

# 2017第二届广东省强网杯线上赛--Nonstandard

原创

[Hk\\_Mayfly](#) 于 2019-11-07 19:48:00 发布 98 收藏

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：[https://blog.csdn.net/qq\\_39542714/article/details/106834802](https://blog.csdn.net/qq_39542714/article/details/106834802)

版权

测试文

件：[http://static2.ichunqiu.com/icq/resources/fileupload/CTF/echunqiu/qwb/Nonstandard\\_26195e1832795caa1](http://static2.ichunqiu.com/icq/resources/fileupload/CTF/echunqiu/qwb/Nonstandard_26195e1832795caa1)

## 1.准备



获得信息：

- 32位文件

## 2.IDA打开

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    FILE *v3; // eax
    FILE *v4; // eax
    FILE *v5; // eax
    char Buf[16]; // [esp+0h] [ebp-24h]
    __int64 v8; // [esp+10h] [ebp-14h]
    int v9; // [esp+18h] [ebp-Ch]
    __int16 v10; // [esp+1Ch] [ebp-8h]

    v9 = 0;
    __mm_storeu_si128((__m128i *)Buf, (__m128i)0i64);
    v10 = 0;
    v8 = 0i64;
    v3 = _iob_func();
    fputs("Place Input Flag:\n", v3 + 1);
    v4 = _iob_func();
    fgets(Buf, 29, v4);
    if ( sub_401480(Buf) == 1 )
    {
        v5 = _iob_func();
        fputs("yes\n", v5 + 1);
    }
    return 0;
}
```

### 3.代码分析

打开 sub\_401480(Buf)函数

```

1 signed int __thiscall sub_401480(const char *this)
2 {
3     const char *v1; // esi
4     const char *v2; // eax
5     unsigned int v3; // eax
6     unsigned int v4; // kr04_4
7     signed int result; // eax
8     char v6; // [esp+4h] [ebp-38h]
9     char Dst; // [esp+5h] [ebp-37h]
10
11     v6 = 0;
12     v1 = this; // this为输入的字符串的地址，长度为28，不包含最后的结束符
13     memset(&Dst, 0, 0x31u); // Dst为大小为49的空间
14     if ( strlen(v1) != 28 ) // 输入字符串长度为28
15         goto LABEL_10;
16     v2 = sub_401070((int)v1, 28u);
17     strncpy_s(&v6, 0x32u, v2, 0x30u);
18     v3 = 0;
19     v4 = strlen(&v6);
20     if ( !v4 )
21         goto LABEL_10;
22     do
23     {
24         if ( byte_402120[v3] != *(&v6 + v3) )
25             break;
26         ++v3;
27     }
28     while ( v3 < v4 );
29     if ( v3 == 48 )
30         result = 1;
31     else
32 LABEL_10:
33         result = -1;
34     return result;
35 }

```

首先，我注意到第24行代码的比

较，`byte_402120[]="AdtxA66nbbdxA71tUAE2AOInnbtrAp1nQzGtAQGtrjC7==="`，这是一段被加密的字符串，而比较的v6数组来自v2，v2是函数`sub_401070((int)v1, 28u)`的返回值，传入的v1是我们的输入字符串。

打开`sub_401070((int)v1, 28u)`

▣`sub_401070((int)v1, 28u)`

这段加密方式是base32

引自：<https://www.ichunqiu.com/writeup/detail/815>

base64编码是用64（2的6次方）个ASCII字符来表示256（2的8次方）个ASCII字符，也就是三位二进制数组经过编码后变为四位的ASCII字符显示，长度比原来增加1/3。

同样，base32就是用32（2的5次方）个特定ASCII码来表示256个ASCII码。所以，5个ASCII字符经过base32编码后会变为8个字符（公约数为40），长度增加3/5。不足8n用"="补足。

base16就是用16（2的4次方）个特定ASCII码表示256个ASCII字符。1个ASCII字符经过base16编码后会变为2个字符，长度增加一倍。不足2n用"="补足

同时我们关注到第41行代码，sub\_401000();函数

```
signed __int16 sub_401000()
{
    signed int v0; // eax
    int v1; // esi
    char *v2; // edx
    char v3; // c1
    signed __int16 result; // ax

    v0 = 1;
    do
    {
        byte_403020[v0] += 32;
        v0 += 2;
    }
    while ( v0 < 26 );
    v1 = 0;
    v2 = &aMnopqrstuvwxyz[13];
    do
    {
        v3 = byte_40301F[++v1];
        byte_40301F[v1] = *v2;
        *v2-- = v3;
    }
    while ( (signed int)v2 > (signed int)aMnopqrstuvwxyz );
    *(_DWORD *)&aMnopqrstuvwxyz[14] = '3567';
    result = '12';
    word_40303E = '12';
    byte_403040 = 0;
    return result;
}
```

这段函数将加密表重新生成，可以在OD动态调试中获得

009110D0	-	0FB681 C8139	movzx eax,byte ptr ds:[ecx+0x9113C8]	
009110D7	-	FF2485 B4139	jmp dword ptr ds:[eax*4+0x9113B4]	
009110DE	>	BF 06000000	mov edi,0x6	Case 8 of switch 009110D0
009110E3	~	EB 13	jmp XNonstand.009110F8	
009110E5	>	BF 04000000	mov edi,0x4	Case 10 of switch 009110D0
009110EA	~	EB 0C	jmp XNonstand.009110F8	
009110EC	>	BF 03000000	mov edi,0x3	Case 18 of switch 009110D0
009110F1	~	EB 05	jmp XNonstand.009110F8	
009110F3	>	BF 01000000	mov edi,0x1	Case 20 of switch 009110D0
009110F8	>	897C24 20	mov dword ptr ss:[esp+0x20],edi	
009110FC	>	8D0CDD 04000	lea ecx,dword ptr ds:[ebx*8+0x4]	Default case of switch 009110D0
00911103	-	B8 CDC00000	mov eax,0xCDC00000	

地址	HEX 数据	ASCII
00913000	00 00 00 00 FE FF FF FF FF FF FF FF	....?jjjjjjj....
00913010	00 00 00 00 7A 59 78 57 76 55 74 53 72 51 70 4F	...zYxWvUtSrQp0
00913020	6E 4D 6C 4B 6A 49 68 47 66 45 64 43 62 41 37 36	nMIKjIhGFEdCbA76
00913030	35 33 32 31 00 00 00 00 00 00 00 00 41 42 43 44	5321.....ABCD
00913040	45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53 54	EFGHIJKLMNOPQRST
00913050	55 56 57 58 59 5A 61 62 63 64 65 66 67 68 69 6A	UVWXYZabcdefghijklmnop
00913060	6B 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 7A	klmnopqrstuvwxyz
00913070	30 31 32 33 34 35 36 37 38 39 2B 2F 3D 00 00 00	0123456789+/=...
00913080	5A 6D 78 68 5A 33 74 6D 62 47 46 6E 58 32 6C 7A	Zmxh23tmbGFnX21z
00913090	58 32 35 76 64 46 39 74 5A 53 46 39 00 00 00 00	X25vdF9tZSF9....
009130A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
009130B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....

因此总体操作就是，将我们输入的字符串，使用新生成的加密表，base32加密，得到AdtxA66nbbdxA71tUAE2AOlnnbtrAp1nQzGtAQGtrjC7===加密字符串

## 4.脚本获取

使用anybase32包来解密：<https://github.com/alanblevins/anybase32>

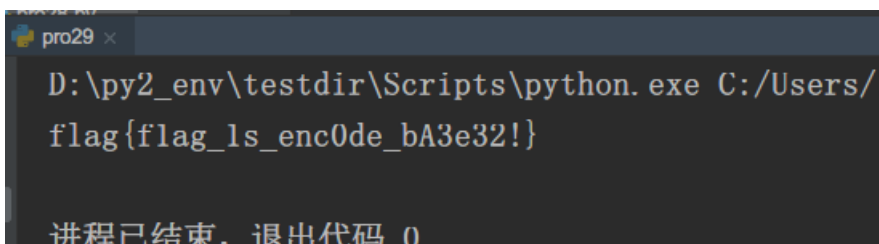
```
from __future__ import print_function
import anybase32

arbitrary_alphabet = b"zYxWvUtSrQpOnMlKjIhGfEdCbA765321"

encoded = b"nAdtxA66nbbdxA71tUAE2AOlnnbtrAp1nQzGtAQGtrjC7"

flag = anybase32.decode(encoded, arbitrary_alphabet)

print(flag)
```



```
pro29 x
D:\py2_env\testdir\Scripts\python.exe C:/Users/
flag{flag_1s_enc0de_bA3e32!}
进程已结束，退出代码 0
```

## 5.get flag!

```
flag{f1ag_1s_enc0de_bA3e32!}
```