红队操作中的搭档——Cobalt Strike 让你的渗透测试更快更持久（上）

原创 丝绸之路 嘶吼专业版 5天前

攻击者通常有一套实现后漏洞利用的工具和方法。Cobalt Strike 就是其中之一，它是一个非常强大的红队指挥与控制(C2)平台，它有许多内置的强大功能。关于 Cobalt Strike ，我最喜欢的一点是可以定制不同的方面来满足你的需求。攻击者脚本（Aggressor scripts）让操作人员有机会添加可能缺失的功能或者他们认为必要的改变(例如，生活质量的改善，OPSEC 的修改)。

在红队操作中，横向移动和持久性是操作人员执行的非常常见的操作。从历史上看，Cobalt Strike 内置的 Windows 横向移动技术有点僵硬; 标准选项包括 PsExec、 PsExec——PowerShell、 WinRM 和 WMI。Cobalt Strike仍然在多个领域中依赖于 PowerShell ，关于这方面的更多信息可以在Raphael Mudge的博客上找到。然而，在最近的4.0更新中，我们看到一些 PowerShell 依赖项被移除了。许多团队和工具已经远离 PowerShell，改为使用 C# 这样的语言来执行相同的活动。

为了摆脱对 PowerShell 的依赖，并仍然执行横向移动或持久化，用户需要以手动方式执行操作。例如，操作人员必须创建其有效载荷，找到将其传递到主机的方法（无论是文件还是

下载者），并远程执行有效载荷。幸运的是，通过一个小的攻击者脚本，可以几乎无缝地完成这项工作。再深入一点，我们可以使用 .Net 以一种几乎完全自动化的方式执行我们的任务。此外，同样的思想也可以应用于持久化操作。

自从Cobalt Strike 3.11版本发布以来，操作人员已经可以通过 Beacon 访问武器化的能力，在内存中执行任意的.Net 程序集。这种能力有助于加速攻击性 .Net 工具的开发，以及随后此类工具的公开发布。

考虑到这一点，我希望为最终用户提供更多的定制，以了解他们如何执行横向移动和持久性。其中许多想法和实现都是公开的，并且已经使用了多年。我对这些项目的目标是将现有的技术简单地结合起来，并提供一种自动执行它们的方法。

今天，我发布了两个项目：MoveKit 和 StayKit。这些项目是攻击者脚本（Aggressor scripts）的组合 .Net 项目，以及用于创建有效载荷的模板。这两个工具都使用现有 Windows 的组合.Net 程序集、 COM 库和 PInvoke 签名，以避免任何第三方库需求。MoveKit 和 StayKit 所需的程序集是 SharpMove、 SharpRDP、 FileWriter 和 SharpStay。这些程序集并不是随项目一起编译的，而是由用户构建它们并将它们放在适当的目录中。

重要的是要记住，尽管这些工具的攻击性方面在整个操作者的体验中起着很大的作用，但所有基础能力都是通过.Net 程序集交付的。因此，Cobalt Strike并不是一个使用任何.Net 项目的必需品。如果你不使用 Cobalt Strike，任何可以执行.Net 程序集的 C2代理都可以充分利用 SharpMove 或 SharpStay 程序集。

MoveKit

把这些想法结合在一起，我创建了一个名为 MoveKit 的工具集，创建这个工具集的想法来源于其他 Cobalt Strike 工具集，这个工具集允许进一步的可定制性(例如，Resource Kit，Artifact Kit，和 Elevate Kit)。这样做的目的是扩展 Cobalt Strike 的横向移动能力。最初这个项目只使用 WMI，因为它的核心是从 SharpWMI 开始的，但是随着时间的推移，它已经内置了其他技术，并进行了各种更改，为操作人员提供了更多的选择。

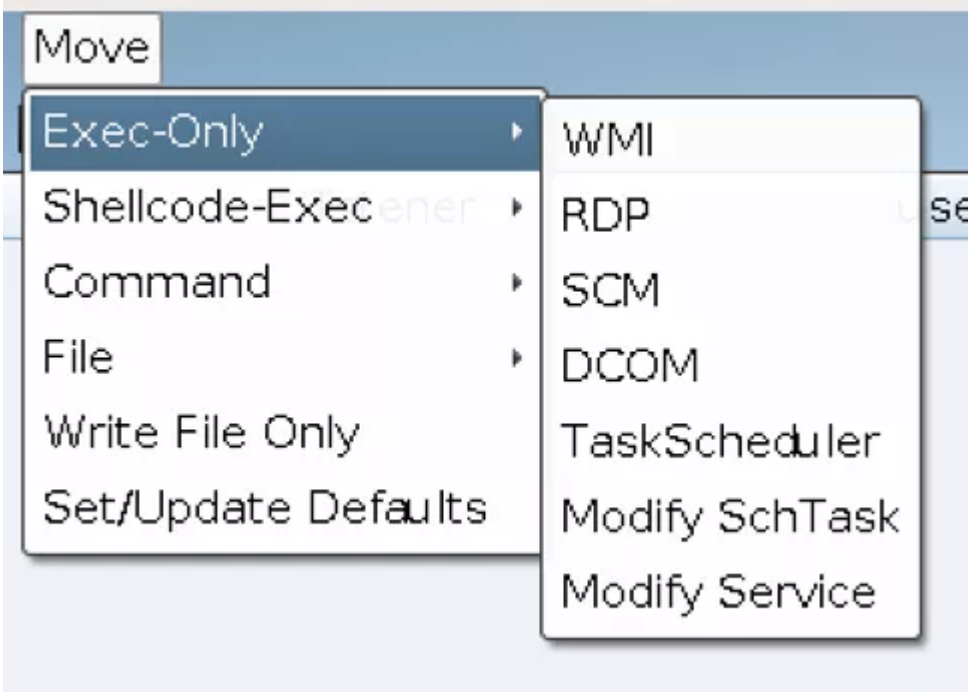

图1— MoveKit 的菜单

这套工具包含一些不同的组件，使一切成为可能。首先，它包含了一些.Net 程序集，包括远程执行和文件移动。第一个程序集 SharpMove 具有所有.Net 代码与远程系统交互以执行 SharpRDP (由于 RDP 文件大小不同，有两个独立的文件)。FileWrite程序集包含了处理向远程系统写入数据或文件的所有 .Net 代码。

其次，它包含了用于构建和执行有效载荷的不同模板源代码文件。最后，攻击者脚本处理所有的菜单，信标命令，有效载荷创建，文件移动等。乍一看，界面有很多选项，但是它们都提供了执行每个操作的不同方式。

攻击者脚本被分解成不同的部分，操作人员可以控制它们想要执行的操作(图1)。

• 第一个菜单，仅用于执行，这意味着没有发生文件移动操作(图1)

• 第二个菜单，用于 shellcode 的执行，它不需要将文件写入磁盘或执行下载者

• 第三个菜单，命令执行(图2)意味着命令在文件写入或加载到内存之前执行，就像下载者一样

• 最后的菜单选择只用于写入文件或数据，不做执行操作

当前在 SharpMove/SharpRDP 和 MoveKit 选项中实现的远程执行技术如下:

• Exec-Only 只执行

• WMI —— 包含了通过 Win32_Process WMI 类执行的能力。此外，还有通过 WMI 事件订阅实现 WMI WQL 查询和 VBS的执行能力。

• SCM——创建或删除服务的能力

• RDP ——通过控制台应用连接 RDP 并执行指定的命令

• DCOM （多个）——实例化远程 COM 服务器和命令执行的调用方法

• Task Scheduler （任务计划程序）——创建和删除计划任务

• 修改服务的 binpath (通过 WMI实现)——使用 Win32_Service WMI 类更新现有服务的现有 binpath，启动服务，暂停服务并将服务重置回原始状态

• 修改计划任务操作——使用命令修改现有计划任务的操作，运行该任务，并将其重置为原始状态

• Shellcode-Exec

• Excel 4 DCOM

• WMI 事件订阅(WIP)

• cmd 命令

• MSHTA

• RegSvr32

• WMIC

• 文件

• 自定义(预制)

• MSBuild

• InstallUtil

• 各种 Hijack

• Service DLL Hijack

• DCOM Hijack（劫持）

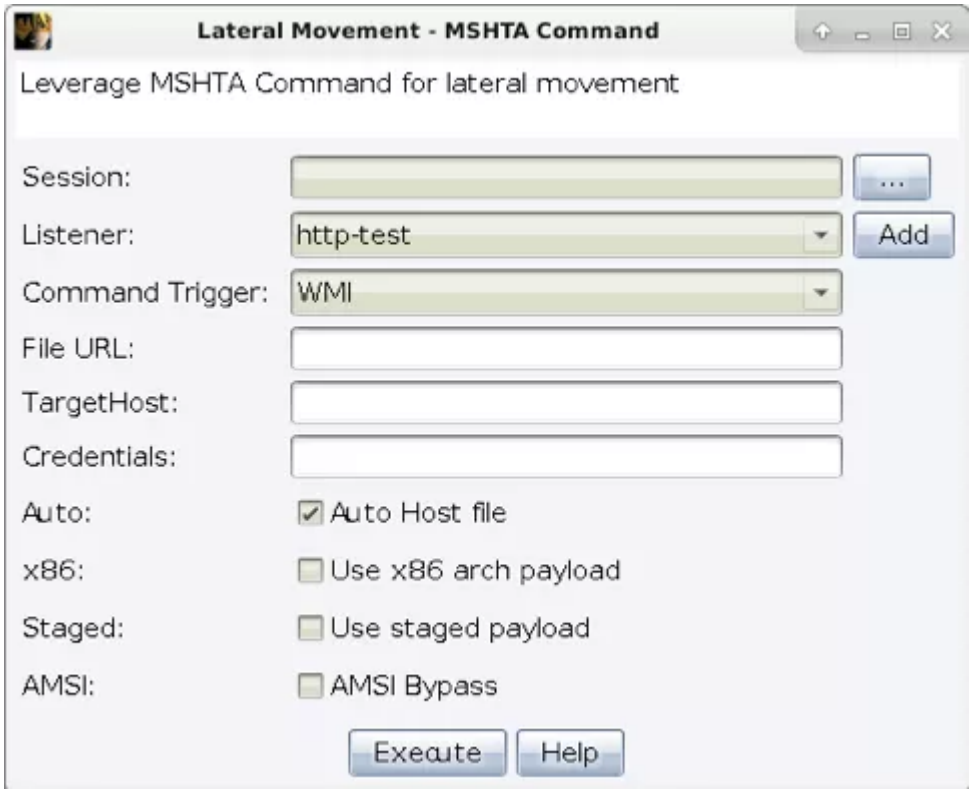

图2— MSHTA 命令菜单

第四个选项(图3)在菜单中称为“File“，它将文件丢到磁盘上，并使用命令(如 MSBuild)执行这个文件。这进一步扩展了执行机制，并为用户提供了更多关于如何触发有效载荷的选项。

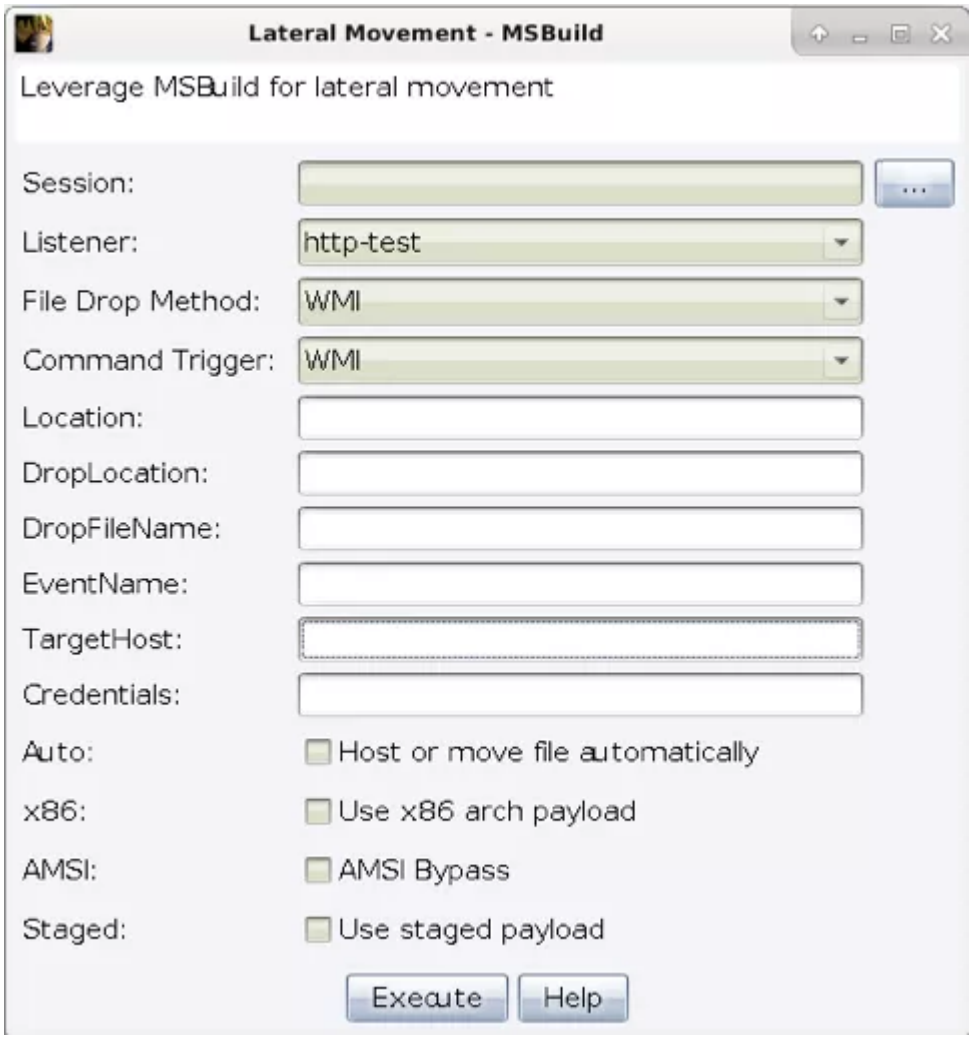

图3— MSBuild 菜单选项

因为“File”选项要求将文件写入目标，所以 FileWrite 程序集会用于这些情况。除了通过 WMI 移动数据之外，攻击者脚本和 FileWrite 程序集都可以使用 SMB 来移动文件，但攻击者脚本将使用 Cobalt Strike 中的上传功能。这样做的目的是为了让操作人员仍然可以选择在Cobalt Strike之外使用这个组装功能。可用的 FileWrite 选项包括：

• SMB ——攻击者脚本使用上传 API 调用到远程 UNC 路径，但程序集可以选择在 Cobalt Strike 之外使用

• WMI 写入文件—— 带有活动脚本消费者的 WMI 事件订阅，执行 VBscript 代码将文件写入磁盘

• WMI 写入注册表项—— WMI 写入注册表数据

• 将 WMI 设置为自定义 WMI 类属性——创建一个新的自定义 WMI 类，并将数据设置为自定义属性的属性值

“File”选项只暴露了“SMB to flat File”和“WMI to flat File”选项，因为需要写入大量文件。要写入注册表项或 WMI 类属性，需要通过“Write File Only（只写文件）”选项来完成。这些数据写入原语的主要用例是存储 shellcode，理想情况下操作人员会将 shellcode 存储在其中的一个位置，并编写一个可以从中读取的文件。

首先查看每种技术的可用选项(图3)，有许多术语表示不同的意思。下面着重说明它们的含义以及如何处理它们以执行特定任务。

• Session 会话——横向运动将起源于信标

• Listener（监听器）—— 将为其生成 shellcode 的监听器。只有在选中“自动”复选框时才使用(可选)

• File Drop method（写入文件的方法）——通过 SMB 或 WMI (写入到平面文件，注册表项 和 WMI 类在这里不公开)写入文件到远程系统的方法。只有在选中“自动”复选框时才使用(可选)

• Command Trigger （命令触发器）——远程触发命令的方式

• Location （位置）——有效载荷的位置，只有当 WMI 被选为“文件写入方法”并且选中“自动”复选框时才使用(可选)

• Drop Location （文件写入位置）—— 要写入文件到远程目标上的位置。只有在选中“自动”复选框时才使用(可选)

• Drop File Name （文件写入名称）——要写入文件到远程主机上的文件名称。只有在选中“自动”复选框时才使用(可选)

• Event Name （事件名称）——事件、计划任务或服务的名称。是否可选取决于命令触发器类型，用于 WMI。

• Target Host （目标主机）—— 试图移动到的主机

• Credentials （凭证）——用于远程系统身份验证的凭证

• Auto （自动）——自动生成和移动/主机文件(取决于机制)。如果不进行检查，就不会生成和移动有效载荷

• x86 —— 要生成的 shellcode 的架构(缺省值为 x64)

• AMSI —— 修改 HKCU\Software\Microsoft\Windows Script\Settings\AmsiEnable 注册表值，只适用于 WSH 而不适用于 PowerShell

• Staged （分段）——使用分段有效载荷(默认是没有分段的)

还应该注意的事情是那些可以在执行后在系统上保持持久性的方法，比如创建的服务、计划任务和事件子任务在执行完成后会自动删除。但是，在目标上写入的文件不会被删除，也没有可用的选项，因此必须手动完成，这是故意这样设计的。

服务控制管理器的选项也被添加到所有技术中，无论它们是否是服务可执行文件。当为 MSBuild 这样的技术选择‘ SCM’作为命令触发器选项时，服务的 binpath 是 cmd.exe，MSBuild 会作为子进程生成。控制管理器调度程序在终止进程之前给出一定的时间来响应，但是，这并不影响子进程。但是这样做会引起对生成 CMD 并为其设置子进程的操作问题。已经为没有意识到这一点或忘记这一点的用户构建了一个安全提示(图4)。此提示将在选择 SCM 和未使用服务可执行文件时重复检查。

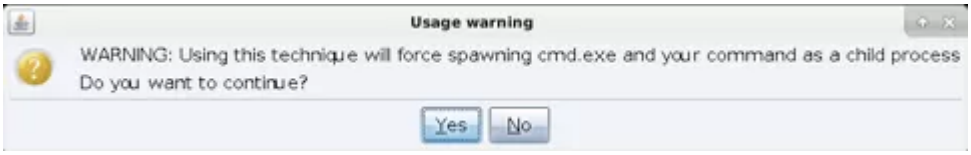

图4 —— 生成 CMD 的安全提示

WMI 文件移动选项之前简单地提到过，但没有完全解释发生了什么。图表将术语‘ WMI 事件’作为一个通用术语，但是所示的技术适用于所有 WMI 选项。“Location”选项指定在何处找到有效载荷，并且只有在文件写入方法为 WMI 时才使用该选项。这可以采用三条独立的路径，每一条路径都在下面的图表中进行了解释。首先，如果输入 URL，信标主机将执行一个向“Location“URI 发出 web 请求的程序集(图5)。请求的响应被保存为一个变量，用于在 WMI 事件中执行 VBScript 并写入主机。

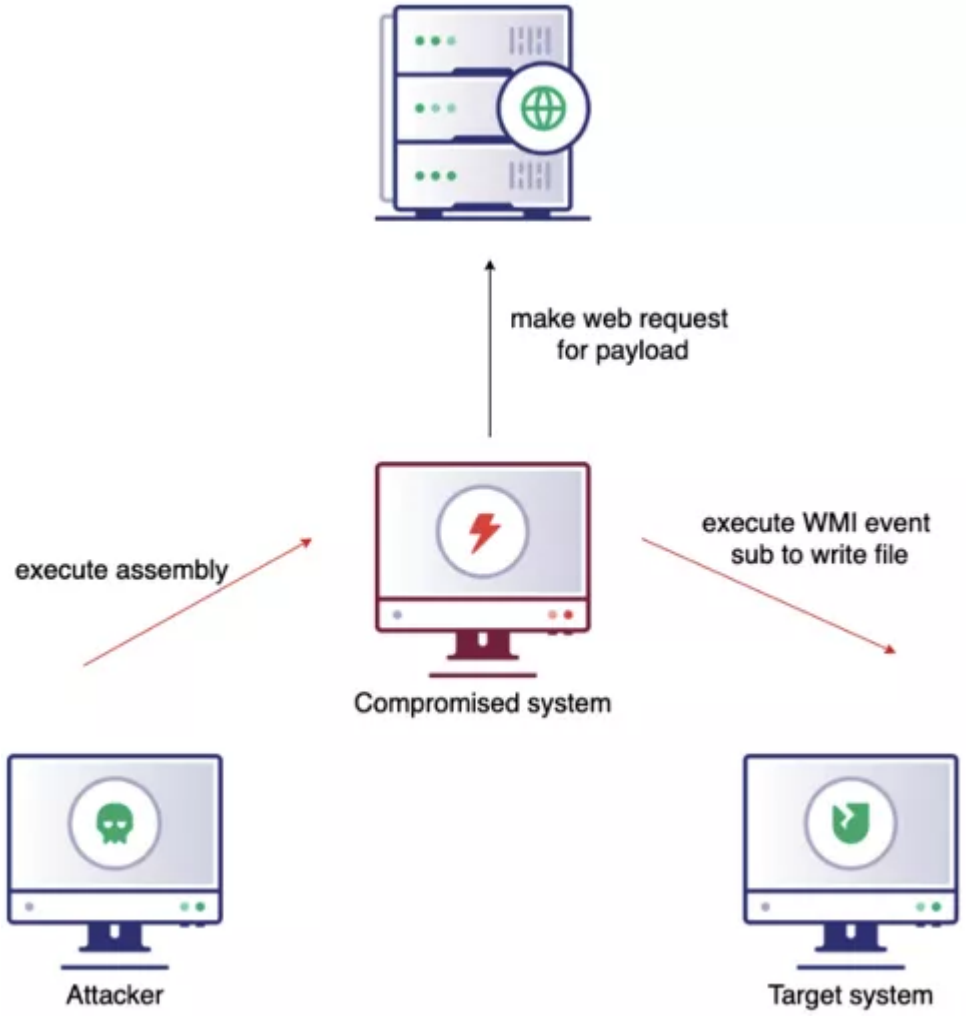

图5— WMI 文件写入方法——URL

其次，“Location”字段可以接受一个 Windows 目录路径，如果找到该路径，则将该路径的文件上传到信标主机(图6)，执行程序集将读取文件内容并将其保存为程序集内的变量，并再次利用 WMI 事件执行 VBScript 并将文件写入到磁盘。这在测试中并不总是最可靠的，这是由于 Cobalt Strike 处理任务的方式所致。

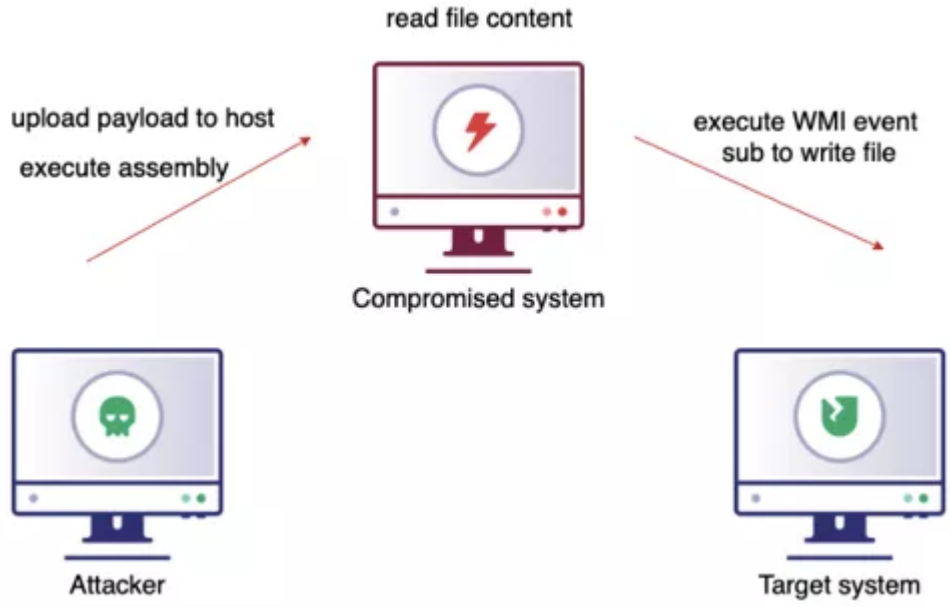

图6— WMI 文件写入方法—— Windows 目录

最后，“Location”可以获取一个 Linux 目录路径，如果找到该路径，它将读取 Cobalt Strike 客户机系统(本地)上的一个文件，对其进行 base64 编码，并将编码后的内容写入 C# 源代码中，然后动态编译该程序集以便在信标主机上执行(图7)。在这三个位置中，这可能是最干净的一种方法，因为没有上传任何文件，也没有从信标主机发出 HTTP 请求。

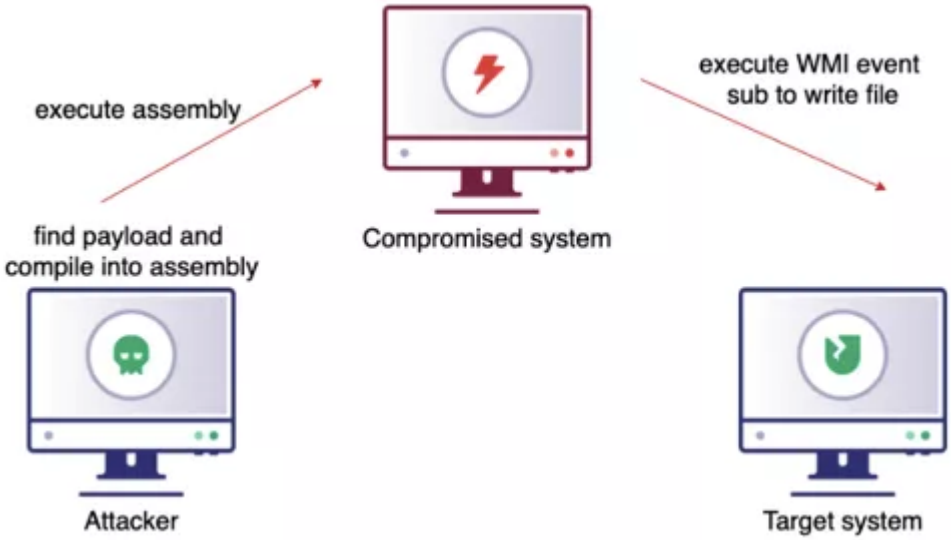

图7— WMI 文件写入方法 —— Linux 目录动态编译

本文参考自：https://posts.specterops.io/move-faster-stay-longer-6b4efab9c644

END

阅读原文