攻防世界 Reverse高手进阶区 3分题 testre

原创

思源湖的鱼 于 2021-02-03 22:53:48 发布 408 收藏

分类专栏： ctf 文章标签： ctf 攻防世界 reverse base58

本文链接：https://blog.csdn.net/weixin_44604541/article/details/113619564

前言

继续ctf的旅程

攻防世界Reverse高手进阶区的3分题

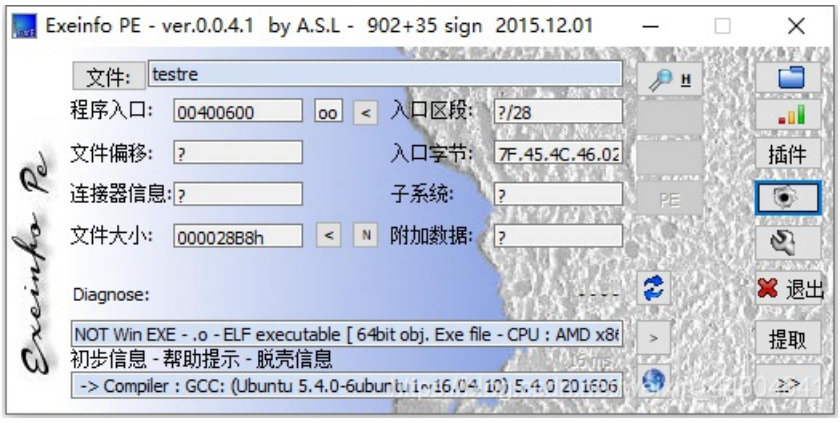
本篇是testre的writeup

发现攻防世界的题目分数是动态的

就仅以做题时的分数为准了

解题过程

PE查壳



扔进IDA

```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  void *ptr; // ST10_8
  __int64 v5; // [rsp+18h] [rbp-28h]
  char v6; // [rsp+20h] [rbp-20h]
  int v7; // [rsp+3Ch] [rbp-4h]

  v7 = 0;
  v5 = 256LL;
  sub_400D00(&v6, 17LL, a3);
  ptr = malloc(0x100uLL);
  sub_400700(ptr, &v5, &v6, 16LL);
  free(ptr);
  return 0LL;
}
```


查看关键函数

```
__int64 __fastcall sub_400D00(__int64 a1, unsigned __int64 a2)
{
  char buf; // [rsp+17h] [rbp-19h]
  unsigned __int64 i; // [rsp+18h] [rbp-18h]
  unsigned __int64 v5; // [rsp+20h] [rbp-10h]
  __int64 v6; // [rsp+28h] [rbp-8h]

  v6 = a1;
  v5 = a2;
  for ( i = 0LL; i < v5; ++i )
  {
    read(0, &buf, 1uLL);
    *(_BYTE *)(v6 + i) = buf;
  }
  *(_BYTE *)(v6 + v5 - 1) = 0;
  fflush(stdout);
  return (unsigned int)i;
}
```


没东西，就是接收输入

看下面这个

```
__int64 __fastcall sub_400700(void *a1, _QWORD *a2, __int64 a3, size_t a4)
{
  unsigned __int8 *v4; // rcx
  __int64 v6; // [rsp+0h] [rbp-C0h]
  int c; // [rsp+8h] [rbp-B8h]
  char v8; // [rsp+Fh] [rbp-B1h]
  int v9; // [rsp+10h] [rbp-B0h]
  bool v10; // [rsp+17h] [rbp-A9h]
  unsigned __int8 *v11; // [rsp+18h] [rbp-A8h]
  char v12; // [rsp+27h] [rbp-99h]
  int v13; // [rsp+28h] [rbp-98h]
  int v14; // [rsp+2Ch] [rbp-94h]
  unsigned __int64 i; // [rsp+30h] [rbp-90h]
  size_t n; // [rsp+38h] [rbp-88h]
  size_t v17; // [rsp+40h] [rbp-80h]
  size_t v18; // [rsp+48h] [rbp-78h]
  size_t j; // [rsp+50h] [rbp-70h]
  size_t v20; // [rsp+58h] [rbp-68h]
  int v21; // [rsp+64h] [rbp-5Ch]
  unsigned __int64 v22; // [rsp+68h] [rbp-58h]
  int v23; // [rsp+74h] [rbp-4Ch]
  __int64 *v24; // [rsp+78h] [rbp-48h]
  __int64 v25; // [rsp+80h] [rbp-40h]
  void *v26; // [rsp+88h] [rbp-38h]
  int v27; // [rsp+94h] [rbp-2Ch]
  size_t v28; // [rsp+98h] [rbp-28h]
```

```
__int64 v29; // [rsp+A0h] [rbp-20h]
_QWORD *v30; // [rsp+A8h] [rbp-18h]
void *s; // [rsp+B0h] [rbp-10h]
char v32; // [rsp+BFh] [rbp-1h]

s = a1;
v30 = a2;
v29 = a3;
v28 = a4;
v27 = -559038737;
v26 = malloc(0x100uLL);
v25 = v29;
v24 = &v6;
v22 = 0LL;
v17 = 0LL;
for ( i = 0LL; i < v28; ++i )
{
  v13 = *(unsigned __int8 *)(v25 + i);
  *((_BYTE *)v26 + i) = byte_400E90[i % 0x1D] ^ v13;
  *((_BYTE *)v26 + i) += *(_BYTE *)(v25 + i);
}
while ( 1 )
{
  v12 = 0;
  if ( v17 < v28 )
    v12 = ~(*(_BYTE *)(v25 + v17) != 0);
  if ( !(v12 & 1) )
    break;
  ++v17;
}
n = ((unsigned __int64)(0x28F5C28F5C28F5C3LL * (unsigned __int128)(138 * (v28 - v17) >> 2) >> 64) >> 2) + 1;
v23 = ((unsigned __int64)(0xAAAAAAAAAAAAAAABLL * (unsigned __int128)((v17 + v28) << 6) >> 64) >> 5) - 1;
v11 = (unsigned __int8 *)&v6
    - ((((unsigned __int64)(0x28F5C28F5C28F5C3LL * (unsigned __int128)(138 * (v28 - v17) >> 2) >> 64) >> 2) +
16) & 0xFFFFFFFFFFFFFFF0LL);
memset(v11, 0, n);
v20 = v17;
v18 = n - 1;
while ( v20 < v28 )
{
  v21 = *(unsigned __int8 *)(v25 + v20);
  for ( j = n - 1; ; --j )
  {
    v10 = 1;
    if ( j <= v18 )
      v10 = v21 != 0;
    if ( !v10 )
      break;
    v22 = v11[j] << 6;
    v21 += v11[j] << 8;
    v9 = 64;
    v11[j] = v21 % 58;
    *((_BYTE *)v26 + j) = v22 & 0x3F;
    v22 >>= 6;
    v21 /= 58;
    v27 /= v9;
    if ( !j )
      break;
  }
```

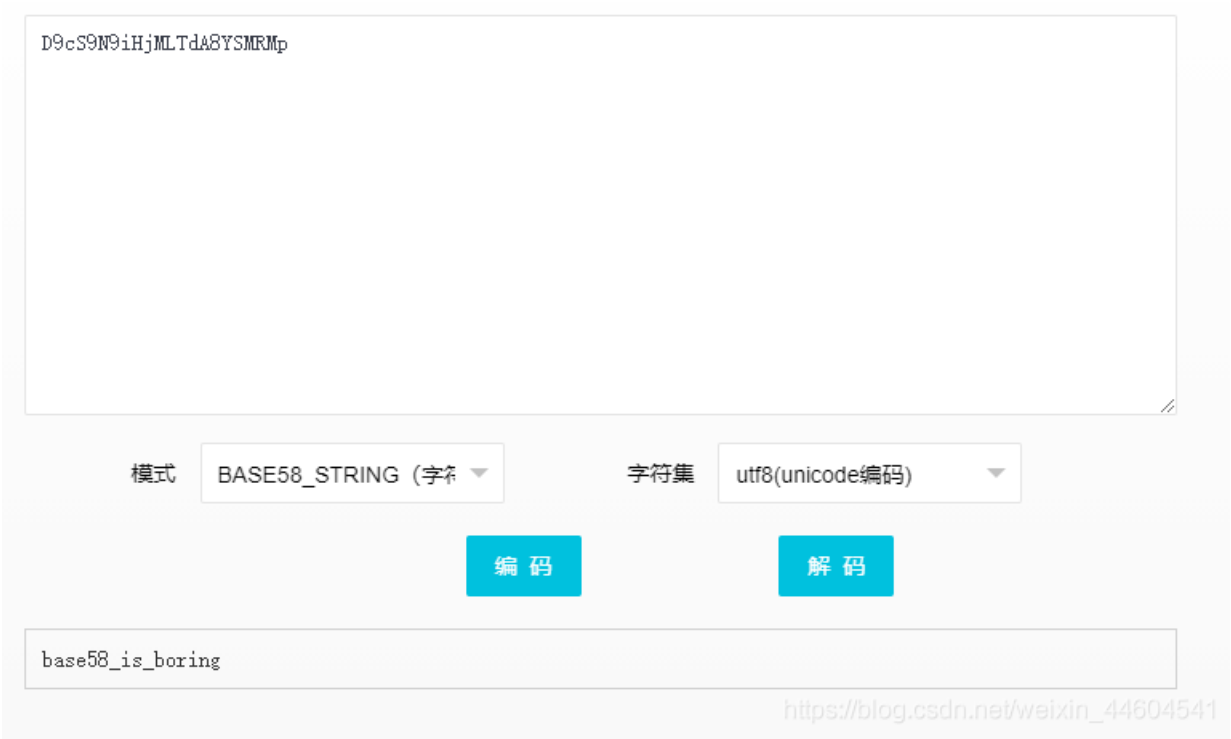
```
    ++v20;
    v18 = j;
  }
  for ( j = 0LL; ; ++j )
  {
    v8 = 0;
    if ( j < n )
      v8 = ~(v11[j] != 0);
    if ( !(v8 & 1) )
      break;
  }
  if ( *v30 > n + v17 - j )
  {
    if ( v17 )
    {
      c = 61;
      memset(s, 49, v17);
      memset(v26, c, v17);
    }
    v20 = v17;
    while ( j < n )
    {
      v4 = v11;
      *((_BYTE *)s + v20) = byte_400EB0[v11[j]];
      *((_BYTE *)v26 + v20++) = byte_400EF0[v4[j++]];
    }
    *((_BYTE *)s + v20) = 0;
    *v30 = v20 + 1;
    if ( !strncmp((const char *)s, "D9", 2uLL)
      && !strncmp((const char *)s + 20, "Mp", 2uLL)
      && !strncmp((const char *)s + 18, "MR", 2uLL)
      && !strncmp((const char *)s + 2, "cS9N", 4uLL)
      && !strncmp((const char *)s + 6, "9iHjM", 5uLL)
      && !strncmp((const char *)s + 11, "LTdA8YS", 7uLL) )
    {
      HIDWORD(v6) = puts("correct!");
    }
    v32 = 1;
    v14 = 1;
  }
  else
  {
    *v30 = n + v17 - j + 1;
    v32 = 0;
    v14 = 1;
  }
  return v32 & 1;
}
```

可以看到最后是s跟`D9cS9N9iHjMLTdA8YSMRMp`作比较
而s来自于v11

```
v11 = (unsigned __int8 *)&v6
    - ((((unsigned __int64)(0x28F5C28F5C28F5C3LL * (unsigned __int128)(138 * (v28 - v17) >> 2) >> 64) >> 2) + 16) & 0xFFFFFFFFFFFFFFF0LL);
memset(v11, 0, n);
v20 = v17;
v18 = n - 1;
while ( v20 < v28 )
{
  v21 = *(unsigned __int8 *)(v25 + v20);
  for ( j = n - 1; ; --j )
  {
    v10 = 1;
    if ( j <= v18 )
      v10 = v21 != 0;
    if ( !v10 )
      break;
    v22 = v11[j] << 6;
    v21 += v11[j] << 8;
    v9 = 64;
    v11[j] = v21 % 58;
    *((_BYTE *)v26 + j) = v22 & 0x3F;
    v22 >>= 6;
    v21 /= 58;
    v27 /= v9;
    if ( !j )
      break;
  }
  ++v20;
  v18 = j;
}
```


分析一下是个base58编码

那直接base58解码



得到flag

结语

base58解码的伪代码
想起之前有道题是base64的伪代码

回头再归纳一下