

订阅DeepL Pro以编辑此演示文稿。  
访问[www.DeepL.com/pro](https://www.deepl.com/pro?cta=edit-document)，了解更多信息。

**探索Cmd.exe混淆和检测技术的深层含义**

熟练的攻击者不断寻找新的攻击载体，同时采用规避技术，在不断变化的防御环境中保持旧载体的有效性。众多的威胁者对常见的脚本语言（如JavaScript和PowerShell）采用混淆框架，以挫败基于签名的对用这些语言编写的常见攻击手段的检测。

然而，随着防御者通过更好的日志做法1和通过微软反恶意软件扫描接口2对这些语言的执行阶段进行内联检查，对这些流行的脚本语言的可视性增加，一些隐蔽的攻击者已经将他们的技术转移到不支持这种额外可视性的语言。至少，有决心的攻击者正在向以前检测到的有效载荷和命令添加简单的混淆符号，以打破僵硬的检测规则。

FireEye的高级实践团队致力于为攻击者在野外使用的高级TTP（工具、技术和程序）开发检测能力。作者作为该团队的高级应用安全研究员，需要研究现有的和新的混淆和规避领域，最终建立更强大的检测能力。列举新的问题空间使人们能够更有效地检测当今威胁者所使用的难以捉摸的伎俩。这种方法也推动了对尚未在野外发现的混淆技术的检测能力。

2017年6月，高级实践团队发现FIN7（一个以金融为目的的威胁行为者，也被称为Carbanak）正在测试cmd.exe中原有的一种新型混淆技术。在这一发现的推动下，作者开始研究cmd.exe支持的混淆技术，并在客户和客户环境以及公共和私人文件库中寻找其使用情况。这些发现代表了9个月的专门研究、检测开发和对全世界1000多万个端点的威胁搜索。

本研究的目的是列举cmd.exe支持的混淆技术的问题空间，以领先于FIN7或其他威胁行为者可能采用的下一个混淆技巧。作者正是以这种防御心态提出这些研究结果，以便其他防御者能够更有效地检测这些混淆和规避技术。

野外的混淆行为

FireEye追踪的众多威胁行为者越来越多地使用混淆技术来试图逃避严格的检测。2017年6月，作者3 与FireEye事件响应经理尼古拉斯-卡尔（Nicholas Carr）共同撰写了一篇博文，概述了他们的团队在野外发现的由三个独立的威胁行为者使用的三种独立的命令行混淆技术。

第一个例子来自于FIN8的一份钓鱼文件，FIN8是一个金融威胁行为者，具有明显的侵略性钓鱼活动。该文件包含一个混淆的宏，它使用进程级环境变量和PowerShell的标准输入命令功能来隐藏winword.exe的子进程cmd.exe及其孙进程powerhell.exe的所有有意义的命令行参数。

来自FIN8网络钓鱼文件的去模糊化的宏（2017年2月）



Mandiant事件响应者在应对2017年4月的APT32（又名OceanLotus）入侵时，在实时事件数据中捕获了第二个例子。这个基于越南的威胁行为者，其动机似乎与越南政府的利益一致，经常使用Invoke-Obfuscation4 PowerShell混淆框架来严重混淆Cobalt Strike Beacon后门下载器，但经常使用被称为 "Squiblydoo "的regsvr32.exe远程下载技术下载这个第二阶段。为了逃避这种技术的刚性签名，这些签名依赖于命令行参数值/i:http:// 或 /i:https:// 的存在，APT32首先使用cmd.exe的转义字符，即圆点（^），然后在这个后来的例子中使用双引号来分解这些参数。

来自APT32的混淆的regsvr32.exe命令（2017年4月）

****

FireEye Advanced Practices Team在一份归属于FIN7的钓鱼文件中发现了最后一个例子。该文件采用了遍布多个有效载荷的新型执行和混淆技术。该文件首先将一个LNK文件丢到磁盘上并执行它。LNK文件将一个混淆的JScript文件写入磁盘，地址是%HOMEPATH%\ md5.txt，并用wscript.exe执行。然后JScript文件通过Word.Application COM对象从原始文档中获取并执行最终的有效载荷。JScript文件包含一个连接（"Wor "+"d.Application"）和ASCII编码的组合，以混淆可疑的评估函数：（this[String.fromCharCode(101)+'va'+'l']）。然而，LNK文件包含以下截图中强调的更新颖的混淆技术。

来自恶意的FIN7 LNK文件的混淆的cmd.exe命令（2017年6月）

****

攻击者在将wscript.exe命令作为标准输入传递给最终的cmd.exe之前，将其设置在一个名为x的进程级环境变量中。攻击者还使用@字符混淆了原始cmd.exe命令中的字符串wscript和/e:jscript。随后，使用 cmd.exe 的本地变量字符串替换功能，将 @ 字符从存储在环境变量 x 中的命令内容中删除。这个字符串替换功能遵循%VariableName:StringToFind=NewString%的形式，其中StringToFind是@字符，NewString是空白，所以@字符被简单地删除。这种字符串替换技术允许LNK文件中被混淆的wscript.exe命令在通过标准输入传递给最终的cmd.exe执行之前，在内存中进行去混淆处理。

下面是这个样本的变量字符串替换技术的简化说明。

变量字符串替换的简化图示

****

这项技术有效地绕过了几个静态检测，并促使作者开始这项探索cmd.exe支持的混淆技术的研究行动。