Simple reverse (không simple cho lắm)

Solver: An Võ

Đầu tiên, ta dùng dnspy để decomplie file C#. Sau đó, dò tìm hàm chính thấy hàm bị mã hoá bằng base64. Giải mã ra, ta sẽ có 1 file mới.

Phân tích file mới này ta thấy 2 hàm mã hoá chính đó là hàm xor và hàm sha1. Hàm sha1 này chính là kết quả sau khi ta mã hoá/giải mã chuỗi ban đầu nên ta không quan tâm và sẽ tập trung vào hàm xor.

```
5
6   namespace \u00A0
7   {
8       // Token: 0x02000002 RID: 2
9       internal class \u00A0
10      {
11          // Token: 0x06000001 RID: 1 RVA: 0x00002050 File Offset: 0x00000250
12          public static string \u00A0(string A_0, string A_1)
13          {
14              StringBuilder stringBuilder = new StringBuilder();
15              for (int i = 0; i < A_1.Length; i++)
16              {
17                  stringBuilder.Append(A_1[i] ^ A_0[i % A_0.Length]);
18              }
19              return stringBuilder.ToString();
20          }
21
22          // Token: 0x06000002 RID: 2 RVA: 0x00002098 File Offset: 0x00000298
23          public static byte[] \u00A0(string A_0)
24          {
25              byte[] result;
26              using (HashAlgorithm hashAlgorithm = SHA1.Create())
27              {
28                  result = hashAlgorithm.ComputeHash(Encoding.UTF8.GetBytes(A_0));
29              }
30              return result;
31          }
```

Hàm xor chính làm hàm để get key, sau đó ta suy ra flag. Tuy nhiên vì file này chỉ cho duy nhất 1 mảng các số đã được mã hoá trước nên ta buộc phải đoán. Mặc dù flag format là wannagame nhưng khi ta tính key để dò ngược thì nó lại bắt đầu bằng flag{}. Do đó, ta viết script để dò thủ công.

Nội dung file solve:

```
key = [ord(i) for i in 'SaY_s0mE_tH1nG']
cipher = [53, 13, 56, 56, 8, 116, 93, 43, 11, 43, 0, 5, 24, 116, 12, 21, 17, 110, 29, 119, 50, 49, 111, 43, 59, 6, 23, 58]
flag = ''
for i in range(28):
        flag += chr(cipher[i] ^ key[i % len(key)])
print(flag)
```