[XCTF-Reverse] 45-48

原创

石氏是时试 于 2022-03-22 00:15:00 发布 162 收藏
分类专栏： CTF reverse 文章标签： reverse

45，2019_DDCTF_Windows_Reverse2

ASPack2.4的壳，平常用的2.1的还解不了，用WSUnpacker通过脱壳机可以脱。

脱壳之后效果并不好，函数都是MEM的地址，不过不影响看。

输入串先通过base16解码后再用base64编码，结果应该是reverse+

```
a = b'reverse+'
from base64 import b64decode,b16encode
b = b64decode(a) #sub_61000
c = b16encode(b)  #sub_61240
print(c)
#ADEBDEAEC7BE
#flag{ADEBDEAEC7BE}
```

46，2019_DDCTF_Windows_Reverse1

跟上题差不多，这个是UPX的壳，解起来很方便

这个就是个查表

```
unsigned int __cdecl sub_401000(const char *a1)
{
  _BYTE *v1; // ecx
  unsigned int v2; // edi
  unsigned int result; // eax
  int v4; // ebx

  v2 = 0;
  result = strlen(a1);
  if ( result )
  {
    v4 = a1 - v1;
    do
    {
      *v1 = byte_402FF8[(char)v1[v4]];
      ++v2;
      ++v1;
      result = strlen(a1);
    }
    while ( v2 < result );
  }
  return result;
}
```

这里边 byte_402FF8 这个位置啥都没有，其实由于0x20以前的字符都不可显示，所以查表的码表在+0x20后（0x403018）

```
a = '~}|{zyxwvutsrqponmlkjihgfedcba`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543210/.-,+*)(\x27&%$#"! '
b = "DDCTF{reverseME}"
c = ''.join([chr(a.index(i)+0x20) for i in b])
print(c)
#ZZ[JX#,9(9,+9QY!
#flag{ZZ[JX#,9(9,+9QY!}
```

47，csaw-ctf-2016-quals_deedeedee

这个解压以后有个.d文件，估计是编译之前的原码，里边有个flag提交就正确

```
string enc_flag = encrypt("flag{t3mplat3_met4pr0gramming_is_gr8_4_3very0n3}");
void main() {
    writeln("Your hexencoded, encrypted flag is: ", enc_flag.hexencode);
    writeln("I generated it at compile time. :)");
    writeln("Can you decrypt it for me?");
}
```

48，csaw-ctf-2016-quals_key

逆了一下出来的不对，还得用动态调试。

在013D1326: call sub_13D20C0下断点，参数就是flag

```
CPU - main thread, module cb505d6217c94b40b6195e8b3eade96

EAX 004FACB0 ASCII "idg_cni~bjbfi|gsxb"   <-------------在 call 013d20c0 下断点，参数
ECX 0024F834 ASCII "sssssssssssss"
EDX 0024F78C
EBX 004FA4F0 ASCII "`[^VZe`uYaY]`s^joY"
ESP 0024F740
EBP 0024F888 ASCII "Øø$"
ESI 0024F774
EDI 0000001F
EIP 013D1326 cb505d6217c94b40b6195e8b3eade96.013D1326
```