分类号 TP309　　密　级

U D C　　编　号 10486

武漢大學

博 士 学 位 论 文

（学术学位）

安全计算机体系结构关键技术研究

研 究 生 姓 名：张　三　三

学　　号：2023100000000

校内导师姓名、职称：李　四　教　授

学科、专业名称：网络空间安全

研 究 方 向：信　息　安　全

二〇二四年四月

Research on Key Techniques for Security Computer Architecture

ZHANG San

论文创新点

本文的创新点主要有以下几个方面：

- ZeRØ[1]提出了独特的内存指令和新颖的元数据编码方案来保护代码和数据指针，仅仅只需要微小的微架构变化。ZeRØ 在 SPEC CPU2017 基准上的性能开销为零，VLSI 测量显示了低功率和面积开销。
- No-FAT[2]将内存分配大小（例如 malloc 大小）作为一个架构特征，来克服传统内存安全方法的许多棘手问题，例如与不安全软件的兼容性和显著的性能下降。No-FAT 在 SPEC CPU2017 基准测试中产生了 8% 的开销，VLSI 测量显示了低功率和面积开销。

目 录

插图索引

表格索引

中英文缩略语对照表

BHB	Branch History Buffer	分支历史缓冲区
BHI	Branch History Injection	分支历史注入
BPU	Branch Prediction Unit	分支预测单元
BRB	Branch Retention Buffer	分支保留缓冲区
BTB	Branch Target Buffer	分支目标缓冲区
CPU	Central Processing Unit	中央处理器
CVE	Common Vulnerabilities and Exposures	通用漏洞披露

摘　要

Intel CET（Control-Flow Enforcement Technology）是 Intel 新推出的一项新的硬件级对策，其主要目的是防止攻击者通过 ROP/JOP/COP 等攻击手段来劫持控制流。对于 ROP 攻击的防护，其基本思想与影子栈（shadow stack）类似，即由操作系统在内存中复制一份程序的内存栈或者是仅仅保留控制流跳转地址。然后，这个影子栈无法由正常的 store、load 指令进行控制，只能通过专门的指令来进行控制。因此，即使攻击者覆盖了软件栈上的返回地址，但是由于影子栈中仍然保存着原始的跳转地址，因此检查失败后会通过抛出异常来终止程序的执行。对于 JOP/COP 攻击的防护，Intel 提出一种叫做 IBT（Indirect Branch Tracking）的技术，其基本思想是通过编译器在合理的间接跳转中　新的指令做标记，然后程序执行时会检查下一条指令是否为新添加的指令（endbr），如果不是则会抛出 #CP 异常。

ARM PAC（Pointer Authentication）是 ARMv8.3 引入的一项新的硬件级对策，其主要通过对指针进行鉴权来防止攻击者通过 ROP/JOP/COP 等攻击手段来劫持控制流。这种防护方法的基本原理是，利用 64 位地址空间中暂时空闲的高 16 位来存放指针的鉴权结果（MAC 码），然后在每次指针使用前，都会对指针进行鉴权，如果鉴权失败，则会抛出异常来终止程序的执行。

关键词：体系结构，硬件安全

ABSTRACT

This is the abstract in English.

Key words: Computer Architecture, Hardware Security

1 绪论

2018 年 1 月 3 日，Google Project Zero 团队的 Jann Horn 等安全研究者公开了两组处理器漏洞，即 Meltdown 漏洞[3]和 Spectre 漏洞[4]。其中 Meltdown 对应的漏洞编号为 CVE-2017-5754（流氓数据缓存加载），这种攻击“熔化”了由硬件来实现的安全边界，允许用户级别的应用程序“越界”访问系统级的内存，从而造成数据泄露。而 Spectre 对应的漏洞编号为 CVE-2017-5753（边界检查绕过）和 CVE-2017-5715（分支目标注入），利用分支预测的错误推测，让攻击者有能力触发受害者访问特定的敏感数据，并通过隐蔽信道泄露信息。

不管是 Meltdown 攻击还是 Spectre 攻击，其本质都是利用现代处理器的优化策略，而这些优化策略在发生错误时，并不会造成架构层面（或者说 ISA 层面）上数据的变化，但是由于现代处理器引入了 cache、TLB、缓冲区等微架构元素，攻击者仍然有能力通过观测微架构的变化来获取信息。而这些利用乱序执行和分支预测等机制，触发处理器执行错误的指令，从而造成秘密数据泄露的方式统称为瞬态执行攻击（Transient Execution Attack）[5]。

2 背景知识

2018 年 1 月 3 日，Google Project Zero 团队的 Jann Horn 等安全研究者公开了两组处理器漏洞，即 Meltdown 漏洞[3]和 Spectre 漏洞[4]。其中 Meltdown 对应的漏洞编号为 CVE-2017-5754（流氓数据缓存加载），这种攻击“熔化”了由硬件来实现的安全边界，允许用户级别的应用程序“越界”访问系统级的内存，从而造成数据泄露。而 Spectre 对应的漏洞编号为 CVE-2017-5753（边界检查绕过）和 CVE-2017-5715（分支目标注入），利用分支预测的错误推测，让攻击者有能力触发受害者访问特定的敏感数据，并通过隐蔽信道泄露信息。

表 2.1: 中英文缩略语对照表

BHB	Branch History Buffer	分支历史缓冲区
BHI	Branch History Injection	分支历史注入
BPU	Branch Prediction Unit	分支预测单元
BRB	Branch Retention Buffer	分支保留缓冲区
BTB	Branch Target Buffer	分支目标缓冲区
CPU	Central Processing Unit	中央处理器
CVE	Common Vulnerabilities and Exposures	通用漏洞披露

不管是 Meltdown 攻击还是 Spectre 攻击，其本质都是利用现代处理器的优化策略，而这些优化策略在发生错误时，并不会造成架构层面（或者说 ISA 层面）上数据的变化，但是由于现代处理器引入了 cache、TLB、缓冲区等微架构元素，攻击者仍然有能力通过观测微架构的变化来获取信息。而这些利用乱序执行和分支预测等机制，触发处理器执行错误的指令，从而造成秘密数据泄露的方式统称为瞬态执行攻击（Transient Execution Attack）[5]。

3 内存安全问题研究

从计算机体系结构的角度来说，内存安全违规（memory safety violation）一般分为两大类：

（1）时间违规（temporal violation）：典型代表为 Use-After-Free（UAF）漏洞，即在释放内存后，再次对该内存进行访问。

（2）空间违规（spatial violation）：典型代表为栈溢出（stack overflow）漏洞，即在栈上分配的内存空间不足以存放当前的数据。

而面向返回编程（Return-Oriented Programming，ROP）、面向跳转编程（Jump-Oriented Programming，JOP）和面向调用编程（Call-Oriented Programming，COP）都是基于空间违规的攻击手段，本文将从攻击基本原理、现有芯片级对策和思考三个方面来介绍 ROP/JOP/COP。

而 ROP 攻击产生的一大原因是因为，现代操作系统为了防止攻击者在栈上发起代码注入漏洞，采用了 $W \oplus X$ 的内存保护机制，即栈上的内存空间只能执行，不能写入。因此，攻击者在栈上发起代码注入漏洞时，只能通过覆盖返回地址来控制程序的执行流程，而这种攻击手段就是 ROP。

4 硬件级安全机制研究

4.1 Intel CET 安全机制分析

Intel CET（Control-Flow Enforcement Technology）是 Intel 新推出的一项新的硬件级对策，其主要目的是防止攻击者通过 ROP/JOP/COP 等攻击手段来劫持控制流。对于 ROP 攻击的防护，其基本思想与影子栈（shadow stack）类似，即由操作系统在内存中复制一份程序的内存栈或者是仅仅保留控制流跳转地址。然后，这个影子栈无法由正常的 store、load 指令进行控制，只能通过专门的指令来进行控制。因此，即使攻击者覆盖了软件栈上的返回地址，但是由于影子栈中仍然保存着原始的跳转地址，因此检查失败后会通过抛出异常来终止程序的执行。对于 JOP/COP 攻击的防护，Intel 提出一种叫做 IBT（Indirect Branch Tracking）的技术，其基本思想是通过编译器在合理的间接跳转中　新的指令做标记，然后程序执行时会检查下一条指令是否为新添加的指令（endbr），如果不是则会抛出 #CP 异常。

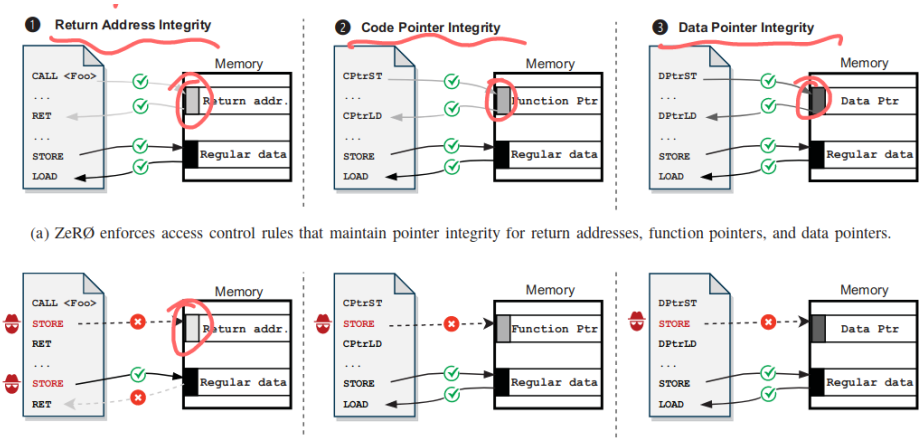
4.2 ARM PAC 安全机制分析

ARM PAC（Pointer Authentication）是 ARMv8.3 引入的一项新的硬件级对策，其主要通过对指针进行鉴权来防止攻击者通过 ROP/JOP/COP 等攻击手段来劫持控制流。这种防护方法的基本原理是，利用 64 位地址空间中暂时空闲的高 16 位来存放指针的鉴权结果（MAC 码），然后在每次指针使用前，都会对指针进行鉴权，如果鉴权失败，则会抛出异常来终止程序的执行。

5 安全架构设计关键技术研究

5.1 ZeRØ 安全架构设计

ZeRØ[1]提出了独特的内存指令和新颖的元数据编码方案来保护代码和数据指针，仅仅只需要微小的微架构变化。ZeRØ 在 SPEC CPU2017 基准上的性能开销为零，VLSI 测量显示了低功率和面积开销。



(a) ZeRØ enforces access control rules that maintain pointer integrity for return addresses, function pointers, and data pointers.

(b) ZeRØ mitigates code-reuse attacks through its use of access control preventing regular STOREs from corrupting pointers.

Fig. 1: A high level overview of how ZeRØ's pointer integrity mechanism works.

图 5.1: ZeRØ 设计框架

5.2 No-FAT 安全架构设计

No-FAT[2]将内存分配大小（例如 malloc 大小）作为一个架构特征，来克服传统内存安全方法的许多棘手问题，例如与不安全软件的兼容性和显著的性能下降。No-FAT 在 SPEC CPU2017 基准测试中产生了 8% 的开销，VLSI 测量显示了低功率和面积开销。

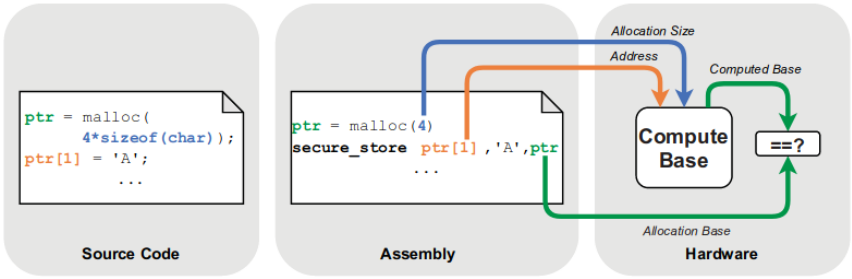


Fig. 1: A high level overview of how No-FAT makes allocation size an architectural feature.

图 5.2: No-FAT 设计框架

6　总结与展望

总结本文的工作，主要包括以下几个方面。

参考文献

[1] Ziad M T I, Arroyo M A, Manzhosov E, et al. ZeRØ: Zero-overhead resilient operation under pointer integrity attacks[C]//2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2021: 999-1012.

[2] Ziad M T I, Arroyo M A, Manzhosov E, et al. No-FAT: Architectural support for low overhead memory safety checks[C]//2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2021: 916-929.

[3] Lipp M, Schwarz M, Gruss D, et al. Meltdown: Reading Kernel Memory from User Space[C]//27th USENIX Security Symposium (USENIX Security 18). USENIX Association, 2018: 973-990.

[4] Kocher P, Horn J, Fogh A, et al. Spectre Attacks: Exploiting Speculative Execution[C]//2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019: 1-19.

[5] Xiong W, Szefer J. Survey of Transient Execution Attacks and Their Mitigations[J]. ACM Computing Surveys, 2021, 54(3): 1-36.

攻博期间发表的科研成果目录

[1] 2024 IEEE International Symposium on XX，2024，第一作者

[2] IEEE Transactions on XX，2023，第一作者

[3] XX 学报，2022，第一作者

[4] ACM Transactions on XX，2021，第一作者

致　谢