2018网鼎杯re--beijing writeup

原创

木木or沐沐 于 2018-08-22 17:56:12 发布 2562 收藏 1
文章标签： 网鼎杯wp 学习note

昨天一上午一直在做这道逆向题，这个题目运行起来输出是乱码，就很烦气，一直在搞这个终端的编码问题，搞了半天结果方向错了，然后就放弃了，其实这道题做出来的人还是很多的，还是自己太菜了，感觉自己的re和pwn还是要多加努力学习才行，今天看到writeup来学习下，做一下学习的笔记哈。

拿到这个binary，用file命令看一下，这个binary的基本信息，这是一个32位的elf文件，放到ubuntu下运行一下发现是乱码。

用ida反汇编逆向一下

```
int __cdecl main()
{
  char v0; // al
  char v1; // al
  char v2; // al
  char v3; // al
  char v4; // al
  char v5; // al
  char v6; // al
  char v7; // al
  char v8; // al
  char v9; // al
  char v10; // al
  char v11; // al
  char v12; // al
  char v13; // al
  char v14; // al
  char v15; // al
  char v16; // al
  char v17; // al
  char v18; // al
  char v19; // al
  char v20; // al

  v0 = encode(dword_804A03C);                   // 6
  printf("%c", v0);
  fflush(stdout);
  v1 = encode(dword_804A044);                   // 9
  printf("%c", v1);
  fflush(stdout);
  v2 = encode(dword_804A0E0);                   // ??
  printf("%c", v2);
  fflush(stdout);
  v3 = encode(dword_804A050);                   // 1
```

```
v3 = encode(dword_804A050);             // 1
printf("%c", v3);
fflush(stdout);
v4 = encode(dword_804A058);             // 0xa
printf("%c", v4);
fflush(stdout);
v5 = encode(dword_804A0E4);             // ??
printf("%c", v5);
fflush(stdout);
v6 = encode(dword_804A064);             // 8
printf("%c", v6);
fflush(stdout);
v7 = encode(word_804A0E8);              // ??
printf("%c", v7);
fflush(stdout);
v8 = encode(dword_804A070);             // 0xb
printf("%c", v8);
fflush(stdout);
v9 = encode(dword_804A078);             // 2
printf("%c", v9);
fflush(stdout);
v10 = encode(dword_804A080);            // 3
printf("%c", v10);
fflush(stdout);
v11 = encode(dword_804A088);            // 1
printf("%c", v11);
fflush(stdout);
v12 = encode(dword_804A090);            // 0xd
printf("%c", v12);
fflush(stdout);
v13 = encode(dword_804A098);            // 4
printf("%c", v13);
fflush(stdout);
v14 = encode(dword_804A0A0);            // 5
printf("%c", v14);
fflush(stdout);
v15 = encode(dword_804A0A8);            // 2
printf("%c", v15);
fflush(stdout);
v16 = encode(dword_804A0B0);            // 7
printf("%c", v16);
fflush(stdout);
v17 = encode(dword_804A0B8);            // 2
printf("%c", v17);
fflush(stdout);
v18 = encode(dword_804A0C0);            // 3
printf("%c", v18);
fflush(stdout);
v19 = encode(dword_804A0C8);            // 1
printf("%c", v19);
fflush(stdout);
```

```
  v20 = encode(dword_804A0D0);                  // 0xc
  printf("%c", v20);
  fflush(stdout);
  printf("\n");
  return 0;
}
```

//ecode函数

```
case 0:
    v2 = byte_804A021 ^ byte_804A020;       // 0x61
    break;
  case 1:
    v2 = byte_804A023 ^ byte_804A022;       // 0x67
    break;
  case 2:
    v2 = byte_804A025 ^ byte_804A024;       // 0x69
    break;
  case 3:
    v2 = byte_804A027 ^ byte_804A026;       // 0x6e
    break;
  case 4:
    v2 = byte_804A029 ^ byte_804A028;       // 0x62
    break;
  case 5:
    v2 = byte_804A02B ^ byte_804A02A;       // 0x65
    break;
  case 6:
    v2 = byte_804A02D ^ byte_804A02C;       // 0x66
    break;
  case 7:
    v2 = byte_804A02F ^ byte_804A02E;       // 0x6a
    break;
  case 8:
    v2 = byte_804A031 ^ byte_804A030;       // 0x6d
    break;
  case 9:
    v2 = byte_804A033 ^ byte_804A032;       // 0x6c
    break;
  case 10:
    v2 = byte_804A035 ^ byte_804A034;       // 0x7b
    break;
  case 11:
    v2 = byte_804A037 ^ byte_804A036;       // 0x7a
    break;
  case 12:
    v2 = byte_804A039 ^ byte_804A038;       // 0x7d
    break;
  case 13:
    v2 = byte_804A03B ^ byte_804A03A;       // 0x5f
    break;
```

经过手工探测发现，发现把异或表达式中的第二个地址的数据按照先前传入的参数所执行case的运算的顺序整理出来再弄成字符就是flag

例如：第一次传入的数据是6，执行case6:chr(66h) --->'f'

但是这里有一个问题，有三个传入的参数是未知的，但是我们可以根据flag字符串的顺序，推断出??=0

然后就得到flag{amazing_beijing}

参考链接：https://blog.csdn.net/SWEET0SWAT/article/details/81879942