Lecture: Security

- Topics: Spectre and Meltdown attacks, information leakage, integrity verification
Hardware Security

• Several types of attacks: physical access to hardware, compromised OS, untrusted co-scheduled applications

• Defenses include: hardware permission checks, encryption, microarchitecture partitions, signature checks, trusted execution environments like Intel SGX

• Information leakage still unresolved – exploited by Meltdown, Spectre, and many subsequent attacks
Meltdown

**Attacker code**

Fill the cache with your own data X

lw  R1  \(\leftarrow\) [illegal address]
lw  ...  \(\leftarrow\) [R1]

Scan through X and record time per access
Spectre: Variant 1

x is controlled by attacker

Victim Code

if (x < array1_size)
y = array2[array1[x]];

Access pattern of array2[ ] betrays the secret

array1[ ] is the secret

Thanks to bpred, x can be anything
Spectre: Variant 2

**Attacker code**

Label0: if (1)

Label1: ...

**Victim code**

R1 ← (from attacker)
R2 ← some secret
Label0: if (...)  

...  ...

Label1:  

lw [R2]
Defenses

- Disable speculation when violations happen (fixes Meltdown)
- Partition resources – has a performance impact
- Several resources involved: bpred, caches, memory controller
- Constant behavior algorithms
Memory Integrity Verification

• Implemented on commercial processors, e.g., Intel SGX

• Confirms that data has not been tampered by malicious agents – attacker with physical access, rogue OS

• Every block has a MAC and a version number

• To prevent a replay attack (attacker sends an old version of data/MAC/counter), a tree of hashes is navigated
Bonsai Merkle Tree