




判断自己的父进程是不是由explorer启动

原创

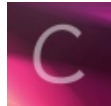
冷风  于 2014-01-20 15:22:16 发布  4889  收藏 1

分类专栏: [木马编写](#)

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/chinafe/article/details/18555455>

版权



[木马编写](#) 专栏收录该内容

86 篇文章 2 订阅

订阅专栏

直接把代码放上。

```
#include "stdafx.h"
#include <windows.h>
#include <tlhelp32.h>
#include <tchar.h>

DWORD get_parent_processid(DWORD pid)//获取指定进程的父进程ID
{
    DWORD ParentProcessID=-1;
    PROCESSENTRY32 pe;
    HANDLE hkz;

    HMODULE hModule = LoadLibrary(_T("Kernel32.dll"));
    if (hModule == NULL)
    {
        OutputDebugString(_T("Load dll error"));
        return -1;
    }

    FARPROC Address = GetProcAddress(hModule, "CreateToolhelp32Snapshot");

    if (Address == NULL)
    {
        OutputDebugString(_T("Get Proc error"));
        return -1;
    }
    _asm
    {
        push 0
        push 2
        call Address
        mov hkz,eax
    }

    pe.dwSize=sizeof(PROCESSENTRY32);
    if (Process32First(hkz,&pe))
    {
        do
        {
```

```

    if (pe.th32ProcessID==pid)//进程ID找到
    {
        ParentProcessID=pe.th32ParentProcessID;
        break;
    }
}
while(Process32Next(hkz,&pe));
}

return ParentProcessID;
}

DWORD get_explorer_processid()
{
    DWORD explorer_id=-1;
    PROCESSENTRY32 pe;
    HANDLE hkz;

    HMODULE hModule = LoadLibrary(_T("Kernel32.dll"));
    if (hModule == NULL)
    {
        OutputDebugString(_T("Load dll error"));
        return -1;
    }

    FARPROC Address = GetProcAddress(hModule, "CreateToolhelp32Snapshot");

    if (Address == NULL)
    {
        OutputDebugString(_T("Get Proc error"));
        return -1;
    }
    _asm
    {
        push 0
        push 2
        call Address
        mov hkz,eax
    }

    pe.dwSize=sizeof(PROCESSENTRY32);
    if (Process32First(hkz,&pe))
    {
        do
        {
            if (_stricmp(pe.szExeFile,"explorer.exe")==0)
            {
                explorer_id=pe.th32ProcessID;
                break;
            }
        }
        while(Process32Next(hkz,&pe));
    }

    return explorer_id;
}

int main(int argc, char* argv[])
{

```

```
DWORD explorer_id=get_explorer_processid();
DWORD parent_id=get_parent_processid(GetCurrentProcessId());

if (explorer_id==parent_id)
{
printf("the parent process is explorer\n");
}else{
printf("the parent process not explorer\n");
}
getchar(0);

return 0;
}
```