PowerShell安全专题之攻击工具篇

嘶吼roartalk / 2017-02-07 03:44:00 / 浏览数 4471 安全技术 技术讨论 顶(0) 踩(0)

PowerShell 成为攻击工具的演变过程

PowerShell 是一个内置在每个受支持的Windows版本中 (Windows 7 / Windows 2008 R2
和更高版本)的命令行程序，它能够提供令人难以置信的灵活性和功能化管理 Windows 系统的能力。这种能力使得 PowerShell
正在吸引攻击者使它逐渐成为一个非常流行且得力的攻击工具。一旦攻击者可以在一台计算机上运行代码，他们就会执行 PowerShell
代码，因为PS代码可以运行在防病毒软件不能看到的地方如内存中。攻击者还可能会下载 PowerShell 脚本文件 (.ps1) 到磁盘中，因为 PowerShell
能够提供从网站下载代码并在内存中运行的能力，不过这常常是不必要。

Dave Kennedy & Josh Kelley 在 DEF CON 18（2010 年） 演讲了攻击者如何利用 PowerShell进行攻击的主题。Matt Graeber 开发了 PowerSploit 并在 Monday.com 发表了博文—— 为什么 PowerShell 是一个强大的攻击平台。"PowerSploit"在 2012 年被发布后，PowerShell
被用来作为攻击平台的趋势快速上升，直到大约一年后 Mimikatz 开始支持 PowerShell 调用 (aka Invoke-Mimikatz) ,这使得 PowerShell
作为攻击工具的使用变得更为流行。PowerShell 提供了强大的能力，因为它可以运行从其他主机（或互联网）上下载的 .Net
代码并动态执行，甚至无需写到磁盘中执行，也能够在内存中执行。这些特点使得 PowerShell 在获得和保持对系统的访问权限时，
就成为了攻击者首选的攻击手段，利用PS的诸多特点攻击者可以持续攻击而不会被轻易发现。PowerShell v5 极大的提高了 PowerShell 的安全性并且在
Windows 10系统上运行 PowerShell 时，PowerShell 的攻击能力将会大大降低。

攻击者的多种选择

这篇文章显然会提到攻击者可以如何颠覆 PowerShell 最新的安全增强特性，包括 PowerShell v5。请记住，攻击者还有更多选择。PowerShell
只是选择之一，exe则是另一种方式。其他的选择包括：

```
■■■■■■■■■ (Exe■Dll ■)
Windows ■■■■■■■CMD■
■■■■■
Sysinternal ■■■■
Windows ■■■■■
VBScript
CScript
JavaScript
■■■■■■
PowerShell
```

PowerShell 的攻击能力

许多的攻击者喜欢使用 PowerShell 的原因如下：

```
■■■■■■■■■■■■■■■■■■■■■■■■■
■■■■■■■■■■■■■■■■■■■
■■■■■■■.Net ■ Windows API■
■■■■■■■■■■■■
CMD.exe ■■■■■■■■■■ PowerShell ■■■■
■■■■■■■■■■■■■■ PowerShell ■■■■■■■
■■■■■■■■■■■■■■■■ PowerShell ■■■■■■■
```

PowerShell 在客户端攻击中常常被利用并且会频繁地调用下列运行选项（通常是编码过的命令（bypasses exec. policy）。

以下是几个典型的 PowerShell 运行选项：

```
WindowsStyle Hidden
NoProfile
ExecutionPolicy Bypass
File
Command
EncodedCommand
```

现实世界中的 PowerShell 攻击工具

PowerSploit

描述： 这是众多 PowerShell 攻击工具中被广泛使用的 PowerShell 后期漏洞利用框架。 使用场景： 信息探测，特权提升，凭证窃取，持久化。 作者： Matt Graeber (@Mattifestation) & Chris Campbell (@obscuresec)

常用的cmdlets：

```
Invoke-DllInjection.ps1
Invoke-Shellcode.ps1
Invoke-WmiCommand.ps1
Get-GPPPassword.ps1
Get-Keystrokes.ps1
Get-TimedScreenshot.ps1
Get-VaultCredential.ps1
Invoke-CredentialInjection.ps1
Invoke-Mimikatz.ps1
Invoke-NinjaCopy.ps1
Invoke-TokenManipulation.ps1
Out-Minidump.ps1
VolumeShadowCopyTools.ps1
Invoke-ReflectivePEInjection.ps1
```

Invoke-Mimikatz

功能特性： 能够在 PowerShell 中执行 Mimikatz，凭证偷窃 & 注入，伪造 Kerberos 票证创建，还有很多很多功能。 使用场景： 凭证窃取 & 重用，持久化 作者： Joseph Bialek (@clymb3r)

PowerView

描述： 一款纯粹的 PowerShell 域/网络态势感知工具。现在是 PowerSploit 的一部分。 使用： 信息探测 作者： Will Harmjoy (@HarmJ0y)

常用的cmslets：

```
Get-NetUser
Get-NetGroup
Get-NetGroupMember
Get-NetLocalGroup
Get-NetSession
Invoke-UserHunter
Get-NetOU
Find-GPOLocation
Get-NetGPOGroup
Get-ObjectACL
Add-ObjectACL
Invoke-ACLScanner
Set-ADObject
Invoke-DowngradeAccount
Get-NetForest
Get-NetForestTrust
Get-NetForestDomain
Get-NetDomainTrust
Get-MapDomainTrust
```

PowerUp

描述： 本地特权提升的一些调用方法，也是 PowerShell Empire 的一部分。 使用： 特权提升 作者： Will Harmjoy (@harmj0y)

```
Get-ServiceUnquoted
Get-ServiceFilePermission
Get-ServicePermission
Invoke-ServiceAbuse
Install-ServiceBinary
Get-RegAutoLogon
Get-VulnAutoRun
Get-VulnSchTask
Get-UnattendedInstallFile
Get-WebConfig
Get-ApplicationHost
Get-RegAlwaysInstallElevated
```

Nishang

描述： PowerShell 渗透测试。 使用场景： 信息探测，凭据窃取，特权提升，持久化。 作者: Nikhil Mitt (@nikhil_mitt)

```
Get-Unconstrained
Add-RegBackdoor
Add-ScrnSaveBackdoor
Gupt-Backdoor
```

```
Invoke-ADSBackdoor
Enabled-DuplicateToken
Invoke-PsUaCme
Remove-Update
Check-VM
Copy-VSS
Get-Information
Get-LSASecret
Get-PassHashes
Invoke-Mimikatz
Show-TargetScreen
Port-Scan
Invoke-PoshRatHttp
Invoke-PowerShellTCP
Invoke-PowerShellWMI
Add-Exfiltration
Add-Persistence
Do-Exfiltration
Start-CaptureServer
```

PowerShell Empire

功能特性：

```
■■ PowerShell ■■■■■■■■
Python ■■■■■■■■■(Kali Linux)
AES ■■■ C2 ■■■■■
■■■■■■■■■■■■■■■■■■■
```

使用场景： 提供前期漏洞利用的集成模块、 信息探测、 凭据窃取 & 重用，以及持久化。 作者: Will Schroeder (@harmj0y) & Justin Warner (@sixdub) & Matt Nelson (@enigma0x3)

模块如下:

```
Code Execution
Collection
Credentials
Exfiltration
Exploitation
Lateral Movement
Management
Persistence
Privilege Escalation
Recon
Situational Awareness
Fun & Trollsploit
```

Cmdlets:

```
Invoke-DllInjection
Invoke-ReflectivePEInjection
Invoke-ShellCode
Get-ChromeDump
Get-ClipboardContents
Get-FoxDump
Get-IndexedItem
Get-Keystrokes
Get-Screenshot
Invoke-Inveigh
Invoke-NetRipper
Invoke-NinjaCopy
Out-Minidump
Invoke-EgressCheck
Invoke-PostExfil
Invoke-PSInject
Invoke-RunAs
MailRaider
New-HoneyHash
Set-MacAttribute
Get-VaultCredential
```

```
Invoke-DCSync
Invoke-Mimikatz
Invoke-PowerDump
Invoke-TokenManipulation
Exploit-Jboss
Invoke-ThunderStruck
Invoke-VoiceTroll
Set-Wallpaper
Invoke-InveighRelay
Invoke-PsExec
Invoke-SSHCommand
Get-SecurityPackages
Install-SSP
Invoke-BackdoorLNK
PowerBreach
Get-GPPPassword
Get-SiteListPassword
Get-System
Invoke-BypassUAC
Invoke-Tater
Invoke-WScriptBypassUAC
PowerUp
PowerView
Get-RickAstley
Find-Fruit
HTTP-Login
Find-TrustedDocuments
Get-ComputerDetails
Get-SystemDNSServer
Invoke-Paranoia
Invoke-WinEnum
Get-SPN
Invoke-ARPScan
Invoke-PortScan
Invoke-ReverseDNSLookup
Invoke-SMBScanner
```

PowerShell 攻击工具的使用

最好的 PowerShell 攻击工具无疑是——Empire，下载 PowerShell Empire zip 文件并解压。 解压之后，在 datamodule_source 中查看 PS1 文件。

PowerShell 并不只是 “PowerShell.exe”

阻止对 PowerShell.exe 的访问是限制 PowerShell 执行的一个比较"简单"的方法，至少，它看上去很简单。但现实情况是 PowerShell 并不仅仅是一个可执行文件这么简单。PowerShell 是 Windows 系统的一个核心组件（且不可移除），它存在于 System.Management.Automation.dll 动态链接库文件 (DLL) 中并且可以附加到不同的运行空间中而且可以有效的进行 PowerShell 实例化（想想 PowerShell.exe 和 PowerShell_ISE.exe）。可以通过代码在自定义的 PowerShell 运行空间实例化，所以 PowerShell 可以通过一个自定义的可执行文件进行启动（例如 MyPowershell.exe）。实际上，已经有一些无需 Powershell.exe 就能运行 PowerShell 代码的几种方法了。Justin Warner (@SixDub) 在 2014 年底发表了一篇博文 如何启动受限的Powershell.exe ，也叫做 PowerPick）。
由于 PowerShell 代码可在不运行 PowerShell.exe 的情况下执行代码，因此限制 Powershell.exe 可执行文件的运行并不是理想的解决方案，也不能很好的阻止攻击。

这里有两个辩论的对立面。

一面是"要限制 PowerShell"。这种做法会有一个积极的结果，由于攻击者在前期无法执行 PS 代码所以一些潜在的攻击就能被阻断。但是由于微软或第三方软件在一些功能上可能会依赖 PowerShell 的支持，因此这个做法有一定的副作用，因为它限制了 windows management 的功能。
另一面是 "不能限制 PowerShell"，因为有其他方法可以限制攻击者执行 PowerShell 代码而无需限制 PowerShell.exe 的运行。可通过 AppLocker 配置 PowerShell 的保护限制。另外 ，将 Powershell 设置为受限的语言模式的这个方法还有待测试。如果要想了解更多此类信息，可以查看后面的章节"限制 PowerShell 的能力"。

```
PowerShell = System.Management.Automation.dll
■■■■■■■■ PowerShell ■■
“PowerShell ps = PowerShell.Create()”
Ben Ten’s AwesomerShell
```

无需 PowerShell.exe 执行 PS 命令

PowerShell v2:
"提供了用于创建管道命令以及同步或异步调用运行空间内的命令的方法。此类还提供了调用命令时生成的且包含数据的输出流的访问。此类主要用于宿主应用程序以编程方式 Windows PowerShell 执行任务。在 Windows PowerShell 2.0 中，介绍了此类。

■■■■■■ Powershell System.Automation.dll ■■■■■ C# ■■■■■■
■■ Automation ■■■■■■■■■ PowerShell ■■■■
■■■■ PowerShell.exe ■■■■■■■

由 Lee Christensen 提出的非托管的 PowerShell 是大多数 PowerShell 攻击工具在脱离 powershell.exe 就能执行 PowerShell 代码的基础。这种方式在非托管进程的 PowerShell 中启动 .NET 并且在内存中加载并执行自定义的 C# 程序集。 自 2016 年 3 月起，Metasploit PowerShell 模块就是利用了非托管的 PowerShell。

另一种利用非托管的 PowerShell 的 PS 项目是 P0wnedShell ，这是一款 "PowerShell 运行空间后期漏洞利用工具包"。它在 powershell 运行空间环境(.NET)内运行 PowerShell 命令和函数，并且包含了许多 PowerShell 攻击工具，其中包括 PowerSploit，PowerCat，Inveigh，Nishang等等，这些攻击工具都包含在一个单一的可执行文件中。

这个项目还提供了一个简单的 PowerShell 攻击工具执行示例——“数字排序”用于简单的执行PS攻击工具。我将它重命名为"Calc.exe"，我从来没有见到运行 Calc 就使用了超过几 MB 的 RAM，当我使用 Mimikatz 执行它时，"Calc" 竟然使用了大于 180 MB的内存。

■■■■■■■adsecurity■■■■■■■■■■■■■■■■ http://www.4hou.com/technology/3134.html

上一篇：PHP各版本的姿势 下一篇：Chrome中”自动填充”安全性研究

1. 0 条回复
   - 动动手指，沙发就是你的了！