# Wifi Detection

import subprocess

results = subprocess.check\_output(["netsh", "wlan", "show", "network"])

results=results.decode("ascii")

results = results.replace("\r"," ")

#print(results)

a=result.count("WPA2-Personal")

print(a)

#latitude and longitude assessment

import geocoder

g = geocoder.ip('me')

print(g.latlng)

Sample Output:

Interface name : Wi-Fi

There are 21 networks currently visible.

SSID 1 : Do not connect

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 2 :

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 3 :

Network type : Infrastructure

Authentication : Open

Encryption : None

SSID 4 :

Network type : Infrastructure

Authentication : WPA2-Enterprise

Encryption : CCMP

SSID 5 : moto g(6)

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 6 : Moto G (5S) Plus 4950

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 7 : AndroidAP5755

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 8 : lil one

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 9 : iguest

Network type : Infrastructure

Authentication : Open

Encryption : None

SSID 10 : LAPTOP-MTKM7VOQ 8848

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 11 : AIGAMER

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 12 : ASUS

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 13 : Raj

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 14 : DIRECT-KpLENOVOmsUR

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 15 : DIRECT-BJDESKTOP-1NOAAB5msJX

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 16 : Venkat's 5G

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 17 : Hotspot

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 18 : Airtel-Hotspot-30E1

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 19 : DIRECT-RADESKTOP-RPO2L76msJN

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 20 : Shrutila

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

SSID 21 : GCR Class Room

Network type : Infrastructure

Authentication : WPA2-Personal

Encryption : CCMP

---------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import tkinter as tk

from tkinter import messagebox

import geocoder

from twilio.rest import Client

from geopy.distance import great\_circle

from geopy.geocoders import Nominatim

import time

import os

import subprocess

lst=[]

def sms(msg,number):

import os

from twilio.rest import Client

account\_sid = 'ACc7d26b30fee179fb40c7759b3307330e'

auth\_token = '840ee22fa776aacf24286166d7173314'

client = Client(account\_sid, auth\_token)

message = client.messages \

.create(

body=msg,

from\_='+12562545424',

#to='+919884920419'

to=number

)

#inp=input("Enter Message : ")

#number=input("Enter Number : ")

#sms(inp,number)

def whatsapp(msg,number):

from twilio.rest import Client

account\_sid = 'ACc7d26b30fee179fb40c7759b3307330e'

auth\_token = '840ee22fa776aacf24286166d7173314'

client = Client(account\_sid, auth\_token)

from\_whatsapp\_number='whatsapp:+14155238886'

to\_whatsapp\_number='whatsapp:'+number

client.messages.create(

from\_=from\_whatsapp\_number,

body=msg,

to=to\_whatsapp\_number)

#inp=input("Enter Message : ")

#number=input("Enter Number : ")

#whatsapp(inp,number)

def sosFunction():

msg = "\nDeviation detected!!!\n" + "Location Coordinates: " + str(currloc)

for number in lst[1:3]:

sms(msg,number)

whatsapp(msg, number)

def showMsg():

messagebox.showinfo('Message', 'You clicked the Submit button!', font=("Helvetica", 32))

sosFunction()

fields = 'Name', 'Emergency contact 1', 'Emergency contact 2', 'Country'

def fetch(entries):

for entry in entries:

field = entry[0]

text = entry[1].get()

print('%s: "%s"' % (field, text))

lst.append(text)

print(lst)

def makeform(root, fields):

entries = []

for field in fields:

row = tk.Frame(root)

lab = tk.Label(row, width=15, text=field, anchor='w', font=("Helvetica", 32))

ent = tk.Entry(row, font=("Helvetica", 32))

row.pack(side=tk.TOP, fill=tk.X, padx=5, pady=5)

lab.pack(side=tk.LEFT)

ent.pack(side=tk.RIGHT, expand=tk.YES, fill=tk.X)

entries.append((field, ent))

return entries

def func():

global currloc

print(source.address)

print()

print(dest.address)

print()

dist = great\_circle(scode, dcode).miles

print(great\_circle(scode, dcode).miles)

t1 = 5

count1 = 0

count2 = 0

it = 0

dev = 0

while(True):

time.sleep(t1)

newloc = list(map(lambda x : x + dev, currloc))

travelled = great\_circle(currloc, newloc).miles

currloc = newloc

newdist = great\_circle(currloc, dcode).miles

if (newdist == 0 ):

print("Destination Reached!!!")

break

if (newdist > dist):

count2 += 1

if(count2 == 4):

msg = "\nDeviation detected!!!\n" + "Location Coordinates: " + str(currloc)

for number in lst[1:3]:

sms(msg,number)

print(msg)

else:

count2 = 0

if(travelled == 0):

count1 += 1

if (count1 == 4):

msg = "\nStagnation detected!!!\n" + "Location Address:" + str(geolocator.reverse(str(currloc[0]) + ", " + str(currloc[1])))

for number in lst[1:3]:

sms(msg,number)

print(msg)

else:

count1 = 0

dist = newdist

print(dist)

it += 1

if( it > 4 ):

dev = 1

if( it > 10 ):

break

root = tk.Tk()

root.title('SHIELD OF SECURITY (S.O.S)')

#root.geometry('400x150')

ents = makeform(root, fields)

b1 = tk.Button(root, text='Submit', font=("Helvetica", 32), command=(lambda e=ents: fetch(e)))

b1.pack(side=tk.LEFT, padx=5, pady=5)

'''

b2 = tk.Button(root, text='Quit', font=("Helvetica", 32), command=root.quit)

b2.pack(side=tk.LEFT, padx=5, pady=5)

'''

root.mainloop()

def showMsg():

messagebox.showinfo('Message', 'SOS ACTIVATED!\nEmergency contacts alearted!')

sosFunction();

fields2='Source','Destination'

tkWindow = tk.Tk()

#tkWindow.geometry('400x150')

tkWindow.title('Shield Of Security (SOS)')

ents = makeform(tkWindow, fields2)

button1 = tk.Button(tkWindow, text='Store', font=("Helvetica", 32), command=(lambda e=ents: fetch(e)))

button1.pack(side=tk.LEFT, padx=5, pady=5)

tkWindow.mainloop()

def message():

results = subprocess.check\_output(["netsh", "wlan", "show", "network"])

results=results.decode("ascii")

results = results.replace("\r"," ")

#print(results)

a=result.count("WPA2-Personal")

return a

messagebox.showinfo('Message', 'There are atleast '+message()+' people near you!')

geolocator = Nominatim(user\_agent="Shield Of Security (SOS)")

source = geolocator.geocode(lst[4]) #"Westin Chennai"

dest = geolocator.geocode(lst[5]) #"tambaram railway station"

scode = (source.latitude, source.longitude)

dcode = (dest.latitude, dest.longitude)

currloc = scode

func()

'''

tkWindow2=tk.Tk()

tkWindow2.title('Shield Of Security (SOS)')

button2 = tk.Button(tkWindow2, text = 'SOS', font=("Helvetica", 32), command = showMsg)

button2.pack()

tkWindow2.mainloop()

'''

'''

sosbutton = tk.Button(tkWindow, font=("Helvetica", 32), text='SOS', width=25, command=sosFunction)

stopbutton = tk.Button(tkWindow, font=("Helvetica", 32), text='Quit', width=25, command=tkWindow.destroy)

sosbutton.pack()

stopbutton.pack()

tkWindow.mainloop()

'''

root.quit

root.mainloop()

'''

import tkinter as tk

def someFunction():

print("working")

tkWindow = tk.Tk()

button = tk.Button(tkWindow, command=someFunction)

'''

def sosFunction():

msg = "\nSOS from "+lst[0]+"\nLocation Coordinates: " + str(currloc)

for number in lst[1:3]:

sms(msg,number)

whatsapp(msg, number)

tkWindow=tk.Tk()

sosbutton = tk.Button(tkWindow, text='SOS', width=25, command=sosFunction)

print("SOS done")

stopbutton = tk.Button(tkWindow, text='Quit', width=25, command=tkWindow.destroy)

sosbutton.pack()

stopbutton.pack()

tkWindow.mainloop()