HYPERVERVISOR-MANAGED LINEAR ADDRESS TRANSLATION

Gao Chao <chao.gao@intel.com>

October 30, 2020
Agenda

Part1: Problem Statement

Part2: HLAT Introduction

Part3: Use HLAT to Enforce Guest Translation Integrity
PART 1: PROBLEM STATEMENT
Problem in Page Table Based Access Control

• Reduce attack surface through access control
  • E.g. executable code must not be writable

• Attackers can bypass access control by overriding page table

• Enforce address translation integrity
  • In a VM, write-protecting CR3 page table leads to high performance penalty
Typical Page Table Overriding Attacks

Alias Mapping

Page Remapping
PART2: HLAT INTRODUCTION
HLAT Overview


Goal: enforce (guest) translation integrity with VMM

Protected linear range (PLR): a range of guest linear address space

**Key Idea**

- GVA (Out PLR) → CR3
- GVA (In PLR) → HLAT
- GPA
- EPT
- HPA

**Benefits**

Security: invulnerable to page remapping attack

Efficiency: no interception to CR3 page table modification (compared with EPT-based page table protection).
HLAT – Nested Page Table Walk

Legacy nested page table walk
HLAT – Paging Structures

- Same as IA32e paging structures
- Support both 5-level and 4-level paging
- Bit 11: “Restart”

HLAT Terminal Fault

- non-present or reserved bits
- new page fault error code: bit 7 – “HLAT Terminal Fault”
EPT Control Bit “Paging-Write”

• “Paging Write” allows CPU to update A/D bits on pages that are non-writable to software.

| EPT Control Bits     | w     | !w    | !w|pw |
|----------------------|-------|-------|-----|
| Software writes      | Allowed | Denied | Denied |
| A/D bits update      | Allowed | Denied | Allowed |

• Improve efficiency of read-only guest page table (under EPT):
  • Reduce VM-exits due to A/D bits update
  • Relieve VMM from A/D bits emulation
EPT Control Bit “Verify Paging-write”

- “Verify Paging-write” enforces that all leaf guest-paging-structure pages encountered during the nested walk have PW set (under EPT), else generates an EPT violation.

Enables VMM to complete CPU A/D bit updates on access-controlled Guest paging structure pages without EPT violation VM exits

Verifies that page-walk occurred through VMM access-controlled pages only
Prevent Alias Mapping with PW & VPW

VPW tagged guest memory can only be accessed with HLAT (PW tagged page table)
PART3: USE HLAT TO ENFORCE GUEST TRANSLATION INTEGRITY
Enforce Guest Translation Integrity

- Prevent page remapping
- Write-protecting HLAT page table
- Prevent alias mapping
- Set VPW for protected guest physical page under EPT
- Set PW for HLAT paging structures pages under EPT
Update HLAT page table

1. Revert write-protection on HLAT through hypercall
2. Update HLAT page table
3. Enforce write-protection on HLAT through hypercall
Security Value

In theory, an attacker with arbitrary memory write capability can make kernel text/rodata writable and then override them.

With this solution, an attacker cannot easily mark kernel text/rodata writable. It must first turn off HLAT or mark HLAT page table writable through hypercalls.
Compare with Write-protecting CR3 Page Table

**Efficient**
- Intercept CR3 switching
- Granularity (less impact to normal mappings)
  - 4KB vs. whole virtual address space

**Clean**
- Small and less intrusive changes to memory management
Demo

```c
static int __init kvmap_test_init(void) {
    int ret;

    printk("kvmap test initing...\n");
    ret = set_memory_rw(target, 1);
    if (ret)
        printk("set memory to rw %d\n", ret);

    printk("attempt to write to %lx\n", target);
    *(unsigned long *)(void *)target = 0;

    return -1;
}
```

![Text section](#)

Non-writable in HLAT. Writable in CR3

![Stack trace](#)
Changes to KVM

- Advertise a PV feature
  - CPUID hypervisor leaf

- New hypercalls
  - Allow guest to set HLAT root page and PLR
  - Allow guest to set VPW/PW/RO flags for guest pages
    - Extend page tracking mechanism

- Handle EPT violations due to guest’s setting
  - Report #VE to guest
Changes to Guest Kernel

Manage HLAT page table and EPT flags

- Hooks in set_memory_ro/rw APIs

Handle #PF

- Handle HLAT terminal fault
- Walks HLAT

Handle #VE

- In general, it means an attack is detected
Status and Plan

Status

• Finished changes on KVM/guest kernel and some tests in kvm-unit-tests
• Verified on simulator

Plan

• Send out RFC patches
• Explore the possibility of using HLAT to enforce integrity for non-executable mapping
THANKS! Q&A