津逮®处理器动态安全监控技术（DSC）白皮书

文档编号：200128

文档版本：1.3

免责声明

本文档包含尚在开发设计阶段的产品信息。因此，本文所含信息会随时变更而不做另行通知。请勿将此类产品信息作为您最终设计方案的依据。

本文档不得被用于对本文档所述的澜起产品进行侵权或其他法律分析之用途，或者为此类用途提供任何便利。并且，对于今后您起草的包含本文档内容的任何专利申请，您均同意向澜起科技授予此专利的非独占、免版税使用许可。

本文档未授予任何知识产权的使用许可（无论是明示还是暗示，禁止反言或其他方式）。

本文档所述产品可能包含致使该产品与发布规格存在偏差的设计缺陷或错误（以勘误表形式记载）。澜起科技会应您的要求提供当前已确认的勘误表。

所有产品及其相关资料均“按现状”并以“包含一切错误”的基础提供。无论是以明示、暗示还是法定保证的形式，澜起科技及其附属公司和供应商明确表示对此不负任何担保责任，包括但不限于对非侵权性、适销性或特定用途适用性的任何默示性担保，以及基于任何提案、规格、样品产生的、或在履约、交易、使用或贸易实践过程中产生的任何担保。

澜起科技公司标识、津逮® / Jintide®、HSDIMM® 以及 Mont-ICMT® 均为澜起科技在中国、美国和/或其他国家的注册商标。其他名称和品牌均为其各自公司所有。

* 本文档所称“澜起科技”系指澜起科技股份有限公司及其附属公司。

目录

图片目录

修订记录

文档版本	修订日期	说明
1.3	2019 年 8 月 19 日	更新“关于澜起科技”和“联系信息”
1.2	2019 年 4 月 10 日	措辞修订
1.1	2019 年 3 月 14 日	更新公司简介
1.0	2018 年 9 月 13 日	正式版第一版

术语表

术语缩写	定义
BIOS	Basic Input and Output System，基本输入输出系统
BMC	Base Management Controller，基板管理控制器
DSC	Dynamic Security Check，动态安全监控
ME	Management Engine，管理引擎
RCP	Reconfigurable Computing Processor，可重构处理器

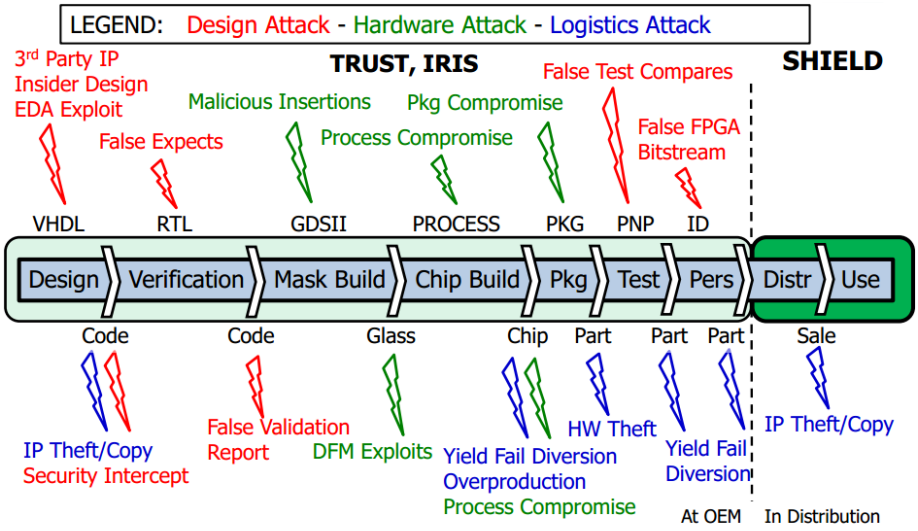
1 处理器硬件安全的重要性

1.1 信息系统安全需从软件扩展到硬件

计算机是信息系统存储和处理的重要工具，而中央处理器是整个计算机的核心，处理器的安全是信息系统安全的基础。当前处理器芯片电路规模庞大（约 10 亿-200 亿颗晶体管），芯片设计中涉及到的环节众多，流程复杂，不管是恶意植入的硬件木马或后门，还是设计人员的疏忽所带来的硬件漏洞或者前门，都会带来潜在的系统安全风险，比如运行非法的硬件指令，窃取关键数据，破坏系统正常运行等。DARPA（Defense Advanced Research Projects Agency，美国国防高级研究计划局）的相关报告总结了芯片在从设计到使用各个环节中面临的各种安全威胁，如图 1 所示。

这样的硬件安全威胁在处理器硬件整个生命周期的各个环节都有可能存在，而且往往都会潜伏很长的时间，直到被一些触发条件激活（如特定的数据或事件等）。而激活前一直存在于处理器硬件系统中，极难被发现。

图 1. 芯片生命周期中的威胁



硬件漏洞的物理本质使得它的潜在风险远高于软件漏洞。软件漏洞可以从受感染的操作系统中彻底清除，而存在硬件漏洞的计算机，仅靠软件的手段无法单独检测到，要想修复硬件的漏洞是非常困难的，更换硬件又耗费人力物力和财力。因此，信息系统安全的边界需要从软件扩展到硬件系统。

1.2 处理器硬件安全面临的风险

1.2.1 硬件木马入侵

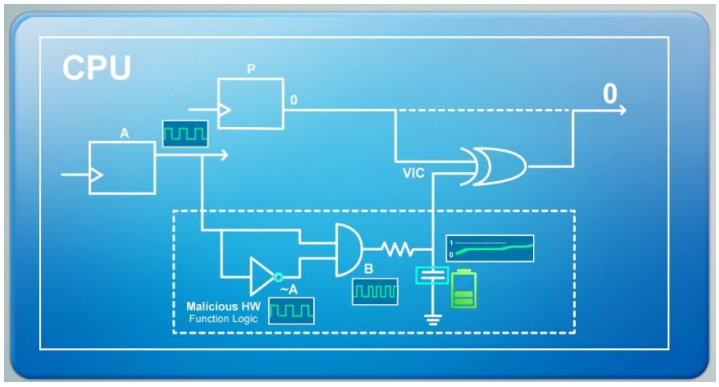
处理器硬件木马是一段为实现恶意行为而设计的电路，它能够在特定的触发激活条件下实现破坏性功能或越权读取处理器芯片内部信息。图 2 即是一个处理器木马的电路设计，虚线部分数十个晶体管的电路是一个动态恶意木马，如果软件频繁的翻转普通用户可以接触的寄存器 A（触发条件），将会让一个寄生电容充电，这个电容可以让特权控制寄存器 P 的输出翻转；那么软件就可以访问高特权等级

才能够访问的内容。而如果停止让 A 进行翻转，该电容由于漏电而降到低电压，此时木马带来的影响就完全消失。

类似的木马电路在很多文献中有讨论，这类电路一旦被触发，可能为系统带来巨大的安全风险，包括：

- 信息泄露：泄露处理器芯片中重要的信息和数据
- 数据篡改：监听并修改芯片内部的数据
- 影响系统正常服务：修改芯片内部电路的结构，导致处理器运行故障或停机等

图 2. 处理器木马电路



1.2.2 主流处理器熔断和幽灵漏洞

Intel 是知名的 x86 处理器厂商，其处理器先进的制造工艺、优异的性能、成熟的行业生态，在其专业领域内几乎难有竞争对手。但是，对于这样一个应用广泛，市场体量巨大的处理器，一旦发生硬件漏洞，影响也是巨大的。2017 年，随着 Intel ME 安全管理漏洞的公布，随后熔断、幽灵等漏洞的爆出，让处理器的安全再次成为一个备受关注的话题。

2018 年 1 月，媒体披露了熔断和幽灵漏洞，该漏洞已经在处理器中存在了 20 年，利用的使处现代处理器性能提高的一些基本技术，包括高速缓存、乱序和推测执行。尤其是幽灵，其影响范围几乎涵盖市场上所有微处理器厂商。随后，Intel 发布了补丁程序，紧接着又通知其用户停止部署该补丁程序，因为该补丁可能会影响系统稳定性，对性能和能耗有潜在影响。随着时间的发展，业界的共识是：要彻底解决该漏洞需要重新设计并更换处理器。这个事件说明，处理器硬件设计中的漏洞一直存在，将来也会存在于系统中，这些漏洞无法依靠软件来解决，软件只能增加漏洞被利用的难度，彻底解决需要处理器架构颠覆性革新。

1.2.3 Intel 公布第三大漏洞 – L1TF 侧信道攻击

2018 年 8 月 16 日，英特尔公布了新漏洞“预兆（Foreshadow）”，这是英特尔芯片继“熔断”和“幽灵”漏洞之后的第三个重要漏洞。这个漏洞可能让攻击者窃取存储在计算机或第三方云上的信息，漏洞由一些研究人员发现并于 1 月向英特尔报告，Intel 将其命名为 L1TF（L1 Terminal Fault，L1 终端故障）。随后，研究人员又发现了两个被叫做“Foreshadow-NG”的类似变体，会攻击代码、操作系统、管理程序软件以及其它微处理器。这三种应用都是与预测执行侧信道缓存计时相关的漏洞，它们的目标是访问 L1 数据高速缓存。对于这种侧信道攻击，Intel 官方网站有如下解释：

计算机系统的物理操作可以从不同的视角观察（例如时间、功耗、甚至声音），侧信道是其中之一。这些因素的统计分析在某些情况下有可能被用于从被操纵的计算设备中收集敏感数据。这些侧信道分析方法不会损坏、修改或删除数据。

大多数现代 CPU 能够预测它们需要为特定进程执行什么代码，并提前运行，以便在需要这些结果之前准备好这些结果。这可以大大提升 CPU 的整体性能和效率，从而提升电脑或移动设备的速度和性能。CPU 有时候会把数据从一个内存位置转移到其它位置，供这些进程使用。虽然系统按照设计精确地运行，在某些特定情况下，这些数据也可能会被此类攻击所观察到。

（资料来源：https://www.intel.cn/content/www/cn/zh/architecture-and-technology/facts-about-side-channel-analysis-and-intel-products.html）

Intel 官方给出了对于 L1TF 漏洞的修复建议：

在系统更新后，我们预计那些运行非虚拟化操作系统的消费者和企业用户（包括大多数的数据中心和个人电脑）所面临的安全风险会降低……针对另外一部分市场——特别是运行传统虚拟技术的细分领域（主要在数据中心领域），我们建议客户或合作伙伴采取额外措施来保护其系统。这主要是为了在 IT 管理员或云服务商无法保证所有虚拟化操作系统都已安装必要更新时，应对并防护该种情况下可能出现的风险。这些措施可以包括启用特定管理程序内核调度功能，或在某些特定场景中选择不使用超线程功能。对于这些特定情况，某些特定负载上的性能或资源利用率可能会受到影响，并相应发生变化。

（资料来源：https://newsroom.intel.cn/news-releases/press-relase-2018-aug-16-02/?_ga=2.43014124.833634112.1536648365-1258383877.1534458801）

1.3 处理器硬件安全解决思路

熔断、幽灵和预兆等漏洞，让硬件安全问题从学术研究领域、或者小众领域走入主流大众的视野，从这个意义上来说，2018 年可谓硬件安全元年。如何解决这些硬件安全问题，也成为一个热点话题。

这和终端用户也是非常相关的。如何保证处理器和与用户息息相关的数据运行在一个安全的环境，尤其对一些安全要求较高的行业，是用户和芯片商不得不考虑的问题。此外，安全需要兼顾到性能，即安全应该支撑应用而不是阻塞应用。因此，在无法彻底放弃现代高速处理器架构的同时，如何能够兼顾它的安全？

回答这个问题之前需要先思考，对于一个处理器，**它的安全如何衡量？**这个问题对于不同的人群有着不同的答案。处理器设计到制造涉及计算机体系架构、微电子、软件等多门学科，不同的专家可能会有不同的考虑：计算机体系结构的专家说，需要从指令集体系架构设计上就要考虑安全性，能够被数学所证明；微电子专家说，需要从模型的建立到实现，保证能够一步一步确认其等价性；软件专家说需要简化的安全的软硬件接口。但是遗憾的是，目前的处理器市场形成的生态环境——X86 和 ARM 分别是通用市场和移动端市场主流的情况下，彻底推到重来不是一件容易的事情。

此外，不同于性能设计，安全是一个需要用户参与其中的设计指标。在熔断和幽灵出现之前，基本没有人——特别是计算机体系结构的专家，会相信处理器中存在如此严重的信息盗取的漏洞；漏洞事件爆出后，很多专家反思业界数十年来追求性能设计，而忽视安全的发展思路是否可取。而我们认为，由于安全对用户的重要价值，以信息泄漏、数据丢失、系统不可用为代表的安全威胁的存在情况，应该是对用户完全透明的。只有用户完全明白其风险，才能够更好地做好应对措施；不能够仅凭借设计者一家之言。

不过，处理器结构复杂、微电子设计环节繁多，普通的用户难以完全理解其所有细节而判定其是否安全。因此，亟需一种技术手段，能够打破这种技术壁垒，为用户提供一种能够验证的处理器硬件安全判定方法。

1.3.1 从行为出发的处理器硬件安全判定标准

为了找到这种方法，我们需要从处理器运行的本质说起。处理器虽然结构复杂，但是其本质是包括了输入输出单元，控制和运算逻辑，以及存储单元的电路。对于用户来说，其核心价值在于能够正确地处理信息，包括：

- 对输入的数据进行正确解读
- 对存储单元的数据进行正确的更新
- 根据输入和存储单元的数据输出正确的数据

其中“正确”表示如下含义：

- 处理器行为是有明确清晰的定义，该定义不应对信息处理造成威胁
- 处理器行为执行是稳定的，这种稳定性体现在：
  - 不同处理器个体
  - 不同的时间
  - 不同的与该行为无关的运算上下文（环境）

我们把上述处理器行为称为“**用户预期的处理器行为**”，以避免“**正确**”这个描述过于笼统。可以说，确保处理器的硬件安全性，就是检查其行为是否符合预期。

1.3.2 处理器安全检测中的四性两态

而为了实现这种检查，处理器需具备“四性两态”，其中四性是指：

- 可观测性：指处理器状态透明，可以观测。
- 可检测性：指处理器的状态转换和输入输出可以被记录，能进行检测。
- 可测试性：可测试性是指处理器行为具有确定性，可以重现。
- 可控制性：可控制性是指处理器的功能行为可以被控制，运行的范围可以被控制。

如果不具备这几点，处理器的安全性就无法检测。例如，如果缺乏可观测性，处理器存在不透明状态，相当于处理器中存在隐藏的、能够影响其行为的寄存器，意味着处理器行为规范并未全部开放给用户；如果不具备可检测性，使处理器的对外的输入输出得到记录，说明处理器输入输出的方法无法得到有

效验证；可测试性决定了处理器行为的稳定性和可复现问题的能力。如果处理器的部分非核心功能无法保证具备完善的上述功能，那么需要增加使能的可控制性，由用户决定是否打开相关的功能。

两态是指：

- 静态
- 动态

在处理器硬件安全检测思路上，有静态和动态两种。静态方法通常是指处理器正式运行信息处理之前进行的检测，包括上电前对固件等进行的检测，以及上电后进行的自检；还包括处理器出厂前进行的测试——厂商自行完成或者第三方机构来完成。静态检测容易执行，比起没有经过任何检测手段来说，对处理器安全进行了一定的验证。但从效果上来说，静态检测有如下几点不足：

1. 难以应对软硬件协同触发的威胁

有些硬件漏洞只有在真实的运行环境中才能够触发，例如被恶意代码触发，或者特定的数据触发。例如熔断和幽灵的漏洞利用，需要部分恶意的或者存在疏漏的代码泄漏信息，然后攻击者才能够通过侧信道手段获取该信息。

2. 覆盖面不足

静态的检测方法通常难以完整地覆盖被检测的区间。例如：如果处理器硬件漏洞由特定的指令触发，那么检测的空间是很容易遍历的，只需要覆盖所有指令即可；如果是由指令组合触发，或者由“指令+数据”的方式触发，那么需要检测的空间就会以指数级迅速增加，从而导致静态检测的方法的覆盖率接近于 0，有效性会大大降低。

3. 难以应对升级带来的威胁

微码是处理器固件的一种，其改变处理器行为的能力不亚于重造一块处理器。随着微码的更新，处理器可能在修复旧的漏洞的同时，引入新的漏洞甚至是“恶意硬件”。

而动态检测是指在处理器真实运行场景中进行的检测，可以很好地弥补静态检测方法的不足。

2 处理器动态安全监控框架简介

基于处理器动态检测的思路，清华大学微电子所凭借多年在芯片和安全行业的经验，提出了一种可用的处理器安全检测的方法，即处理器动态安全监控（DSC），它可以针对处理器硬件安全进行检测、分析和控制，增强处理器的硬件安全。

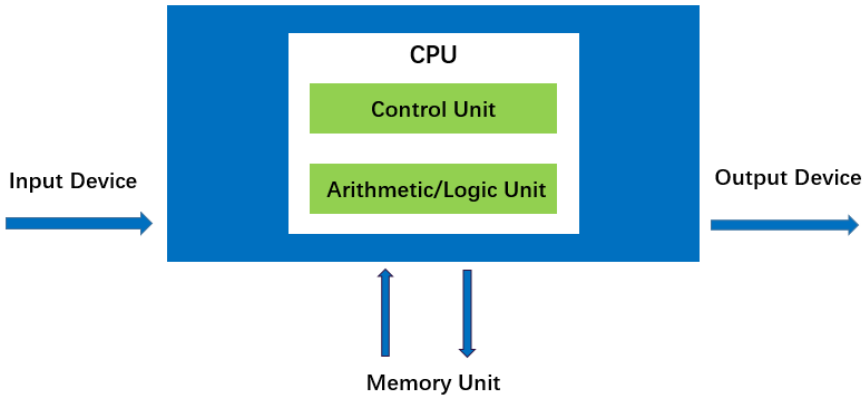
2.1 处理器动态安全监控原理

2.1.1 经典计算机模型和有限状态机 FSM

如图 3 所示，冯诺依曼体系结构的计算机主要包括 5 大部件，分别为：控制器、运算器、存储器和输入/输出设备。运算器和控制器在处理器内部，存储器即内存，输入/输出设备为 I/O 外设。近几十年来，虽然计算机的运行速度、方式有了很大的改变，但是其基本原理并没有变化。

现代计算机是使用有限状态机 FSM（Finite State Machine）作为计算模型的，即前一个时刻系统的状态和中间过程的 I/O 输入，完全决定了下一个时刻系统状态和 I/O 输出。

图 3. 现代计算机模型（冯.诺依曼）



依据该模型，只要能够完整的获取系统的状态和输入，结合系统的行为模型，即可准确的判断系统下一时刻的状态和输出。而反之，如果获取了系统当前状态和输入，结合系统下一时刻的输出和状态，可以判断系统在这个区间中的行为是否符合某一行为模型。这一原理应用于处理器，就是处理器安全检测方法的理论基础。

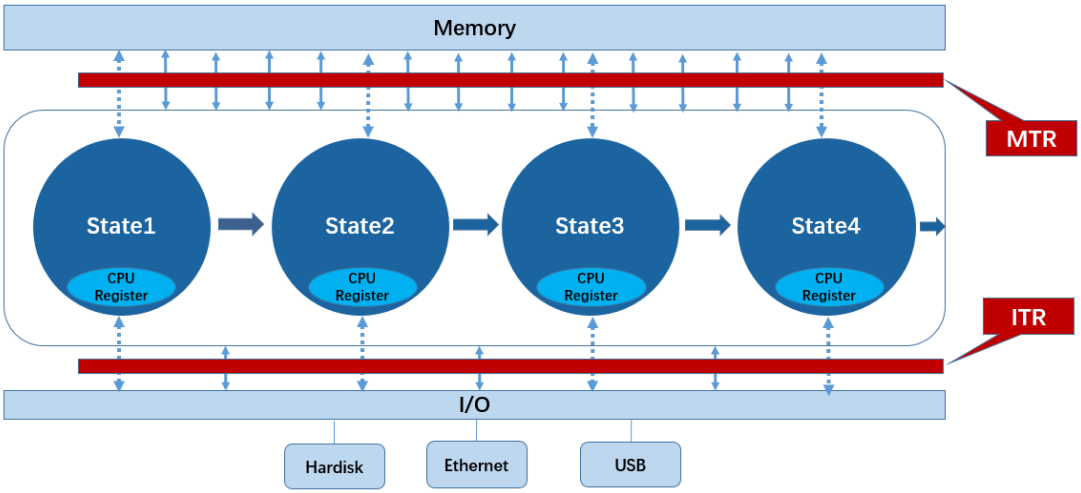
2.1.2 处理器数据采集

作为一个数字电路的一种，处理器在任何一个时刻的状态是确定的，决定处理器输出和下一时刻状态的因素为：

- 前一个时刻处理器的状态（包括处理器状态和内存）
- 前一个时刻到当前时刻所有的输入（I/O 中的输入）

如果处理器运行中存在非预期行为，产生一个非预期的状态，一定会在处理器内部寄存器、IO 中的输出表现出来，如：透过网络进行发送非预期数据。因此，要想对处理器行为进行安全检测，应该针对 Memory、I/O、CPU Register 进行综合分析。如图 4 所示，其中 ITR 和 MTR 都代表一种检测方法，ITR 代表 IO 数据跟踪和检测，MTR 代表 Memory 数据跟踪和检测。

图 4. 决定处理器状态的要素



有了处理器中采集到的状态和输入信息，是否可以唯一的判断处理器的非预期行为呢？这需要对采集数据的完备性有一定保证：

任何影响状态改变或者将来输出的状态（寄存器）都需要进行采集。

例如，处理器中的一个寄存器可能不会直接影响到处理器的输出，但是其数值可能会转移到其他寄存器中，从而影响到输出（例如寄存器批量保存在堆栈中）。如果一个处理器内核引脚的数据最终会影响到处理器的状态转移或者输出，那么这个引脚需要纳入到处理器 IO 采集的范围。因此通常处理器数据采集需要对处理器几乎全部的引脚进行管控和记录。

准确全面的处理器数据采集是进行安全分析的前提。

2.1.3 处理器安全分析

以上介绍了处理器安全检测需要分析的检测项，然而对于一个内部设计未知的处理器，如何进行分析来判断处理器是否安全，是影响最终检测结果的关键。

对于现代处理器来说，超标量、流水线、乱序执行等技术的引入导致了处理器对外虽然行为上和一个“老式”处理器——寄存器和体系结构定义一一对应、一条一条执行指令——相同，但是其内部结构已经大不相同，一个典型的例子是其寄存器数量可能由于寄存器重命名的存在而大大多于体系结构的定义。

在这种情况下，针对不同的安全分析需求，DSC 可以采集不同的数据。对于一般的应用，虽然处理器设计本身可能采用了寄存器重命名等技术，但是我们依旧可以采集出来某一时刻的指令集体系结构中定义的寄存器数值，就好像“处理器中真的仅仅存在静态分配的这些寄存器”一样，进而进行分析。

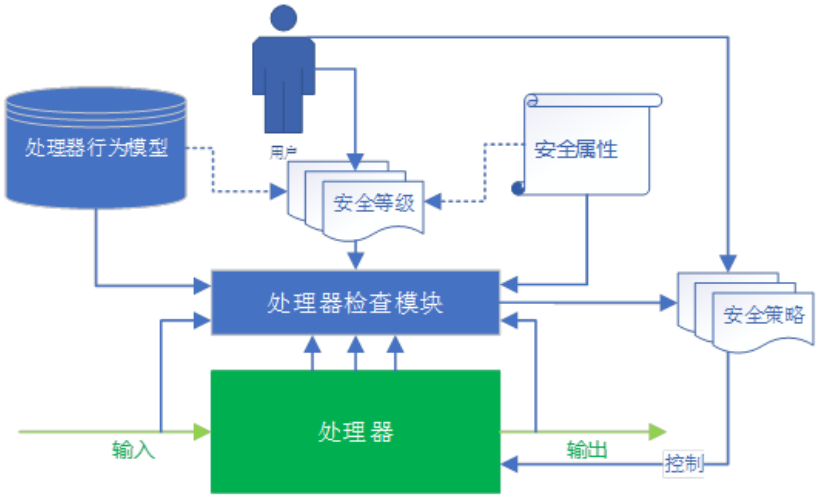
而在一些特殊的案例中，指令集体系结构的数据对于安全判定是不够的，可能需要部分微体系结构的数据支持，例如 Cache 的状态等。

图 5 展示了一个通用的处理器安全分析的框架。根据用户定义的不同的安全等级，一般被检查模块所采用的安全分析方法包括：

1. 处理器行为模型：利用黄金的处理器模型和采集到的处理器行为进行对比。这种模型可能包括处理器指令集模型，微架构模型，甚至更加精确的精确到时钟周期的模型等。

2. 安全属性或者规则：用户定义的一系列行为准则，以简化安全判定或者弥补处理器行为模型的不足。例如在侧信道攻击的案例中，由于处理器的设计缺陷，特定的代码片段会造成信息泄漏，这些可以被安全检查模块所捕获。

图 5. 通用处理器安全分析框架



2.1.4 实时在线的动态采样检测方法

如前文所讨论，动态检测是处理器真实运行场景中进行的检测，在处理器运行终生都可以进行检测，静态检测是指检测机制在处理器出厂前一段时间内运行，通过测试后检测机制移除。动态检测和静态检测都具有一定的安全性，如：

- 都可以一定程度上检验处理器行为规范的完备性
- 通过检测可以发现硬件威胁，并进行分析、管控

同时，动态安全检测还有如下优势：

- 覆盖范围更广。动态检测更加贴近真实应用场景，因为待测环境就是真实应用场景本身。因此动态检测的时间相比静态检测的有限测试时间来说更长，覆盖面也更广。
- 管控作用。可以有效管控真实场景中的异常行为，减小安全风险带来的损失。

然而，处理器的行为检测会不可避免地对正在运行的处理器产生影响，包括其状态、行为的采集，行为符合预期的判定等。特别是处理器作为运算行为密集的部件，其行为的判定本身复杂度都需要数倍

于其计算能力的运算器件才能够完成；要实现处理器全部行为的完全检测，其实现复杂度会很高，而且由此带来的性能影响、功耗代价均较为难以接受。

为了解决这一问题，需要在安全性和可实现性上采取平衡。为此，DSC 采用了基于采样的动态处理器检测方法。对于运行中的处理器，采样可以由第三方随机发起，采用较小的采样比例——例如采样率千分之一，对处理器运行过程和轨迹进行检测。

采样的方法也是有限制性的，比如会遗漏掉一些偶发的个体的硬件安全问题，例如制造缺陷引发的单一处理器中的硬件“漏洞”，在处理器中如果只产生单次的影响，那么检测出的概率就只有采样率。

但是，由于处理器是成批生产，微码的更新也是针对大量处理器，当处理器行为中存在恶意行为的时候，通常会有大量的恶意行为的产生。一个典型的例子是：熔断和幽灵漏洞的利用，需要反复利用处理器的预测执行才能够获得一个字节的数据，反复执行的目的是为了提高准确率。根据公开的代码和文献，通常要在 100 次以上的执行中才能够达到高于 99%的泄漏的数据的准确读取。在反复执行的场景中，采样的方法有着较高的概率，可以发现处理器中的硬件问题。

为了说明采样的有效性，可以用一个简单的数学模型来说明：假设某一型号的处理器部署了基于采样的动态行为安全检测技术，采样率是 r=1/1000，意味着处理器的行为（以采样的段为基本颗粒度）中有 1/1000 接受了检测；如果一项非预期行为发生次数为 N，假设数次发生之间间隔足够长（大于以此检测的区间长度），数次发生之间是随机的，且相互独立。那么，该种类的行为被随机采样的动态检测方案查出的概率不低于：

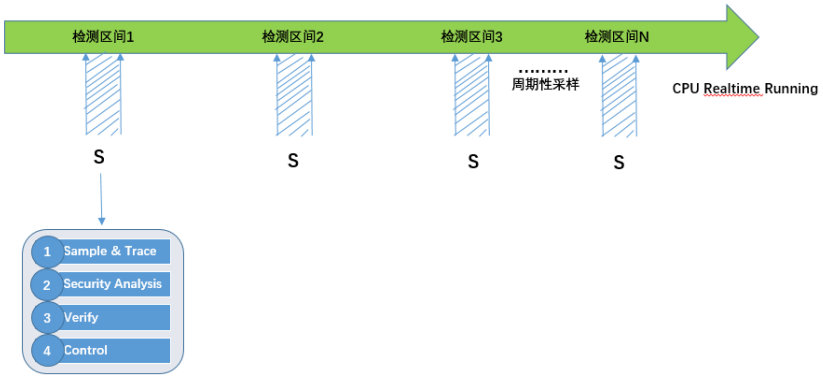
$$P = 1-(1-r)^N$$

计算表明，如果恶意行为发生一次 N=1，那么检查出来的概率即为 r=0.001；当恶意行为发生的次数 N=1000 时候，检出概率 P=0.63；如果恶意行为发生的次数 N=10000，检出率 P=0.99996，已经非常接近 1。

在熔断和幽灵的案例中，漏洞被反复的利用才能够读出泄漏的数据，虽然和上述模型有一定区别，但是如果合适地确定采样长度和频度，漏洞利用的行为几乎一定会被发现。

一个基于周期采样的处理器动态检查方案如图 6 所示，可以将处理器运行过程划分为不同的区间，从而对部分区间进行检测，包括采样开始和数据记录开始，分析和验证等。

图 6. 动态周期采样模型



以上介绍了处理器动态安全监控的几个关键点，包括安全检测项、安全分析、动态采样等，这是处理器动态安全检测技术实现的基础。下一章描述处理器动态安全检测的流程。

2.2 处理器动态安全检测流程

如图 7 所示，处理器动态安全检测流程（Dynamic CPU Check）一次检测周期共包括 4 个步骤，分别是：采样和数据获取、数据分析、数据验证以及安全控制。

图 7. 处理器动态安全检测流程

- 采样和数据获取

数据采样是检测被触发之后的第一个步骤，它会根据采样周期的长短，确定两个检测点，在这两个检测点分别提取处理器的状态（内存和处理器寄存器值），并记录检测区间的 IO 事件。这些数据集会进行打包，交给下一步进行分析。

在数据采样阶段，需要解决的一个问题是：数据中的无关信息去除，以及数据时间的确定。

对于不同层次的数据分析过程，所需要的“有效”数据是不相同的。例如，对于一般的处理器指令集架构层面的分析，存储器的访问可以简化为检测区间内对存储器的第一次读取和最后一次写入（如果存在写入的话）的信息；但是对于微架构模型，访问需要细化到缓存，因此每一次的读取和写入都是需要记录的。在保存有效数据的基础上，去除掉其他的数据，可以节省存储空间，提高后续数据分析和处理的效率。

此外，对于保存的记录数据，特别是 IO 和存储器的访问数据，需要和决定处理器行为的主要因素：处理器运行的指令流对应，并解决后续分析过程中由于 IO 数据和指令流在时序上的关系造成的处理器行为的不确定性。

- 数据分析

上一个步骤中采集到的两个数据集会在这个环节被分析。其主要的思路是通过分析处理器在给定的外部输入和起始状态的情况下，预测其运行流程和输出信息。

一种常用的分析方法是，根据起始检测点的状态，经过一个理想的处理器（符合该处理器指令集或芯片手册定义的行为模型）运行，输入同样的 IO 事件的输入激励，得到一个理论上安全的检测终点状态，以及在运行过程中的安全的输出。

上述分析方法中的行为模型可以来自于模拟器，或者用户自定义的模型；该模型的精度决定了安全性。例如，漏洞利用中采用的侧信道攻击方法，采用一般的指令集模型是无法分析的，原因是指令集模型除了有限的缓存操作指令，缓存本身是对指令透明的，数据在缓存中的状态不体现在模型中。所以要检测侧信道攻击，需要借用相对精确的缓存模型。

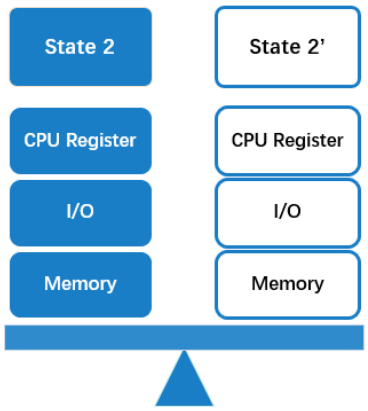
分析方法也可以借用约束方法，即通过某些特征来确定恶意行为，例如检测存储器翻转攻击可以使用“连续向同一地址频繁交替写入 0 和 1”这一特征来检测。

- 数据验证

数据验证是处理器动态监控技术的核心环节。在准确获得处理器指令执行结果的基础上，对理想的处理器状态数据集进行对齐、解析、翻译和筛选。对处理器行为的检测结果可以分为匹配和异常两种结果；根据不同的异常类型，或用户定义的安全策略，可以通过控制系统，进行下一步处理。

数据的验证中，对于异常结果的判定是核心。例如，处理器的推测执行通常会引入对存储器的提前访问——这种访问可能是程序运行流程不需要的，如果是在推测错误的情况下——在软件分析中，这些访问是不可见的，由于推测执行本身是一种硬件机制，对软件完全透明；但是在处理器安全检查中，这些数据作为对外的访问是处理器的输入，会被记录。额外的访问可能带来信息的泄漏，通过判定额外访问的目标地址的权限等级可以初步筛查熔断攻击，而对比预测执行路径和正常路径，可以获取幽灵攻击的部分特征。

图 8. 安全数据验证



- 安全管控

  安全管控主要分为两大类别：

  - **威胁管控**：是当发现处理器行为异常时，可以采取一些措施来减少安全风险，比如报警，并保存现场以便进一步分析，严重安全问题甚至可以关闭服务器等。

    此外，针对一些软硬件联合的威胁，其依赖于软件恶意代码对于硬件漏洞的利用，当安全检测发现相关威胁后，可以提取软件特征码，软件安全手段，例如病毒防火墙等可以将其识别为病毒从而阻止其在系统中运行和造成进一步的危害。

  - **风险管控**：得益于处理器数据采集框架中对于处理器输入和输出数据的全面记录和采集机制，对于处理器中或者处理器外的子系统、外设，如果有对处理器运算系统的访问和修改行为，上述机制可以有效发现，根据用户的设定进行预警；如果采集机制的硬件可以切断部分数据通路，可以从物理上阻断子系统对于运算环境的侵害。

    风险管控可以应用的一个实例是 Intel ME 和 AMD PSP，这些子系统通常存在于主板芯片组，提供带外的管理接口方便对系统的管理访问。但是随着近年来数个严重漏洞被曝光，其安全性受到普遍质疑，特别是部分漏洞可以允许黑客远程打开这些子系统更是让从未使能过这些功能的计算机暴露在攻击威胁之下。

如欲了解更多技术和产品信息，敬请联系：globalsales@montage-tech.com

关于澜起科技

作为业界领先的集成电路设计公司之一，澜起科技致力于为云计算和人工智能领域提供高性能芯片解决方案。公司在内存接口芯片市场深耕十余年，先后推出了 DDR2、DDR3、DDR4 系列高速、大容量内存缓冲解决方案，以满足云计算数据中心对数据速率和容量日益增长的需求。澜起科技发明的 DDR4 全缓冲“1+9”架构被 JEDEC 采纳为国际标准，其相关产品已成功进入全球主流内存、服务器和云计算领域，占据国际市场的主要份额。

2016 年以来，澜起科技与清华大学、英特尔鼎力合作，研发出津逮®系列 CPU。基于津逮®CPU 及澜起科技的安全内存模组而搭建的津逮®服务器平台，实现了芯片级实时安全监控功能，为云计算数据中心提供更为安全、可靠的运算平台。此平台还融合了先进的异构计算与互联技术，可为大数据及人工智能时代的各种应用提供强大的综合数据处理及计算力支撑。

澜起科技成立于 2004 年，总部设在上海并在昆山、澳门、美国硅谷和韩国首尔设有分支机构。

联系信息

总部
地址：上海市宜山路 900 号科技大楼 A 栋 6 楼，邮编：200233
电话：+86-21-5467-9038
传真：+86-21-5426-3132

销售联系
邮箱：globalsales@montage-tech.com