# MELTDOWN & SPECTRE

Juarez Aguilar Osmar

# Puntos a abordar

- Introducción
- Breve descripción de Meltdown y Spectre
- ¿De dónde surgen estas vulnerabilidades que explota Meltdown y Spectre?
- Conceptos importantes
- Descripción del funcionamiento de estas vulnerabilidades

# Introducción

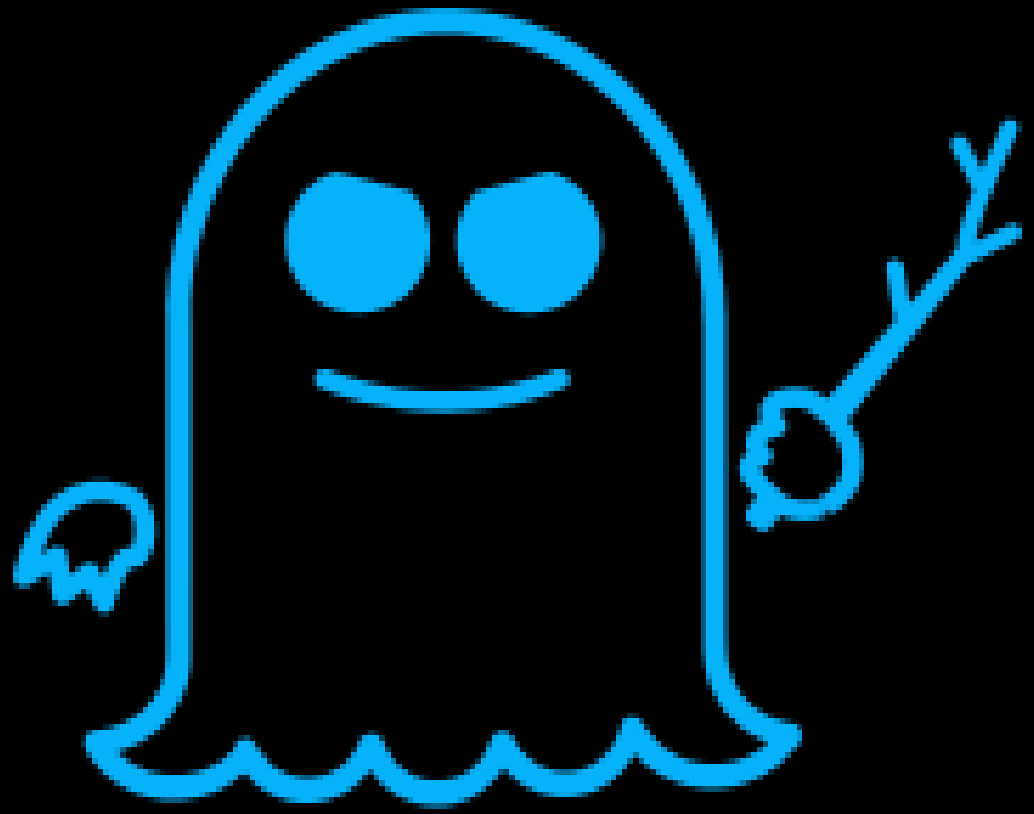
¿QUIÉN Y CUANDO SE DESCUBRIERON ESTAS VULNERABILIDADES?


- Estos ataques de seguridad se encontraron por un grupo de ingenieros de Project Zero en el verano del 2017.
- Project Zero es un Proyecto desarrollado por Google para encontrar vulnerabilidades en diferentes sistemas e informar de estos fallos para tomar medidas en el asunto
- Su objetivo es crear una informática segura con cero errores y vulnerabilidades.

# MELTDOWN

# MELTDOWN

- Meltdown rompe todas las garantías de seguridad proporcionadas por las capacidades de aislamiento de memoria de la CPU.
- Meltdown permite que un proceso sin privilegios lea los datos asignados en el espacio de direcciones del kernel, incluida toda la memoria física.
- Afecta a procesadores Intel y unos cuantos ARM.
- Es un rootkit de tipo kernel-mode.
- Usa ataques de canal lateral.
- Se han creado medidas de mitigación contra Meltdown



SPECTRE

# SPECTRE

- Ataque que se basa en vulnerabilidades de la arquitectura permitiendo a los programas alojados en el sistema operativo del usuario acceder a una dirección arbitraria del espacio de memoria de un programa.
- Spectre afecta a todas las arquitecturas de procesadores, tanto a Intel como AMD y ARM.
- Explota mas vulnerabilidades que Meltdown.
- Usa ataques sincronizados de canal lateral.
- Hasta el momento no hay una solución definitiva para librarse de Spectre

# Todo empezó aproximadamente hace 20 años

La arquitectura de los procesadores a partir de 1993 (Intel Pentium)

- Empezó con la integración de la APIC a la microarquitectura del procesador.
- El APIC es el Controlador de Interrupciones Programables Avanzado.
- El APIC se encarga de administrar los eventos de interrupción enviados al procesador.
- Al principio el APIC era un circuito separado.

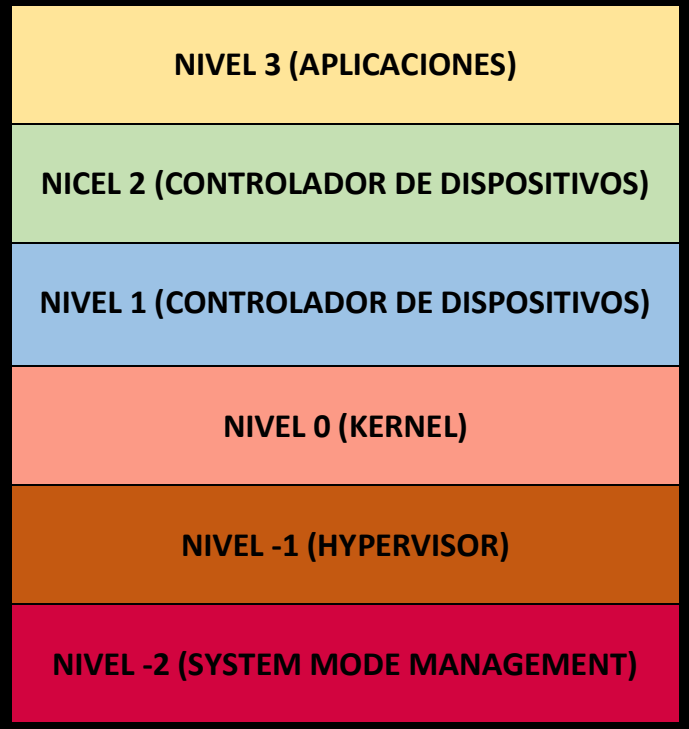
# ¿Por qué esto supone el principio de un fallo en la arquitectura de un procesador?

- Para permitir un acceso rápido y flexibilidad en la administración del APIC, los registros del chip se asignaron a la memoria del procesador en la región de 4KB entre 0xFEE00000 y 0xFEE01000.
- Esto causó inadvertidamente conflictos con el software el cual usaba este rango de memoria para otros fines.

# El Segundo fallo en la arquitectura

- En el siguiente procesador de Intel (Pentium Pro 1995) se extendió el APIC para permitir la reasignación de los registros a otra región de la memoria.
- Es decir, que a partir de esta generación de procesadores Intel un programador con el conocimiento suficiente podía mover el espacio donde se encontraba el APIC a oro espacio de memoria.
- La capacidad de reubicar los registros APIC introduce una vulnerabilidad compleja en el modo de gestión del sistema (SMM).

- El Modo de Gestion del Sistema SMM (por sus siglas en ingles). Es el nivel -2 de seguridad, el cual es el que realmente está a cargo del procesador, es decir este nivel de seguridad es el que se encarga de controlar el hardware, firmware y la mayoría de las validaciones criticas de seguridad




# SYSTEM MANAGEMENT MODE

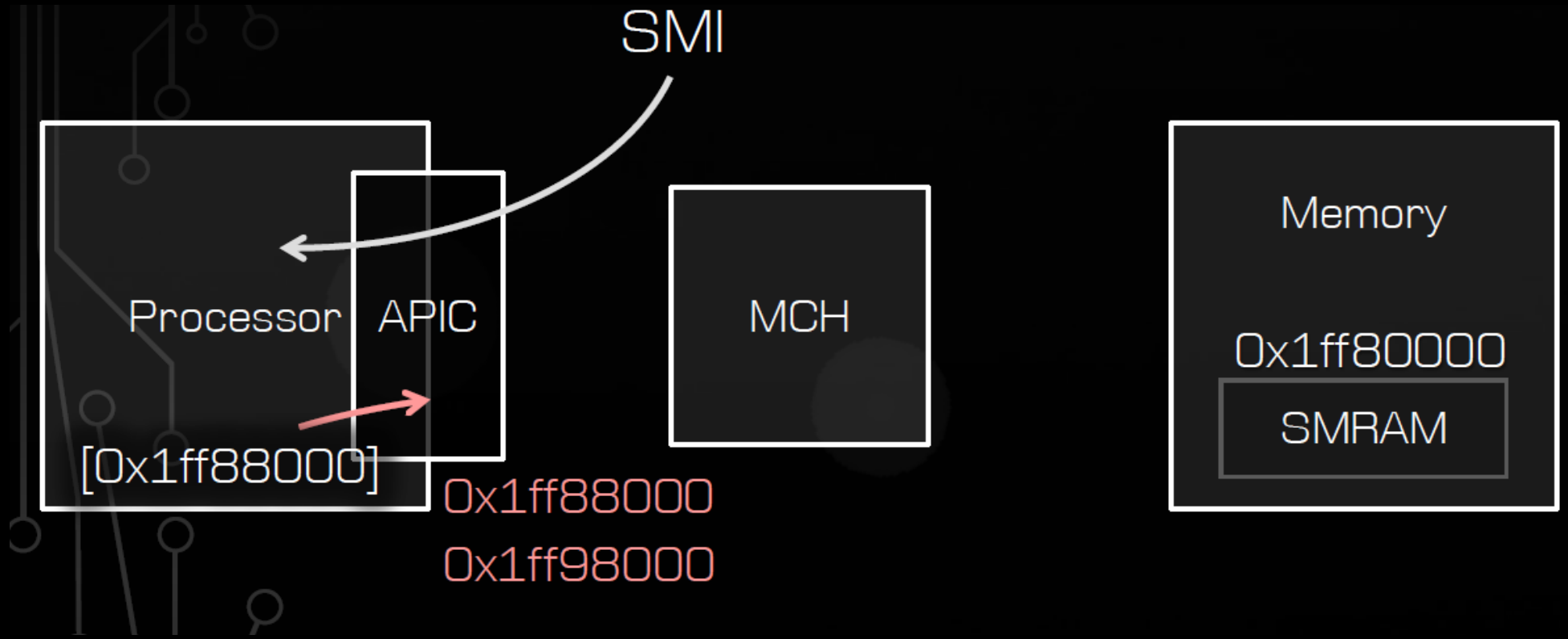
# ¿QUÉ PASA SI LOS REGISTROS DEL APIC SE MUEVEN?

- Se pueden mover para superponerse al rango definido para la SMRAM (espacio de memoria restringido solo para el Gestor del Sistema)
- Esto proporciona a un código de anillo **0** una pequeña influencia indirecta sobre el SMM y viola la separación arquitectónica fundamental de los dos modos de ejecución.

## ¿HAY ALGUIEN QUE CONTROLE EL ACCESO A ESTE ESPACIO EN MEMORIA RESTRINGIDO?

- El acceso al SMRAM es controlado por el MCH o Memory Controller Hub
- El MCH se ubica entre el núcleo del procesador y la memoria.
- Si el procesador no está en SMM, el MCH bloquea el acceso a la SMRAM.

- Al mover el espacio de memoria del APIC, los accesos de memoria que deben enviarse al MCH para su validación son aceptados prematuramente por el propio APIC, y nunca recibidos por el MCH.

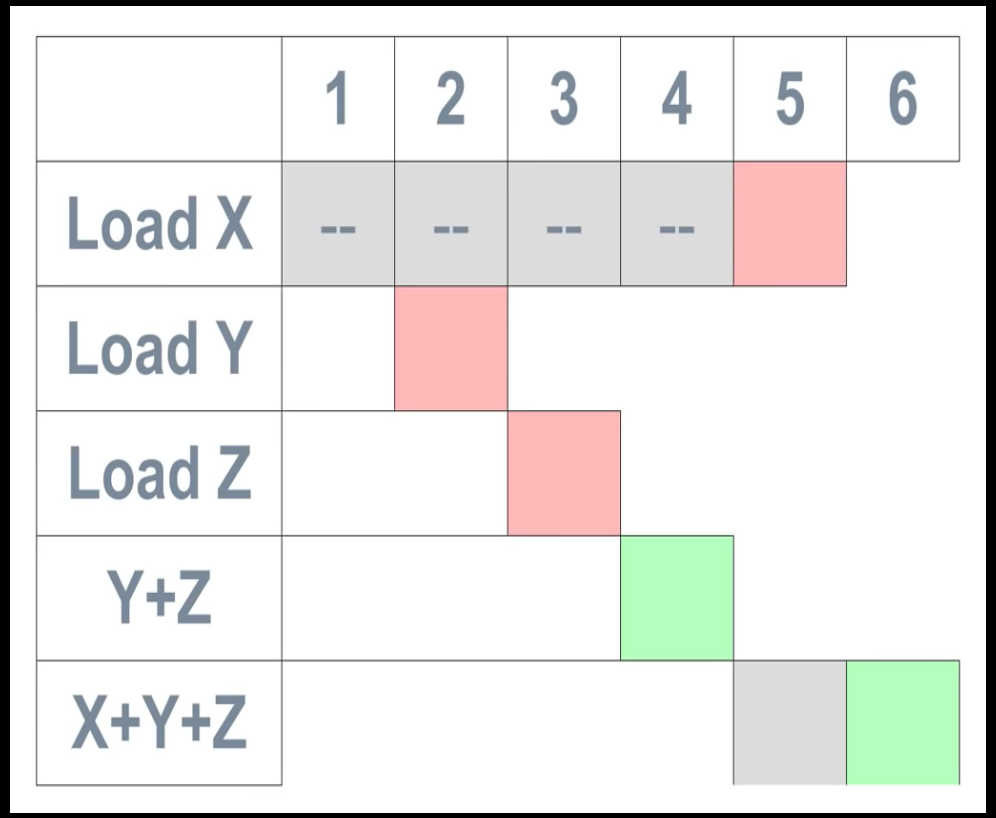


# Conceptos Importantes

EJECUCION FUERA DE ORDEN

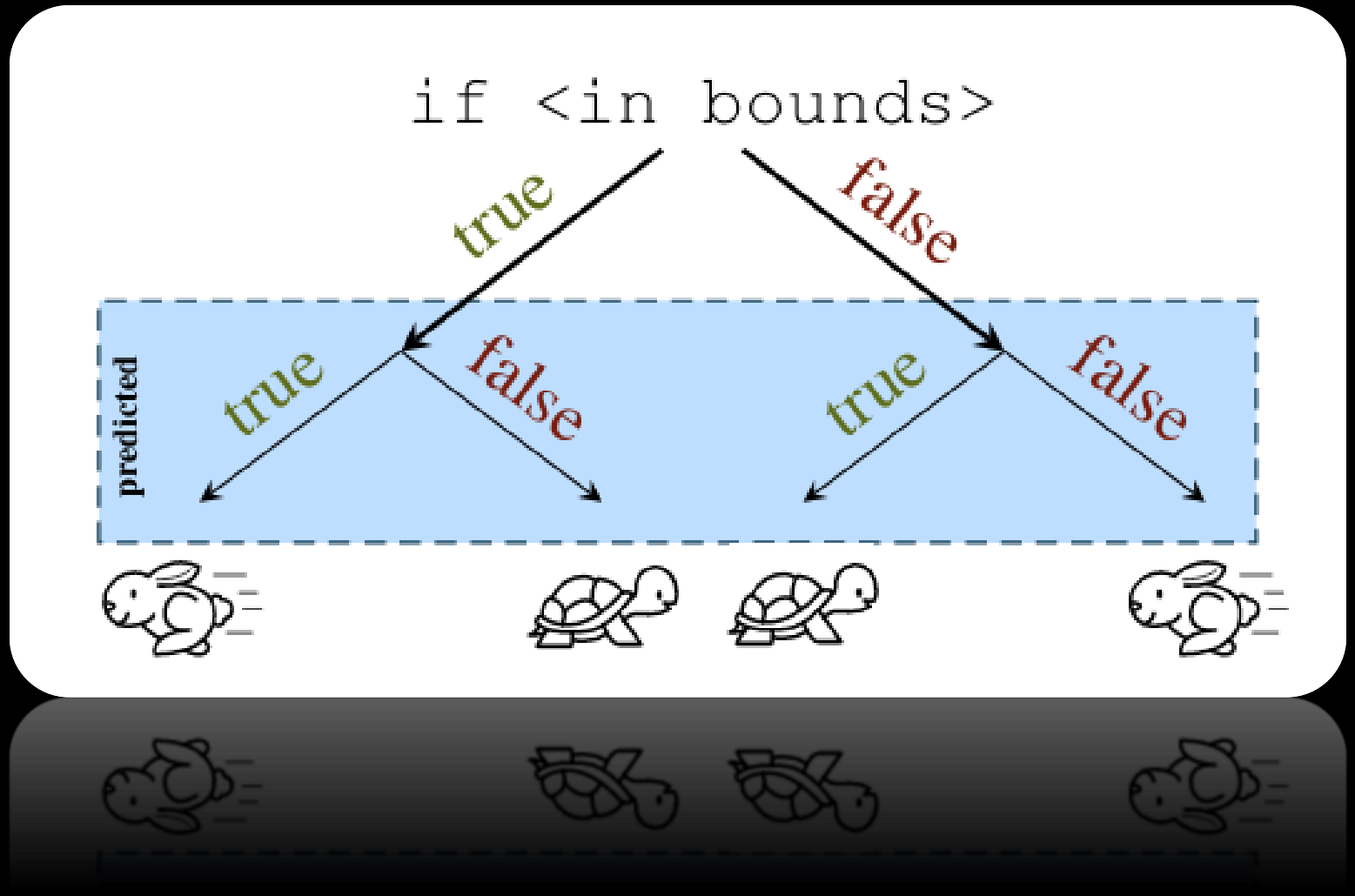
- Característica de rendimiento importante de los procesadores de hoy en día para superar las latencias de las unidades de ejecución ocupadas.
- Funciona mirando hacia adelante y programando las operaciones subsiguientes a las unidades de ejecución inactivas del núcleo.
- Mientras una sentencia se ejecuta y si la siguiente no tiene que esperar un valor por la anterior esta se empieza a ejecutar

1. Load X
2. Load Y
3. Load Z
4. temp = Y + Z
5. X = X + temp



## EJECUCION ESPECULATIVA

- Llevar a cabo un trabajo antes de saber si será realmente necesario con la intención de evitar el retraso que supondría realizarlo *después* de saber que sí es necesario.
- Ejecutar pedazos de código anticipadamente y ver todos los posibles escenarios que supondrían una sentencia de cuyo valor no se sabe exactamente hasta después de determinado tiempo.


if <in bounds>
true
false
predicted
true
false
true
false

- El objetivo de esta técnica es proporcionar una mayor concurrencia en caso de disponer de más recursos.
- Un punto importante es que toda la Información desechada por la ejecución especulativa se guarda también en la chache

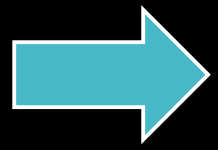
MELTDOWN

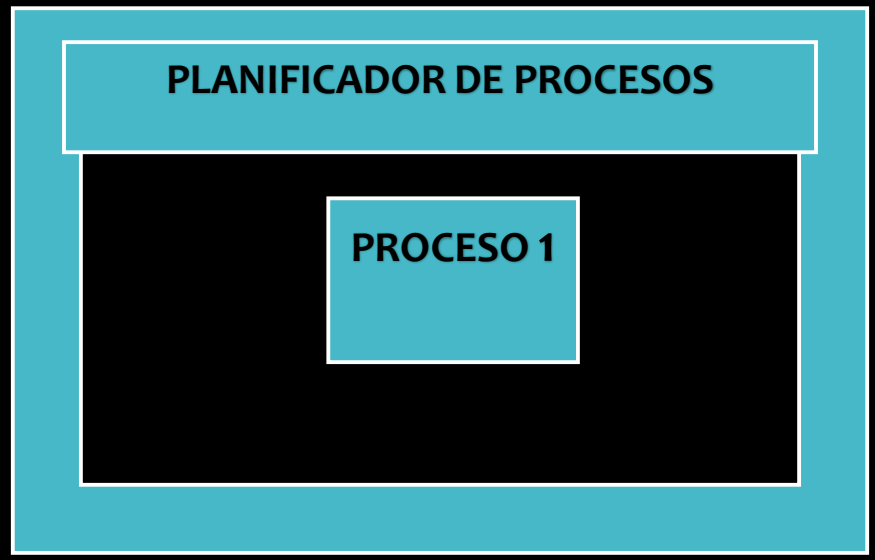
¿CÓMO FUNCIONA?

- Meltdown se basa en una condición de carrera de la CPU que puede surgir entre la ejecución de instrucciones y la comprobación de privilegios, con el fin de leer sin autorización información mapeada en memoria de una forma detectable .

**PROCESO 1**

Instrucción de leer un espacio de memoria no autorizado



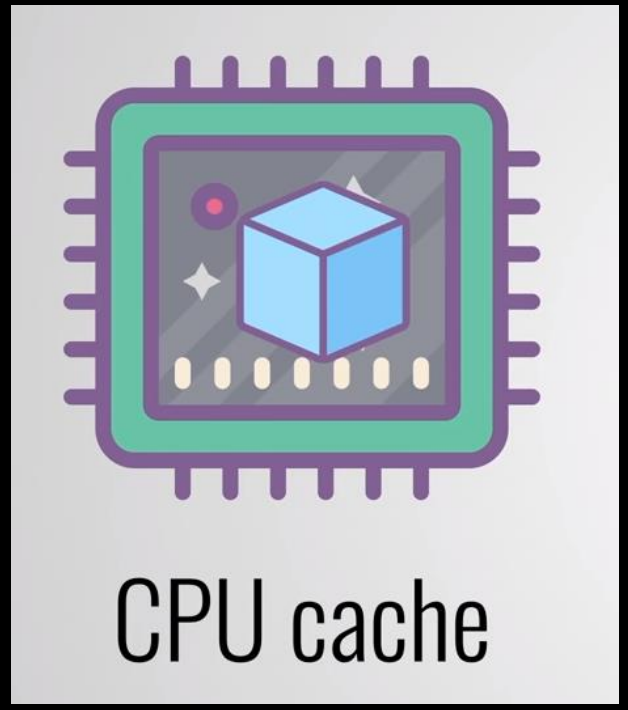


# EN EJECUCIÓN

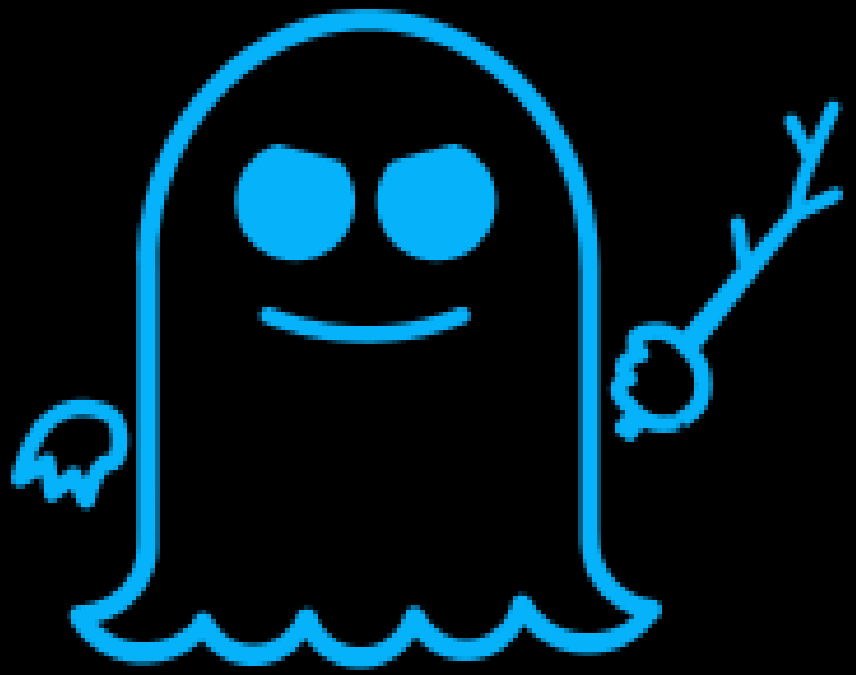
**PROCESO 1**

| | | |
|---|---|---|
| **Suposición 1: P1 SI tiene los privilegios** | | |
| **Suposición 2: P1 NO tiene los privilegios** | | **Termina el proceso 1** |
| **Verificación de Privilegios** | **No cuenta con los privilegios** | |

El proceso 1 termina y se le avisa que no cuenta con los privilegios necesarios por lo que no se le entrega nada de la información que quería leer.




Durante la suposición 1 se cargo a cache lo que pedía el proceso 1 aunque no contaba con los privilegios



SPECTRE

¿CÓMO FUNCIONA?

- Este ataque se basa en explotar los efectos secundarios de la ejecución especulativa.
- Spectre se centra en la predicción de saltos, un caso especial de la ejecución especulativa.
- Spectre explota todas las vulnerabilidades de una arquitectura.

EN EJECUCIÓN

**PROCESO 1**

**Con datos/instruccines escondidos**

| | | |
|---|---|---|
| Suposición 1: P1 SI tiene los privilegios | Empieza a cargar lo necesario | |
| Suposición 2: P1 NO tiene los privilegios | | Termina el proceso 1 |
| Verificación de Privilegios | No cuenta con los privilegios | |

- Dentro de los ataques de este tipo se esconden datos o instrucciones protegidas en su espacio de memoria creado.
- Recordemos que a cada proceso se le asigna un espacio de memoria virtual para que se ejecute. Esto no tiene nada que ver con privilegios.
- Esta parte escondida o secreta dentro de este proceso puede hacer que mediante la ejecución especulativa se cree un canal secreto que extraiga toda la información mapeada en nuestra memoria.
- Todo esto ocurriría antes de que se llegase al punto en que el procesador avise que este proceso no tiene privilegios suficientes.

# Diferencias entre Meltdown y Spectre

- Meltdown es un ataque de microarquitectura relacionado que explota la ejecución fuera de orden.
- Meltdown se basa en la observación de que cuando una instrucción causa una trampa, las siguientes instrucciones se ejecutan fuera de orden antes de ser terminadas.
- Meltdown explota una vulnerabilidad específica de muchos procesadores Intel y algunos procesadores ARM que permite que ciertas instrucciones ejecutadas de manera especulativa eviten la protección de la memoria.
- Spectre se basa en la ejecución especulativa y sus consecuencias como la predicción de saltos.

# Referencias


- Lipp, Moritz; Schwarz, Michael; Gruss, et al. (2018) *Meltdown: Reading Kernel Memory from User Space. (PDF)* p. 8 sec. 5.1. Recuperado de: https://meltdownattack.com/meltdown.pdf Consultado el 16 de marzo de 2019.
- Paul Kocher, Jann Horn, Anders Fogh, et al. (2018) *Spectre Attacks: Exploiting Speculative Execution. (PDF)* Recuperado de: https://spectreattack.com/spectre.pdf Consultado el 16 de marzo de 2019


**Videos Sugeridos:**

- **https://www.youtube.com/watch?v=syAdX44pokE**
- **https://www.theregister.co.uk/2015/08/11/memory_hole_roots_intel_processors/?page=1**
- **https://www.youtube.com/watch?v=Uv6lDgcUAC0**