登录

PowerShell安全专题之 PS5 安全增强功能

嘶吼roartalk / 2017-02-13 03:41:00 / 浏览数 3252 安全技术 技术讨论 顶(0) 踩(0)

PowerShell v5 是一个 RTM版本。（截至 2015/12/18)。在此之前的八月起，有一个"生产预览版"可用，这意味着它不是最后的版本。随着 PowerShell v5 最终版的正式发布，我强烈建议你下载 PowerShell v5 并开始测试准备生产部署。

PowerShell 为系统管理员提供了广泛的管理能力，但是反过来看，这种能力也可以被攻击者利用，进行企业内网渗透和持久化控制。
Microsoft 在下载网站上的提供了以下几点关于 PowerShell v5 的优势和更新说明：

Windows 管理框架 (WMF) 5.0 从 WMF 4.0 带来了已更新的功能。WMF 5.0 是仅可用于安装在 Windows Server 2012 R2、 Windows Server 2012，Windows 2008 R2，Windows 8.1 和 Windows 7 SP1。其他一些在此版本中新增和更新的功能说明包括如下：

1.■■ Windows PowerShell ■■
2.■■■■■■■■ (JEA)
3.■■■■■■■■■■■■■■■■■■■■
4.■ Windows PowerShell ■■■■■■■■■■■■■
5.PowerShell ■■■■ 6.■■■■■■■■■■■■■■■■■■■■■■ cmdlet
7.ODataUtils ■■■■■ OData ■■■■ Windows PowerShell cmdlet
8.■■■■■ cmdlet ■■ ZIP ■■
9.■■■■■■cmdlet ■■■■■■■■■■
10.■ Windows PowerShell ISE ■■■■■ DSC ■■
11.DSC ■■■■■■■■ 32 ■
12.■■■■■■■■■■■■■■■■ Windows PowerShell ■■
13.■■■■■■■■■■■■ DSC ■■■■■■■■■
14.■ DSC ■■■■■■■■■■■■■
15.■■ DSC ■■■■■■■■■■
16.■ DSC ■■■■■■■■■■■■■
17.■ DSC ■■■■■■■■■■■■■■■■■■■■
18.■ DSC ■■■■■■■■■■ -■
19.■■ DSC RunAsCredential
20.■■■DSC LCM ■■■■■
21.DSC ■■■■ PowerShell ■■■■■■■■
22.PSDesiredStateConfiguration ■■■■■■■■ 1.1
23.DSC ■■■■■■■■■■■■■■■■■■ 24.■■■■■ PackageManagement ■■■■■■■■
25.■■■■■ PowerShellGet ■■ PowerShell ■■■■ PowerShell ■■■■ DSC ■■
26.■■■■■ Windows PowerShell■■■■■■■■
27.■■■■■■■ (SIL)

在 PowerShell v5 中有几个令人信服的安全功能，使得它很有必要去部署（恕我直言）。我曾在2015年的几个安全会议上提到过这些安全功能。

这些安全功能包括：

■■■■■■■■
■■■■■■■■■■
■■■■■
■■■■■■■■■■ AMSI■Windows 10■

脚本块日志记录

脚本块日志提供了在事件日志中记录反混淆的 PowerShell 代码的能力。大多数的攻击工具都会进行混淆处理，通常会使用 Base64 编码，在执行代码之前很难发现或确认这些代码实际上会做些什么事情。由于脚本块日志会在实际的代码传递到 PowerShell 引擎之前进行记录，这就使得在代码执行之前就能进行日志记录，因为脚本代码在执行之前需要进行反混淆处理。

由于许多 PowerShell 攻击攻击都对攻击代码进行了混淆处理，所以很难识别脚本代码的具体功能。脚本块日志会对要执行的代码进行反混淆和记录。由于代码已经被反混淆且进行了记录，所以当

识别带有攻击性的 PowerShell 代码的一个关键挑战是大多数情况下代码都是混淆过的（Base64，Base64 + XOR 等）。这使得几乎不可能实现实时分析，因为没有触发警报的关键字消息。

更深度的脚本块记录可疑记录它处理过的脚本文件内容也就是在执行时所生成的脚本的文件内容。

Microsoft 提供了一个经过混淆处理的命令代码示例：

```
## Malware
function SuperDecrypt
{
param($script)
$bytes = [Convert]::FromBase64String($script)
```

```
## XOR “encryption”
$xorKey = 0x42
for($counter = 0; $counter -lt   $bytes.Length; $counter++)
{
$bytes[$counter] = $bytes[$counter] -bxor $xorKey
}
[System.Text.Encoding]::Unicode.GetString($bytes)
}
$decrypted = SuperDecrypt “FUIwQitCNkInQm9CCkItQjFCNkJiQmVCEkI1QixCJkJlQg==”
Invoke-Expression $decrypted
```

传递给 PowerShell 处理的原始的脚本块内容（如上文所述）就是 PowerShell 执行的实际命令。
请注意，脚本块记录默认是启用的。

完整的脚本副本

完整的脚本副本功能可以通过组策略启用并为系统上每个用户在该系统上执行的每个 PowerShell
命令和代码块提供一个"更加丰富"的副本文件。这份副本可以定向传输到”只写”的网络共享上为后续的分析和 SIEM 工具进行读取。

另外，PowerShell 具有将控制台输出的文本信息写入到一个副本文件中，这需要用户或脚本在运行时使用 “start-transcript
$FileName”执行。这就提供了一个简单的脚本日志文件。这种方法的缺点是在同一时间只能有一个副本记录活动。PowerShell ISE
编辑器不支持副本记录，Start-Transcript 必须被添加到每个用户的 PowerShell 配置文件中以便按顺序保存记录所运行的命令。

完整的脚本副本提供简单的方法把所有的 PowerShell 命令（包括那些内部运行的或其他位置的脚本）
都写入到一个特定于计算机上的且存储在网络共享中的记录文件中。这样就可以以接近实时分析的效果对 PowerShell
的活动进行快速分析并能确认已知的安全风险。

完整的脚本副本记录功能可以通过组策略启用，标头中会包含以下信息：

```
Start time
User Name
RunAs User
Machine (Operating System)
Host Application
Process ID
```

参数：

```
IncludeInvocationHeader —— ■■■■■■■■■■■■■■■■
OutputDirectory —— ■■■■■■■■■■■■■■■■■■■■■■■■
```

Microsoft 提供了一个 PowerShell 脚本配置 中央副本共享 ACL 的示例：

```
md c:Transcripts
## Kill all inherited permissions
$acl = Get-Acl c:Transcripts
$acl.SetAccessRuleProtection($true, $false)
## Grant Administrators full control
$administrators = [System.Security.Principal.NTAccount] “Administrators”
$permission = $administrators,”FullControl”,”ObjectInherit,ContainerInherit”,”None”,”Allow”
$accessRule = New-Object System.Security.AccessControl.FileSystemAccessRule $permission
$acl.AddAccessRule($accessRule)
## Grant everyone else Write and ReadAttributes. This prevents users from listing
## transcripts from other machines on the domain.
$everyone = [System.Security.Principal.NTAccount] “Everyone”
$permission = $everyone,”Write,ReadAttributes”,”ObjectInherit,ContainerInherit”,”None”,”Allow”
$accessRule = New-Object System.Security.AccessControl.FileSystemAccessRule $permission
$acl.AddAccessRule($accessRule)
## Deny “Creator Owner” everything. This prevents users from
## viewing the content of previously written files.
$creatorOwner = [System.Security.Principal.NTAccount] “Creator Owner”
$permission = $creatorOwner,”FullControl”,”ObjectInherit,ContainerInherit”,”InheritOnly”,”Deny”
$accessRule = New-Object System.Security.AccessControl.FileSystemAccessRule $permission
$acl.AddAccessRule($accessRule)
## Set the ACL
$acl | Set-Acl c:Transcripts
## Create the SMB Share, granting Everyone the right to read and write files. Specific
## actions will actually be enforced by the ACL on the file folder.
New-SmbShare -Name Transcripts -Path c:Transcripts -ChangeAccess Everyone
```

通过组策略启用完整的脚本副本记录功能的操作步骤：

```
Windows Components■Windows ■■■ -> Administrative Templates ■■■■■■ -> Windows PowerShell -> Turn on PowerShell Transcription
```

该组策略配置对应的注册表路径为：

```
HKLM:SoftwarePoliciesMicrosoftWindowsPowerShellTranscription
```

PowerShell 的约束模式

PowerShell 支持多种"语言模式"。其中有个比较有趣的语言模式——"受限的语言模式"，它会将 PowerShell 锁定为基本功能模式。

PowerShell v5 也同样支持自动锁定降级，这需要 AppLocker 部署在"允许"模式才行。Applocker
允许模式是真正的程序白名单，它可以有效防止未经授权的任何二进制文件执行。当 PowerShell v5 检测到 Applocker 在允许模式下时，PowerShell
会自动将其语言模式设置为约束模式，这就极大地限制了系统上的受攻击面。在Applocker 允许模式开启并且 PowerShell
是在约束模式下运行的时候，攻击者不可能将 PowerShell 的语言模式更改为完整的模式也无法运行任何 PowerShell 攻击工具。当 AppLocker
配置在"允许模式"时，PowerShell 会将自身功能降级到"约束模式"，只允许交互式输入以及用户编写的脚本的功能。约束模式下的 PowerShell 只允许核心的
PowerShell 功能目的是防止执行那些经常使用扩展语言特点的且带攻击性的 PowerShell 工具 (如：操作 .NET 的脚本，通过 Add-Type cmdlet 调用 Win32 API
以及与 COM 对象进行交互的脚本) 。

反恶意软件集成 (Windows 10)

新的 Windows 10 反恶意软件扫描接口（AMSI）要求所有的脚本引擎 （PowerShell，VBScript 和 JScript ）
对脚本文件，在命令行中键入的命令甚至是从互联网下载并在内存中执行的代码进行动态内容分析。这样就可以在计算机上执行 PowerShell
代码之前进行安全扫描。当代码被传递到 PowerShell"引擎"(System.Management.Automation.dll) 时，它会将代码发送到 AMSI
进行反恶意软件检查。系统上安装的反恶意软件解决方案需要支持 AMSI 以便于能进行代码扫描。Windows Defender 支持 Windows 10 AMSI。扫描后，如果
AMSI 返回了 OK，则代码会被执行。反之，则不会执行代码。

这意味着，只要反病毒/反恶意软件解决方案支持 AMSI，那么在 Windows 10 计算机上就可以阻止 PowerShell 攻击代码的执行。

反恶意软件扫描接口 (AMSI)
是一种允许应用程序和服务集成在一台机器上的任何反恶意软件产品的泛型接口标准。它为用户和他们的数据、应用程序以及工作负载提供了增强的恶意软件防护。
AMSI
是反恶意软件供应商不可知论者，它是一个现代反恶意软件产品，并被设计针对最常见的恶意软件的扫描也可以集成其他应用程序所提供的保护技术。它支持文件和内存以及
URL/IP 信誉检查和其他技术。 AMSI
还支持会议的概念，以便反恶意软件供应商可以将不同的扫描请求相关联。例如，通过将孤立的片段相关联后就可以针对不同的碎片化的恶意 payload
达到更明智的决定。

```
■■■■■■■■ ■■■■■ ■■■■■■■■■■■■■■■■■■http://www.4hou.com/technology/3144.html
```


上一篇：小议验证码 下一篇：我的WafBypass之道（Misc篇）

1. 0 条回复
   - 动动手指，沙发就是你的了！