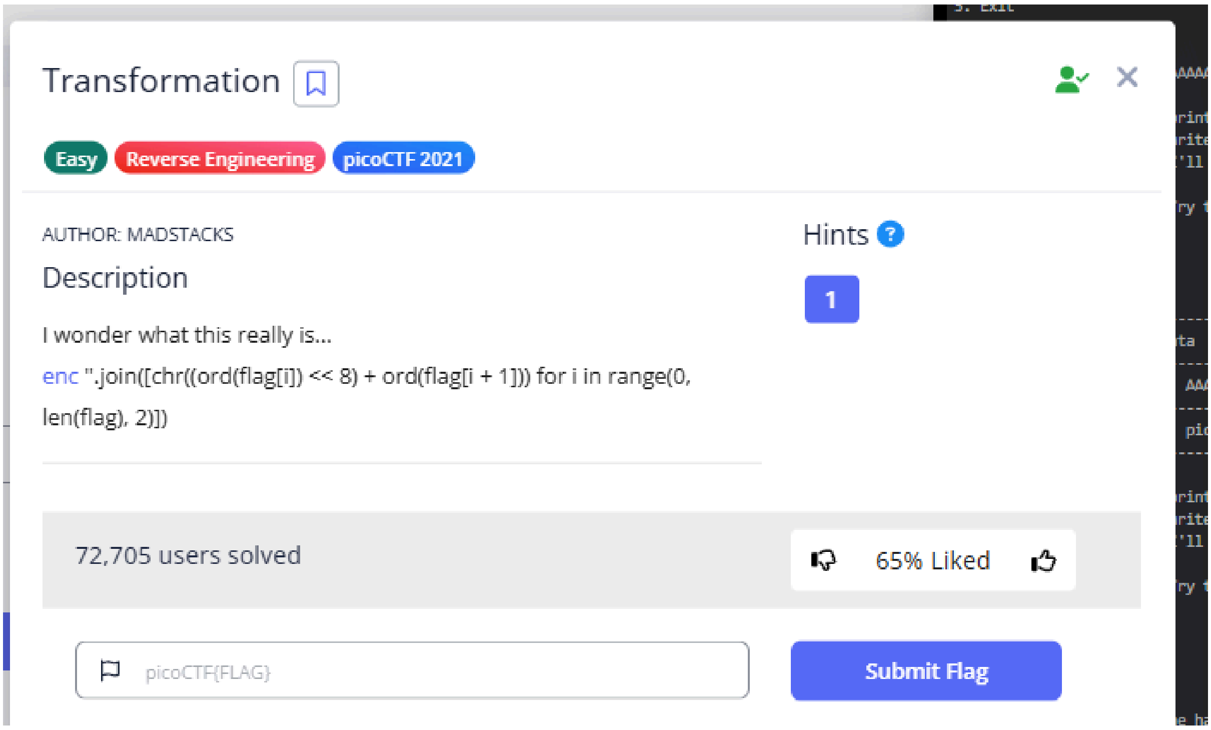
JASMINE C. OMANDAM

picoCTF - picoGym Challenges



Transformation

Easy | Reverse Engineering | picoCTF 2021

AUTHOR: MADSTACKS

Description

I wonder what this really is...

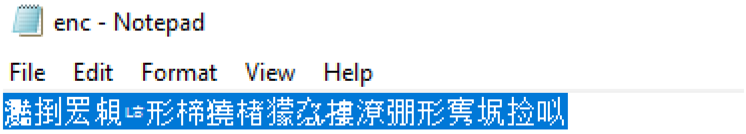
enc ''.join([chr((ord(flag[i]) << 8) + ord(flag[i + 1])) for i in range(0, len(flag), 2)])

Hints ?

1

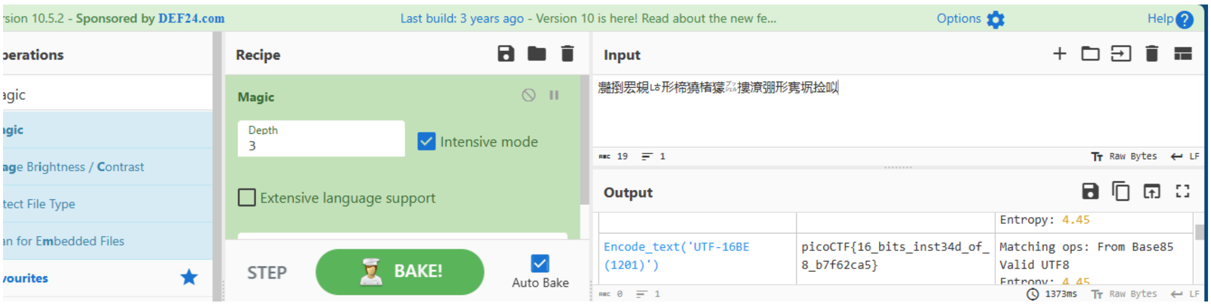
72,705 users solved

65% Liked



enc - Notepad

灩捯䍔䙻ㄶ形楴獟楮獴㌴摟潦弸形㝦㘲捡㕽



rsion 10.5.2 - Sponsored by DEF24.com
Last build: 3 years ago - Version 10 is here! Read about the new fe...
Options
Help
perations
agic
agic
age Brightness / Contrast
tect File Type
an for Embedded Files
vourites
Recipe
Input
Output
STEP
BAKE!
Auto Bake

Magic

Depth 3

☑ Intensive mode

□ Extensive language support

灩捯䍔䙻ㄶ形楴獟楮獴㌴摟潦弸形㝦㘲捡㕽

```
19    1                                   Raw Bytes   LF
```

```
                                         Entropy: 4.45
Encode_text('UTF-16BE   picoCTF{16_bits_inst34d_of_   Matching ops: From Base85
(1201)')                8_b7f62ca5}                   Valid UTF8
                                                      Entropy: 4.45
0    1                                   1373ms   Raw Bytes   LF
```

Write-Up: Transformation

When I opened the challenge, the Python code immediately stood out. It was taking two characters at a time, shifting one by 8 bits, adding them together, and turning that into a single Unicode character. That explained why the encoded output looked like a string of Chinese characters — the flag had been transformed from **8-bit ASCII** into **16-bit Unicode**.

Instead of writing a decoder myself, I turned to **CyberChef**, a tool I often use for quick transformations. I pasted the encoded text into CyberChef and ran the **Magic** operation with deeper analysis enabled. Within seconds, the tool recognized the transformation and produced the decoded flag:

picoCTF{16_bits_inst34d_of_8_b7f62ca5}

Reflection

This challenge reminded me that even simple bitwise operations can make text look completely foreign. At first, the Chinese characters seemed confusing, but once I understood the encoding logic, the solution became clear. Using CyberChef saved time and showed the value of knowing when to rely on tools versus coding from scratch. It reinforced the idea that problem-solving in CTFs is often about combining understanding with resourcefulness.