无文件攻击技术浅析

发表于 2021-03-27 | 分类于 红队渗透

无文件攻击（Fileless attack）是不向磁盘写入可执行文件的攻击方法，是APT惯用攻击手段，本文介绍其攻击手段、攻击类型及检测技术...

0x00 技术简介

1. Fileless attack

无文件攻击（Fileless attack）是不向磁盘写入可执行文件的攻击方法，使得攻击者可以攻陷带有反病毒和应用白名单策略防护的系统，绕过常规检测手段。

当攻击者通过消除将PE/ELF（可移植可执行文件）复制到磁盘驱动器的传统步骤来逃避检测时，就会发生无文件或无恶意软件的攻击。

2. Living Off The Land

Living off the Land (LotL) 攻击描述了一种网络攻击，其中入侵者使用系统中可用的合法软件和功能对其执行恶意操作，攻击者使用已经安装的工具在目标计算机上执行指令，或使用正在运行的脚本，或直接在内存中加载shellcode执行指令。

LotL 攻击通常被归类为无文件攻击，因为它们不会留下任何痕迹。

LOLBAS：Windows平台Living off the land Binaries

GTFOBins：Unix平台Living off the land Binaries

0x01 攻击手段

1. 恶意文档

在众多被分类为无文件攻击的案例中，大部分都用到文本文件。在此类攻击中，对方往往是利用恶意文本文档（最常见的是电子邮件的附件），以达到如下目的：

- 文本文档可作为其他文件的灵活载体、容器。

例如，攻击者把 JavaScript 文件嵌入 Microsoft Office 文档中，利用社会工程学诱使文档接收者双击执行脚本。可内嵌文件的其它文档类型包括PDF、RTF。这些特性是各个应用程序的正常特性，通常情况下，反恶意软件技术也不会对这些功能的使用进行干扰。

- 文本文档可以携带执行恶意代码的漏洞利用。

当前，文档功能的复杂性，为漏洞利用提供了广阔的攻击面。通常，exploit 可以触发捆绑的 shellcode 在应用程序的内存空间执行。攻击者无需将恶意代码存放到本地文件系统上即可取得终端设备的控制权。

- 文本文档可以执行恶意感染的逻辑。

现代文档支持强大的脚本能力，例如 Microsoft Office 支持 VBA 宏。此类功能特性，利用大多数反恶意软件无法区分恶意脚本与良性脚本的弱点，使得攻击者无需通过编译程序即可实施恶意感染。文档脚本拥有的能力包括了执行程序和下载恶意代码。

尽管文本文档本身还是存储在终端设备的本地文件系统中，但避免了传统的直接将恶意可执行文件存放于磁盘之上。在大多数情况下，文本文档引导恶意代码直接在终端设备的内存中执行。

2. 恶意脚本

恶意软件创作者倾向于”无文件”方式的脚本运行攻击，而非传统的将代码编译成可执行二进制文件。除了上面提到的文本文档对脚本的原生支持外，直接在 Microsoft Windows 下运行脚本还有以下优点：

- 直接与系统进行交互，无需受其他应用程序的限制，比如浏览器对脚本的检查。
- 与编译的可执行恶意文件相比，更不容易被反恶意软件技术探测到。
- 为躲避应用行为检测，可以灵活地将恶意逻辑分割到数个进程中。
- 可以利用混淆技术，延缓和更好地避免被反恶意软件技术分析、检测。

Microsoft Windows 自带的脚本解释器包括 PowerShell, VBScript, 批处理文件和 JavaScript。对应的应用程序为 powershell.exe,cscript.exe,cmd.exe 和 mshta.exe

攻击者可以利用各种框架工具轻易地混淆攻击脚本。这些混淆措施包括：Daniel Bohannon 针对 PowerShell 的 Invoke-Obfuscation 和 Invoke-DOSfuscation 框架

3. Living off the Land

对无文件攻击的讨论，通常包括对 Microsoft Windows 众多内建工具的滥用。这些工具使得攻击者轻松地从一个阶段”跳转”到另一个阶段，无需执行任何编译的二进制可执行文件。这种操作模式有时被称为”Living off the Land”

一旦攻击者的恶意代码能与本地程序进行交互，那么攻击者就有可能利用系统的原生工具进行下一步攻击，包括下载附带的其他恶意代码，启动程序，执行脚本，窃取数据，横向移动，维持权限等。攻击者为达到这些目的而调用的工具包括 regsvr32.exe, rundll32.exe, certutil.exe, schtasks.exe。更全面地了解系统中被利用的内建程序，库和脚本，参考 LOLBAS project

Windows 管理规范（WMI），是 Windows 系统内建工具，为攻击者提供了”Living off the Land”的绝好机会。WMI 借助运行 wmic.exe 程序和执行脚本（如，PowerShell ），使得攻击者可以操作设备的绝大部分配置。这些工具都是系统自带的、受信任的，所以反恶意软件技术也难以检测和限制，利用 WMI 进行无文件攻击可参考Abusing WMI to Build a Persistent,Asyncronous, and Fileless Backdoor

4. 内存恶意代码

现今的反病毒产品，对于检测磁盘上的恶意文件，是比较成熟的；然而，要想检测到只存在于内存中的恶意代码，往往是事倍功半。内存的易失性、动态性，使得恶意软件很容易地改变形态；同时，可以自由地运行，无需顾忌任何反恶意技术的侦测。

一旦攻击者开始在终端设备上执行恶意代码，那么就有可能将恶意软件解包到内存，无需在磁盘上留下丝毫痕迹。此过程可能包括解压代码到进程自身的内存空间。其他情况下，恶意程序可能将代码注入到其他受信任的进程或者其他正常进程。

内存攻击技术有如下类型：

- 在不利用系统脆弱性的情况下，利用系统功能的内存注入。

例如，包括 VirtualAllocEx 和 WriteProcessMemory 在内的 API 常被恶意软件滥用于内存注入。详细操作，参阅 Gal Bitensky 的 overview of the AZORult attack

- 攻击者可能将编译的可执行程序捆绑在脚本里面，从而在脚本运行时把恶意攻击载荷释放到内存。

该攻击手段的常用工具为 PowerSploit, 操作过程可以阅读 Asaf Aprozper 和 Gal Bitensky 的 the GhostMiner analysis。 Chris Truncer 的 Veil Framework 是另一实例。

- Process Doppelgänging 是另一种无文件技术，且无需涉及经典的内存注入。

攻击者利用 Microsoft Windows NTFS 的事物（访问）特性：临时修改受信任的内存文件而不同步更改至磁盘。Anton Ivanov, Fedor Sinitsyn 和 Orkhan Mamedov 在 SynAck malware中解析了该技术

内存滞留技术，使攻击者可以绕过众多的反恶意技术的控制，包括应用白名单策略。尽管反病毒工具试图追踪内存注入行为，但是攻击者的感染手段使得其防不胜防。Asaf Aprozper的 coffeeshot-avoid-detection-with-memory-injection 利用JAVA实现的一个注入方法可绕过各种主流检测方法

0x02 无文件恶意软件

无文件恶意软件经常使用的工具和技术包括：

- 漏洞利用工具包
- 利用合法工具，如WMI和PowerShell
- 使用被盗凭证
- 注册表驻留恶意软件
- 内存型（Memory-only）恶意软件

1）漏洞利用工具包（Exploit kits）

漏洞利用是一种允许攻击者利用操作系统或应用程序漏洞来访问系统的技术。漏洞利用是一种高效的无文件技术，因为它们可以直接注入内存中，而无需将任何内容写入磁盘。

通过允许攻击者自动化和大规模执行初始突破，漏洞利用工具包使得攻击者的生活更轻松、工作更高效。所需要做的只是，诱使受害者进入漏洞利用工具包服务器，办法通常是网络钓鱼或社会工程。

这些工具包通常提供对许多漏洞的攻击，以及一个管理控制台，一旦成功利用漏洞，攻击者就可以控制失陷的系统。有些漏洞利用工具包甚至提供了扫描受害者系统中的漏洞的功能，因此可以快速构建并启动成功的漏洞攻击。

2）注册表驻留恶意软件

注册表驻留恶意软件是安装在Windows注册表中的恶意软件，以便在逃避检测的同时，保持持久性。第一种是Poweliks，此后就出现了许多变体。一些变体，如Kovter，使用了类似的注册表隐藏技术，来保持不被发现。Poweliks调用C2（命令和控制）服务器，攻击者可以从该服务器向受损系统发送进一步的指令。所有这些操作，都可以在没有任何文件写入磁盘的情况下进行。

3）内存型（Memory-only）恶意软件

有些恶意软件只存在于内存中，以逃避检测。新版本的Duqu蠕虫就是这种情况，它只驻留在内存中，不会被发现。Duqu 2.0有两个版本：第一个是后门，它允许攻击者在组织中站稳脚跟。如果攻击者认为目标值得攻击，他可以使用Duqu 2.0的高级版本，该版本提供了诸如侦察、横向移动、数据渗出等附加功能。

0x03 检测技术简析

1. 常规检测手段

如下为一些常见的端点保护技术，多数方法极难检测到无文件攻击：

1）传统防病毒（AV）

旨在寻找已知恶意软件的特征码。由于无文件攻击没有恶意软件，所以AV没有可检测的特征码。

2）基于机器学习（ML）的反恶意软件方法

在应对无文件攻击时，面临着与传统AV相同的挑战。ML动态分析未知文件，并将其区分为好的或坏的。但是我们已经注意到，在无文件攻击中，没有要分析的文件，因此ML无法提供帮助。

3）白名单方法

包括列出一台机器上所有良好的进程，以防止未知进程执行。无文件攻击的问题在于，它们利用易受攻击的合法白名单应用程序，并利用内置的操作系统可执行文件。阻止用户和操作系统共同依赖的应用程序，并不是一个好的选项。

4）使用失陷指标（IOC）工具

本质上，IOC类似于传统的AV签名，因为它们是攻击者留下的已知恶意制品。然而，由于它们利用合法的进程，并且在内存中操作，所以无文件攻击不会留下制品，因此IOC工具几乎找不到任何东西。

5）沙箱检测技术

可以采取多种形式，包括基于网络的爆破和微虚拟化。由于无文件攻击不使用PE文件，因此沙盒没有什么可爆破的。即便真有东西被发送到沙箱，因为无文件攻击通常会劫持合法进程，大多数沙箱也都会忽略它

2. IOA检测技术

依赖基于签名的方法、沙盒、白名单甚至机器学习等方法，检测无文件技术非常有挑战性，CrowdStrike提出了一种攻击指标（IOA，Indicators of Attack）的检测方法。

IOA提供了针对无文件攻击的独特的主动预防能力。IOA寻找攻击可能正在进行的迹象，而不是关心攻击的步骤是如何执行的。这些迹象可以包括代码执行、隐藏规避、横向移动等。如何启动或执行这些步骤对IOAs来说并不重要。例如，对于IOA来说，一个活动是从驱动器上复制的文件启动的，还是从无文件技术启动的，都无关紧要。**IOA关注的是所执行的行为、它们之间的关系、它们的顺序、它们的依赖性，将它们视为揭示一系列事件背后真实意图和目的的指标。**IOA不关注攻击者使用的特定工具和恶意软件。

在无文件攻击的情况下，恶意代码可以利用诸如PowerShell之类的合法脚本语言，而无需写入磁盘。正如我们所看到的，这对于基于签名的方法、白名单、沙箱甚至机器学习来说都是一个挑战。相比之下，IOA检测恶意软件或攻击完成其任务所必须执行的事件序列。这可以暴露最隐秘的无文件方法，因此它们可以被迅速处理。

由于IOA会查看意图、上下文、活动序列，因此即使恶意活动是使用合法帐户实施的，也可以检测和阻止这些活动，攻击者使用窃取的凭据时通常会出现这种情况。

所有这些使得IOA成为防止无文件恶意软件攻击的突破口。IOA不再基于磁盘上可执行文件的存在与否，来开展无文件攻击这场徒劳的战斗，而是在造成任何损害之前监视、检测、阻止此类攻击的影响。

0xFF Reference

- Living off the land and fileless attack techniques
- https://paper.bobylive.com/Security/crowdstrike-fileless-wp.pdf
- https://conference.apnic.net/48/assets/files/APIC778/Living-off-theLand-An-APT-case-study%20.pdf
- https://minerva-labs.com/blog/deconstructing-fileless-attacks-into-4-underlying-techniques/
- https://www.cybereason.com/blog/fileless-malware
- https://securelist.com/fileless-attacks-against-enterprise-networks/77403/
- https://learn.microsoft.com/zh-cn/microsoft-365/security/intelligence/fileless-threats
- https://lolbas-project.github.io/
- https://gtfobins.github.io/

FILELESS ATTACK LIVING OFF THE LAND