



MELTDOWN: READING FOR BEGINNERS

2

8



provided by *instruction*

1 Check access

2 Read into register

1 *Magic*

up to 100 orders by execution

2

Read into register

1

Magic

1

Check access

Reordering is not a problem because the CPU will ensure that  is only seen *iff*  succeeds.

Unless  is able to hide the secret in such a way that the attacker can find it later.

The re-ordering on the right happens, when the “forbidden data” is already cached (because cache access is so fast).

In our burger example:

- 1. Customer orders a burger**
- 2. Customer gets his burger**
- 3. Customer has not enough money**
- 4. Customer runs away with burger**

MELTDOWN: READING FORBIDDEN DATA



μOPs ordered by *instruction*

1 Check access

2 Read into register

1 *Magic*



μOPs ordered by *execution*


2 Read into register

1 *Magic*

1 Check access

The re-ordering on the right happens, when the “forbidden data” is already cached (because cache access is so fast).

Reordering is not a problem because that  is only seen *iff*  succeeds

Unless  is able to hide the secret, an attacker can find it later.

In our burger example:

1. Customer orders a burger
2. Customer gets his burger
3. Customer has not enough money
4. Customer runs away with burger

MELTDOWN



For Meltdown two actors are needed

The a **spy** and a **collector**.

110011010 The **spy** will “steal” the secret and stash it away.
010111010
111100100
000101101 The CPU will kill him for accessing the secret
100110010 information.
Spy

110011010
010111010
111100100
000101101
100110010

The **collector** will find the stashed away secret.

Collector