IOPMP Task Group Meeting

July 6, 2023

Video link

Minutes

- Replace “Source ID” to “Domain ID” (DID)
  - Due to too many IDs, use an abstract name as the ID.
- Lite configuration extension
  - There are still a lot of 32-bit systems that will use IOPMPs.
  - Current registers definition is for 64 bits, we would suffer from some problems
- Parallel rule match (Resolving the overlap between WG and IOPMP)
  - Proposal-1 and Proposal-2

Why Lite Configuration?

- The registers is designed for 64-bit systems.
- For 32-bit systems, it may cause some problems:
  - Take 2 transactions to update a 64-bit register ➔ Not atomicity
  - <32 MDs ➔ 2-transaction update wastes time and space
  - 32-bit addr. ➔ 64-bit address register waste circuits and space

Requirement to Use LC.

- Systems to use the lite configuration should
  - Address width of input/output port: ≤ 34 bits
    - The address register of a rule is shrunk to 32 bits.
  - Data width of control port: ≥ 32 bits
    - Ensure atomic updates: SRCMD, MDMSK, MD_STALL, …
  - Number of memory domain: ≤ 31
    - A 32-bit register can contain up to 31 MDs only.
  - Number of SID: ≤ 64
    - Replace RULE_OFFSET by 0x0800, so SRCMD space shrunk to 1KB.

Shrunk Registers

- List of the register whose width is shrunk to 32 bits:
  - SRCMD_EN(s); optional SRCMD_R(s) and SRCMD_W(s)
    - MSB is still the lock bit; the reset bits support up to 31 MDs.
  - ENTRY_ADDR(i): the same as the *pmpcfg*-style for XLEN=32
  - ENTRY_CFG(i): *r* (1bit), *w* (1bit), *x* (1bit), *a* (2bit), *user-defined*(? bit)
  - MDMSK:
    - MSB is still the lock bit; the reset bits support up to 31 MDs.
  - MD_STALL:
    - MSB is still the EXEMPT bit; the reset bits support up to 31 MDs.
  - ERR_REQADDR: illegal address, ADDR[33:2].
- RULE_OFFSET: removed, replaced by a const `0x800`.

Offset of LC Registers

- The offsets are moved forward: capacity is limited
  - `0x0000`: INFO/HWCFG
  - `0x0200`: MDCFG: up to 31 MDs, each takes 4B
  - `0x0400`: SRCMD: up to 64 SIDs, each takes 16B
  - `0x0800`: Entry Array: each takes 8B

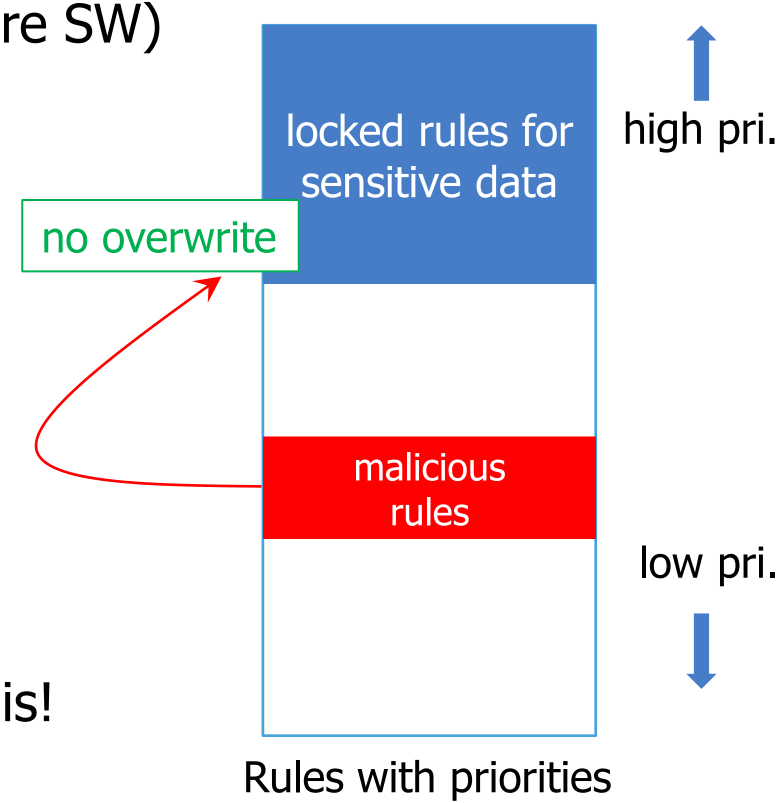
- When only 256 entries or fewer, the IOPMP control space can only occupy a 4KB area.

Agenda

- Replace “Source ID” to “Domain ID” (DID)
- Lite configuration extension
- Parallel rule match extension

Priority Matching not Just Drill a Little Hole

- A malicious rule (set by compromised secure SW) CAN'T overwrite the higher priority rules.
- That is, why "lock" can work well.

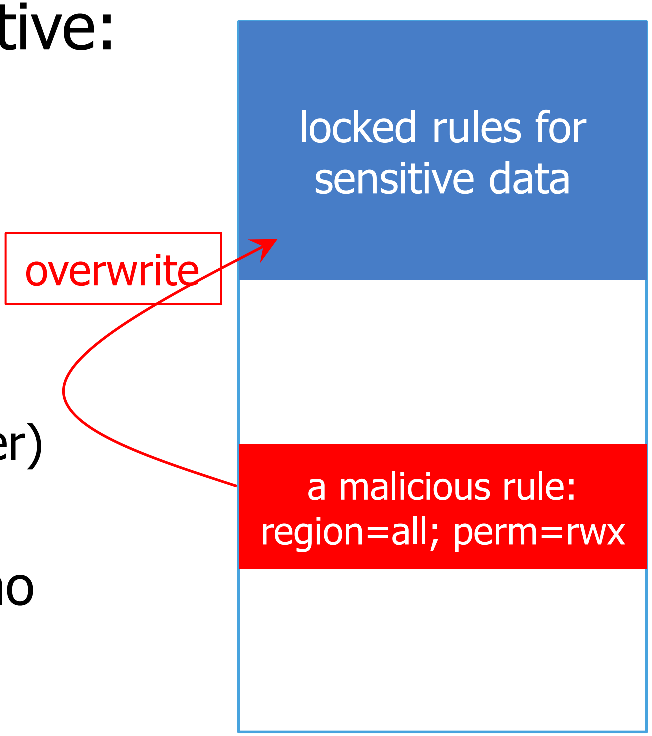


- Can secure SW be compromised? Check this!
  - A survey: sp2020-tees.pdf (purdue.edu)

Parallel Can’t Stop Overwriting

- A malicious rule makes a checker ineffective:
  - rule with whole region and all permissions.

- Lock all rules at the beginning?
  - Memory regions locked progressively.
    - secure boot locks some (e.g, anti-rollback cnter)
    - secure monitor locks more.
  - Memory of a device may change scope, but no enough rules
  - Plug/play devices (PCIe/USB) have BARs.
    - Allocate space only when they are plugged.


locked rules for sensitive data
overwrite
a malicious rule: region=all; perm=rwx

Parallel Rule

Issues on Priority Rule Match

- Timing could be the issue of priority rule match:
  - a lower clock rate and/or
  - extra cycles.

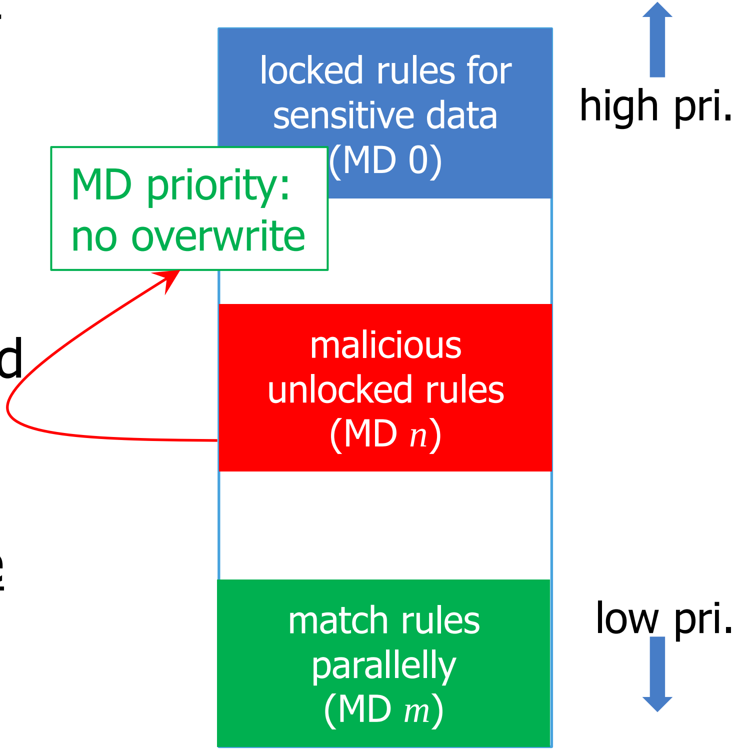
- Not ideal for latency-sensitive devices, e.g., GPU/DSP/TPU.

Proposal-1

- A read-only bit, Parallel Rule Match, *PRM* indicates if the implementation matches rules parallel.
- From overwrite prevention to overwrite detection.
- An optional sticky bit *MMCE* for Multiple Match Capturer:
  - When *MMCE*=1: active an output signal *MMC* when multiple rules grant a transaction with inconsistent permissions.
  - *MMC* indicates a malicious or improper rule setting found.
  - *MMC* connects the system reset to prevent the system from being further explored.

Proposal-2

- Keep the priority between different MDs, but use parallel rules match within a MD.
- Still can drill a hole, but in another MD.
- Timing of rule match within a MD can be relaxed
- Sensitive data can be protected by the locked rules in the higher-priority MD. No overwrite can happen!
- Always works this way; no PRM, <u>no software fragmentation</u>.


locked rules for sensitive data (MD 0)
MD priority: no overwrite
malicious unlocked rules (MD *n*)
match rules parallelly (MD *m*)
high pri.
low pri.

MDs with priorities

Summary:

- Proposal-1:
  - Shifts from overwrite protection to overwrite detection
  - Either totally parallel rule match or totally priority rule match.
  - Additional signal to reset/interrupt/exception on the overwrite detected.
  - PRM indicates the parallel or priority rule match.
- Proposal-2:
  - Overwrite protection
  - Mix parallel rule match and .
  - No additional signal
  - Spec always works this way. No SW fragmentation.

Thank You

RISC-V®