ISCC2025 WriteUp 提交模板

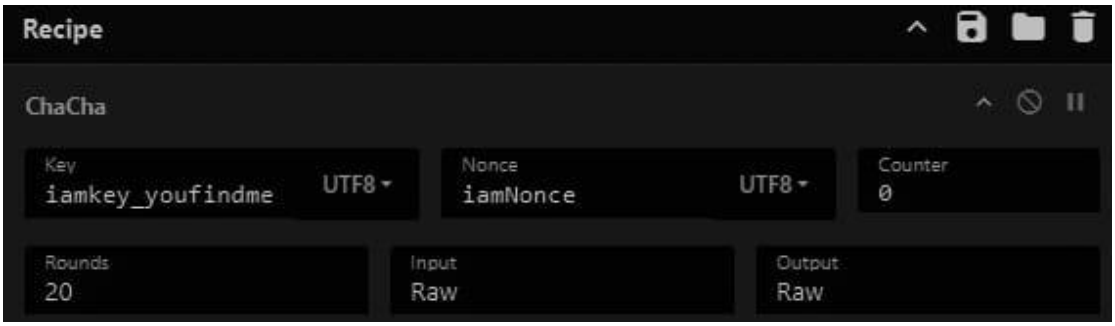
REVERSE+真？复杂

解题思路

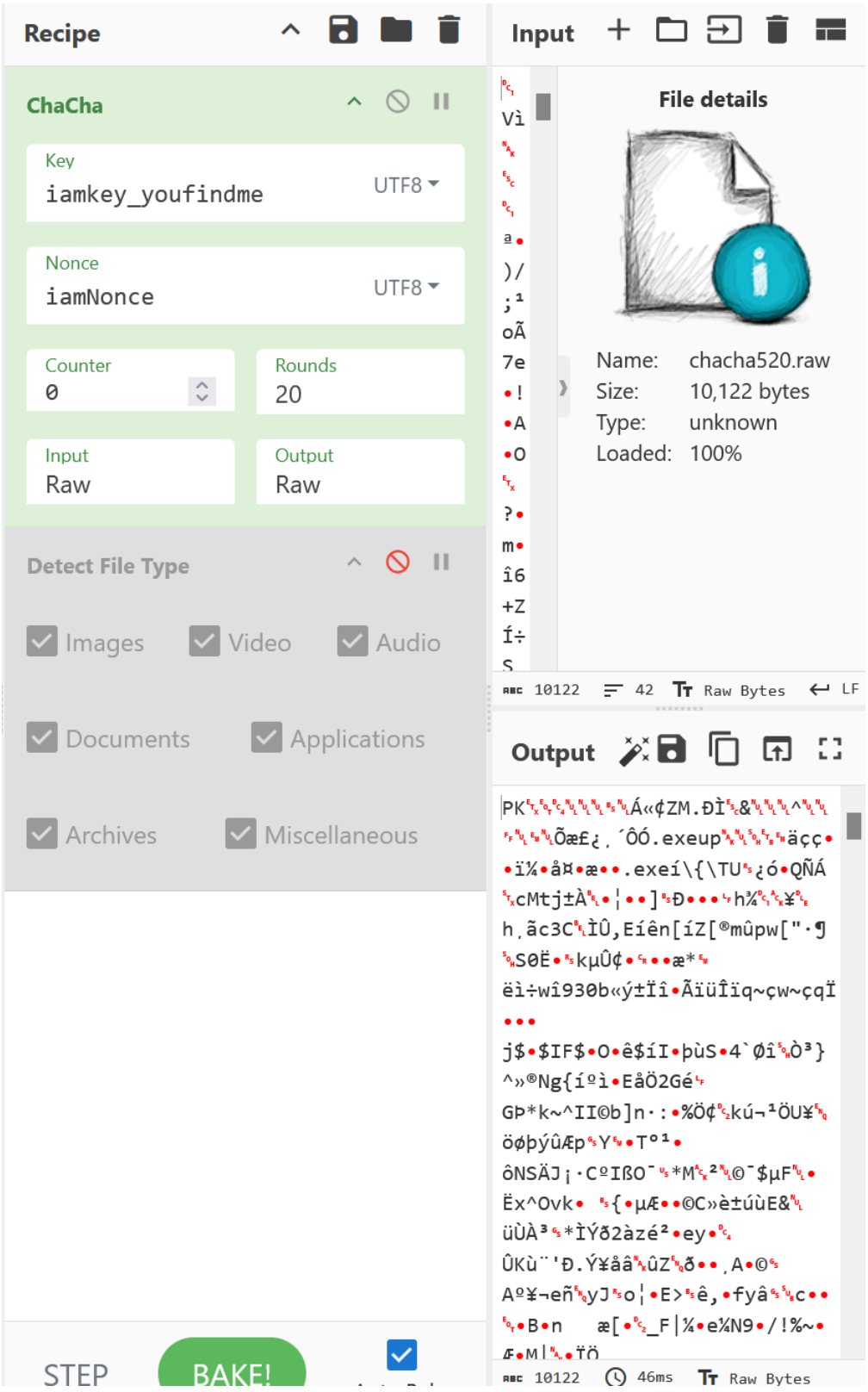
1. 拿到一个 raw 文件，用 010 打开，发现文件头也不是 raw 的文件头，我们看看这个 raw 文件里是不是包含了这些信息。

2. 我们使用常用压缩包缀搜索，没有，图片后缀呢，也没有，但是意外发现了一个 JDIF 头，，，jpg 图片？是的了因为有 FF D8 E0，我 0 们提取出来修复文件头 FF D8FFE0 ， 截止到 FF 09 为止

3. 得到图片，chacha20，是一个流密码算法。要进行加密解密的话我们还需要密文



4. 将源文件图片提取出来之后，把图片部分删掉，剩余内容当作密文去 chacha 解密



得到压缩包

5. 得到一个 falg.txt.enc 和.exe，用 ida 打开，发现有很多混淆函数，最重要的是 sub_40182E 函数，对 flag.txt 中的内容进行变换保存到 flag.txt.enc。

sub_40182E 函数是一个状态机，通过 v5 变量作为状态码，每个状态执行不同的操作。参数 a1 是输入缓冲区，a2 是长度，a3 是种子值，这里主函数中传入的是 1131796。状态机从状态 0 开始，根据条件转移到不同的状态，每个状态对当前字节进行处理。

```
318           sub_401550();
319         if ( (unsigned int)sub_401550() )
320           sub_4015DE();
321         if ( (unsigned int)sub_401682() )
322           sub_4015DE();
323         if ( (unsigned int)sub_401550() )
324           sub_40172A();
325         if ( (unsigned int)sub_4015DE() )
326           sub_401682();
327         if ( (unsigned int)sub_40172A() )
328           sub_4015DE();
329         goto LABEL_213;
330       case 15:
331         rand();
332         rand();
333         *(_BYTE *)(a1 + v4) -= v4;
334         v5 = 16;
335         continue;
336       case 16:
337         rand();
338         rand();
339         *(_BYTE *)(a1 + v4) = (*(_BYTE *)(a1 + v4) ^ byte_409970[v4 % 16]) + v4;
340         v5 = 17;
341         continue;
342       case 17:
343         if ( (rand() & 1) != 0 )
344         {
345           if ( (rand() & 1) != 0 )
346           {
347             if ( (v4 & 1) != 0 )
348               v5 = 21;
349             else
350               v5 = 18;
351           }
352           else if ( (v4 & 1) != 0 )
353           {
354             v5 = 21;
355           }
356           else
357           {
358             v5 = 18;
359           }
360         }
361         else if ( (rand() & 1) != 0 )
```

现在需要逆向这些操作，即从加密后的字节推导出原始字节。

6. 我们可以知道生成随机数函数是 4017D8，种子数是 1131796LL

```
1 void __fastcall sub_4017D8(unsigned int a1)
2 {
3   int i; // [rsp+2Ch] [rbp-4h]
4
5   srand(a1);
6   for ( i = 0; i <= 15; ++i )
7     byte_409970[i] = rand();
8 }
```

我们生成 16 个作为 key

srand(1131796);

for (int i = 0; i < 16; i++) {

printf("随机数 #%d: %u\n", i+1, rand());

得到

25992,22147,9379,1918,9706,10657,9658,28453,8050,22479,17949,7022,18297,22864,21527,15440

7. 然后丢给 gpt 写脚本解密就行

```
E:\reverse\python\venv\Scripts\python.exe E:\reverse\python\test\iscc\sm4.py
ISCC{1t_1s_F@ke_011vm!}
```

Exp

```
from pathlib import Path
def custom_decrypt(data):
    key = [25992,22147,9379,1918,9706,10657,9658,28453,8050,22479,17949,7022,18297,22864,21527,15440]
    result = bytearray()

    for idx, byte in enumerate(data):
        k = key[idx % len(key)]

        if idx % 2 == 0:
```

```
            # Even index: bitwise NOT → add index → XOR key → subtract
index
            tmp = (~byte & 0xFF)
            tmp = (tmp + idx) & 0xFF
            tmp ^= k
            tmp = (tmp - idx) & 0xFF
        else:
            # Odd index: XOR with index → subtract index → XOR key → add
index
            tmp = (byte ^ idx) & 0xFF
            tmp = (tmp - idx) & 0xFF
            tmp ^= k
            tmp = (tmp + idx) & 0xFF

        result.append(tmp)

    return bytes(result)

if __name__ == "__main__":
    encrypted_file = Path("flag.txt.enc")
    encrypted_bytes = encrypted_file.read_bytes()
    decrypted_bytes = custom_decrypt(encrypted_bytes)

    try:
        print(decrypted_bytes.decode("utf-8"))
    except UnicodeDecodeError:
        print(decrypted_bytes)
```