#!/usr/bin/python

import numpy as np

data\_base = []

hosts = []

freq\_final = []

freq\_200\_final = []

freq\_302\_final = []

freq\_404\_final = []

perc\_succ\_req = []

def feature\_extraction():

#Loading the file

file\_obj = open('attack\_data\_log\_sample.txt','r+')

i = 0

for line in file\_obj:

record = {}

list\_1 = line.split(' - - ')

list\_2 = list\_1[1].split('"')

list\_2[0] = list\_2[0].strip()

list\_2[2] = list\_2[2].strip()

list\_3 = list\_2[2].split()

list\_2.remove(list\_2[2])

list\_1.remove(list\_1[1])

list\_info = list\_1 + list\_2 + list\_3

i = i + 1

record['serial\_no'] = i

record['host'] = list\_info[0]

record['timestamp'] = list\_info[1]

record['request'] = list\_info[2]

record['HTTP\_reply\_code'] = int(list\_info[3])

record['Size'] = list\_info[4]

data\_base.append(record)

file\_obj.close()

def host\_seperation():

#Seperating hosts

for i in range(len(data\_base)):

hosts.append(data\_base[i]['host'])

def freq\_of\_HTTP\_req():

#Frequencies

freq\_set = set()

for i in range(len(hosts)):

freq\_set.add((hosts[i],hosts.count(hosts[i])))

freq\_set = list(freq\_set)

print (len(freq\_set))

for i in range(len(freq\_set)):

freq\_final.append(freq\_set[i][1])

def freq\_unique\_url():

#Freq unique URL

URL\_requests = []

uniq\_url = []

for i in range(len(data\_base)):

URL\_requests.append(data\_base[i]['request'])

for i in range(len(URL\_requests)):

if URL\_requests.count(URL\_requests[i]) == 1:

uniq\_url.append(URL\_requests[i])

freq\_uniq\_URL = len(uniq\_url)

return (freq\_uniq\_URL)

def freq\_200\_resp():

#hosts whose reply code was 200

freq\_200 = []

for i in range(len(data\_base)):

if data\_base[i]['HTTP\_reply\_code'] == 200:

freq\_200.append([data\_base[i]['host'],data\_base[i]['HTTP\_reply\_code']])

freq\_200\_set = set()

for i in range(len(freq\_200)):

freq\_200\_set.add((freq\_200[i][0], freq\_200.count(freq\_200[i])))

freq\_200\_set = list(freq\_200\_set)

for i in range(len(freq\_200\_set)):

freq\_200\_final.append(freq\_200\_set[i][1])

def freq\_302\_resp():

#hosts whose reply code was 302

freq\_302 = []

for i in range(len(data\_base)):

if data\_base[i]['HTTP\_reply\_code'] == 302:

freq\_302.append([data\_base[i]['host'],data\_base[i]['HTTP\_reply\_code']])

freq\_302\_set = set()

for i in range(len(freq\_302)):

freq\_302\_set.add((freq\_302[i][0], freq\_302.count(freq\_302[i])))

freq\_302\_set = list(freq\_302\_set)

for i in range(len(freq\_302\_set)):

freq\_302\_final.append(freq\_302\_set[i][1])

def freq\_404\_resp():

#hosts whose reply code was 404

freq\_404 = []

for i in range(len(data\_base)):

if data\_base[i]['HTTP\_reply\_code'] == 404:

freq\_404.append([data\_base[i]['host'],data\_base[i]['HTTP\_reply\_code']])

freq\_404\_set = set()

for i in range(len(freq\_404)):

freq\_404\_set.add((freq\_404[i][0], freq\_404.count(freq\_404[i])))

freq\_404\_set = list(freq\_404\_set)

for i in range(len(freq\_404\_set)):

freq\_404\_final.append(freq\_404\_set[i][1])

def Percentage\_diff\_req():

#Percentage of different request made

total\_freq = 0

for i in range(len(hosts)):

temp = {}

temp[hosts[i]] = hosts.count(hosts[i])

total\_freq += hosts.count(hosts[i])

perc\_diff\_req = round((((freq\_unique\_url()/total\_freq)\*100)),2)

return (perc\_diff\_req)

def Percentage\_succ\_resp():

#Percentage of successful response received by a user

freq\_200 = []

for i in range(len(data\_base)):

if data\_base[i]['HTTP\_reply\_code'] == 200:

freq\_200.append([data\_base[i]['host'],data\_base[i]['HTTP\_reply\_code']])

freq\_200\_set = set()

for i in range(len(freq\_200)):

freq\_200\_set.add((freq\_200[i][0], freq\_200.count(freq\_200[i])))

freq\_200\_set = list(freq\_200\_set)

temp\_freq = set()

for i in range(len(data\_base)):

if data\_base[i]['HTTP\_reply\_code'] == 200:

temp\_freq.add((hosts[i],hosts.count(hosts[i])))

temp\_freq = list(temp\_freq)

for i in range(len(temp\_freq)):

for j in range(len(temp\_freq)):

if (temp\_freq[i][0] == freq\_200\_set[j][0]):

perc\_succ\_req.append(round(((freq\_200\_set[j][1]/temp\_freq[i][1])\*100),2))

l=len(freq\_final)

mat=np.zeros((10,5))

def output\_display():

print ('Features extracted -- Host, Timestamp, Request, HTTP\_reply\_code, Size in bytes\n')

for i in range(len(data\_base)):

print (data\_base[i])

print ('\nExtra features\n')

print ('Freq of HTTP requests --\n')

for i in range(len(freq\_final)):

print (freq\_final[i])

mat[i,0]=freq\_final[i]

print ('\nFreq of unique HTTP request -- ', freq\_unique\_url())

print ('\nFreq of 200 code response --\n')

for i in range(len(freq\_200\_final)):

print (freq\_200\_final[i])

mat[i,1]=freq\_200\_final[i]

print ('\nFreq of 302 code response --\n')

for i in range(len(freq\_302\_final)):

print (freq\_302\_final[i])

mat[i,2]=freq\_302\_final[i]

print ('\nFreq of 404 code response --\n')

for i in range(len(freq\_404\_final)):

print (freq\_404\_final[i])

mat[i,3]=freq\_404\_final[i]

print ('\nPercentage of different requests made -- ', Percentage\_diff\_req())

print ('\nPercentage of successful requests --\n')

for i in range(len(perc\_succ\_req)):

print (perc\_succ\_req[i])

mat[i,4]=perc\_succ\_req[i]

print mat

def main():

feature\_extraction()

host\_seperation()

freq\_of\_HTTP\_req()

freq\_unique\_url()

freq\_200\_resp()

freq\_302\_resp()

freq\_404\_resp()

Percentage\_diff\_req()

Percentage\_succ\_resp()

output\_display()

return mat

print ('matrix is \n \n')

print (mat)