 'The following URLs returned status code 200 and should be valid'

for l in uncoveredURLs:

print l.rstrip()

else:

print ''

print 'RESULT'

print 'Number of requests made: '+str(len(requestCounter))

print 'Number of Valid URLs found: '+str(len(uncoveredURLs))

print 'The server returned no 200 responses. Try another folder and file name list'

# A very quick and dirty way of "handling" missing command line arguments

if(len(sys.argv) < 3):

displayHelp()

try:

hostname = str(sys.argv[1])

foldernames = str(sys.argv[2])

filenames = str(sys.argv[3])

openFolderFiles(foldernames,filenames,hostname)

printResults()

except:

print ''

### NSLOOKUP LOOP SOURCE CODE

#! /bin/sh

while true

do

nslookup example.com

sleep 5

done

### POLLOCK SOURCE CODE

import requests

### DEFINE FUNCTIONS

# Get geo data on the ip address

def getGeoInfo(ipAddress):

r = requests.get('http://ip-api.com/json/'+ipAddress)

geoData = r.json()

return geoData

# Iterate the data and grab the things we need

def filterGeoData(geoData):

lon =''

lat =''

ipAddress =''

country =''

city =''

isp =''

# print geoData

for key,value in geoData.iteritems():

if key == 'lon':

lon = value

if key == 'lat':

lat = value

if key == 'query':

ipAddress = value

if key == 'country':

country = value

if key == 'city':

city = value

if key == 'isp':

isp = value

return lon, lat, ipAddress, country, city, isp

###

# Open file of IP addresses from ip.txt in the local directory

try:

fh = open('ip.txt','r')

except IOError as e:

print e

raise

for x in fh:

x = x.rstrip()

# Send the IP addresses to the API at ip-api.com to get a JSON object back

geoData = getGeoInfo(x)

# Parse the JSON object to grab only what we need

lon, lat, ipAddress, country, city, isp = filterGeoData(geoData)

# ...and print the results

print ' IP Address: '+str(ipAddress)+'\t Country: '+str(country)+'\t City: '+str(city)+'\t Lon: '+str(lon)+'\t Lat: '+str(lat)+'\t ISP: '+str(isp)