# 队伍名称：siyah

# 队伍 ID：00007f

## TEST NC

Kali 虚拟机中输入 nc 119.45.235.21 30838

```
┌──(siyah㉿Kali)-[~]
└─$ nc 119.45.235.21 30838
ipconfig
/bin/sh: ipconfig: not found
ls
bin
dev
etc
flag
home
lib
media
mnt
opt
proc
root
run
sbin
srv
start.sh
sys
tmp
usr
var
cat /flag
hgame{yOUr_C4n-C0NN3Ct_TO_TH3_rEM0t3_3Nvir0nM3nt-TO_Get-FLag0}
```

## Compress dot new

```python
import json


# 解析霍夫曼树
def parse_huffman_tree(json_data):
    if 's' in json_data:
        return {'type': 'leaf', 'char': str(json_data['s'])}  # 确保字符是字符串
    else:
        return {
            'type': 'node',
            'left': parse_huffman_tree(json_data['a']),
            'right': parse_huffman_tree(json_data['b'])
        }
# 解码二进制字符串
def decode_huffman(tree, binary_string):
    result = []
    current_node = tree
    for bit in binary_string:
        if bit == '0':
```

```
            current_node = current_node['left']
        else:
            current_node = current_node['right']

        if current_node['type'] == 'leaf':
            result.append(str(current_node['char']))  # 确保添加的是字符串
            current_node = tree  # 重置到根节点
    return ''.join(result)
# 读取 enc.txt 文件
with open('enc.txt', 'r') as f:
    data = f.read().split('\n')
    json_data = json.loads(data[0])
    binary_string = data[1]
# 解析霍夫曼树
huffman_tree = parse_huffman_tree(json_data)
# 解码二进制字符串
original_data = decode_huffman(huffman_tree, binary_string)
# 输出原始数据
print(original_data)
```

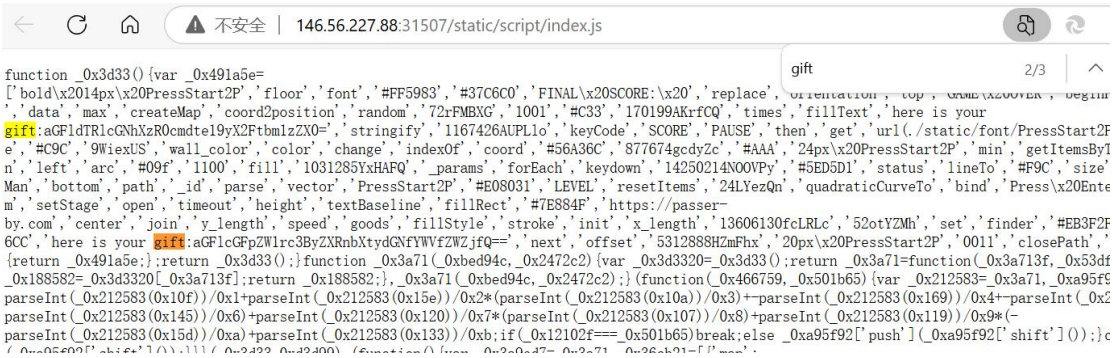
输出：

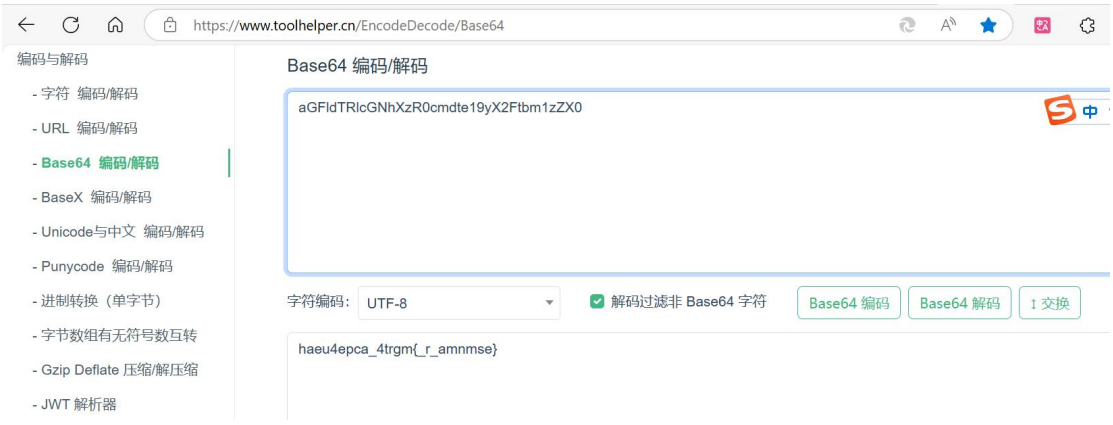
104103971091011237811745831041011081084511599114491121161154597114514549110116511145111511649110103451164845119114491161104103971091011237811745831041011081084511599114491121161154597114514549110116511145111511649110103451164845119114491161014538451171151013312513107611111410110932105112115117109321001111081111143211510511632971091011164432991111101151019911610111611711432971001051121051159910511010332101108105116461310781171081089732110101993210810510311710897321101011131171014538451171151013312513107611111410110932105112115117109321001111081111143211510511632971091011164432991111101151019911610111611711432971001051121051159910511010332101108105116461310781171081089732110101993210810510311710897321101011131171610111611711432971001051120511599105110103321011081051164613107811710810897321101019932108105103117108973211010111311710146326911610597109321011163211810511810111411497321101171109944321181011083298105981011101001171093211410511511711546326811111010199461
**hgame{Nu-Shell-scr1pts-ar3-1nt3r3st1ng-t0-wr1te-&-use!}**

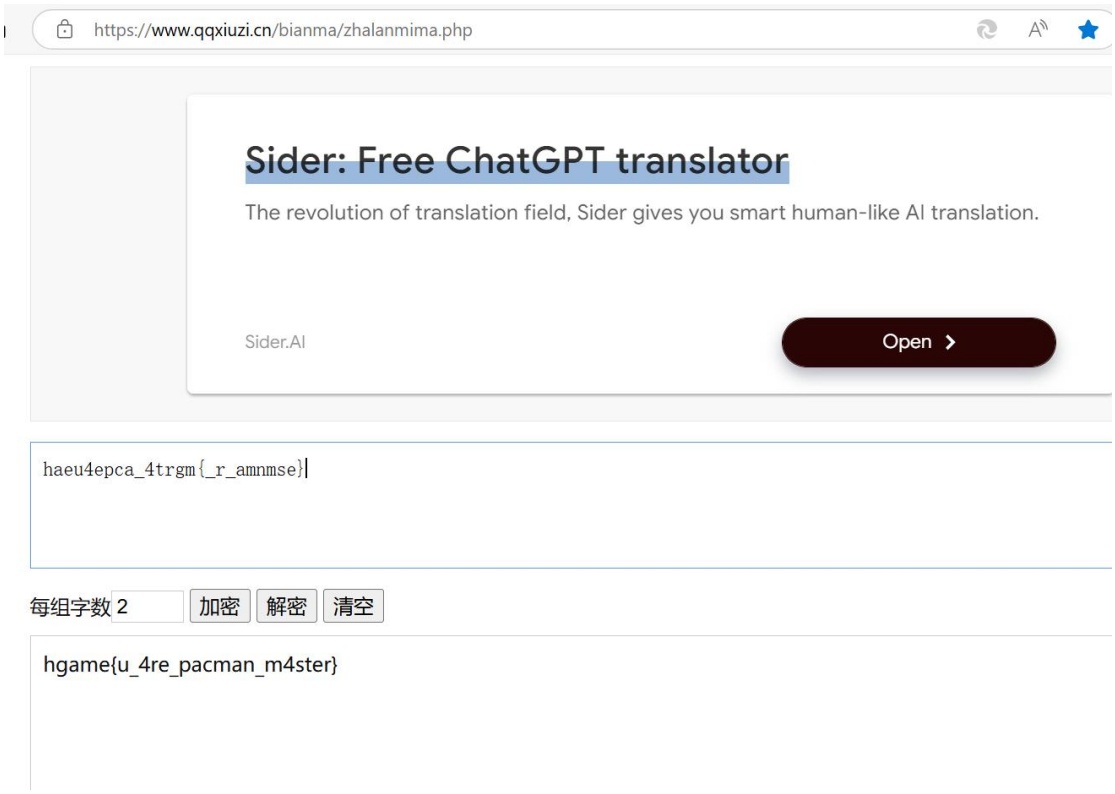
## Level 24 Pacman

游戏失败


从网页源代码进入 index.js



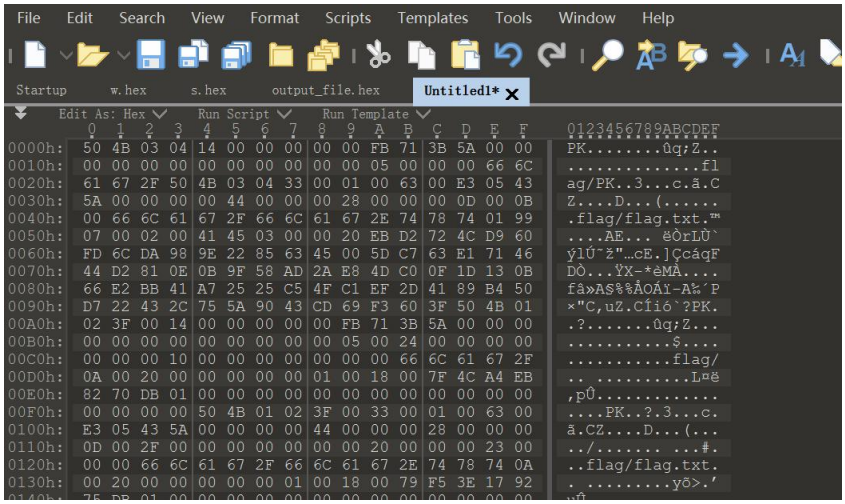






flag:hgame{u_4re_pacman_m4ster}

## Hakuya Want A Girl Friend

010editor 打开附件




分成两个文件，压缩包和图片

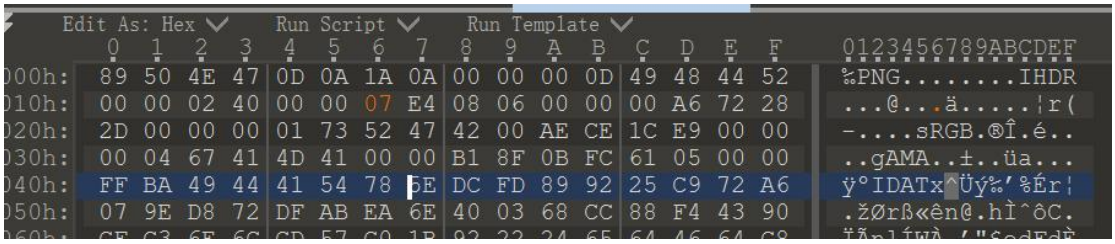
图片文件逆序

```
def reverse_hex_file(input_file, output_file):
    # 以二进制方式打开输入文件
    with open(input_file, 'rb') as f:
        # 读取文件内容
        file_data = f.read()

    # 将文件内容逆序
```

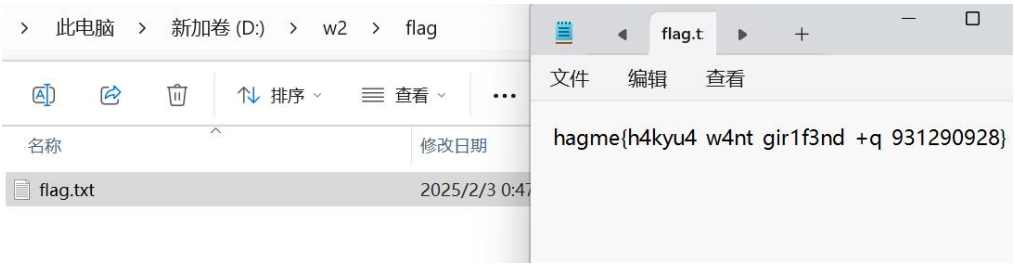
```
    reversed_data = file_data[::-1]

    # 将逆序后的内容写入到输出文件
    with open(output_file, 'wb') as f:
        f.write(reversed_data)

    print(f"文件已成功逆序并保存到 {output_file}")

# 示例使用
input_file = "input_file.hex"  # 输入文件
output_file = "output_file.hex"  # 输出文件

reverse_hex_file(input_file, output_file)
```

打开图片


修改高度




压缩包解压输入密码 To_f1nd_th3 _QQ


修改前缀

hgame{h4kyu4_w4nt_gir1f3nd_+q_931290928}