# 英特尔对预测执行侧信道的分析
# 白皮书

***1.0 版***

***2018 年 1 月***

英特尔技术的特性和优势取决于系统配置，可能需要激活支持的硬件、软件或服务。性能的变化依赖于系统配置。没有任何计算机系统是绝对安全的。请联系您的系统制造商或零售商，或者访问 www.intel.com，了解详情。

本文档不明确或暗含地授予任何知识产权。

本文档包含正在开发的产品、服务和/或制程的信息。本文中提及的所有信息可能变更，请见谅不另行通知。请联系您的英特尔代表，了解最新的预测、计划、规格和路线图。

所述产品和服务可能包含不足或错误（被称为勘误表），可能与已公布规范有偏差，目前的勘误表可应请求提供。

英特尔不承认所有的默认保证，包括并不限于暗含地保证适销性、适于特定用途、非侵权，以及履行惯例、交易惯例或贸易惯例中出现的任何保证。

本文中提及的所有信息可能变更，请见谅不另行通知。请联系您的英特尔代表以获取最新的英特尔产品规格与路线图。

欲获取有序列号并且在本文中提及的文档副本，请致电 1-800-548-4725 或访问 www.intel.com/design/literature.htm。

# 目录



# 修改历史

| 文档编号 | 版本号 | 说明 | 日期 |
|---|---|---|---|
| 336983-001 | 1.0 | 第一版 | 2018 年 1 月 |

# 1 引言

英特尔致力于通过软件和硬件提升计算机系统的整体安全性。正如谷歌 Project Zero 项目（https://googleprojectzero.blogspot.com/）所详细描述的，一系列新的侧信道分析方法已经被发现，可能会造成未经授权的信息访问。这些方法依赖于高性能微处理器和现代操作系统的共同属性，而潜在的安全问题不仅限于英特尔处理器，而且并不意味着处理器运行时超出了其设计的功能规范。所有这些方法均利用了预测执行，这是处理器中的一个常用技巧，旨在实现高性能。

英特尔正在与生态系统合作伙伴以及其它受影响的芯片厂商密切合作，针对这些方法而设计防御方案。

本白皮书提供了谷歌 Project Zero 所提到了这些方法的信息，并介绍了英特尔正在为每一种方法开发的防御方案。

# 2 预测执行侧信道方法

## 2.1 预测执行

预测执行是大多数现代高性能处理器为了提高性能而采取的主要技术之一。预测执行的概念是，CPU 基于经验预先执行了后续可能执行的指令。如果没有预测执行，处理器在执行后续指令之前需要等待指令解析完成。通过预测性地执行指令，可以减小延迟，提高并行性，从而提高性能。预测执行的一个缺点是，如果发现预测执行的指令根本没有必要，那么已经执行得出的结果可能会被丢弃。

最常见的预测执行形式是一个程序的控制流。处理器会利用一套高度复杂的机制来预测控制流，而不是等待所有分支指令的解析来确定需要执行哪些操作。这些预测通常是正确的，从而可以通过隐藏决定控制流的操作延迟，并通过更大的指令池提高处理器的并行处

理能力来实现高性能。然而，如果一个预测发生错误，那么已经根据预测而执行的工作就会被放弃，而处理器需要重新按照正确的指令路径执行。

虽然预测操作并不影响处理器的架构状态，但是它们会影响微架构状态，例如存储在 TLB（**旁路快表缓冲**）和 L1 缓存中的信息。本白皮书中描述的侧信道分析方法就是利用了预测执行会影响缓存中的内容这个事实。

### 2.2 侧信道缓存方法

侧信道方法的工作原理是通过监测系统来获取信息——例如测量系统的微架构属性。不同于缓冲区溢出或其它安全漏洞，侧信道既不直接影响程序的执行，也不允许修改或删除数据。

缓存计时侧信道涉及一个代理程序去检测一段数据是否存在于处理器特定层级的缓存中，从而用于推断其它部分的信息。一种用来检测目标数据是否存在于缓存的方法是使用计时器去测量该内存地址的访问延迟。如果访问这块内存的时间很短，那么数据肯定会在相应的缓存中。如果访问需要更长时间，那么数据可能不在相应的缓存中。

谷歌 Project Zero 项目发现了三种方法。其中，缓存计时侧信道可能会被用来泄漏秘密信息。

#### 2.2.1 边界检查旁路

“边界检查旁路”方法利用了条件分支指令之后的预测执行。攻击者发现并创建“混淆代理”代码，从而获取到正常情况下无法访问的信息。

边界检查旁路进行预测操作发生在处理器检查输入是否处于边界时，例如检查一个被读取的阵列的索引的值是否在可接受的范围之内。这种方法利用了在边界检查解析之前预测性地执行对界外内存的访问。这些内存访问在特定情况下会向攻击者泄漏信息。

如果攻击者识别到一个特权“混淆代理”，攻击者能够利用该代理来推断该代理可以访问

（攻击者无法访问）的内存内容。

### 2.2.2 分支目标注入

“分支目标注入”方法利用了处理器内部的间接分支预测器，后者用于指导预测性地执行哪些操作。通过影响间接分支预测器的操作方式，攻击者可以让恶意代码被预测性地执行，从而推断数据内容。

对于条件性直接分支，只有两个有关预测性地执行哪些代码的选项——分支的目标或该分支直接的后续指令。除此之外，攻击者无法让代码被预测性地执行。然而，间接分支可以导致更广泛的目标集上的代码被预测性地执行。这种方法的原理是让间接分支预测性地执行一个恶意程序，此应用程序根据受害者可用的敏感数据而创建一个侧信道。

利用对处理器预测器的干扰引发一个侧信道高度依赖于微架构的实现，因此确切的方法因不同的处理器系列和不同代的处理器而异。例如，某些处理器实现中的间接分支预测器只使用了全部地址的一个子集对预测器进行索引。如果攻击者可以辨别使用了哪些子集，他们就可以利用这些信息通过混淆来制造干扰。类似地，在支持超线程的处理器上，一个线程的行为是否可以影响其它线程的预测是一个考量因素。分支目标注入方法只能用于近程间接分支指令。

### 2.2.3 恶意数据缓存加载

最后一个方法是恶意数据缓存加载。这种方法涉及到攻击者直接探测内核（管理程序）的内存。这种操作通常会导致程序错误（由于页表权限导致的页面错误）。然而，在特定实现中，这种操作可能在特定条件下被预测性地执行。例如，在某些实现中，如果数据驻留在最低层数据缓存（L1）上，这种预测性操作只会把数据传递给后续操作。这让应用软件能够查询问题数据，从而导致侧信道泄漏管理程序数据。这种方法只适用于页表仅指定给管理程序的内存块，而不适用于指定为不存在的内存。

## 3 防御方案

针对上述三种侧信道攻击的防御，英特尔已经在与生态系统合作伙伴——包括其它处理器

厂商和软件开发商——密切合作，并将考虑同时适用于已上市产品和研发中产品的技术。防御方案旨在抵御这些攻击方法，同时平衡性能影响和实现复杂性。开启现有安全特性如防止超级用户访问以及禁止执行位能大幅提高攻击系统的难度并降低其它防御方案对性能的影响。有关这些安全特性的更多细节见“相关的英特尔安全特性”部分。

英特尔一直在与操作系统厂商、VMM 厂商和其它软件开发商合作，抵御这些攻击。

作为我们常规开发流程的一部分，英特尔在未来的处理器中可能会加强这些防御方案的性能和效果。

### 3.1 边界检查旁路的防御

对于边界检查旁路攻击，英特尔的防御重心是软件修正。

英特尔推荐的软件防御是在恰当的地方插入屏障来阻止预测。具体来说，就是推荐使用 LFENCE 指令。即使在预测性执行的情况下，序列化及 LFENCE 指令将在旧指令执行完毕前阻止较新指令的执行，但 LFENCE 是一个比其它序列化指令性能更高的解决方案。边界检查插入的 LFENCE 指令会防止边界检查之前去执行新的指令操作。注意，必须在恰当的地方插入 LFENCE；如果随便使用，性能可能会大幅下降。

可以创建一组静态分析规则用来查找软件中需要插入屏障的地方。例如，英特尔对 Linux 内核的分析发现了只有少数可能需要插入 LFENCE 指令的地方，因此把对性能的影响降至最低。就像所有静态分析工具一样，结果中可能存在误报，因此推荐人工检测。

### 3.2 分支目标注入的防御

对于分支目标注入攻击，已经开发了两种防御技巧。软件厂商可以选择适合自己安全、性能和兼容性目标的方案。

第一种技术是在处理器和系统软件之间引入一个新的接口。这个接口提供的机制能够让系统软件防止攻击者控制受害者的间接分支预测，例如在合适的时间清除间接分支预测器以抵御此类攻击。这个接口的详细信息将在以后的英特尔®64 和 IA-32 架构软件开发者手册

中提供。对于很多现有的处理器，支持这个新的接口需要同时加载系统软件和 CPU 微码的更新，将来的英特尔处理器也将支持这个新接口。这种防御策略实现了以下三种新功能的支持。如果更新了合适的 CPU 微码，这些功能将适用于现有产品以及将来的产品，从而降低这些防御措施对性能的影响。具体来说，这些功能包括：

- 间接分支受限推测（IBRS）：限制对间接分支的推测。
- 单线程间接分支预测器（STIBP）：防止间接分支预测被同核的超线程所控制。
- 间接分支预测器屏障（IBPB）：确保前期代码的行为不会控制后续间接分支预测。

第二种技术引入了“return trampoline”概念，也被称为“retpoline”。本质上，软件利用一个代码序列来取代本地间接跳转和调用指令。该代码序列包括把存疑的分支的目标推向堆栈，然后执行一个 Return（RET）指令跳转到该位置，因为 Return 指令通常通过这种方法来保护。对于很多当前英特尔处理器上的特定工作负载，这种技术比第一种技术的更有效。

英特尔已经在与各种开源编译器合作，确保对“return trampoline”的支持，并与操作系统厂商合作，以确保对这些技术的支持。对于 Broadwell 一代以及后续的英特尔®酷睿™处理器，这个 retpoline 防御策略还需要更新 CPU 微码，才能让这种防御完全发挥作用。

### 3.3 恶意数据缓存加载的防御

对于恶意数据缓存加载的攻击，操作系统软件会确保在执行用户代码时不会映射特权页面，以防止在用户模式下访问内核特权页面。操作系统为每个用户进程创建两个根页表结构（CR3 值）：

- “用户”页面结构应当映射该进程的所有应用页面，但只包括最少的特权页面以保证常规处理器操作以及内核空间和用户空间的切换。
- “特权”页面结构应当映射所有内核页面。为了便于访问，它可能希望也映射应用页面。

“KASLR 已死：KASLR1 万岁”论文中，针对侧信道攻击的防御策略，提出了这种应用内核地址空间布局随机化（KASLR）技术的双页表结构，并被称作“KAISER”。这种方法也可

防御恶意数据缓存加载。英特尔已经与各种操作系统厂商合作，在其操作系统中启用双页表结构。

实现这种双页表结构的操作系统可能需要利用处理器进程上下文标识符（PCID）的特性，支持进程上下文标识符（PCID）特性的处理器可以大幅降低 TLB 刷新（在用户空间和内核空间切换时频繁重新加载 CR3）对性能的影响。

未来的英特尔处理器还将拥有针对防御恶意数据缓存加载的硬件支持。

## 4 相关的英特尔安全特性和技术

现有英特尔产品或未来产品中有一些安全特性和技术，可降低前文所述攻击的有效性。

### 4.1 英特尔® OS Guard

英特尔® OS Guard 也称为管理模式执行保护 (SMEP)。这项特性启用时，操作系统不能直接执行应用代码（包括推测执行）。这会迫使攻击者在操作系统代码中查找漏洞，从而增加在操作系统上进行分支目标注入攻击的难度。同时这也会增加应用程序训练操作系统代码以求跳转至漏洞位置的难度。主流操作系统默认启用 SMEP。

### 4.2 执行禁用位（Execute Disable Bit）

执行禁用位是一种基于硬件的安全特性，可帮助减少系统受到病毒和恶意代码攻击的风险。执行禁用位允许处理器对应用代码可以执行或无法执行（包括推测执行）的内存区域进行划分。这可缩小漏洞范围，从而增加分支目标注入攻击的难度。主流操作系统默认启用执行禁用位。

### 4.3 控制流执行技术 (Control flow Enforcement Technology)

在未来的英特尔处理器中，控制流执行技术 (CET)将允许限制近程间接跳转和调用指令，使其仅针对 ENDBRANCH 指令。这一特性可减少允许的非 ENDBRANCH 指令预测。这可以缩小漏洞范围，从而增加分支目标注入攻击的难度。

如欲了解有关 CET 的更多信息，请参见此处的控制流执行技术预览：

https://software.intel.com/sites/default/files/managed/4d/2a/control-flow-enforcement-

technology-preview.pdf。

### 4.4 保护密钥 (Protection Keys)

未来的英特尔处理器，将配备用于减少恶意数据高速缓存负载 (Rogue Data Cache Load) 的硬件支持和保护密钥支持。保护密钥可限制软件对数据的访问。这可以被用于防止分支目标注入或边界校验绕道攻击暴露内存地址。

### 4.5 管理模式访问保护 (Supervisor-Mode Access Prevention)

管理模式访问保护 (SMAP) 迫使攻击内核的应用程序去使用内核内存空间用于侧信道，限制可使用的基于高速缓存的侧信道的内存地址。这加大了应用在内核上实施攻击的难度，因为确定内核是否进行了高速缓存比确定应用是否进行了高速缓存更加困难。

## 5 结论

英特尔协同受到该攻击影响的其他平台厂商，与软件厂商、设备制造商和生态系统合作伙伴紧密合作，开发可保护系统免受这些方法影响的软件和固件更新。最终用户和系统管理员应及时咨询操作系统厂商和系统制造商，尽快应用任何有效的可用更新。

恶意软件会通过这些方法在本地破坏系统安全性。英特尔建议采取系统安全措施全面防范恶意软件对这些分析方法的不良使用。

威胁环境在持续演变。英特尔致力于加强产品安全性和可靠性方面的投资，与行业安全研究人员等有关各方开展建设性合作，全力保护用户的敏感信息。如需了解更多详情，请访问英特尔安全中心[2]。英特尔将继续研究架构和/或微架构变化，帮助抵御这类攻击，同时保持处理器的卓越性能。

[2] https://security-center.intel.com/