

# 【reversing.kr逆向之旅】Direct3D FPS的writeup

原创

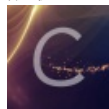
iqiqiya 于 2018-11-15 18:35:02 发布 446 收藏

分类专栏: [我的逆向之路](#) [我的CTF之路](#) [-----reversing.kr](#) 文章标签: [【reversing.kr逆向之旅】Direct3D FPS](#) [【reversing.kr逆向之旅】Direct3D FPS writeup](#) [逆向](#)

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/xiangshangbashaonian/article/details/84108105>

版权



[我的逆向之路](#) 同时被 3 个专栏收录

108 篇文章 10 订阅

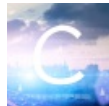
订阅专栏



[我的CTF之路](#)

92 篇文章 5 订阅

订阅专栏



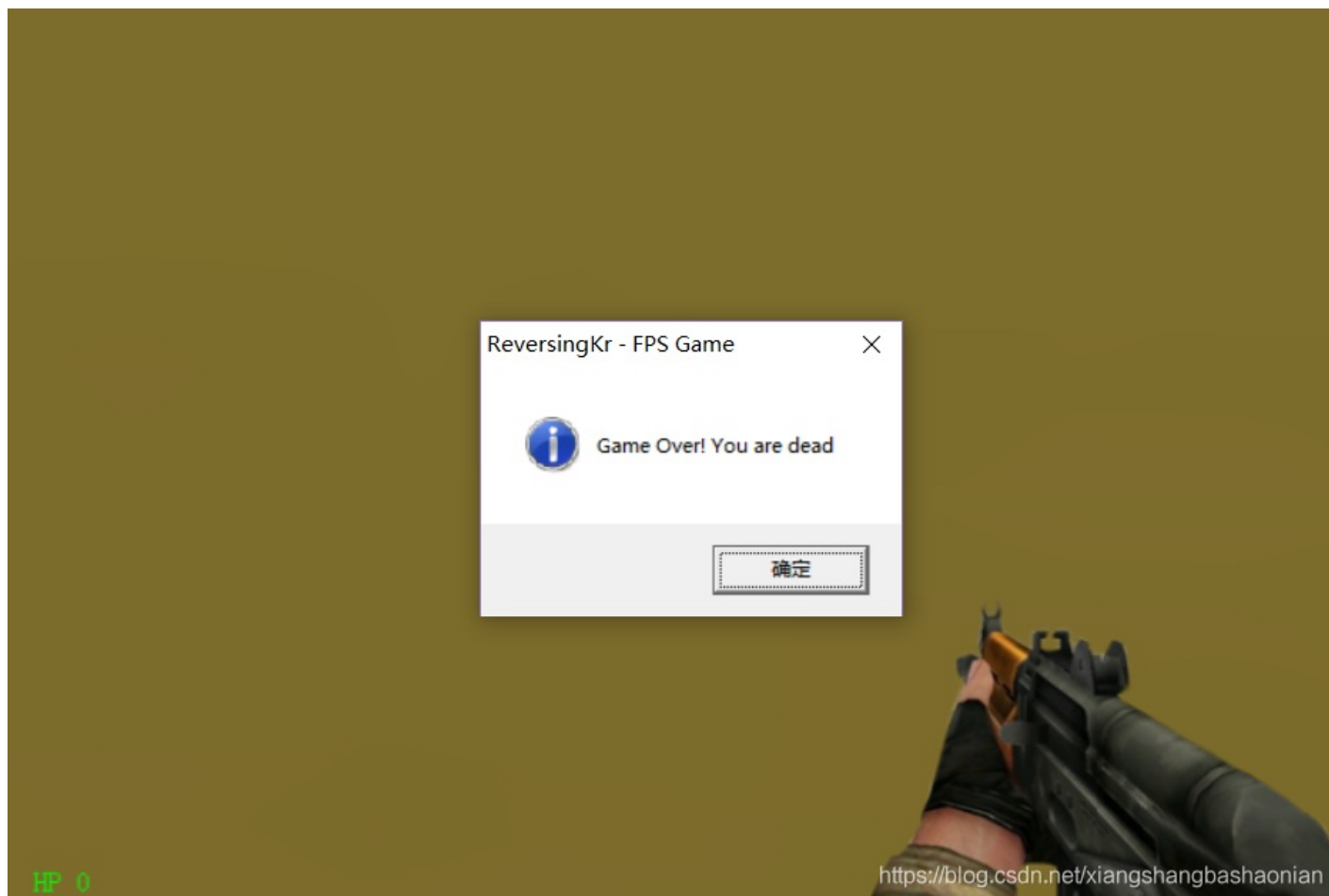
[-----reversing.kr](#)

11 篇文章 0 订阅

订阅专栏

Direct3D FPS 看到这个名字就感觉会不会是个射击游戏 然后运行果然是

玩了一会儿 看到有个几个小胖人 打不死 走过去就Game Over了



PEiD看到使用C++写的

直接载入IDA

分析字符串

```
.rdata:00000036 C [Error] LoadFile - 'Skybox.x' 颇老闹 佬瘤 给沁嚼聪促.  
.rdata:00000018 C Game Over! You are dead  
.rdata:0000000F C data\\Shoot.wav  
.rdata:00000006 C HP %d  
.rdata:00000015 C ReversingKr / FPS %d  
.rdata:0000000C C Game Clear!  
.rdata:0000000D C KERNEL32.dll  
.rdata:0000000B C USER32.dll  
.rdata:00000009 C GDI32.dll
```

很明显与Game Clear! 就是与失败相对应的成功

双击

```
.rdata:00405598 ; DATA XREF: WinMain(x,x,x,x)+80C↑  
.rdata:004055AD align 10h  
.rdata:004055B0 ; CHAR aGameClear[]  
.rdata:004055B0 aGameClear db 'Game Clear!',0 ; DATA XREF: sub_4039C0+10↑  
.rdata:004055BC align 10h
```

查看交叉引用

```
mov     eax, hWnd  
push   40h ; uType  
push   offset aGameClear ; "Game Clear!"  
push   offset byte_407028 ; lpText  
push   eax ; hWnd  
call   ds:MessageBoxA  
mov     ecx, hWnd  
push   0 ; lParam  
push   0 ; wParam  
push   2 ; Msg  
push   ecx ; hWnd  
call   ds:SendMessageA
```

Direct	Ty	Address	Text
Up	w	sub_403400+2D	xor byte_407028[eax], cl
o		sub_4039C0+22	push offset byte_407028, lpText

然后F5得到

```

int sub_403400()
{
    int i; // eax
    int v2; // ecx
    int v3; // edx

    i = sub_403440();
    if ( i != -1 )
    {
        v2 = 0x84 * i;
        v3 = dword_409190[0x84 * i];
        if ( v3 > 0 )
        {
            dword_409190[v2] = v3 - 2;
        }
        else
        {
            dword_409194[v2] = 0;
            flag[i] ^= byte_409184[v2 * 4];
        }
    }
    return i;
}

```

这里再对`flag[i] ^= byte_409184[v2 * 4];`这句进行动态分析 可知就是`i*4`

写脚本

```

s = [0x43, 0x6B, 0x66, 0x6B, 0x62, 0x75, 0x6C, 0x69, 0x4C, 0x45,
    0x5C, 0x45, 0x5F, 0x5A, 0x46, 0x1C, 0x07, 0x25, 0x25, 0x29,
    0x70, 0x17, 0x34, 0x39, 0x01, 0x16, 0x49, 0x4C, 0x20, 0x15,
    0x0B, 0x0F, 0xF7, 0xEB, 0xFA, 0xE8, 0xB0, 0xFD, 0xEB, 0xBC,
    0xF4, 0xCC, 0xDA, 0x9F, 0xF5, 0xF0, 0xE8, 0xCE, 0xF0, 0xA9]
flag = ''
for i in range(len(s)):
    flag+=chr(s[i] ^ (i*4))
print flag
#Congratulation~ Game Clear! Password is Thr3EDPr0m

```