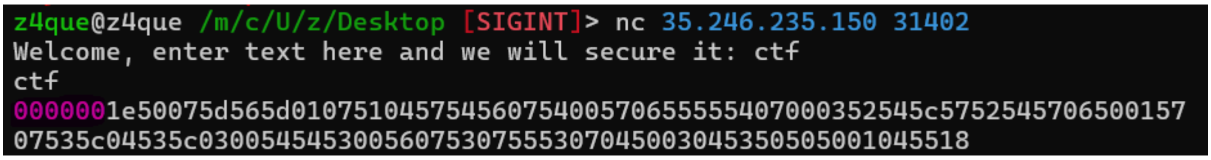
# When we write the correct character of the flag, we get “00” from the output. For example, If I write **ctf,** I can get the following output, with six zeros :



```
z4que@z4que /m/c/U/z/Desktop [SIGINT]> nc 35.246.235.150 31402
Welcome, enter text here and we will secure it: ctf
ctf
0000001e50075d565d01075104575456075400570655555407000352545c5752545706500157
07535c04535c03005454530056075307555307045003045350505001045518
```

# So we have to write a python code to try each character (from a to f and from 0 to 9 because it’s a sha256) and when we get another pair of zeros, we need to keep the character. The python code :

```
from pwn import *

def send_input(input_data):
    r = remote(host, port)
    r.recvuntil(": ")
    r.sendline(input_data)
    r.recvline()
    response = r.recvline().decode().strip()
    r.close()
    return response

def find_flag():
    flag = "ctf{"
    while True:
        for char in valid_chars:
            response = send_input(flag + char)
            n = len(flag) + 1
            if response[:2 * n] == "0" * (2 * n):
                flag += char
                print("NEW CHAR : ", char)
                break
        else:
            print(f"Flag : {flag}" + "}")
            break

def main() :
    valid_chars = "abcdef0123456789"
    host='35.246.235.150'
    port = 31402
    find_flag()

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

THE FLAG :
ctf{5b838db4a213b1e2c001bef7192712c5d2b69a69fe116e3b6b06ba5fa6555da0}
~Z4que