Spectre y Meltdown

Vulnerabilidades de hardware de los procesadores Intel, AMD y ARM

Expositores:

Osorio Robles Sergio de Jesús

Medina Mora Fernando Arturo


MELTDOWN

Descubrimiento

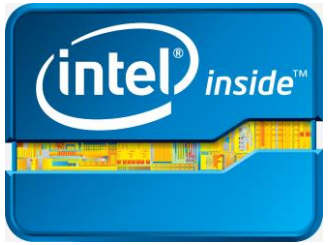
- 1995 (Documento): La Arquitectura del procesador Intel 80x86:trampas para los sistemas seguros

- 2016 (Documento): ARMagedón: Ataques a la caché en dispositivos móviles

- 2016 (Presentación): "¿Qué podría salir mal al usar <inserte aquí una instrucción x86>?”

Hardware afectado

- Afecta principalmente



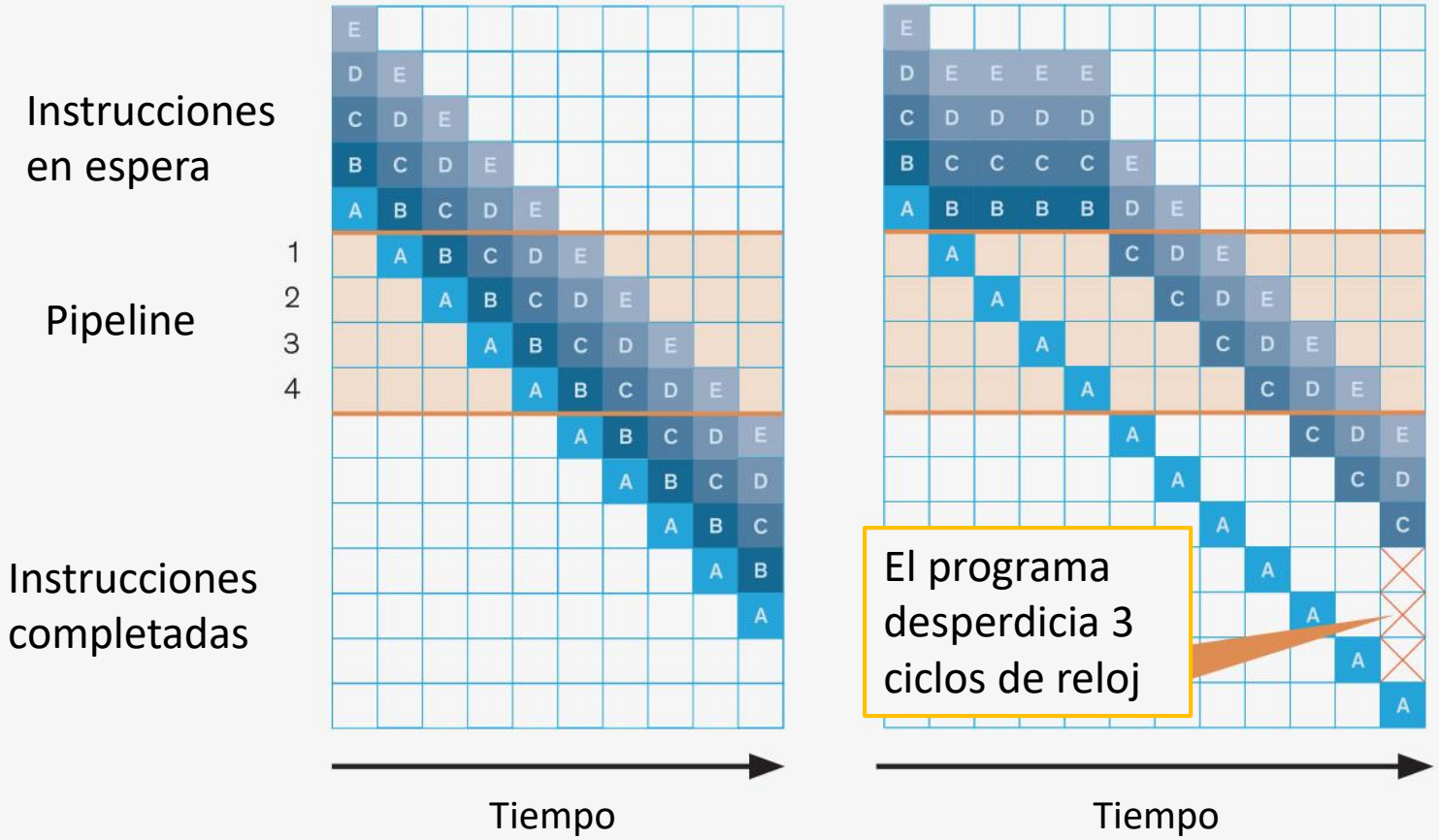
- Parcialmente afectados

- No afectados

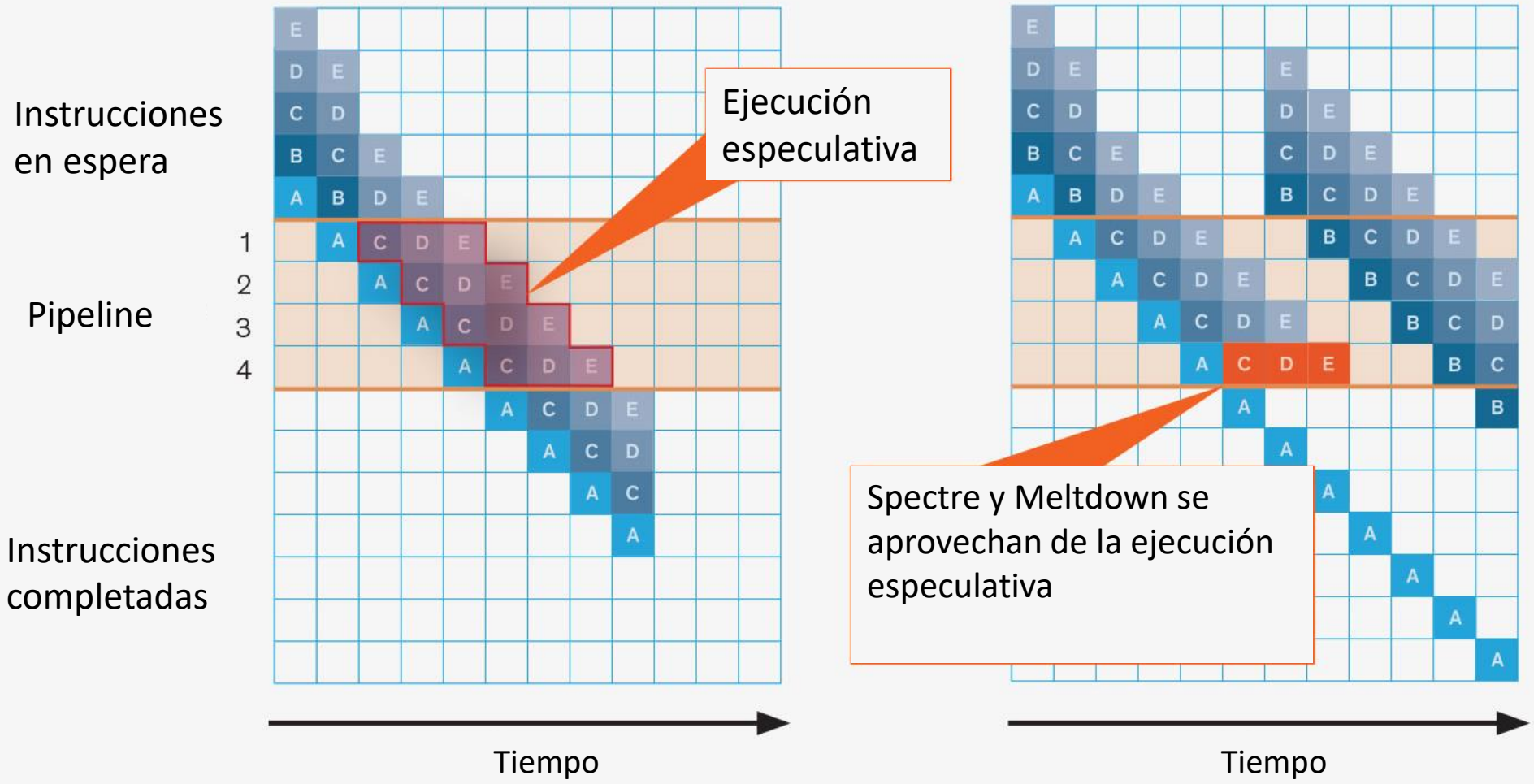
Ejecución especulativa

- Es una forma de optimización en la que un sistema operativo realiza una tarea que podría no ser necesaria
- El objetivo de esta técnica es proporcionar una mayor concurrencia en caso de disponer de más recursos.

Ejecución especulativa en un pipeline



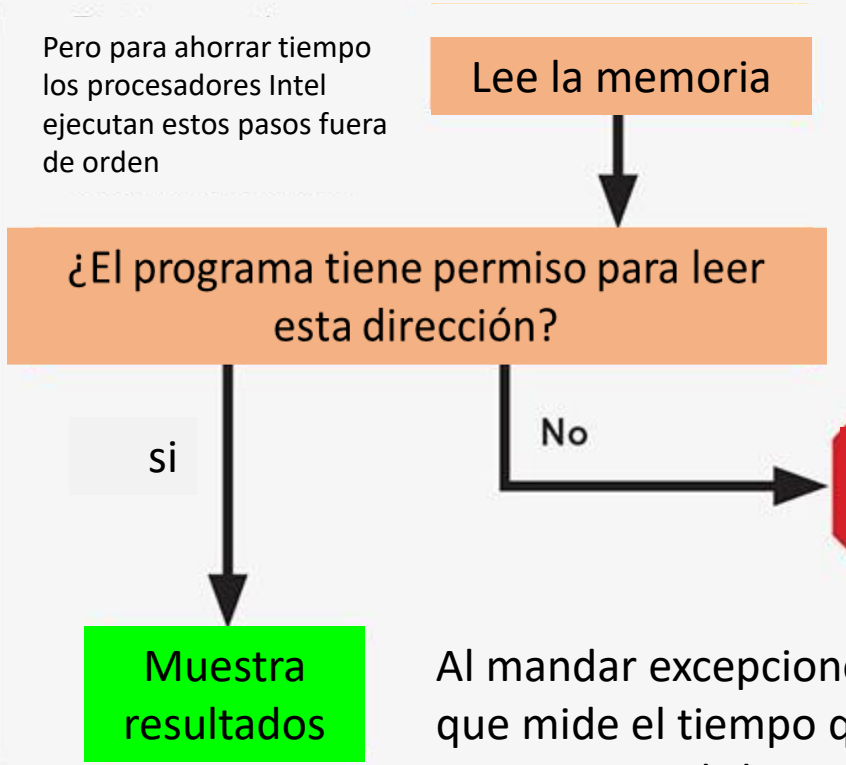
Ejecución especulativa en un pipeline



Mecanismo de acción

Lógicamente una dirección de memoria debería leerse así:

Pero para ahorrar tiempo los procesadores Intel ejecutan estos pasos fuera de orden



Al mandar excepciones y utilizar un canal lateral que mide el tiempo que lleva leer desde la memoria, Meltdown puede robar datos sensibles.

“…básicamente lo que hace es derretir las barreras de seguridad que normalmente coloca el hardware.”

-Moritz Lipp y Michael Schwarz autores de *Meltdown: Reading Kernel Memory from User Space*

Impacto

- Todo procesador Intel que implemente ejecución fuera de orden esta potencialmente afectado.

  (Intel ha fabricado chips con ejecución fuera de orden desde 1995)

- Gran impacto sobre grandes proveedores de servicios de computo en la nube.

Impacto

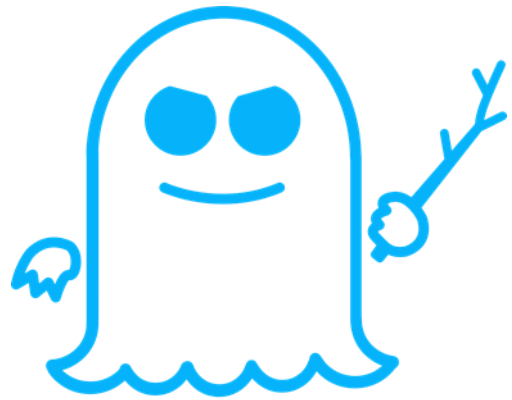
- El ataque puede revelar el contenido de cualquier espacio de memoria mapeado en el espacio de direcciones de usuario, aún en el caso de estar protegido de otro modo.

- Los contenedores Docker también son vulnerables.

- Sistemas Operativos virtualizados no pueden afectar al Sistema anfitrión.

Mitigación

- Publicación de un parche para Windows 10, 3 Noviembre 2017.
- Publicación de un parche para macOS 10.13, 3 Diciembre 2017.
- Publicación en medios de comunicación de las vulnerabilidades descubiertas, 3 Enero 2018.
- Ubuntu y demás distribuciones actualizan sus kernel, 10 Enero 2018.



SPECTRE

Descubrimiento

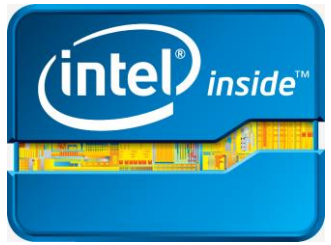
- 2002-2003: Yukiyasu Tsunoo usaron la vulnerabilidad para romper los algoritmos de cifrado MISTY1 y DES.

- 2017: Anders Fogh realizó una presentación acerca de esta vulnerabilidad en procesadores con ejecución especulativa.

- 2017: Las compañías fabricantes analizan sus procesadores para saber si son vulnerables

Hardware afectado

- Afecta principalmente



AMD

- Parcialmente afectados

ARM

- No afectados

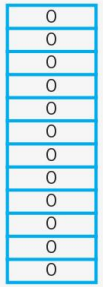
Spectre

Se ejemplifica con el siguiente código

```
If (x<256){
        secret=array1[x]
        y=array2[secret]
}
```

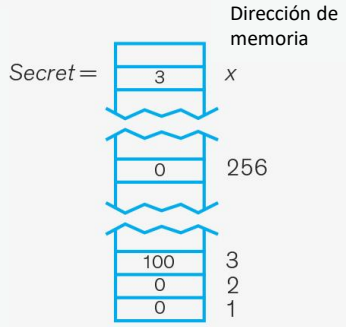
1. El código se ejecuta varias veces con x menor a 256. Esto prepara al predictor de rama para que espere que x sea menor que 256 la próxima vez

Memoria caché

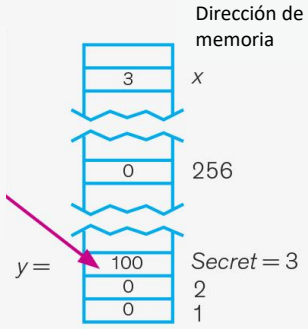


2. El atacante vacía la memoria caché del procesador utilizando la instrucción flush, de modo que cualquier dato leído por el programa debe ingresarse desde la memoria principal

3. El atacante ejecuta el código con x > 256. El procesador comienza a ejecutar "especulativamente" el resto del código como si x fuera menor a 256. La dirección de memoria en x contiene datos secretos que el atacante desea. El software asigna estos datos a la variable "secret"

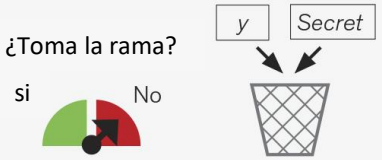


Traído de memoria principal porque el caché está vacío

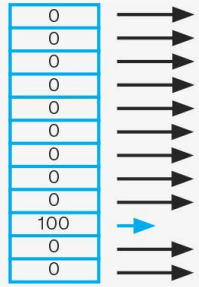


4. Luego, el código usa el valor de "secret" como una dirección de memoria. Lee los datos en esa dirección y trae esos datos al caché desde la memoria principal

5. Finalmente, el procesador se da cuenta de que no debería haber tomado la rama. Y hace que las variables "y" y "secret" no sean visibles para el programa.



Memoria caché



Tiempo

6. El atacante accede a cada dirección en el caché.
Debido a que la dirección de memoria de "secret" es la única cuyos datos ya han sido traídos de la memoria principal a la caché.

“ La vulnerabilidad fue denominada "Spectre" (espectro) porque está basada en la causa de base, la ejecución especulativa. Puesto que no es fácil de corregir, será algo que nos persiga durante mucho tiempo (como un fantasma)”

-Paul Kocher and Jann Horn autores de *Spectre Attacks: Exploiting Speculative Execution*

Impacto

- Spectre tiene el potencial de tener un mayor impacto en los proveedores de la nube que Meltdown.

- Spectre puede permitir que programas maliciosos induzcan a un hipervisor a transmitir los datos a un sistema invitado que se ejecute en él, como es el caso de una máquina virtual.

Mitigación

- Lo más probable es que no pueda haber un único parche general
- Se ha informado de que los parches para Spectre degradan el rendimiento de forma significativa (2%-14%).
- El coste de la mitigación puede aliviarse en procesadores que soporten la limpieza selectiva del TLB (como la arquitectura Alpha).

Mitigación

- El 14 de julio de 2018 fue liberada la versión 9.5 de Debian la cual incluye una mitigación a la variante dos de Spectre por medio de un parche al kernel.

Referencias

- Abu-Ghazaleh, N., Ponomarev , D., & Evtyushkin, D. (2019). How the Spectre and Meltdown Hacks Really Worked. *IEEE Spectrum*, 42-49.
- {Moritz Lipp and Michael Schwarz. (2018). Meltdown: Reading Kernel Memory from User Space. *Security Symposium* (pág. 18). Baltimore: USENIX.
- Graz University of Technology. (30 de Enero de 2018). *Meltdown and Spectre*. Obtenido de Vulnerabilities in modern computers leak passwords and sensitive data: https://meltdownattack.com/
- Paul Kocher and Jann Horn. (19 de Enero de 2019). Spectre Attacks: Exploiting Speculative Executio. *40th IEEE Symposium on Security and Privacy.* Nueva York: IEEE. Obtenido de Meltdownattack.
- "Out-of-order Execution". pcguide.com. Obtenido 17-01-2014. This flexibility improves performance since it allows execution with less 'waiting' time.
- Lazy and Speculative Execution Butler Lampson Microsoft Research OPODIS, Bordeaux, Francia 12 Diciembre 2006