easy_Maze 攻防世界

原创

北风~ 于 2020-04-28 18:09:29 发布 1112 收藏 1
分类专栏： 逆向与保护
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/weixin_45055269/article/details/105819835
版权

逆向与保护 专栏收录该内容
65 篇文章 4 订阅
订阅专栏

工具

IDA

思路展开

maze类：1.内存中画出一张地图（地图变换） 2.明确起点和终点 3.（四个字符对应上下左右）flag就是走出的路径

题目提示是maze类的，找上面三个关键点

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  __int64 v3; // rax
  int v5[7]; // [rsp+0h] [rbp-270h]
  int v6; // [rsp+C0h] [rbp-1B0h]
  int v7[7]; // [rsp+D0h] [rbp-1A0h]
  int v8; // [rsp+190h] [rbp-E0h]
  int v9[7]; // [rsp+1A0h] [rbp-D0h]
  int v10; // [rsp+1BCh] [rbp-B4h]
  int v11; // [rsp+1C0h] [rbp-B0h]
  int v12; // [rsp+1C4h] [rbp-ACh]
  int v13; // [rsp+1C8h] [rbp-A8h]
  int v14; // [rsp+1CCh] [rbp-A4h]
  int v15; // [rsp+1D0h] [rbp-A0h]
  int v16; // [rsp+1D4h] [rbp-9Ch]
  int v17; // [rsp+1D8h] [rbp-98h]
  int v18; // [rsp+1DCh] [rbp-94h]
  int v19; // [rsp+1E0h] [rbp-90h]
  int v20; // [rsp+1E4h] [rbp-8Ch]
  int v21; // [rsp+1E8h] [rbp-88h]
  int v22; // [rsp+1ECh] [rbp-84h]
  int v23; // [rsp+1F0h] [rbp-80h]
  int v24; // [rsp+1F4h] [rbp-7Ch]
  int v25; // [rsp+1F8h] [rbp-78h]
int v25; // [rsp+1F8h] [rbp-78h]
int v26; // [rsp+1FCh] [rbp-74h]
int v27; // [rsp+200h] [rbp-70h]
int v28; // [rsp+204h] [rbp-6Ch]
int v29; // [rsp+208h] [rbp-68h]
int v30; // [rsp+20Ch] [rbp-64h]
int v31; // [rsp+210h] [rbp-60h]
int v32; // [rsp+214h] [rbp-5Ch]
int v33; // [rsp+218h] [rbp-58h]
int v34; // [rsp+21Ch] [rbp-54h]
int v35; // [rsp+220h] [rbp-50h]
int v36; // [rsp+224h] [rbp-4Ch]
int v37; // [rsp+228h] [rbp-48h]
int v38; // [rsp+22Ch] [rbp-44h]
int v39; // [rsp+230h] [rbp-40h]
int v40; // [rsp+234h] [rbp-3Ch]
int v41; // [rsp+238h] [rbp-38h]
int v42; // [rsp+23Ch] [rbp-34h]
int v43; // [rsp+240h] [rbp-30h]
int v44; // [rsp+244h] [rbp-2Ch]
int v45; // [rsp+248h] [rbp-28h]
int v46; // [rsp+24Ch] [rbp-24h]
int v47; // [rsp+250h] [rbp-20h]
int v48; // [rsp+254h] [rbp-1Ch]
int v49; // [rsp+258h] [rbp-18h]
int v50; // [rsp+25Ch] [rbp-14h]
int v51; // [rsp+260h] [rbp-10h]

v9[0] = 1;
v9[1] = 1;
v9[2] = -1;
v9[3] = 1;
v9[4] = -1;
v9[5] = 1;
v9[6] = -1;
v10 = 0;
v11 = 0;
v12 = 0;
v13 = 0;
v14 = 1;
v15 = -1;
v16 = 0;
v17 = 0;
v18 = 1;
v19 = 0;
v20 = 0;
v21 = 1;
v22 = 0;
v23 = -1;
v24 = -1;
v25 = 0;
v26 = 1;
v27 = 0;
v28 = 1;
v29 = -1;
v30 = 0;
v31 = -1;
v32 = 0;
v33 = 0;
v34 = 0;
```

```
  v35 = 0;
  v36 = 0;
  v37 = 1;
  v38 = -1;
  v39 = -1;
  v40 = 1;
  v41 = -1;
  v42 = 0;
  v43 = -1;
  v44 = 2;
  v45 = 1;
  v46 = -1;
  v47 = 0;
  v48 = 0;
  v49 = -1;
  v50 = 1;
  v51 = 0;
  memset(v7, 0, 0xC0uLL);
  v8 = 0;
  memset(v5, 0, 0xC0uLL);
  v6 = 0;
  Step_0((int (*)[7])v9, 7, (int (*)[7])v7); #地图变换
  Step_1((int (*)[7])v7, 7, (int (*)[7])v5); #地图变换
  v3 = std::operator<<<std::char_traits<char>>(&_bss_start, "Please help me out!");
  std::ostream::operator<<(v3, &std::endl<char,std::char_traits<char>>);
  Step_2((int (*)[7])v5); #输入，验证
  system("pause");
  return 0;
}
```

进入Step_2函数

```
__int64 __fastcall Step_2(int (*a1)[7])
{
  int v1; // eax
  __int64 v2; // rax
  __int64 v3; // rax
  __int64 result; // rax
  __int64 v5; // rax
  char v6[35]; // [rsp+10h] [rbp-30h]
  char v7; // [rsp+33h] [rbp-Dh]
  int v8; // [rsp+34h] [rbp-Ch]
  int v9; // [rsp+38h] [rbp-8h]
  int v10; // [rsp+3Ch] [rbp-4h]

  v10 = 0;
  v9 = 0;
  v8 = 0;   #初始位置[0][0]
  while ( v8 <= 29 && (*a1)[7 * v10 + v9] == 1 )  #最多30步，走1
  {
    std::operator>><char,std::char_traits<char>>(&std::cin, &v7);
    v1 = v8++;
    v6[v1] = v7;
    if ( v7 == 'd' ) #向右
    {
      ++v9;
    }
    else if ( v7 > 'd' )
    {
      if ( v7 == 's' ) #向下
```

```
   if ( v7 == 's' ) #向下
   {
    ++v10;
   }
   else
   {
    if ( v7 != 'w' ) #向上
     goto LABEL_14;
    --v10;
   }
  }
  else if ( v7 == 'a' ) #向左
  {
   --v9;
  }
  else
  {
LABEL_14:
   v2 = std::operator<<<std::char_traits<char>>(&_bss_start, "include illegal words.");
   std::ostream::operator<<(v2, &std::endl<char,std::char_traits<char>>);
  }
 }
 if ( v10 != 6 || v9 != 6 ) #结束位置[6][6]
 {
  v5 = std::operator<<<std::char_traits<char>>(&_bss_start, "Oh no!,Please try again~~");
  std::ostream::operator<<(v5, &std::endl<char,std::char_traits<char>>);
  result = 0LL;
 }
 else
 {
  v3 = std::operator<<<std::char_traits<char>>(&_bss_start, "Congratulations!");
  std::ostream::operator<<(v3, &std::endl<char,std::char_traits<char>>);
  output(v6, v8);
  result = 1LL;
 }
 return result;
}
```

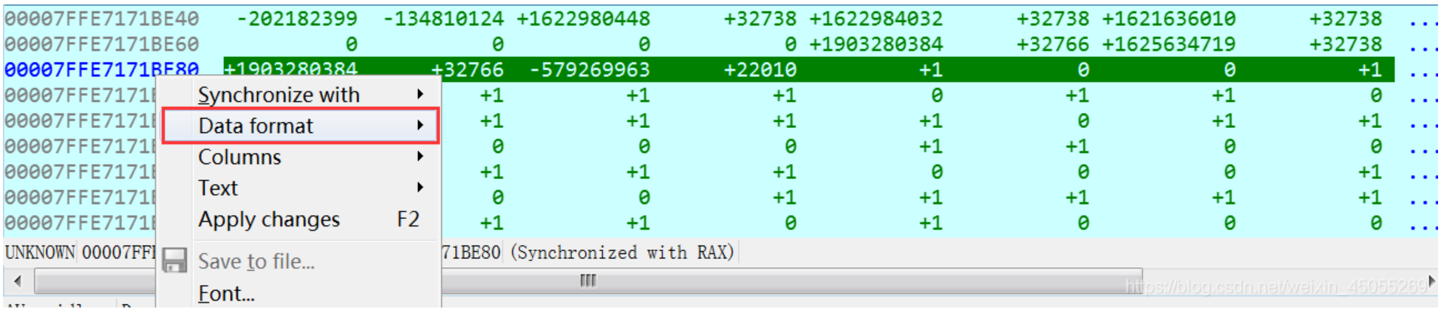
综上：

1.地图两次变换（可通过动态调试找到）

2.起点[0][0]到[6][6]

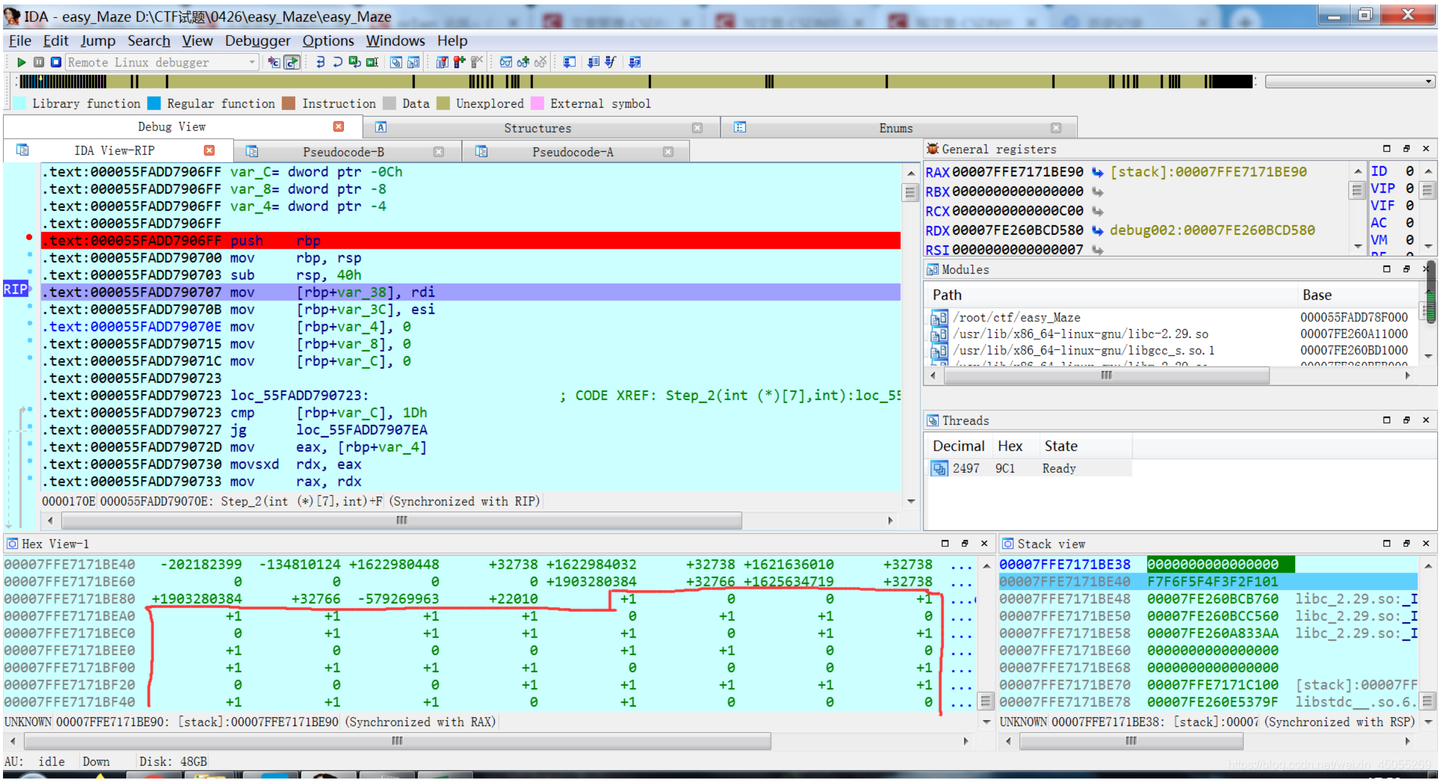
3.wasd 对应上左下右

动态调试地图（这里小坑，需要设置显示的格式（下图））（设置成4byte_Integer和Signed）



```
00007FFE7171BE40  -202182399 -134810124 +1622980448   +32738 +1622984032   +32738 +1621636010   +32738 ...
00007FFE7171BE60           0          0           0        0 +1903280384   +32766 +1625634719   +32738 ...
00007FFE7171BE80 +1903280384     +32766  -579269963   +22010          +1        0           0       +1 ...
00007FFE7171I                        +1          +1       +1           0       +1          +1        0 ...
00007FFE7171I                        +1          +1       +1          +1        0          +1       +1 ...
00007FFE7171I                         0           0        0          +1       +1           0        0 ...
00007FFE7171I                        +1          +1       +1           0        0           0       +1 ...
00007FFE7171I                         0           0       +1          +1       +1          +1       +1 ...
00007FFE7171I                        +1          +1        0          +1        0           0        0 ...
UNKNOWN 00007FFI                    71BE80 (Synchronized with RAX)
```

动调dump出地图



```
.text:000055FADD7906FF var_C= dword ptr -0Ch
.text:000055FADD7906FF var_8= dword ptr -8
.text:000055FADD7906FF var_4= dword ptr -4
.text:000055FADD7906FF
.text:000055FADD7906FF push    rbp
.text:000055FADD790700 mov     rbp, rsp
.text:000055FADD790703 sub     rsp, 40h
.text:000055FADD790707 mov     [rbp+var_38], rdi
.text:000055FADD79070B mov     [rbp+var_3C], esi
.text:000055FADD79070E mov     [rbp+var_4], 0
.text:000055FADD790715 mov     [rbp+var_8], 0
.text:000055FADD79071C mov     [rbp+var_C], 0
.text:000055FADD790723
.text:000055FADD790723 loc_55FADD790723:                       ; CODE XREF: Step_2(int (*)[7],int):loc_55
.text:000055FADD790723 cmp     [rbp+var_C], 1Dh
.text:000055FADD790727 jg      loc_55FADD7907EA
.text:000055FADD79072D mov     eax, [rbp+var_4]
.text:000055FADD790730 movsxd  rdx, eax
.text:000055FADD790733 mov     rax, rdx
0000170E 000055FADD79070E: Step_2(int (*)[7],int)+F (Synchronized with RIP)
```

```
RAX 00007FFE7171BE90 ↳ [stack]:00007FFE7171BE90    ID  0
RBX 0000000000000000 ↳                             VIP 0
RCX 00000000000000C00 ↳                            VIF 0
RDX 00007FE260BCD580 ↳ debug002:00007FE260BCD580   AC  0
RSI 0000000000000007 ↳                             VM  0
```

Path	Base
/root/ctf/easy_Maze	000055FADD78F000
/usr/lib/x86_64-linux-gnu/libc-2.29.so	00007FE260A11000
/usr/lib/x86_64-linux-gnu/libgcc_s.so.1	00007FE260BD1000

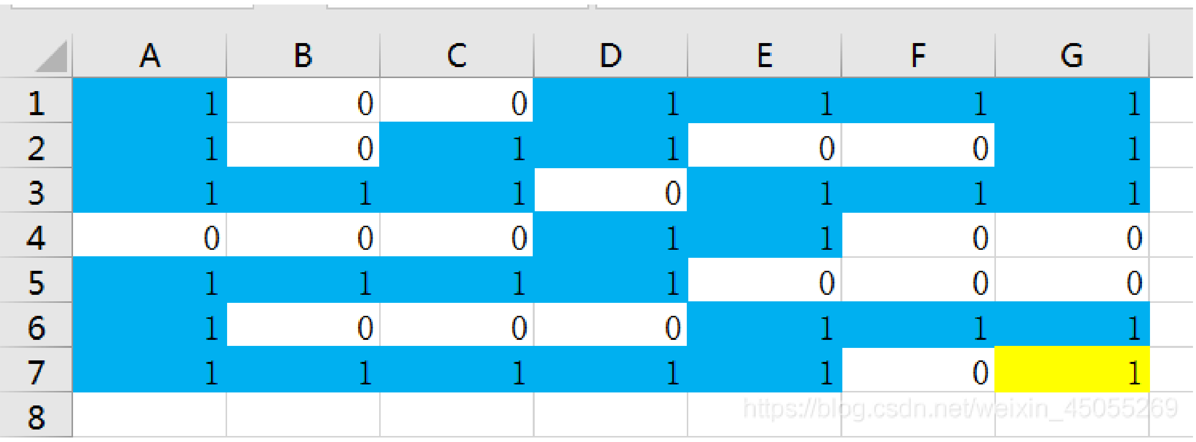
Decimal	Hex	State
2497	9C1	Ready

```
00007FFE7171BE40  -202182399 -134810124 +1622980448   +32738 +1622984032   +32738 +1621636010   +32738 ...
00007FFE7171BE60           0          0           0        0 +1903280384   +32766 +1625634719   +32738 ...
00007FFE7171BE80 +1903280384     +32766  -579269963   +22010          +1        0           0       +1 ...
00007FFE7171BEA0          +1         +1          +1       +1           0       +1          +1        0 ...
00007FFE7171BEC0           0         +1          +1       +1          +1        0          +1       +1 ...
00007FFE7171BEE0          +1          0           0        0          +1       +1           0        0 ...
00007FFE7171BF00          +1         +1          +1        0          +1        0           0       +1 ...
00007FFE7171BF20           0          0           0       +1          +1       +1          +1       +1 ...
00007FFE7171BF40          +1         +1          +1        0          +1        0           0        0 ...
UNKNOWN 00007FFE7171BE90: [stack]:00007FFE7171BE90 (Synchronized with RAX)
```

```
00007FFE7171BE38  0000000000000000
00007FFE7171BE40  F7F6F5F4F3F2F101
00007FFE7171BE48  00007FE260BCB760  libc_2.29.so:_I
00007FFE7171BE50  00007FE260BCC560  libc_2.29.so:_I
00007FFE7171BE58  00007FE260A833AA  libc_2.29.so:_I
00007FFE7171BE60  0000000000000000
00007FFE7171BE68  0000000000000000
00007FFE7171BE70  00007FFE7171C100  [stack]:00007FF
00007FFE7171BE78  00007FE260E5379F  libstdc__.so.6.
UNKNOWN 00007FFE7171BE38: [stack]:0000? (Synchronized with RSP)
```

AU: idle Down Disk: 48GB

地图整出来，走法



	A	B	C	D	E	F	G
1	1	0	0	1	1	1	1
2	1	0	1	1	0	0	1
3	1	1	1	0	1	1	1
4	0	0	0	1	1	0	0
5	1	1	1	1	0	0	0
6	1	0	0	0	1	1	1
7	1	1	1	1	1	0	1
8							

ssddwdwdddssaasasaaassddddwdds

UNCTF{ssddwdwdddssaasasaaassddddwdds}

https://blog.csdn.net/weixin_45055269