picoCTF - RE - Transformation writeup

原创

yyyayo 于 2021-07-09 14:10:14 发布 188 收藏

分类专栏： CTF

本文链接：https://blog.csdn.net/yyyayo/article/details/118604297

- 一份 binary 文件
- `''.join([chr((ord(flag[i]) << 8) + ord(flag[i + 1])) for i in range(0, len(flag), 2)])`

上面是把 flag 转化为 binary 的过程，需要还原。

解题如下：

```
enc = open("enc", "r")
buf = enc.read()
for c in buf:
 print(chr(ord(c)>>8),end="")
 print(chr(ord(c)-(ord(c)>>8<<8)),end="")
```

要点：

1. 擅用 `chr` 和 `ord` 函数。
   ord() in Python Given a string of length one, return an integer representing the Unicode code point of the character when the argument is a unicode object, or the value of the byte when the argument is an 8-bit string. 即把 unicode 和 ascii 转化为数字。
   chr() return a string of one character whose ASCII code or unicode is the integer i.
2. 字符编码和读取、写入文件。
   注意这道题的 flag 在编码之后，每个字符是16位，也就是得到的是 unicode 字符，然后写入文件。所以读取的时候也要按 str 读取，不能是以 bytes `rb` 读取。
   另外，ASCII 码对应的 bytes 和它本身是一样的，但 Unicode 不是。比如一个字符 `a`，它存储到文件中，`hexdump` 得到的结果是 `\x61`，但是一个 Unicode 字符 `\u7069`，它存储到文件中 bytes 内容为 `\xe7\x81\xa9`。

关于 Unicode encoding 的内容可以参考 UTF-8 Wikipedia

First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
U+0000	U+007F	0xxxxxxx			
U+0080	U+07FF	110xxxxx	10xxxxxx		
U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
U+10000	U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx