# Side-Channel Attack

Leonardo Costa Santos - 10783142

Lucas Paiolla Forastiere - 11221911

Julia Leite - 11221797

2 de dezembro de 2020

# 1 Introdução

Side-Channel Attacks, que do inglês significa, Ataque por Canal Lateral são jeitos explorar vunerabilidades físicas de componentes eletrônicos, como a CPU de um computador.

O nome vem do fato deles não atacarem "pela porta da frente", mas sim algum "rastro"físico que um componente deixa ao fazer determinadas ações.

Um SCA não necessariamente tem a ver com componentes eletrônicos, pois podemos, por exemplo, descifrar uma senha de alguém captando os sons do teclado. Em geral, o SCA ataca um ponto fraco de um componente que não tem nada a ver com o seu funcionamento em si (como no exemplo do teclado, o teclado teoricamente não tem a responsabilidade de deixar os barulhos de cada tecla iguaizinhos). Daí então o nome *canal lateral*.

Os dois principais SCA, que tornaram o "ramo" famoso foram o *Meltdown* e o *Spectre*, descobertos por independentemente por uma série de pesquisadores, mas destacando-se o grupo Project Zero da Google [**Lipp2018meltdown**].

Entretanto, existem muitas classes de SCA, como ataques ao cache (que é o caso dos dois exemplos citados), ataques que monitoram a energia consumida pelo computador, ataques que monitoram o eletromagnetismo eminido, ataques que monitoram o som emitido (como o exemplo do teclado), ataques que recuperam dados excluidos do disco entre muitos outros.

# 2 História

Primeira pessoa a usar o termo (Side Channel Cryptanalysis of Product Ciphers) Primeiro ataque descoberto (TEMPEST 1942) Popularização do termo Mais descobertas de ataques

# 3 Classificações de Side-Channel Attacks

## 3.1 Classificação por (?)

## 3.2 Classificação por grau da invasão

# 4 Exemplos de Side-Channel Attacks

## 4.1 Meltdown

Teste citando[**Lipp2018meltdown**]

## 4.2 Spectre

## 4.3 CacheOut

CacheOut é um ataque de *Microarchitectural Data Sampling* (MDS), ou Amostragem de Dados Microarquitetural, capaz de evitar as medidas de segurança contra MDSs dos processadores da Intel. Este ataque é capaz de vazar dados atravéz de barreiras de segurança como o isomento de mémoria entre processos, entre *user/kernel spaces*, *SGX enclaves* e máquinas virtuais. Ao causar contenção de linhas de cache, CacheOut causa a expulsão de dados do cache e lê estes dados dos LFBs com um ataque TAA, conseguindo vazar páginas inteiras de memória. [**schaik2020cacheout**]

### 4.3.1 Line Fill Buffers

*Line Fill Buffers* (LFBs) são *buffers* microarquiteturais usados para armazenar dados durante acessos ao cache L1, tratando de pedidos ao cache em *cache misses* e temporariamente armazenando dados em acessos à memória e operações de I/O. Também podem ser usados em *cache hits* e para encaminhar dados para operações de leitura e escrita no cache. [**IntelMDS**]

### 4.3.2 Transactional Synchronization Extensions

*Transactional Synchronization Extensions* (TSX) é uma implementação de transações de memória, que agrupa instruções em transacões executadas de modo atômico, executando todas as instruções da transação especulativamente e consolidando os resultados apenas após a execução de sua última instrução. Se alguma instrução causa uma *memory fault*, a transação inteira é descartada. [**schaik2020cacheout**]

### 4.3.3 TSX Asynchronous Abort

*TSX Asynchronous Abort* é um tipo de ataque que zera linhas de cache antes de uma transação que carrega dados dessas linhas, causando uma falha na transação. Alocando espaço no LFB antes da transação, os dados do LFB são encaminhados para a instrução que causa a *fault*. Como a transação não é completada, a instrução é executada com dados de uma transação anterior, permitindo a amostragem desses dados. [**schaik2020cacheout**]

### 4.4 SGAxe

### 4.5 ZombieLoad

### 4.6 Foreshadow

## 5 Há jeito de se previmir?

## 6 Há como saber se estou sofrendo um SCA?

No. =(=

# 7 Conclusão