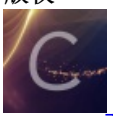
【WhaleCTF逆向题】第一期大骰子writeup & ebCTF 2013 BIN100writeup

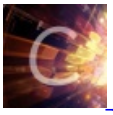
原创

iqiqiya 于 2018-09-18 18:04:41 发布 1028 收藏 4
分类专栏： 我的逆向之路 我的CTF之路 ------WhaleCTF 我的CTF进阶之路 文章标签： 【WhaleCTF逆向题】第一期大骰子writeup 大骰子writeup ebCTF 2013 BIN100writeup reverse CTF

本文链接：https://blog.csdn.net/xiangshangbashaonian/article/details/82761791


我的逆向之路 同时被 3 个专栏收录
108 篇文章 10 订阅
订阅专栏


我的CTF之路
92 篇文章 5 订阅
订阅专栏


------WhaleCTF
8 篇文章 0 订阅
订阅专栏

【20181026更新】 这道题是有两种方法的

0x01:一种就是不管游戏 直接找与flag有关的变量 操作过程 最后直接py脚本计算flag即可(ps:有机会补坑)

0x02:一种就是下面这样按照计算出正常的输入 让程序帮我们计算flag输出

程序有两个反调试的点

0x01: 这里调用isDebuggerPresent()方法 如果检测到正在被动态调试 就将v86的值改为66

```
if ( isDebuggerPresent() )
   v86 = 66;
 else
   v86 = 16;
```

0x02:如果程序执行的时间间隔大于两秒 就改变v85的值

```
    v24 = time(0);
    v79 = v24;
    v81 = v24 - v80;
    if ( v24 - v80 > 2 )
      v85 *= 2;
```

题目信息：

大骰子

4

据说赌神可以获得绝世的宝贝，可是我相信赌神也不能投出7点吧哈哈！答案格式ebCTF{xxxx}

Dice.zip

运行一下

出现

```
[*] ebCTF 2013 Teaser - BIN100 - Dice Game
    To get the flag you will need to throw the correct numbers.

[*] You will first need to throw a three, press enter to throw a dice!
```

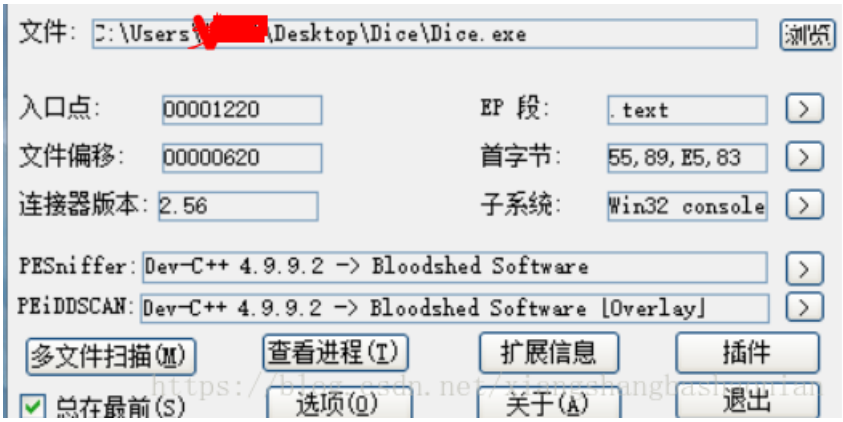

我们可以看出这是ebCTF 2013 BIN100的题目

dice　骰子的意思　　提示我们要拿到flag就得输入正确的数字（骰子要投出相应的数字）

但是骰子哪有7点？？？

那就得爆破了

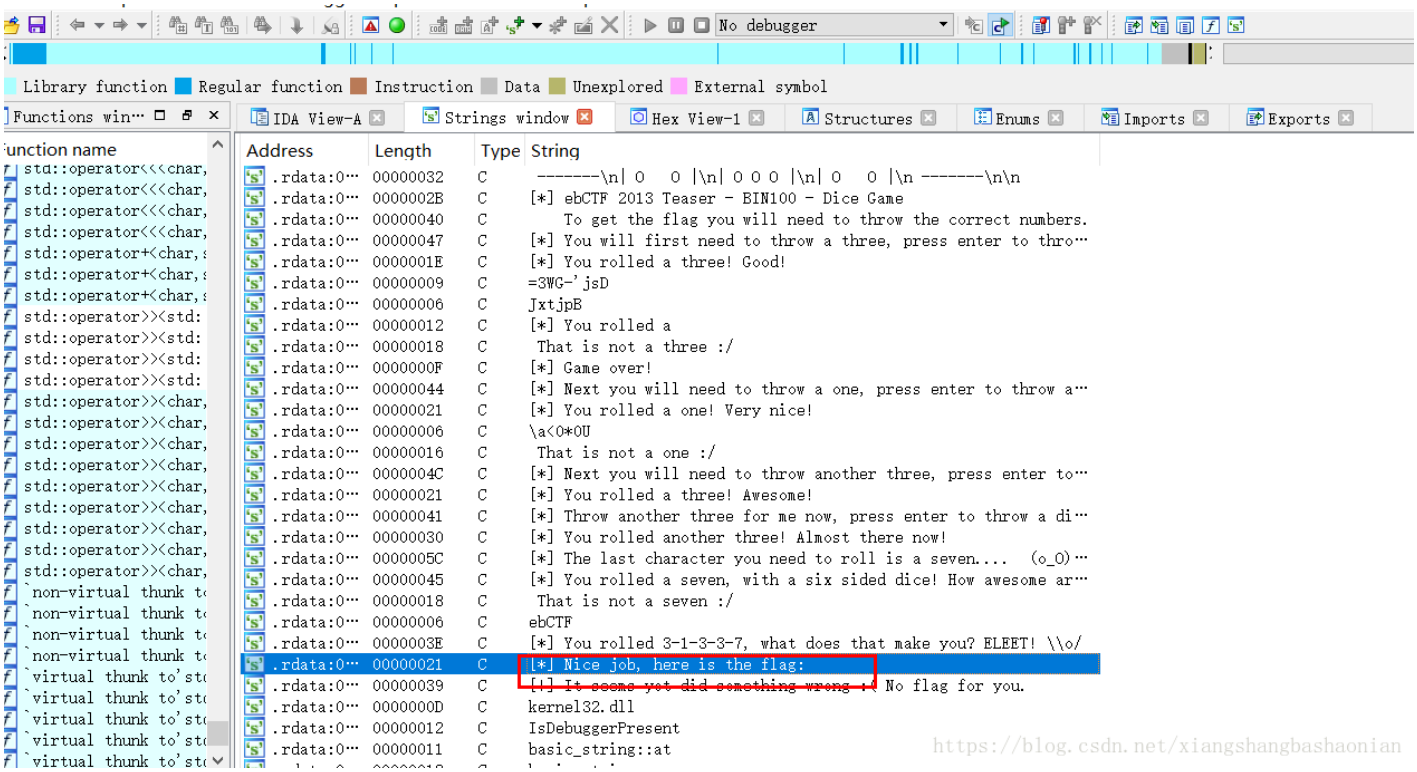
查壳无壳　并且是Dev-C++编写的



Dev-C++ 编辑

Dev-C++是一个Windows环境下的一个适合于初学者使用的轻量级 C/C++ 集成开发环境（IDE）。它是一款自由软件，遵守GPL许可协议分发源代码。它集合了MinGW中的GCC编译器、GDB调试器和 AStyle格式整理器等众多自由软件。原开发公司Bloodshed 在开发完 4.9.9.2 后停止开发，所以现在由 Orwell 公司继续更新开发，最新版本：5.11。

IDA载入 查看字符串或者直接查看WinMain()也可以找到关键



双击进去 再双击引用

再F5一下

```
int __stdcall WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)
{
  unsigned int v4; // eax
  int v5; // eax
  int v6; // eax
  std::ostream *v7; // eax
  int v8; // eax
  std::ostream *v9; // eax
  std::ostream *v10; // eax
  int v11; // eax
  std::ostream *v12; // eax
  int v13; // eax
  std::ostream *v14; // eax
  std::ostream *v15; // eax
  time_t v16; // eax
  std::ostream *v17; // eax
  std::ostream *v18; // eax
  int v19; // eax
  std::ostream *v20; // eax
  int v21; // eax
  std::ostream *v22; // eax
```

```
std::ostream *v23; // eax
time_t v24; // eax
std::ostream *v25; // eax
std::ostream *v26; // eax
int v27; // eax
std::ostream *v28; // eax
int v29; // eax
std::ostream *v30; // eax
std::ostream *v31; // eax
time_t v32; // eax
std::ostream *v33; // eax
std::ostream *v34; // eax
int v35; // eax
unsigned int v36; // eax
_BYTE *v37; // eax
std::ostream *v38; // eax
int v39; // eax
std::ostream *v40; // eax
std::ostream *v41; // eax
std::ostream *v42; // eax
std::ostream *v43; // eax
int v44; // eax
std::ostream *v45; // eax
int v46; // eax
std::ostream *v47; // eax
std::ostream *v48; // eax
time_t v49; // eax
unsigned int v50; // eax
_BYTE *v51; // eax
unsigned int v52; // eax
_BYTE *v53; // eax
unsigned int v54; // eax
std::ostream *v55; // eax
std::ostream *v56; // eax
int v57; // eax
int v58; // eax
int v59; // eax
std::ostream *v60; // eax
int v61; // eax
int v63; // [esp+0h] [ebp-188h]
_BYTE *v64; // [esp+3Ch] [ebp-14Ch]
int v65; // [esp+60h] [ebp-128h]
struct SjLj_Function_Context fctx; // [esp+64h] [ebp-124h]
void *v67; // [esp+88h] [ebp-100h]
int *v68; // [esp+8Ch] [ebp-FCh]
int j; // [esp+98h] [ebp-F0h]
int i; // [esp+9Ch] [ebp-ECh]
int v71; // [esp+A0h] [ebp-E8h]
int v72; // [esp+B0h] [ebp-D8h]
int v73; // [esp+C0h] [ebp-C8h]
int v74; // [esp+D0h] [ebp-B8h]
int v75; // [esp+E0h] [ebp-A8h]
int v76; // [esp+F0h] [ebp-98h]
int v77; // [esp+100h] [ebp-88h]
int v78; // [esp+110h] [ebp-78h]
time_t v79; // [esp+120h] [ebp-68h]
time_t v80; // [esp+124h] [ebp-64h]
int v81; // [esp+128h] [ebp-60h]
int v82; // [esp+12Ch] [ebp-5Ch]
```

```
char v83; // [esp+130h] [ebp-58h]
char v84; // [esp+140h] [ebp-48h]
int v85; // [esp+158h] [ebp-30h]
int v86; // [esp+15Ch] [ebp-2Ch]
char v87; // [esp+160h] [ebp-28h]
char v88; // [esp+170h] [ebp-18h]

fctx.personality = __gxx_personality_sj0;
fctx.lsda = dword_44176C;
fctx.jbuf[0] = &v88;
v67 = &loc_402B5C;
v68 = &v63;
_Unwind_SjLj_Register(&fctx);
fctx.call_site = -1;
std::string::string(&v87);
v85 = 6;
fctx.call_site = 17;
std::string::string(&v84);
fctx.call_site = 16;
std::string::string(&v83);
v4 = time(0);
srand(v4);
std::allocator<char>::allocator(&v77);
fctx.call_site = 14;
std::string::string(&v78, " -------\n|       |\n|   O   |\n|       |\n -------\n\n", &v77);
std::allocator<char>::~allocator(&v77);
std::allocator<char>::allocator(&v76);
fctx.call_site = 12;
std::string::string(&v77, " -------\n|     O |\n|       |\n| O     |\n -------\n\n", &v76);
std::allocator<char>::~allocator(&v76);
std::allocator<char>::allocator(&v75);
fctx.call_site = 10;
std::string::string(&v76, " -------\n|     O |\n|   O   |\n| O     |\n -------\n\n", &v75);
std::allocator<char>::~allocator(&v75);
std::allocator<char>::allocator(&v74);
fctx.call_site = 8;
std::string::string(&v75, " -------\n| O   O |\n|       |\n| O   O |\n -------\n\n", &v74);
std::allocator<char>::~allocator(&v74);
std::allocator<char>::allocator(&v73);
fctx.call_site = 6;
std::string::string(&v74, " -------\n| O   O |\n|   O   |\n| O   O |\n -------\n\n", &v73);
std::allocator<char>::~allocator(&v73);
std::allocator<char>::allocator(&v72);
fctx.call_site = 4;
std::string::string(&v73, " -------\n| O   O |\n| O   O |\n| O   O |\n -------\n\n", &v72);
std::allocator<char>::~allocator(&v72);
std::allocator<char>::allocator(&v71);
fctx.call_site = 2;
std::string::string(&v72, " -------\n| O   O |\n| O O O |\n| O   O |\n -------\n\n", &v71);
std::allocator<char>::~allocator(&v71);
fctx.call_site = 1;
if ( isDebuggerPresent() )
  v86 = 66;
else
  v86 = 16;
fctx.call_site = 1;
v5 = std::ostream::operator<<(&std::cout, std::endl<char,std::char_traits<char>>);
std::operator<<<std::char_traits<char>>(v5, "[*] ebCTF 2013 Teaser - BIN100 - Dice Game");
v6 = std::ostream::operator<<(&std::cout, std::endl<char,std::char_traits<char>>);
v7 = std::operator<<<std::char_traits<char>>(v6, "    To get the flag you will need to throw the correct
```

```
v8 = std::ostream::operator<<(v7, std::endl<char,std::char_traits<char>>);
std::ostream::operator<<(v8, std::endl<char,std::char_traits<char>>);
v9 = std::operator<<<std::char_traits<char>>(
       &std::cout,
       "[*] You will first need to throw a three, press enter to throw a dice!");
std::ostream::operator<<(v9, std::endl<char,std::char_traits<char>>);
std::getline<char,std::char_traits<char>,std::allocator<char>>(&std::cin, &v83);
v80 = time(0);
v82 = rand() % 6 + 1;                         // 取随机数除6取余，然后逐个和1-6进行对比，再和目标数字(3-1-3-3-7
if ( v82 == 1 )
  std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v78);
if ( v82 == 2 )
{
  fctx.call_site = 1;
  std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v77);
}
if ( v82 == 3 )
{
  fctx.call_site = 1;
  std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v76);
}
if ( v82 == 4 )
{
  fctx.call_site = 1;
  std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v75);
}
if ( v82 == 5 )
{
  fctx.call_site = 1;
  std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v74);
}
if ( v82 == 6 )
{
  fctx.call_site = 1;
  std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v73);
}
if ( v82 == 3 )                               // 第一次投掷为3
{
  fctx.call_site = 1;
  v10 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] You rolled a three! Good!");
  v11 = std::ostream::operator<<(v10, std::endl<char,std::char_traits<char>>);
  std::ostream::operator<<(v11, std::endl<char,std::char_traits<char>>);
  v85 *= 2;
  std::string::operator=(&v84, &byte_444240);
  v16 = time(0);
  v79 = v16;
  v81 = v16 - v80;
  if ( v16 - v80 > 2 )                        // 第一次的时间减去第二次的时间 如果大于2
    v85 *= 2;
  fctx.call_site = 1;
  v17 = std::operator<<<std::char_traits<char>>(
          &std::cout,
          "[*] Next you will need to throw a one, press enter to throw a dice!");
  std::ostream::operator<<(v17, std::endl<char,std::char_traits<char>>);
  std::getline<char,std::char_traits<char>,std::allocator<char>>(&std::cin, &v83);
  v80 = time(0);
  v82 = rand() % 6 + 1;
  if ( v82 == 1 )
    std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v78);
```

```
  if ( v82 == 2 )
  {
    fctx.call_site = 1;
    std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v77);
  }
  if ( v82 == 3 )
  {
    fctx.call_site = 1;
    std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v76);
  }
  if ( v82 == 4 )
  {
    fctx.call_site = 1;
    std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v75);
  }
  if ( v82 == 5 )
  {
    fctx.call_site = 1;
    std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v74);
  }
  if ( v82 == 6 )
  {
    fctx.call_site = 1;
    std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v73);
  }
  if ( v82 == 1 )                               // 第二次投掷为1
  {
    fctx.call_site = 1;
    v18 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] You rolled a one! Very nice!");
    v19 = std::ostream::operator<<(v18, std::endl<char,std::char_traits<char>>);
    std::ostream::operator<<(v19, std::endl<char,std::char_traits<char>>);
    v85 += 4;
    std::string::operator=(&v87, &byte_444309);
    v24 = time(0);
    v79 = v24;
    v81 = v24 - v80;
    if ( v24 - v80 > 2 )
      v85 *= 2;
    fctx.call_site = 1;
    v25 = std::operator<<<std::char_traits<char>>(
            &std::cout,
            "[*] Next you will need to throw another three, press enter to throw a dice!");
    std::ostream::operator<<(v25, std::endl<char,std::char_traits<char>>);
    std::getline<char,std::char_traits<char>,std::allocator<char>>(&std::cin, &v83);
    v80 = time(0);
    v82 = rand() % 6 + 1;
    if ( v82 == 1 )
      std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v78);
    if ( v82 == 2 )
    {
      fctx.call_site = 1;
      std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v77);
    }
    if ( v82 == 3 )
    {
      fctx.call_site = 1;
      std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v76);
    }
    if ( v82 == 4 )
    {
```

```
{
  fctx.call_site = 1;
  std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v75);
}
if ( v82 == 5 )
{
  fctx.call_site = 1;
  std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v74);
}
if ( v82 == 6 )
{
  fctx.call_site = 1;
  std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v73);
}
if ( v82 == 3 )                         // 第三次投掷为3
{
  fctx.call_site = 1;
  v26 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] You rolled a three! Awesome!");
  v27 = std::ostream::operator<<(v26, std::endl<char,std::char_traits<char>>);
  std::ostream::operator<<(v27, std::endl<char,std::char_traits<char>>);
  v85 *= 3;
  v32 = time(0);
  v79 = v32;
  v81 = v32 - v80;
  if ( v32 - v80 > 2 )
    v85 *= 2;
  fctx.call_site = 1;
  v33 = std::operator<<<std::char_traits<char>>(
          &std::cout,
          "[*] Throw another three for me now, press enter to throw a dice!");
  std::ostream::operator<<(v33, std::endl<char,std::char_traits<char>>);
  std::getline<char,std::char_traits<char>,std::allocator<char>>(&std::cin, &v83);
  v80 = time(0);
  v82 = rand() % 6 + 1;
  if ( v82 == 1 )
    std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v78);
  if ( v82 == 2 )
  {
    fctx.call_site = 1;
    std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v77);
  }
  if ( v82 == 3 )
  {
    fctx.call_site = 1;
    std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v76);
  }
  if ( v82 == 4 )
  {
    fctx.call_site = 1;
    std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v75);
  }
  if ( v82 == 5 )
  {
    fctx.call_site = 1;
    std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v74);
  }
  if ( v82 == 6 )
  {
    fctx.call_site = 1;
    std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v73);
```

```
  }
  if ( v82 == 3 )                       // 第四次投掷为3
  {
    fctx.call_site = 1;
    v34 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] You rolled another three! Almost t
    v35 = std::ostream::operator<<(v34, std::endl<char,std::char_traits<char>>);
    std::ostream::operator<<(v35, std::endl<char,std::char_traits<char>>);
    v85 += 2;
    for ( i = 0; ; ++i )
    {
      fctx.call_site = 1;
      v36 = std::string::size(&v87);
      if ( i >= v36 )
        break;
      v37 = std::string::operator[](&v87, i);
      *v37 ^= v86;
    }
    v79 = time(0);
    v81 = v79 - v80;
    if ( v79 - v80 > 2 )
      v85 *= 2;
    fctx.call_site = 1;
    v42 = std::operator<<<std::char_traits<char>>(
            &std::cout,
            "[*] The last character you need to roll is a seven....  (o_O)  Press enter to throw a di
    std::ostream::operator<<(v42, std::endl<char,std::char_traits<char>>);
    std::getline<char,std::char_traits<char>,std::allocator<char>>(&std::cin, &v83);
    v80 = time(0);
    if ( v82 == 1 )
      std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v78);
    if ( v82 == 2 )
    {
      fctx.call_site = 1;
      std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v77);
    }
    if ( v82 == 3 )
    {
      fctx.call_site = 1;
      std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v76);
    }
    if ( v82 == 4 )
    {
      fctx.call_site = 1;
      std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v75);
    }
    if ( v82 == 5 )
    {
      fctx.call_site = 1;
      std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v74);
    }
    if ( v82 == 6 )
    {
      fctx.call_site = 1;
      std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v73);
    }
    if ( v82 == 7 )
    {
      fctx.call_site = 1;
      std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v72);
    }
```

```
}
if ( v82 == 7 )                         // 第五次投掷为7
{
  fctx.call_site = 1;
  v43 = std::operator<<<std::char_traits<char>>(
          &std::cout,
          "[*] You rolled a seven, with a six sided dice! How awesome are you?!");
  v44 = std::ostream::operator<<(v43, std::endl<char,std::char_traits<char>>);
  std::ostream::operator<<(v44, std::endl<char,std::char_traits<char>>);
  v85 *= 2;
  v85 *= 50;
  v85 /= 50;
  v85 += 65;
  v85 -= 65;
  v85 *= 42;
  v85 /= 42;
  v49 = time(0);
  v79 = v49;
  v81 = v49 - v80;
  if ( v49 - v80 > 2 )
    v85 *= 2;
  for ( i = 0; ; ++i )
  {
    fctx.call_site = 1;
    v50 = std::string::size(&v87);
    if ( i >= v50 )
      break;
    v51 = std::string::operator[](&v87, i);
    *v51 ^= v85;
  }
  i = 0;
  for ( j = 0; ; ++j )
  {
    fctx.call_site = 1;
    v52 = std::string::size(&v84);
    if ( j >= v52 )
      break;
    v64 = std::string::operator[](&v84, j);
    v53 = std::string::operator[](&v87, i);
    *v64 ^= *v53;
    ++i;
    v54 = std::string::length(&v87);
    if ( i >= v54 )
      i = 0;
  }
  fctx.call_site = 1;
  if ( std::string::find(&v84, "ebCTF", 0) == -1 )
  {
    fctx.call_site = 1;
    v59 = std::ostream::operator<<(&std::cout, std::endl<char,std::char_traits<char>>);
    v60 = std::operator<<<std::char_traits<char>>(
            v59,
            "[!] It seems yot did something wrong :( No flag for you.");
    v61 = std::ostream::operator<<(v60, std::endl<char,std::char_traits<char>>);
    std::ostream::operator<<(v61, std::endl<char,std::char_traits<char>>);
    fctx.call_site = 3;
    std::string::~string(&v72);
    fctx.call_site = 5;
    std::string::~string(&v73);
    fctx.call_site = 7;
```

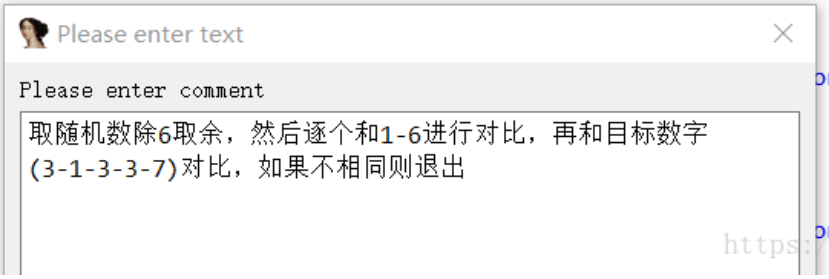
```
      std::string::~string(&v74);
      fctx.call_site = 9;
      std::string::~string(&v75);
      fctx.call_site = 11;
      std::string::~string(&v76);
      fctx.call_site = 13;
      std::string::~string(&v77);
      fctx.call_site = 15;
      std::string::~string(&v78);
      fctx.call_site = 16;
      std::string::~string(&v83);
      fctx.call_site = 17;
      std::string::~string(&v84);
      fctx.call_site = -1;
      std::string::~string(&v87);
      v65 = 0;
    }
    else
    {
      v55 = std::operator<<<std::char_traits<char>>(
              &std::cout,
              "[*] You rolled 3-1-3-3-7, what does that make you? ELEET! \\o/");
      std::ostream::operator<<(v55, std::endl<char,std::char_traits<char>>);
      v56 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] Nice job, here is the flag: ")
      v57 = std::operator<<<char,std::char_traits<char>,std::allocator<char>>(v56, &v84);
      v58 = std::ostream::operator<<(v57, std::endl<char,std::char_traits<char>>);
      std::ostream::operator<<(v58, std::endl<char,std::char_traits<char>>);
      fctx.call_site = 3;
      std::string::~string(&v72);
      fctx.call_site = 5;
      std::string::~string(&v73);
      fctx.call_site = 7;
      std::string::~string(&v74);
      fctx.call_site = 9;
      std::string::~string(&v75);
      fctx.call_site = 11;
      std::string::~string(&v76);
      fctx.call_site = 13;
      std::string::~string(&v77);
      fctx.call_site = 15;
      std::string::~string(&v78);
      fctx.call_site = 16;
      std::string::~string(&v83);
      fctx.call_site = 17;
      std::string::~string(&v84);
      fctx.call_site = -1;
      std::string::~string(&v87);
    }
  }
  else
  {
    fctx.call_site = 1;
    v45 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] You rolled a ");
    v46 = std::ostream::operator<<(v45, v82);
    v47 = std::operator<<<std::char_traits<char>>(v46, " That is not a seven :/");
    std::ostream::operator<<(v47, std::endl<char,std::char_traits<char>>);
    v48 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] Game over!");
    std::ostream::operator<<(v48, std::endl<char,std::char_traits<char>>);
    fctx.call_site = 3;
```

```
        std::string::~string(&v72);
        fctx.call_site = 5;
        std::string::~string(&v73);
        fctx.call_site = 7;
        std::string::~string(&v74);
        fctx.call_site = 9;
        std::string::~string(&v75);
        fctx.call_site = 11;
        std::string::~string(&v76);
        fctx.call_site = 13;
        std::string::~string(&v77);
        fctx.call_site = 15;
        std::string::~string(&v78);
        fctx.call_site = 16;
        std::string::~string(&v83);
        fctx.call_site = 17;
        std::string::~string(&v84);
        fctx.call_site = -1;
        std::string::~string(&v87);
        v65 = 0;
      }
    }
    else
    {
      fctx.call_site = 1;
      v38 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] You rolled a ");
      v39 = std::ostream::operator<<(v38, v82);
      v40 = std::operator<<<std::char_traits<char>>(v39, " That is not a three :/");
      std::ostream::operator<<(v40, std::endl<char,std::char_traits<char>>);
      v41 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] Game over!");
      std::ostream::operator<<(v41, std::endl<char,std::char_traits<char>>);
      fctx.call_site = 3;
      std::string::~string(&v72);
      fctx.call_site = 5;
      std::string::~string(&v73);
      fctx.call_site = 7;
      std::string::~string(&v74);
      fctx.call_site = 9;
      std::string::~string(&v75);
      fctx.call_site = 11;
      std::string::~string(&v76);
      fctx.call_site = 13;
      std::string::~string(&v77);
      fctx.call_site = 15;
      std::string::~string(&v78);
      fctx.call_site = 16;
      std::string::~string(&v83);
      fctx.call_site = 17;
      std::string::~string(&v84);
      fctx.call_site = -1;
      std::string::~string(&v87);
      v65 = 0;
    }
  }
  else
  {
    fctx.call_site = 1;
    v28 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] You rolled a ");
    v29 = std::ostream::operator<<(v28, v82);
    v30 = std::operator<<<std::char_traits<char>>(v29, " That is not a three :/");
```

```
      std::ostream::operator<<(v30, std::endl<char,std::char_traits<char>>);
      v31 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] Game over!");
      std::ostream::operator<<(v31, std::endl<char,std::char_traits<char>>);
      fctx.call_site = 3;
      std::string::~string(&v72);
      fctx.call_site = 5;
      std::string::~string(&v73);
      fctx.call_site = 7;
      std::string::~string(&v74);
      fctx.call_site = 9;
      std::string::~string(&v75);
      fctx.call_site = 11;
      std::string::~string(&v76);
      fctx.call_site = 13;
      std::string::~string(&v77);
      fctx.call_site = 15;
      std::string::~string(&v78);
      fctx.call_site = 16;
      std::string::~string(&v83);
      fctx.call_site = 17;
      std::string::~string(&v84);
      fctx.call_site = -1;
      std::string::~string(&v87);
      v65 = 0;
    }
  }
  else
  {
    fctx.call_site = 1;
    v20 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] You rolled a ");
    v21 = std::ostream::operator<<(v20, v82);
    v22 = std::operator<<<std::char_traits<char>>(v21, " That is not a one :/");
    std::ostream::operator<<(v22, std::endl<char,std::char_traits<char>>);
    v23 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] Game over!");
    std::ostream::operator<<(v23, std::endl<char,std::char_traits<char>>);
    fctx.call_site = 3;
    std::string::~string(&v72);
    fctx.call_site = 5;
    std::string::~string(&v73);
    fctx.call_site = 7;
    std::string::~string(&v74);
    fctx.call_site = 9;
    std::string::~string(&v75);
    fctx.call_site = 11;
    std::string::~string(&v76);
    fctx.call_site = 13;
    std::string::~string(&v77);
    fctx.call_site = 15;
    std::string::~string(&v78);
    fctx.call_site = 16;
    std::string::~string(&v83);
    fctx.call_site = 17;
    std::string::~string(&v84);
    fctx.call_site = -1;
    std::string::~string(&v87);
    v65 = 0;
  }
}
else
```

```
  {
    fctx.call_site = 1;
    v12 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] You rolled a ");
    v13 = std::ostream::operator<<(v12, v82);
    v14 = std::operator<<<std::char_traits<char>>(v13, " That is not a three :/");
    std::ostream::operator<<(v14, std::endl<char,std::char_traits<char>>);
    v15 = std::operator<<<std::char_traits<char>>(&std::cout, "[*] Game over!");
    std::ostream::operator<<(v15, std::endl<char,std::char_traits<char>>);
    fctx.call_site = 3;
    std::string::~string(&v72);
    fctx.call_site = 5;
    std::string::~string(&v73);
    fctx.call_site = 7;
    std::string::~string(&v74);
    fctx.call_site = 9;
    std::string::~string(&v75);
    fctx.call_site = 11;
    std::string::~string(&v76);
    fctx.call_site = 13;
    std::string::~string(&v77);
    fctx.call_site = 15;
    std::string::~string(&v78);
    fctx.call_site = 16;
    std::string::~string(&v83);
    fctx.call_site = 17;
    std::string::~string(&v84);
    fctx.call_site = -1;
    std::string::~string(&v87);
    v65 = 0;
  }
  _Unwind_SjLj_Unregister(&fctx);
  return v65;
}
```

```
std::getline<char,std::char_traits<char>,std::allocator<char>>(&std::cin, &v83);
v80 = time(0);
v82 = rand() % 6 + 1;                    // 取随机数除6取余，然后逐个和1-6进行对比，再和目标数字(
if ( v82 == 1 )
  std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &v78);
if ( v82 == 2 )
```




```
or<char>>(&std::cout, &v77);
or<char>>(&std::cout, &v76);
```


具体分析　上边代码都有

可以看出程序是每次读取我们的一个输入

先与1~6进行比较

接着再与3-1-3-3-7尽心比较

判断他们的跳转都是jnz

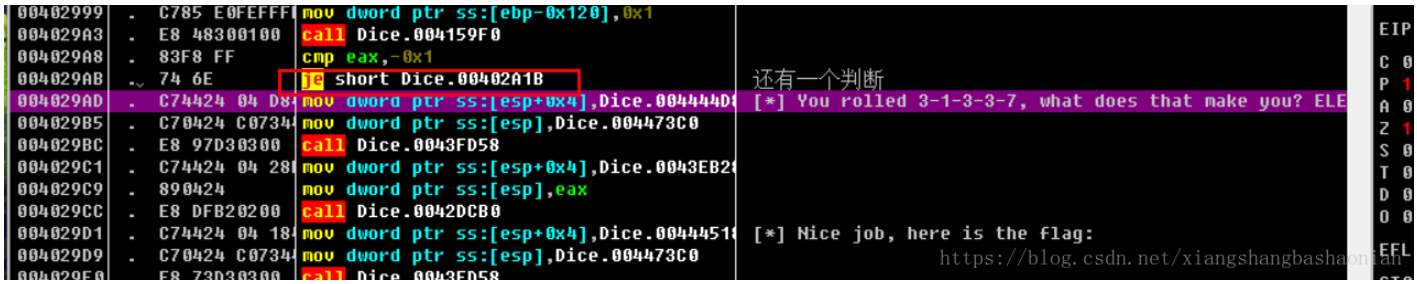
我们只需要将他们都nop掉保存即可

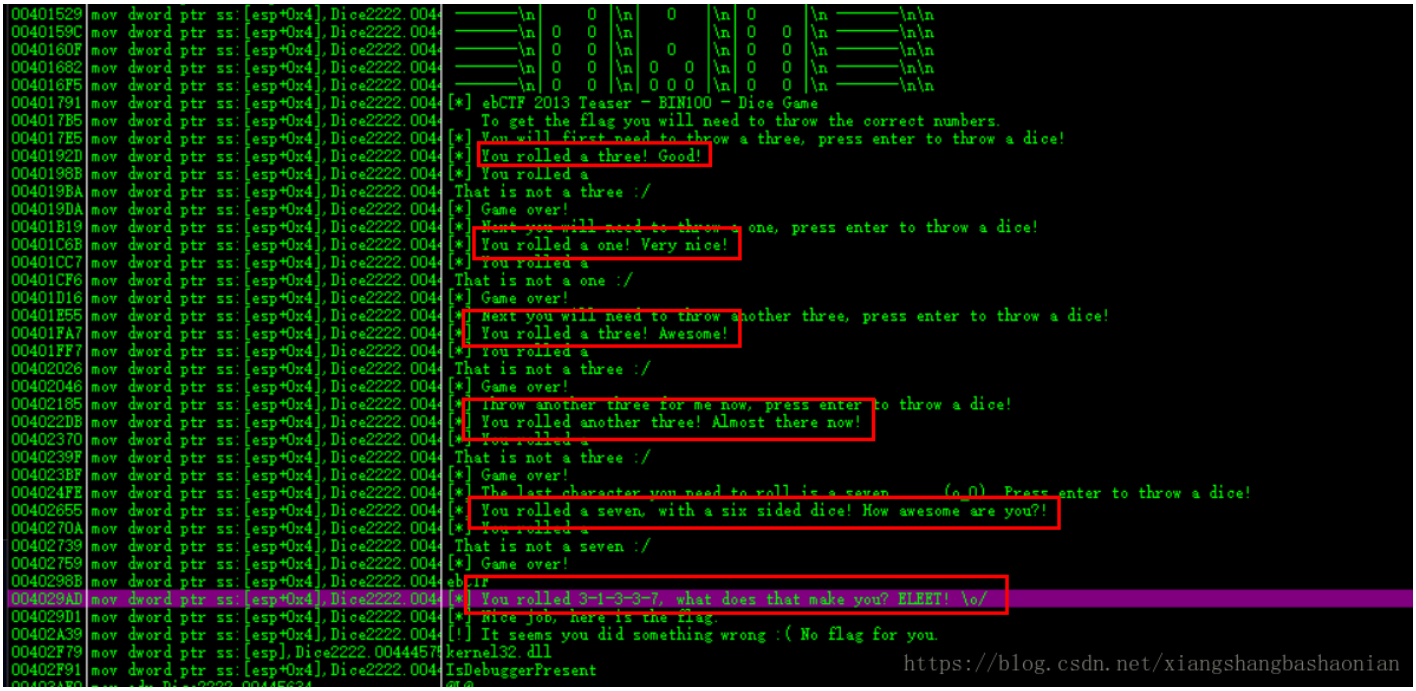
最简单的方法就是直接在IDA中改字节码 （以后再补坑。）

最直接的就是OD搜索ASCII码　从字符串向上找到关键调转jnz　nop

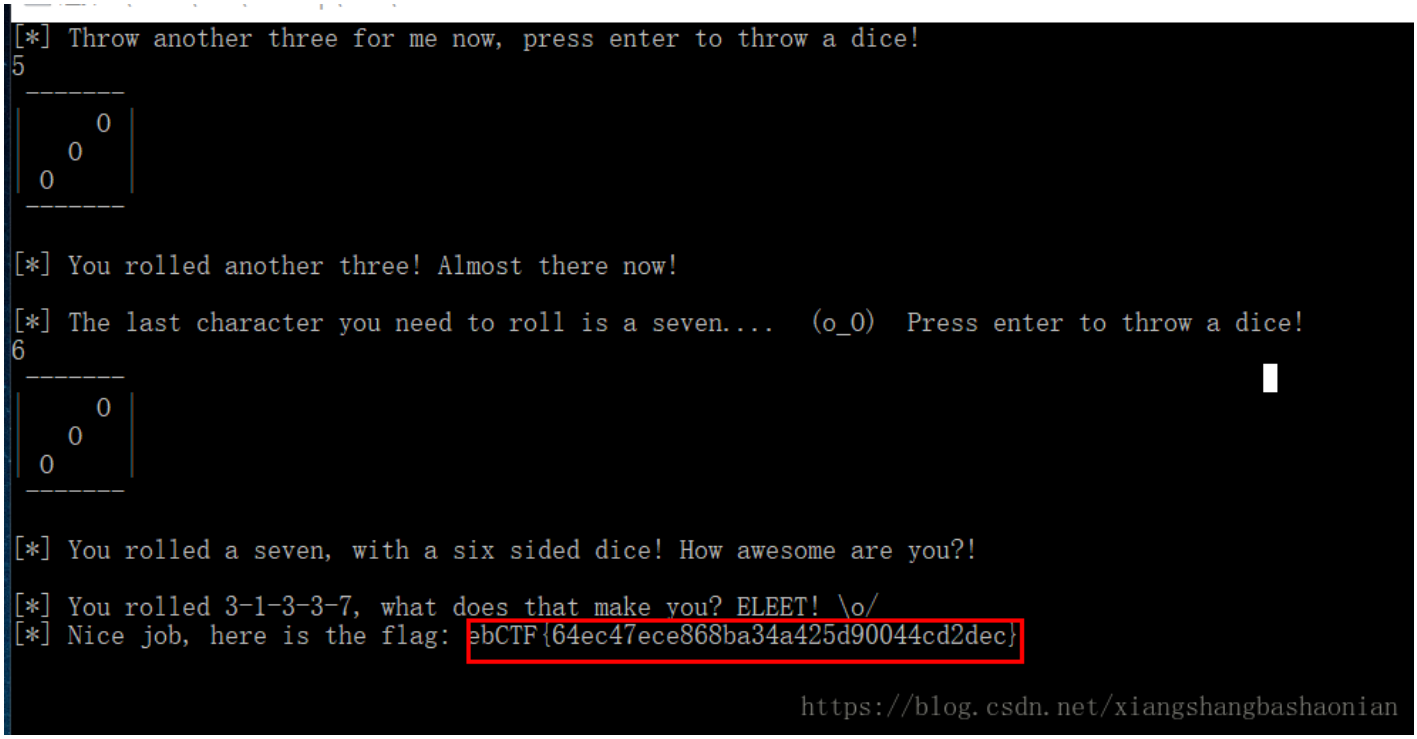
掉即可

需要注意的是还有一个je跳转　也需要nop掉





最后保存 运行　随意输入 即可得到flag



留个坑：里边还有time()这个东西　我还没搞明白 。