

# Meltdown: Reading Kernel Memory from User Space

Moritz Lipp<sup>1</sup>, Michael Schwarz<sup>1</sup>, Daniel Gruss<sup>1</sup>, Thomas Prescher<sup>2</sup>,  
Werner Haas<sup>2</sup>, Anders Fogh<sup>3</sup>, Jann Horn<sup>4</sup>, Stefan Mangard<sup>1</sup>,  
Paul Kocher<sup>5</sup>, Daniel Genkin<sup>6,9</sup>, Yuval Yarom<sup>7</sup>, Mike Hamburg<sup>8</sup>

<sup>1</sup>*Graz University of Technology*, <sup>2</sup>*Cyberus Technology GmbH*,

<sup>3</sup>*G-Data Advanced Analytics*, <sup>4</sup>*Google Project Zero*,

<sup>5</sup>*Independent (www.paulkocher.com)*, <sup>6</sup>*University of Michigan*,

<sup>7</sup>*University of Adelaide & Data61*, <sup>8</sup>*Rambus, Cryptography Research Division*

## Abstract

The security of computer systems fundamentally relies on memory isolation, e.g., kernel address ranges are marked as non-accessible and are protected from user access. In this paper, we present Meltdown. Meltdown exploits side effects of out-of-order execution on modern processors to read arbitrary kernel-memory locations including personal data and passwords. Out-of-order execution is an indispensable performance feature and present in a wide range of modern processors. The attack is independent of the operating system, and it does not rely on any software vulnerabilities. Meltdown breaks all security guarantees provided by address space isolation as well as paravirtualized environments and, thus, every security mechanism building upon this foundation. On affected systems, Meltdown enables an adversary to read memory of other processes or virtual machines in the cloud without any permissions or privileges, affecting millions of customers and virtually every user of a personal computer. We show that the KAISER defense mechanism for KASLR has the important (but inadvertent) side effect of impeding Meltdown. We stress that KAISER must be deployed immediately to prevent large-scale exploitation of this severe information leakage.

## 1 Introduction

A central security feature of today’s operating systems is memory isolation. Operating systems ensure that user programs cannot access each other’s memory or kernel memory. This isolation is a cornerstone of our computing environments and allows running multiple applications at the same time on personal devices or executing processes of multiple users on a single machine in the cloud.

On modern processors, the isolation between the kernel and user processes is typically realized by a supervi-

sor bit of the processor that defines whether a memory page of the kernel can be accessed or not. The basic idea is that this bit can only be set when entering kernel code and it is cleared when switching to user processes. This hardware feature allows operating systems to map the kernel into the address space of every process and to have very efficient transitions from the user process to the kernel, e.g., for interrupt handling. Consequently, in practice, there is no change of the memory mapping when switching from a user process to the kernel.

In this work, we present Meltdown<sup>10</sup>. Meltdown is a novel attack that allows overcoming memory isolation completely by providing a simple way for any user process to read the entire kernel memory of the machine it executes on, including all physical memory mapped in the kernel region. Meltdown does not exploit any software vulnerability, *i.e.*, it works on all major operating systems. Instead, Meltdown exploits side-channel information available on most modern processors, e.g., modern Intel microarchitectures since 2010 and potentially on other CPUs of other vendors.

While side-channel attacks typically require very specific knowledge about the target application and are tailored to only leak information about its secrets, Meltdown allows an adversary who can run code on the vulnerable processor to obtain a dump of the entire kernel address space, including any mapped physical memory. The root cause of the simplicity and strength of Meltdown are side effects caused by *out-of-order execution*.

Out-of-order execution is an important performance feature of today’s processors in order to overcome latencies of busy execution units, e.g., a memory fetch unit needs to wait for data arrival from memory. Instead of stalling the execution, modern processors run operations

<sup>9</sup>Work was partially done while the author was affiliated to University of Pennsylvania and University of Maryland.

<sup>10</sup>Using the practice of responsible disclosure, disjoint groups of authors of this paper provided preliminary versions of our results to partially overlapping groups of CPU vendors and other affected companies. In coordination with industry, the authors participated in an embargo of the results. Meltdown is documented under CVE-2017-5754.