攻防世界_Reverse_game

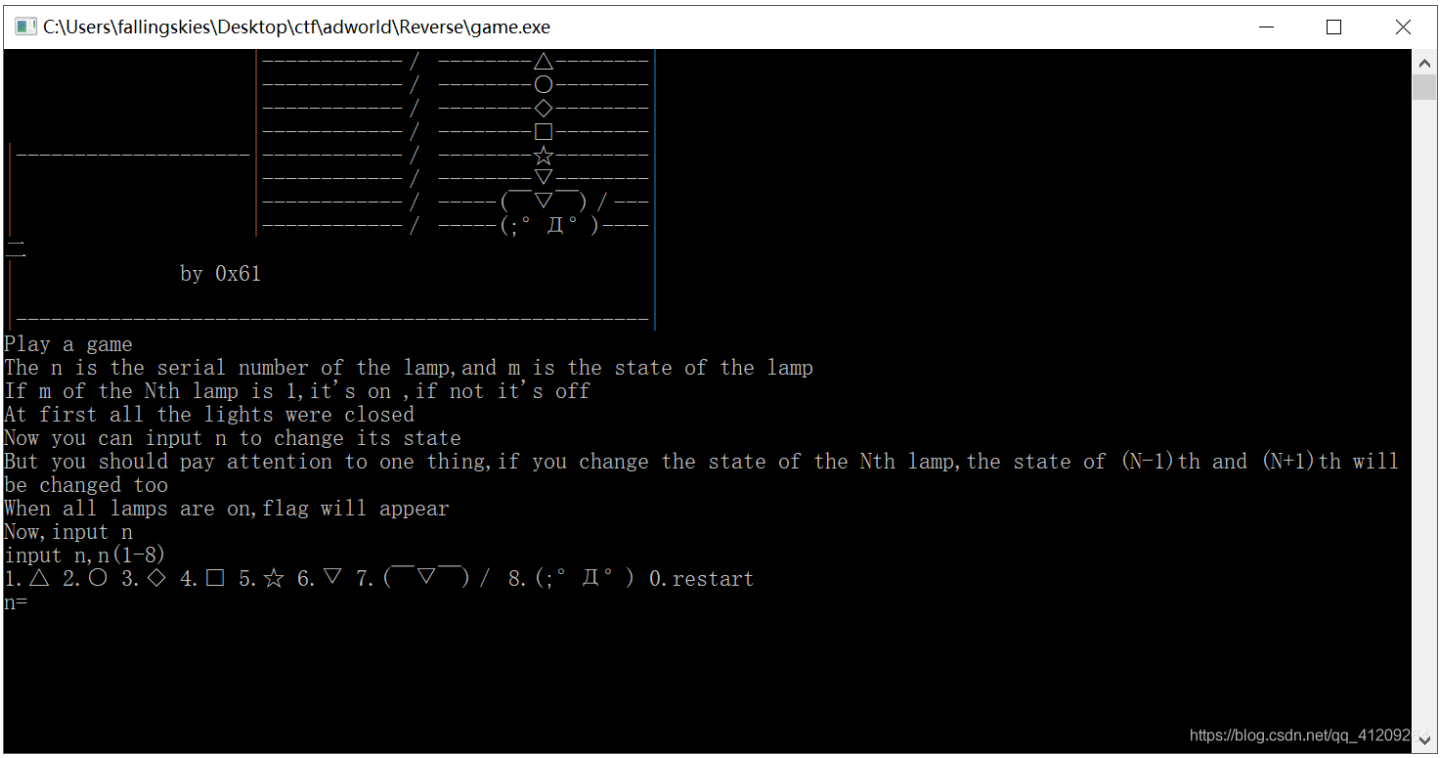
原创

fallingskies22 于 2019-05-11 16:28:11 发布 3524 收藏 2

文章标签： ctf 攻防世界 reverse

本文链接：https://blog.csdn.net/qq_41209264/article/details/90111409

打开发现是一个游戏，挺简单的，直接输入12345678就能直接拿到flag



使用IDA静态分析

在函数中搜索main，进入main函数,按F5进行反汇编

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
int result; // eax

main_0();
return result;
}
```

跳转到main_0函数

```
void __cdecl main_0()
{
  signed int i; // [esp+DCh] [ebp-20h]
  int v1; // [esp+F4h] [ebp-8h]

  sub_13FA7BE(&unk_14AB110);
  sub_13FA7BE(&unk_14AB158);
```

```
  sub_13FA7BE(&unk_14AB1A0);
  sub_13FA7BE(&unk_14AB1E8);
  sub_13FA7BE(&unk_14AB230);
  sub_13FA7BE(&unk_14AB278);
  sub_13FA7BE(&unk_14AB2C0);
  sub_13FA7BE(&unk_14AB308);
  sub_13FA7BE(&unk_14AAFD0);
  sub_13FA7BE("|              by 0x61                                |\n");
  sub_13FA7BE("|                                                     |\n");
  sub_13FA7BE("|-----------------------------------------------------|\n");
  sub_13FA7BE(
    "Play a game\n"
    "The n is the serial number of the lamp,and m is the state of the lamp\n"
    "If m of the Nth lamp is 1,it's on ,if not it's off\n"
    "At first all the lights were closed\n");
  sub_13FA7BE("Now you can input n to change its state\n");
  sub_13FA7BE(
    "But you should pay attention to one thing,if you change the state of the Nth lamp,the state of (N-1)th and
(N+1)th w"
    "ill be changed too\n");
  sub_13FA7BE("When all lamps are on,flag will appear\n");
  sub_13FA7BE("Now,input n \n");
  while ( 1 )
  {
    while ( 1 )
    {
      sub_13FA7BE("input n,n(1-8)\n");
      sub_13F9418();
      sub_13FA7BE("n=");
      sub_13F96D4("%d", &v1);
      sub_13FA7BE("\n");
      if ( v1 >= 0 && v1 <= 8 )
        break;
      sub_13FA7BE("sorry,n error,try again\n");
    }
    if ( v1 )
    {
      sub_13F76D6(v1 - 1);
    }
    else
    {
      for ( i = 0; i < 8; ++i )
      {
        if ( (unsigned int)i >= 9 )
          j____report_rangecheckfailure();
        byte_14D2E28[i] = 0;
      }
    }
    j__system("CLS");
    sub_13F8054();
    if ( byte_14D2E28[0] == 1
      && byte_14D2E28[1] == 1
      && byte_14D2E28[2] == 1
      && byte_14D2E28[3] == 1
      && byte_14D2E28[4] == 1
      && byte_14D2E28[5] == 1
      && byte_14D2E28[6] == 1
      && byte_14D2E28[7] == 1 )
    {
```

```
      sub_13F7AB4();
    }
  }
}
```

分析代码，当所有灯点亮时，进入sub_13F7AB4(),跳转到该函数

```
int sub_13F7AB4(void)
{
  return sub_13FE940();
}
```

继续跳转

```
int sub_13FE940()
{
  signed int i; // [esp+D0h] [ebp-94h]
  char v2; // [esp+DCh] [ebp-88h]
  char v3; // [esp+DDh] [ebp-87h]
  char v4; // [esp+DEh] [ebp-86h]
  char v5; // [esp+DFh] [ebp-85h]
  char v6; // [esp+E0h] [ebp-84h]
  char v7; // [esp+E1h] [ebp-83h]
  char v8; // [esp+E2h] [ebp-82h]
  char v9; // [esp+E3h] [ebp-81h]
  char v10; // [esp+E4h] [ebp-80h]
  char v11; // [esp+E5h] [ebp-7Fh]
  char v12; // [esp+E6h] [ebp-7Eh]
  char v13; // [esp+E7h] [ebp-7Dh]
  char v14; // [esp+E8h] [ebp-7Ch]
  char v15; // [esp+E9h] [ebp-7Bh]
  char v16; // [esp+EAh] [ebp-7Ah]
  char v17; // [esp+EBh] [ebp-79h]
  char v18; // [esp+ECh] [ebp-78h]
  char v19; // [esp+EDh] [ebp-77h]
  char v20; // [esp+EEh] [ebp-76h]
  char v21; // [esp+EFh] [ebp-75h]
  char v22; // [esp+F0h] [ebp-74h]
  char v23; // [esp+F1h] [ebp-73h]
  char v24; // [esp+F2h] [ebp-72h]
  char v25; // [esp+F3h] [ebp-71h]
  char v26; // [esp+F4h] [ebp-70h]
  char v27; // [esp+F5h] [ebp-6Fh]
  char v28; // [esp+F6h] [ebp-6Eh]
  char v29; // [esp+F7h] [ebp-6Dh]
  char v30; // [esp+F8h] [ebp-6Ch]
  char v31; // [esp+F9h] [ebp-6Bh]
  char v32; // [esp+FAh] [ebp-6Ah]
  char v33; // [esp+FBh] [ebp-69h]
  char v34; // [esp+FCh] [ebp-68h]
  char v35; // [esp+FDh] [ebp-67h]
  char v36; // [esp+FEh] [ebp-66h]
  char v37; // [esp+FFh] [ebp-65h]
  char v38; // [esp+100h] [ebp-64h]
  char v39; // [esp+101h] [ebp-63h]
  char v40; // [esp+102h] [ebp-62h]
  char v41; // [esp+103h] [ebp-61h]
  char v42; // [esp+104h] [ebp-60h]
  char v43; // [esp+105h] [ebp-5Fh]
  char v44; // [esp+106h] [ebp-5Eh]
```

```
char v45; // [esp+107h] [ebp-5Dh]
char v46; // [esp+108h] [ebp-5Ch]
char v47; // [esp+109h] [ebp-5Bh]
char v48; // [esp+10Ah] [ebp-5Ah]
char v49; // [esp+10Bh] [ebp-59h]
char v50; // [esp+10Ch] [ebp-58h]
char v51; // [esp+10Dh] [ebp-57h]
char v52; // [esp+10Eh] [ebp-56h]
char v53; // [esp+10Fh] [ebp-55h]
char v54; // [esp+110h] [ebp-54h]
char v55; // [esp+111h] [ebp-53h]
char v56; // [esp+112h] [ebp-52h]
char v57; // [esp+113h] [ebp-51h]
char v58; // [esp+114h] [ebp-50h]
char v59; // [esp+120h] [ebp-44h]
char v60; // [esp+121h] [ebp-43h]
char v61; // [esp+122h] [ebp-42h]
char v62; // [esp+123h] [ebp-41h]
char v63; // [esp+124h] [ebp-40h]
char v64; // [esp+125h] [ebp-3Fh]
char v65; // [esp+126h] [ebp-3Eh]
char v66; // [esp+127h] [ebp-3Dh]
char v67; // [esp+128h] [ebp-3Ch]
char v68; // [esp+129h] [ebp-3Bh]
char v69; // [esp+12Ah] [ebp-3Ah]
char v70; // [esp+12Bh] [ebp-39h]
char v71; // [esp+12Ch] [ebp-38h]
char v72; // [esp+12Dh] [ebp-37h]
char v73; // [esp+12Eh] [ebp-36h]
char v74; // [esp+12Fh] [ebp-35h]
char v75; // [esp+130h] [ebp-34h]
char v76; // [esp+131h] [ebp-33h]
char v77; // [esp+132h] [ebp-32h]
char v78; // [esp+133h] [ebp-31h]
char v79; // [esp+134h] [ebp-30h]
char v80; // [esp+135h] [ebp-2Fh]
char v81; // [esp+136h] [ebp-2Eh]
char v82; // [esp+137h] [ebp-2Dh]
char v83; // [esp+138h] [ebp-2Ch]
char v84; // [esp+139h] [ebp-2Bh]
char v85; // [esp+13Ah] [ebp-2Ah]
char v86; // [esp+13Bh] [ebp-29h]
char v87; // [esp+13Ch] [ebp-28h]
char v88; // [esp+13Dh] [ebp-27h]
char v89; // [esp+13Eh] [ebp-26h]
char v90; // [esp+13Fh] [ebp-25h]
char v91; // [esp+140h] [ebp-24h]
char v92; // [esp+141h] [ebp-23h]
char v93; // [esp+142h] [ebp-22h]
char v94; // [esp+143h] [ebp-21h]
char v95; // [esp+144h] [ebp-20h]
char v96; // [esp+145h] [ebp-1Fh]
char v97; // [esp+146h] [ebp-1Eh]
char v98; // [esp+147h] [ebp-1Dh]
char v99; // [esp+148h] [ebp-1Ch]
char v100; // [esp+149h] [ebp-1Bh]
char v101; // [esp+14Ah] [ebp-1Ah]
char v102; // [esp+14Bh] [ebp-19h]
char v103; // [esp+14Ch] [ebp-18h]
```

```
char v104; // [esp+14Dh] [ebp-17h]
char v105; // [esp+14Eh] [ebp-16h]
char v106; // [esp+14Fh] [ebp-15h]
char v107; // [esp+150h] [ebp-14h]
char v108; // [esp+151h] [ebp-13h]
char v109; // [esp+152h] [ebp-12h]
char v110; // [esp+153h] [ebp-11h]
char v111; // [esp+154h] [ebp-10h]
char v112; // [esp+155h] [ebp-Fh]
char v113; // [esp+156h] [ebp-Eh]
char v114; // [esp+157h] [ebp-Dh]
char v115; // [esp+158h] [ebp-Ch]

sub_13FA7BE("done!!! the flag is ");
v59 = 18;
v60 = 64;
v61 = 98;
v62 = 5;
v63 = 2;
v64 = 4;
v65 = 6;
v66 = 3;
v67 = 6;
v68 = 48;
v69 = 49;
v70 = 65;
v71 = 32;
v72 = 12;
v73 = 48;
v74 = 65;
v75 = 31;
v76 = 78;
v77 = 62;
v78 = 32;
v79 = 49;
v80 = 32;
v81 = 1;
v82 = 57;
v83 = 96;
v84 = 3;
v85 = 21;
v86 = 9;
v87 = 4;
v88 = 62;
v89 = 3;
v90 = 5;
v91 = 4;
v92 = 1;
v93 = 2;
v94 = 3;
v95 = 44;
v96 = 65;
v97 = 78;
v98 = 32;
v99 = 16;
v100 = 97;
v101 = 54;
v102 = 16;
v103 = 44;
v104 = 52;
```

```
v105 = 32;
v106 = 64;
v107 = 89;
v108 = 45;
v109 = 32;
v110 = 65;
v111 = 15;
v112 = 34;
v113 = 18;
v114 = 16;
v115 = 0;
v2 = 123;
v3 = 32;
v4 = 18;
v5 = 98;
v6 = 119;
v7 = 108;
v8 = 65;
v9 = 41;
v10 = 124;
v11 = 80;
v12 = 125;
v13 = 38;
v14 = 124;
v15 = 111;
v16 = 74;
v17 = 49;
v18 = 83;
v19 = 108;
v20 = 94;
v21 = 108;
v22 = 84;
v23 = 6;
v24 = 96;
v25 = 83;
v26 = 44;
v27 = 121;
v28 = 104;
v29 = 110;
v30 = 32;
v31 = 95;
v32 = 117;
v33 = 101;
v34 = 99;
v35 = 123;
v36 = 127;
v37 = 119;
v38 = 96;
v39 = 48;
v40 = 107;
v41 = 71;
v42 = 92;
v43 = 29;
v44 = 81;
v45 = 107;
v46 = 90;
v47 = 85;
v48 = 64;
v49 = 12;
```

```
  v50 = 43;
  v51 = 76;
  v52 = 86;
  v53 = 13;
  v54 = 114;
  v55 = 1;
  v56 = 117;
  v57 = 126;
  v58 = 0;
  for ( i = 0; i < 56; ++i )
  {
    *(&v2 + i) ^= *(&v59 + i);
    *(&v2 + i) ^= 0x13u;
  }
  return sub_13FA7BE("%s\n");
}
```

很明显，这是关键的输出flag的函数。

分析代码，发现是创建了两个长度为57的字符串，并进行运算。

编写python脚本

```
a=[18,64,98,5,2,4,6,3,6,48,49,65,32,12,48,65,31,78,62,32,49,32,
   1,57,96,3,21,9,4,62,3,5,4,1,2,3,44,65,78,32,16,97,54,16,44,
   52,32,64,89,45,32,65,15,34,18,16,0]
b=[123,32,18,98,119,108,65,41,124,80,125,38,124,111,74,49,
   83,108,94,108,84,6,96,83,44,121,104,110,32,95,117,101,99,
   123,127,119,96,48,107,71,92,29,81,107,90,85,64,12,43,76,86,
   13,114,1,117,126,0]
i=0
c=''
while (i<56):
    a[i]^=b[i]
    a[i]^=19
    c=c+chr(a[i])
    i=i+1
print (c)
```

运行即可得到flag

zsctf{T9is_tOpic_1s_v5ry_int7resting_b6t_others_are_n0t}