Spectre and Meltdown Attack Variants

N V S S Hari Krishna Nama - 170050077
Dileep Kumar Reddy Chagam - 170050080
Tarun Somavarapu - 170050093

Problem Statement/Goals:

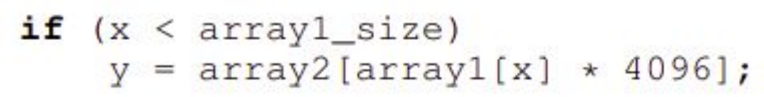
In appetite for speed, we lost security.

Spectre and Meltdown attacks exploit the crucial vulnerabilities in modern processors. While programs are typically not permitted to read data from other programs, a malicious program can exploit Meltdown and Spectre to get hold of secrets stored in the memory of other running program. This might include your passwords stored in a password manager or browser, your personal photos, emails, instant messages, and even business-critical documents.

The main goal of the project is to understand this attacks and implementing them. At the moment, there are many variants available to these attacks, we are going to show some of these variants in the demo.

Spectre:

Variant 1: In this variant of Spectre attacks, the attacker mistrains the CPU’s branch predictor into mispredicting the direction of a branch, causing the CPU to temporarily violate program semantics by executing code that would not have been executed otherwise.



```
if (x < array1_size)
    y = array2[array1[x] * 4096];
```

Variant 2: In this variant, the attacker chooses a gadget from the victim’s address space and influences the victim to speculatively execute the gadget. The attacker does not rely on a vulnerability in the victim code. Instead, the attacker trains the Branch Target Buffer (BTB) to mispredict a branch from an indirect branch instruction to the address of the gadget, resulting in speculative execution of the gadget.

Meltdown:

Meltdown exploits out-of-order execution and side channels on modern processors to read arbitrary kernel memory from an unprivileged user space program.

Unlike Spectre, Meltdown does not use branch prediction. Instead, it relies on the observation that when an instruction causes a trap, following instructions are executed out-of-order before being terminated.

Work done

- Read the papers explaining the spectre[Ref: 1] and meltdown[Ref: 2] attacks
- Modified the existing implementations[Ref: 4, 5, 6] of spectre and meltdown attacks (variants mentioned in this presentation) to work on our personal devices

In Progress

- Trying to fetch secret data from chromium browsers using spectre attack
- Trying to fetch data from a different process by directly reading physical memory

Observations

- Every Intel processor which implements **out-of-order** execution is potentially affected, which is effectively every processor since 1995 (except Intel Itanium and Intel Atom before 2013). It is unclear whether AMD processors are also affected. According to ARM, some of their processors are also affected
- Almost **every system** is affected by Spectre **including smartphones**. Spectre is verified on Intel, AMD, and ARM processors
- A software patch named KAISER works as a shorterm workaround for Meltdown until hardware is fixed
- To mitigate Spectre, we need to disable speculative execution, which is expected to reduce the system speed by 10 to 30 percent

Trials and tribulations

- Original paper's ATTACK PATTERN is easily predictable by the system's stride prediction. So we had to randomise it further and increase the number of training loops.
- After disabling the KAISER patch on Linux(PTI), we tried dumping memory from another process but in vain. Later tried leaking Kernel Debugging Symbols, which was successful. Now we are working on leaking memory of other processes/kernel modules.

References

1. https://spectreattack.com/spectre.pdf
2. https://meltdownattack.com/meltdown.pdf
3. https://meltdownattack.com/
4. https://gist.github.com/anonymous/99a72c9c1003f8ae0707b4927ec1bd8a
5. https://github.com/Anton-Cao/spectrev2-poc
6. https://github.com/paboldin/meltdown-exploit

Demo

Thank you