LA VULNERABILITA' MELTDOWN

1. Quando un processo cerca di accedere a una zona di memoria riservata, l'istruzione viene schedulata nella pipeline della CPU. Prima del completamento dell'istruzione, il controllo dei privilegi di accesso fallirà, l'esecuzione verrà annullata, nessun dato verrà fornito in uscita

2. Se l'istruzione che cerca di accedere a una zona di memoria riservata si trova in un ramo del codice (cioè se la sua effettiva esecuzione dipende ad esempio da un controllo di un valore), l'esecuzione speculativa eseguirà comunque l'istruzione di lettura del dato protetto, per poi segnalare l'errore solo nel caso in cui l'istruzione viene davvero eseguita.

3. Nel caso in cui il ramo che contiene l'istruzione incriminata viene ignorato, il dato segreto viene ignorato, ma non senza lasciare traccia!

4. Il dato infatti, essendo stato da poco utilizzato, è rimasto nella cache! Pur essendo la memoria cache non direttamente leggibile, si può risalire all'effettiva presenza di una locazione di memoria nella cache. L'idea è quella che se il dato è presente nella cache diventa subito disponibile (come abbiamo visto, le momorie cache sono estremamente veloci),

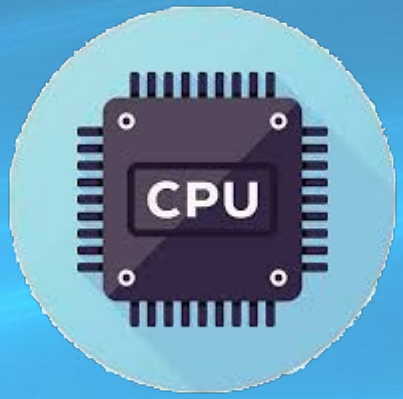
Meltdown può utilizzare questa tecnica per leggere potenzialmente ogni locazione di memoria del computer in tempi rapidi, in questo modo un processo malevolo può leggere dati riservati come password, dati cifrati o ogni tipo di dato presente in memoria

HARDWARE E SICUREZZA

L' hardware ha quindi un ruolo indispensabile nel garantire la sicurezza, intesa come riservatezza, integrità e disponibilità di dati e risorse.

Abbiamo visto che, data la complessità dei processori moderni, possono nascere vulnerabilità inaspettate. Questo però non deve preoccuparci oltremodo: tutti i processori e altre componenti hardware, prima di essere messe in commercio, sono sottoposti a lunghi processi di verifica della sicurezza.

Dobbiamo, tuttavia, sempre tenere a mente un principio sempre valido: la sicurezza al 100% non esiste!
E' questa la lezione più importante che Meltdown e Spectre ci hanno insegnato, rimanendo nascoste nei processori di tutto il mondo per più di 20 anni.



RIFERIMENTI

- https://www.c3t.it/projects/awareness/articoli&brochure/attacchi_processori/

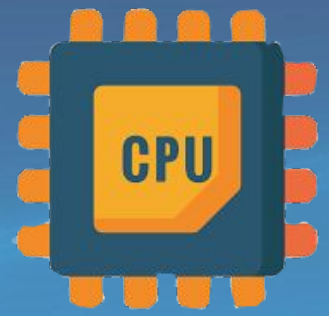
POR FESR 2014-2020 obiettivo Crescita e Occupazione (CreO)

ATTACCHI AI PROCESSORI

Scopri come anche i processori talvolta possono presentare vulnerabilità e quali sono

I moderni processori utilizzano una varietà di tecniche di ottimizzazioni al fine di ottenere prestazioni elevate. Può capitare, tuttavia, che nella loro complessità si nascondano vulnerabilità inaspettate. E' questo il caso delle vulnerabilità **Spectre** e **Meltdown.**

Le presenza delle vulnerabilità Spectre e Meltdown è stata resa nota il 3 gennaio 2018 da un team di ricercatori di Google. Si tratta di vulnerabilità presenti nell'Hardware e risolvibili definitivamente solo con la sostituzione o modifica dell'Hardware stesso, che può comportare un importante decadimento delle prestazioni.
Per capire come si è arrivati a "Spectre" e "Meltdown", perché sono così importanti e pericolosi, e apprezzare i principi su cui si basano, è utile ripercorrere i principi su cui si basano le moderne CPU.



Pipeline ed esecuzione speculativa: Al fine di migliorare le prestazioni, le moderne CPU sono in grado di eseguire più istruzioni in parallelo. In alcuni casi, per garantire la correttezza delle operazioni, è necessario modificare l'ordine di esecuzione di alcune istruzioni. I moderni processori sono dotati di uno scheduler analizza le istruzioni da elaborare e decide se alcune istruzioni possono essere eseguite in parallelo oppure se è necessario per questioni di efficienza cambiare l'ordine di esecuzione delle istruzioni. Nel caso di un'istruzione di diramazione (se C è 0 esegui l'istruzione A altrimenti l'istruzione B) lo scheduler può decidere di eseguire entrambe le istruzioni (A e B) e di mantenere le istruzioni in sospeso fino a quando non sarà noto il valore di C. Quando C sarà noto, l'istruzione corretta verrà mantenuta, mentre quella eseguita per errore verrà annullata. Questa strada usualmente è più rapida dell'aspettare di conoscere il reale valore di C

La cache: Le cache sono delle memorie di piccole dimensioni ma molto veloci su cui si memorizzano dati utilizzati frequentemente. Essendo il tempo di accesso alla memoria principale molto lungo, quando un dato viene utilizzato viene salvato nella cache, in modo che se fosse necessario nuovamente il suo utilizzo, sarebbe disponibile in tempo molto minore.

Memoria virtuale: Quando il programma cerca di accedere a una locazione di memoria il processore traduce un indirizzo "virtuale" (quello noto al programma) con l'indirizzo fisico (la locazione reale di memoria). Questo processo si utilizza per questione di efficienza e sicurezza. Quando si prova ad accedere ad una locazione non consentita (ad esempio la memoria kernel), viene generato un'eccezione e nessun dato sulla locazione di memoria viene resa nota al processo.