SOPHOS

解释漏洞攻击：

综合漏洞攻击预防

漏洞攻击利用合法软件产品（如 Adobe Flash 和 Microsoft Office）的漏洞感染计算机实现犯罪用途。网络罪犯通常利用漏洞攻击渗透企业防御。罪犯的目的多种多样：盗窃数据或劫持勒索，执行搜索，或者作为部署更传统恶意软件的方式。

通常可以发现漏洞攻击是网络攻击的一部分：90% 以上 报告数据泄漏中，在攻击链的一个或多个点采用漏洞攻击。将漏洞攻击预防加入综合安全防御无疑具有重要意义。

漏洞攻击已经问世超过 30 年，所以并不奇怪，几乎所有主要安全供应商都声称一定程度的漏洞攻击预防。但是，不同供应商对于该防护的广度和深度差异巨大。对于有些供应商只是随意而为；对于有些供应商则是重要焦点。阅读本白皮书，更多了解漏洞攻击，以及主流安全产品中的漏洞攻击预防水平。

目录

漏洞攻击行业:犯罪软件即服务

借助漏洞攻击工具,恶意软件作者无需担心如何找到 Java 或 Silverlight 或 Flash 中的缺陷;如何在有效漏洞攻击中利用这些缺陷;如何找到不安全的 Web 服务器以植入漏洞攻击;如何吸引潜在受害者 访问存在陷阱的网页。

同样,漏洞攻击工具作者不用担心编写完整恶意软件;他们不用运行服务器来跟踪受感染的计算机或从各个 受害者手中盗窃财富;无需参与泄漏盗窃数据 或出售这些数据。

网络犯罪现在已经成为一个价值数十亿美元的行业,预计 到 2019 年造成的破坏将达到 2 万亿美元,攻击的每个方面都实现了产业化。罪犯只需擅长威胁蓝图中的一个或多个部分,我们开玩笑地称为 CaaS – 或"犯罪软件即服务"。

在这个暴利行业中,漏洞攻击经纪人已经出现:他们 从发现漏洞的人手中购买漏洞,然后出售给希望利用的人, 无论是政府机构还是非法黑客。

始终只有买家知道自己的目的。正如 Kevin Mitnick, Mitnick 的 Absolute Zero Day Exploit Exchange 创始人向 Wired 提到的,"当我们 有客户出于任何原因需要零日漏洞时,我们不问原因,事实上他们也不会说。研究人员找到漏洞,出售给我们获得 X 回报,我们 再出售给客户获得 Y 回报,中间就产生了利润。"

> "当我们有客户出于某些原因想要购买零日漏洞时,我们不会问具体原因,实际上他们也不会告诉我们具体原因。漏洞研究者以 X价格出售给我们,我们以 Y价格出售给客户,从中获利。"
>
> Kevin Mitnick

漏洞攻击减轻技术

每天产生超过 400,000 个独特恶意软件样本,每年发现数以千计的新漏洞,预防恶意攻击 的难度大得难以形象。恶意软件变种的爆炸式增长需要 新的创新方法来防御网络罪犯。

仔细研究现代网络犯罪行业,我们可以看到 不对称防御的机会。顾名思义,虽然新攻击看起来无穷无尽,但可以用于漏洞攻击软件的技术只有二十来种。所以,能够应对这些漏洞攻击技术 – 而不是 针对每个漏洞攻击 – 的方法格外有用。

此外:根据漏洞,攻击者往往必须组合 多个漏洞攻击技术才能到达投放恶意软件的阶段。多年来,这些技术并没有太大变化:可能只增加了一到两种新技巧。

评估主要安全产品时,很奇怪缺少重要漏洞攻击技术减轻。有些较新的供应商宣称提供 下一代技术,对漏洞攻击减轻具有更广泛支持, 但他们的覆盖面漏洞百出。下面是旨在消除整个类别或漏洞攻击, 击败网络罪犯以及各国所使用漏洞攻击技术的漏洞攻击减轻列表。

每种技术的减轻依据供应商而不同。务必知道，当供应商宣称预防漏洞攻击时，大多数供应商只是 防范少部分常用漏洞攻击方法，而且通常在 64 位应用程序上 无效。只有 Sophos 提供真正综合漏洞攻击预防。

实施数据执行预防 (DEP)

数据执行预防 (DEP) 是一组硬件和软件技术，对内存执行额外检查以帮助避免缓冲区溢出。如果没有 DEP，攻击者可以尝试跳跃到由攻击者控制的数据所在内存位置的恶意代码 (shellcode)，如堆或堆栈，尝试利用软件漏洞。如果没有 DEP，这些区域通常标记为可执行，这样恶意代码将能够运行。

DEP 是 Windows XP 及更高版本的决定加入选项，必须由软件供应商在开发应用程序时设置。此外，攻击可绕过内置 DEP 保护，因此不建议依赖操作系统实施。

强制地址空间布局随机化 (ASLR)

一些漏洞攻击将已知与特定进程关联的内存位置作为目标。在较旧版本的 Windows 中（包括 Windows XP），核心进程通常在系统启动时加载到可预测的内存位置。地址空间布局随机化 (ASLR) 随机化系统文件和其他程序使用的内存位置，使攻击者更加难以正确猜测给定进程的位置，包括可执行文件的基础，堆栈、堆和库的位置。

ASLR 仅在 Windows Vista 及更高版本中可用，和 DEP 一样，必须由软件供应商在开发应用程序时设置。和 DEP 一样，攻击者可以绕过内置 ASLR 保护，因此，不建议依赖操作系统实施。

由下至上 ASLR

如果启用，由下至上 ASLR 可以改进强制 ASLR 的熵或随机性。

Sophos Intercept X 中的强制 ASLR 和由下至上 ASLR 的主要优点在于，应用程序的基础地址不仅在每次重新启动时随机化，而且在每次启动受保护应用程序时随机化。

空页面（空参考解除保护）

从 Windows 8 开始，Microsoft 拒绝程序分配和/或映射“空页面”的功能（在地址空间的虚拟地址 0x00000000 驻留的内存）。Microsoft 以这种方式成功减轻了一整类“空指针参考解除”漏洞的直接攻击。

在 Windows XP、Windows Vista 和 Windows 7 上，此类缺陷的漏洞攻击将允许攻击者在内核上下文中执行代码（ring0 CPU 权限等级以下），导致权限提升到一个最高等级。此类漏洞攻击允许攻击者防护几乎操作系统的所有部分。

Heap Spray 预分配

Heap spray 技术实际不利用漏洞，而是用于让漏洞更容易被利用。利用称为 Heap Feng Shui[1] 的技术，攻击者可以可靠定位堆上的预期数据结构或 shellcode，从而方便可靠攻击软件漏洞。

典型 heap spray 减轻包括保留或预分配 常用内存地址，这样无法用于承载有效载荷。更有创意的攻击者知道这些方法，因此在实际攻击中，此减轻几乎没有效果。Heap spray 预分配也称为反 HeapSpray 实施或 Shellcode 预分配，通常可有效抵御 测试机构使用的默认漏洞攻击。

动态 Heap Spray

相比静态 Heap Spray 预分配，动态 Heap Spray 减轻通常由内存消耗的突然增加触发。

动态 heap spray 减轻实际分析最近内存分配的内容，检测指示 heap spray 的模式，包括 NOP sled、多形态 NOP sled、JavaScript 数组以及堆上用于方便漏洞攻击的其他可疑序列。

Stack Pivot

应用程序堆栈是一个内存区域，包含内存地址位置列表（称为返回地址）。这些位置包含处理器需要在稍后执行的实际代码。

Stack pivoting 被漏洞攻击广泛用于绕过 DEP 等保护，例如在返回方向的编程攻击中链接 ROP 工具。利用 stack pivoting，攻击者可以从实际堆栈移动到新的假堆栈，可以是受攻击者控制的缓冲区，如堆，攻击者可以在这里控制未来程序执行流。

1 https://cansecwest.com/slides/2014/The%20Art%20of%20Leaks%20-%20read%20version%20-%20Yoyo.pdf

Stack Exec (MemProt)

在正常情况下，堆栈包含数据以及指向处理器将要执行代码的地址。利用堆栈缓冲区溢出[2]，攻击者可以用任意代码改写堆栈。要在处理器上运行此代码，堆栈的内存区域必须可执行以绕过 DEP。执行堆栈内存后，攻击者很容易提供并运行程序代码。

基于堆栈的 ROP 减轻（调用方）

为了打败类似数据执行预防 (DEP) 的安全技术和解决空间布局随机化 (ASLR)，攻击者通常劫持存在漏洞的互联网应用程序的控制流。此类内存攻击对防病毒产品、大部分“下一代”产品以及其他网络防御不可见，因为不涉及恶意文件。攻击在运行时发起，将 现有应用程序（如 Internet Explorer 和 Adobe Flash Player）的短良性代码段组合 – 所谓的代码重新使用或返回方向编程 (ROP) 攻击。

在正常控制流中，敏感 API 功能 – 如 VirtualAlloc 和 CreateProcess – 由 CALL 指令调用。调用敏感 API 时，通常 ROP 防御停止代码执行，使用位于堆栈顶部的“return”地址确定 API 调用地址。如果 API 调用地址指令不是 CALL，则终止进程。

由于堆栈内容可写入，攻击者可以写入在堆栈写入特定值，误导基于堆栈的 ROP 防御分析。基于堆栈的 ROP 防御无法确定堆栈内容是良性还是被攻击者操纵。

基于分支的 ROP 减轻（硬件增强控制流完整性）

正如之前所述，基于堆栈的返回方向编程 (ROP) 防御是粗粒度的，容易受到操纵。为了改进，防御者需要更细粒度且防操纵的数据来进行运行时分析。

Sophos Intercept X 推出硬件增强控制流完整性 (CFI)，发挥主流 Intel® 处理器（2008 及更新版本）的未使用的硬件功能。处理器硬件本身提供只读数据，增强运行时复杂漏洞攻击的检测。采用硬件跟踪（分支）记录相比基于软件堆栈的方法具有明显安全优势。可从这些记录检索的分支信息不仅标识分支目标，而且标识来源。因此，实际显示导致控制流更改的位置。使用基于堆栈的解决方案（如 Microsoft EMET 或 Palo Alto Networks Traps），无法获得相同置信水平的特定信息。

2 https://en.wikipedia.org/wiki/Stack_buffer_overflow

硬件跟踪记录中的分支信息无法被操纵；攻击者无法用于控制数据改写。基于堆栈的方法依赖受攻击者控制的堆栈数据，尤其是 ROP 攻击情况下，攻击者反过来可以误导防御者。相比之下，Sophos Intercept X 检查的硬件跟踪数据更加可靠防篡改。

Endgame 的另一种硬件辅助控制流完整性实施 (HA-CFI) 依据训练普通控制流，检测与编程人员预期代码路径的偏差。必须持续训练以建立有效代码指针地址白名单，反映受保护应用程序的所有可能功能和版本。Sophos Intercept X 不需要训练，而且在线程上下文切换和动态频率缩放时正确工作。

Sophos Intercept X 检测到 Intel® Core™ i3、i5 或 i7 处理器 (CPU) 时，将自动采用硬件增强控制流跟踪。如果未检测到支持的处理器硬件，Sophos Intercept X 将自动回滚基于纯软件堆栈的完整性检查。

Sophos Intercept X 不仅利用硬件跟踪记录增强 ROP 检测，而且用于导入地址过滤 (IAF) 以保护受保护应用程序的导入地址表。

注：用于修复与 Intel CPU 硬件内的分支预测程序有关的 Spectre 漏洞的补丁不影响 Sophos Intercept X 的正确功能。

结构化异常句柄覆盖防护 (SEHOP)

攻击者可以用控制值改写堆栈上的异常记录句柄指针。发生异常时，操作系统将遍历异常记录链，调用每个异常记录的所有句柄。由于攻击者控制其中一个记录，操作系统将跳跃至攻击者希望的位置，给予攻击者对执行流的控制权。

SEHOP 是 Windows Vista 及更高版本的决定加入选项，必须由软件供应商在开发应用程序时设置。攻击可用于绕过内置 SEHOP 保护，因此不建议依赖操作系统实施。

导入地址表访问过滤 (IAF)

攻击者最终需要特定系统功能（如 kernel32!VirtualProtect）的地址才能执行恶意活动。可以从不同来源检索这些地址，加载模块的导入地址表 (IAT) 就是其中一个来源。应用程序在不同模块中调用功能时，IAT 用作查找表。由于编译程序无法知道所依赖的库的内存位置，进行 API 调用时需要间接跳跃。动态链接工具加载模块并链接在一起时，将实际地址写入到 IAT 插槽，从而指向相应库功能的内存位置。

Sophos Intercept X 推出硬件增强导入地址表访问过滤，利用主流 Intel® 处理器（2008 及更新版本）的硬件功能。除了实施控制流完整性的硬件跟踪分支记录，Sophos Intercept X 还利用硬件分支预测，进一步增强导入地址表的保护。

注：用于修复与 Intel CPU 硬件内的分支预测程序有关的 Spectre 漏洞的补丁不影响 Sophos Intercept X 的正确功能。

加载库

攻击者可以将恶意库放在 UNC 路径上，尝试加载。监测对 LoadLibrary API 的所有调用可用于预防此类库加载。

反射 DLL 注入

通常在 Windows 中加载 DLL 时，您调用 API 功能 LoadLibrary。LoadLibrary 将 DLL 的文件路径作为输入，加载到内存中。

反射 DLL 加载指从内存而不是磁盘加载 DLL。Windows 没有支持此功能的 LoadLibrary 功能，因此要实现此功能，必须自己编写。自己编写功能的一个优点是可以忽略 Windows 的一些通常操作，如注册 DLL作为进程中的加载模块，使反射加载工具更加隐秘。Meterpreter 是使用反射加载隐藏自己的一个工具示例。分析 DLL 是否反射加载在内存中，执行减轻。

Shellcode

Shellcode 是一小段机器代码，用作软件漏洞攻击的有效载荷。称为“shellcode”的原因是历史上将启动一个命令 shell，攻击者可以从这里控制受威胁的机器，但任何执行类似任务的代码段都可以称为 shellcode。

漏洞攻击通常在利用漏洞获得处理器指令指针 (EIP/RIP) 控制权前或同时，将 shellcode 注入目标进程。调整指令指针指向 shellcode，然后执行代码，完成任务。

VBScript 上帝模式

在 Windows 中，VBScript 可以用于浏览器或本地 shell。如果用于浏览器，出于安全原因限制 VBScript 的功能。此限制受 safemode 标志控制。如果修改此标志，HTML 中的 VBScript 可以像在本地 shell 一样执行任何操作。这样，攻击者可以在 VBScript 中轻松写入恶意代码。在 Web 浏览器中操纵 VBScript 的 safemode 标志称为上帝模式[3]。

例如，攻击者可以利用CVE-2014-6332 漏洞[4]修改 safemode 标志值，此缺陷由调整Internet Explorer VBScript 引擎中的数组大小时不当操作导致。在上帝模式下，用 VBScript 写入的任意代码可破坏浏览器沙箱。借助上帝模式，数据执行预防 (DEP)、地址空间布局随机化 (ASLR) 和控制流防御 (CFG) 保护不起作用。

WoW64

Microsoft 通过“Windows on Windows”(WoW) 层在 64 位版本 Windows 上提供对 32 位软件的向后兼容性。WoW 实施的各个方面为攻击者提供有趣的方法进行复杂化动态分析，二进制解包以及绕过漏洞攻击减轻。

32 位应用程序在 WoW64 环境下的行为与真正的 32 位系统存在许多方面的不同。运行时切换执行模式的功能可以为攻击者提供漏洞攻击、模糊化和防模拟方法，例如：

- 32 位代码中没有的其他 ROP 工具
- 混合执行模式有效载荷编码器
- 减弱呈现器减轻效果的执行环境功能
- 绕过安全软件插入的挂钩，仅在 32 位用户空间中

3 https://en.wikipedia.org/wiki/Glossary_of_video_game_terms#God_mode
4 https://www.rapid7.com/db/modules/exploit/windows/browser/ms14_064_ole_code_execution

大多数端点保护软件将仅当在 WoW64 下运行进程时，挂钩 32 位用户内存空间中的敏感 API 功能。如果攻击者能够切换为 64 位模式，可以获得访问权解钩在 32 位模式下挂钩的 64 位版本敏感 API 功能。

在 64 位版本 Windows 上，Sophos Intercept X 禁止程序代码直接从 32 位切换为 64 位模式（例如使用 ROP），同时仍支持 WoW64 层执行此过渡。

有关滥用 WoW64 的更多信息，请参见 Duo Security 的研究：WoW64 和 So Can You[5] 以及减轻 Wow64 漏洞攻击[6]。

Syscall

Syscall（或系统调用）是一种编程方式，计算机程序可在其中申请操作系统内核的服务。这包括硬件相关服务，如访问本地磁盘，创建和执行新进程。

通常，操作系统提供普通应用程序编程结构 (API)，位于正常程序和操作系统之间。在正常情况下，应用程序将始终调用 API，从内核申请特定任务。安全软件在敏感 API 功能中放入挂钩，拦截并执行类似放大版扫描的检查，然后才允许内核服务请求。

攻击者可以利用下面的事实：

- 不是所有 API 功能都被安全软件挂钩；仅敏感功能。
- 用于调用内核功能的存根非常类似；仅功能索引唯一。

在偏移调用非监测非敏感功能存根（以有意解决敏感内核服务），攻击者可以有效回避大多数安全软件或沙箱分析。

Sophos Intercept X 采用全新方法阻止攻击者通过非保护 API 功能解决敏感内核功能。

有关滥用 syscall 的更多信息，请参见 BreakDev.org 博客文章，从内部防御防病毒实时防护[7]。

Process Hollowing

在 Process hollowing 技术中，受信任应用程序 – 如 explorer.exe 或 svchost.exe – 仅加载在系统上，充当恶意代码容器。空心进程通常以挂起状态创建，内存取消映射，并替换为恶意代码。与代码注入类似，恶意代码的执行掩蔽在合法进程下，可避开防御和检测分析。

5 https://duo.com/blog/wow64-and-so-can-you
6 https://hitmanpro.wordpress.com/2015/11/10/mitigating-wow64-exploit-attacks
7 https://breakdev.org/defeating-antivirus-real-time-protection-from-the-inside

Process Doppelgänging

大多数 Windows 计算机使用 NTFS 文件系统。2007 年，Microsoft 推出一个称为 Transactional NTFS (TxF) 的新功能。此功能允许将多个文件操作作为一个整体处理：可以作为整体成功 – 提交，或者作为整体失败 – 回滚。这样应用程序可能对磁盘上的多个文件进行多个改动，如果检测到错误，所有文件回滚到原始状态。TxF 的最常用用途是在安装 Windows 更新时。

Process Doppelgänging 利用 TxF 机制隐藏恶意软件。它选择无辜文件，改写并通过低级 API 运行恶意软件，例如模仿受信任文件（类似于 process hollowing）。允许恶意软件运行前 – 拒绝或回滚所有改动，从而阻止防病毒软件扫描实际执行的文件内容。如果打开，磁盘上的文件不包含可疑内容。此外，此文件可以是数字签名的已知应用程序。

DLL 劫持

借助通常称为 DLL 劫持、DLL 欺骗、DLL 预加载或二进制植入的漏洞，许多程序将加载并执行与此类程序打开的数据文件处于相同文件夹中的恶意 DLL。

动态数据交换 (DDE)

Windows 动态数据交换 (DDE) 是一种客户端服务器协议，用于应用程序之间的进程间通信 (IPC)。攻击者可以利用 DDE 执行任意命令。例如，Microsoft Office 文档可能被 DDEAUTO 命令中毒，用于通过针对性网络钓鱼广告或植入 Web 内容提供 PowerShell 命令执行，避免使用 Visual Basic for Applications (VBA) 宏。还可以将 DDEAUTO 命令嵌入在 电子邮件或会议请求的邮件正文中，在 Microsoft Outlook 中回复或接受时执行。

由于采用应用程序锁定减轻的设计，Sophos Intercept X 固有阻止通过动态数据交换执行恶意代码。

应用程序锁定

如果攻击者成功利用漏洞并绕过所有内存和代码减轻，Sophos Intercept X 可以限制攻击者的能力。此功能称为应用程序锁定，目的是阻止攻击者引入有害代码。

应用程序锁定阻止通常不依赖应用程序中软件缺陷的攻击。例如，此类攻击可以使用（针对性）网络钓鱼电子邮件附加的 office 文档中的伪造（恶意）宏。文档中的宏采用 Visual Basic for Applications (VBA) 编程语言创建，包含从 Web 下载并运行二进制文件的功能，并允许使用 PowerShell和其他受信任应用程序，具有潜在危险。

这种意外功能（或逻辑缺陷漏洞攻击）为攻击者带来明显优势，攻击者无需利用软件缺陷或寻找方法绕过代码和内存防御来感染计算机。他们只需滥用广泛使用的受信任应用程序的标准功能，利用社交工程说服受害者打开专门编制的文档。

无需维持文件夹黑名单，Sophos Intercept X 将根据行为自动终止受保护的应用程序；例如，利用 office 应用程序启动 PowerShell，访问 WMI，运行宏以安装任意代码或操纵关键系统区域时，Sophos Intercept X 将阻止恶意操作 – 即使攻击不产生子进程。

Java 锁定

以前，漏洞攻击工具是顺便下载恶意软件的重要实现途径。它们利用 Java 运行时环境 (JRE) 的漏洞投放Windows PE 有效载荷。JRE 作为插件或外接程序加载在主流浏览器中。

Sophos Intercept X 阻止 JRE 运行非 Java 应用程序。例如，Sophos Intercept X 将终止尝试引入和 运行 Windows PE 二进制文件的 Java 应用程序。此外，攻击者无法滥用 Java 操纵 自动启动位置，包括 startup 文件夹、Run、RunOnce 和其他注册表项。

注：2014 年推出 Java 8 更新 20 后，Java 应用程序的安全等级默认设置为高。这样攻击者更难以用足够权限运行 Java 漏洞以感染端点。因此，Java 漏洞不再是漏洞攻击工具的最爱，这使得 Java 锁定的减轻意义有一定削弱。

代码洞

对手利用代码洞技术，修改得像合法软件一样以包含额外应用程序。此额外应用程序插入在代码洞，即程序不使用的目标应用程序文件段。代码洞存在于大多数应用程序中，在这些部分加入代码不得破坏主要应用程序的行为。

通常插入代码洞的执行代码只是远程 shell 启动程序或后门程序；体积可能非常小，仅授予对手对可执行其他操作的端点的访问权。此类攻击需要攻击者在端点建立存在，这样可以部署后门应用程序或者欺骗用户下载并安装已经利用代码洞漏洞的应用程序。

对手使用代码洞的一个主要原因是躲避普通用户和管理员检测。预期应用程序仍将正常工作，但插入的应用程序也运行。

如果已修改的应用程序是管理员在设备上需要的合法商业工具，当传统防病毒产品检测到问题时，他们不太可能将其视为恶意软件。管理员可能将其添加到豁免列表，认为防病毒引擎生成误报。这样，对手在端点建立存留，甚至欺骗管理员允许其插入的应用程序运行。

在所谓的供应链攻击中，攻击者还可以突破软件更新服务器，为更新程序编织恶意代码，用勒索软件或 wiper 恶意软件静默感染客户。

Sophos Intercept X 自动阻止执行编织后门的应用程序。如果代码执行不流动到代码洞或感染 PE 文件中的添加段，甚至可以检测到添加的 shellcode。它提供对 Shellter 和 Backdoor Factory 等 shellcode 注入工具的广泛防护。

进程迁移 – 远程反射 DLL 注入

对手通常利用进程迁移技术首次在设备上建立存在，并移动到其他进程以提升权限或获得更持久的访问权。对手不希望最终用户关闭浏览器或终止已威胁进程后失去控制，因此需要迁移到系统进程。

远程反射 DLL 攻击类似于进程迁移。对手已经威胁一个进程，并从此进程操纵其他进程以加载 DLL 和运行任意代码。

本地权限提升 (LPE)

Sophos Intercept X 阻止低权限进程通过从高权限进程盗窃的令牌提升权限。此技术通常与其他漏洞一起使用，以系统权限成功投放和运行攻击者的恶意代码。

DoublePulsar 代码注入

DoublePulsar 最初是美国国家安全局 (NSA) 人权小组开发的后门植入工具，2017 年初被 The Shadow Brokers 泄漏。植入工具包含多个 NSA 漏洞部分的新注入技术，包括 EternalBlue 和 EternalRomance。这些漏洞还用于 WannaCry 和 NotPetya 爆发中的自我传播蠕虫病毒部分。

DoublePulsar 代码注入技术采用异步程序调用 (APC) 在通常受信任进程中运行任意代码 (shellcode)。Sophos Intercept X 本质上破坏 DoublePulsar 采用的基本方法，从而阻止依赖相同技术进行代码注入的攻击。

AtomBombing 代码注入

异步程序调用 (APC) 注入将恶意代码连接到进程线程的 APC 队列。队列 APC 功能在线程进入可修改状态后执行。AtomBombing 是利用 APC调用以前写入到全局 atom 表的恶意代码的变种。

DoubleAgent 代码注入

DoubleAgent 利用 Windows 的一个合法工具 Microsoft 应用程序检验工具。此工具包含在所有版本的 Microsoft Windows 中，用作运行时检验工具以发现和修复应用程序的缺陷。应用程序检验工具还可设置为从磁盘加载任何库，从而可以加载恶意库，获得受害者进程的权限。

DoubleAgent 设计为防病毒产品上的漏洞和零日攻击，但实际上应用程序检验工具的目的是将任意代码加载到任何选择的应用程序中，包括信任的生产力和 Windows 进程。

Sophos Intercept X 阻止通过滥用应用程序检验工具注入代码。

Intercept X 功能

功能	
漏洞防御	
实施数据执行预防	✓
强制地址空间布局随机化	✓
由下至上 ASLR	✓
空页面 (空服从保护)	✓
Heap Spray 分配	✓
动态 Heap Spray	✓
Stack Pivot	✓
Stack Exec (MemProt)	✓
基于堆栈的 ROP 减轻 (调用方)	✓
基于分支的 ROP 减轻 (硬件辅助)	✓
结构化异常句柄覆盖 (SEHOP)	✓
导入地址表过滤 (IAF)	✓
加载库	✓
反射 DLL 注入	✓
Shellcode	✓
VBScript 上帝模式	✓
Wow64	✓
Syscall	✓
Hollow 进程	✓
DLL 劫持	✓
Squiblydoo Applocker 绕过	✓
APC 保护 (Double Pulsar / AtomBombing)	✓
进程权限提升	✓
活跃对手减轻	
凭据盗窃保护	✓
代码洞减轻	✓
Man-in-the-Browser 保护 (安全浏览)	✓
恶意流量监测	✓
Meterpreter Shell 检测	✓

功能	
防病毒预防	
勒索软件文件保护 (CryptoGuard)	✓
自动文件恢复 (CryptoGuard)	✓
磁盘和引导记录保护 (WipeGuard)	✓
应用程序锁定	
Web 浏览器 (包括 HTA)	✓
Web 浏览器插件	✓
Java	✓
媒体应用程序	✓
办公应用程序	✓
深入学习	
深入学习恶意软件检测	✓
深入学习阻止潜在不需要的应用程序 (PUA)	✓
误报禁止	✓
实时保护	✓
应对 调查 移除	
根本原因分析	✓
Sophos Clean清理方案	✓
同步化安全心跳	✓
部署	
可作为独立代理运行	✓
可随现有防病毒程序运行	✓
可作为现有 Sophos Endpoint 代理组件运行	✓
Windows 7	✓
Windows 8	✓
Windows 8.1	✓
Windows 10	✓
macOS*	✓

* 功能支持 CryptoGuard、恶意流量检测、同步安全心跳、根本原因分析

免费试用 Sophos Intercept X

sophos.cn/intercept-x

北京：
电话：4006506598
+86 13552376911
电子邮件：salescn@sophos.com

上海：
电话：+86 18521070801
+86 18901838899
电子邮件：salescn@sophos.com

华南：
电话：+86 13859998247 （深圳）
+86 13602416506 （广州）
电子邮件：salescn@sophos.com

18-03-06 WP-ZHCN (DD)

SOPHOS