

# The title told us this is a challenge with a **zlib** file. I connected with **netcat** to the server and I got :

```
z4que@Zemf-PC /m/c/U/Z/Desktop [SIGINT]> nc 34.89.251.121 31905
Incoming work proof!!!
eJwrzy_Kj18oys9PszW3NEpOSUtlTTXMSEnLMMlMy8hIs8ioSiw0S05LTEuzBABi7Q_a
Insert work proof:
```

# After I uploaded the text on **CyberChef**, I found out that we are talking about a Base64 code, compressed with **zlib** :

The screenshot shows the CyberChef interface with three main sections: 'From Base64', 'Zlib Inflate', and 'From Quoted Printable'. In the 'From Base64' section, the input is a Base64 string: 'eJwrzy\_Kj18oys9PszW3NEpOSUtlTTXMSEnLMMlMy8hIs8ioSiw0S05LTEuzBABi7Q\_a'. Below it, under 'Zlib Inflate', are settings for 'Start index' (0), 'Initial output buffer size' (0), and 'Buffer expansion type' (Adaptive). Under 'From Quoted Printable', there is a checkbox for 'Verify result'. On the right side, the 'Output' panel shows the decompressed result: 'work\_proofy2cdffee1hdfh4ifhhf8hzaq6cfaff9'.

# We got :

**work\_proofy2cdffee1hdfh4ifhhf8hzaq6cfaff9**

# But the time is very short when the server is getting closed. We have almost no time to go on CyberChef and use the decoded text, so I wrote a program in Python :

```
from pwn import *
import base64, zlib

def connect_server(ip, port) :
    remote_access = remote(ip, port)

    for i in range(1000) :
        print(remote_access.recvline())
        data = remote_access.recvline().decode("utf-8").strip()

        print("DATA : ", data)
        print("Remote Access : ", remote_access.recvline())
```

```
decode_the_message =
zlib.decompress(base64.urlsafe_b64decode(data)).decode("utf-8")
    print("Sent : ", decode_the_message)

remote_access.sendline(decode_the_message)

remote_access.close()

def main() :
    connect_server("34.89.251.121", 31905)

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

# When the program is finished, we get an EOF error and also the flag

THE FLAG :

CTF{a7550246d72f8c7946a9248b3b9eee93461ac30f53ac8ca9749c9590b4ed1a2b}  
~Z4que