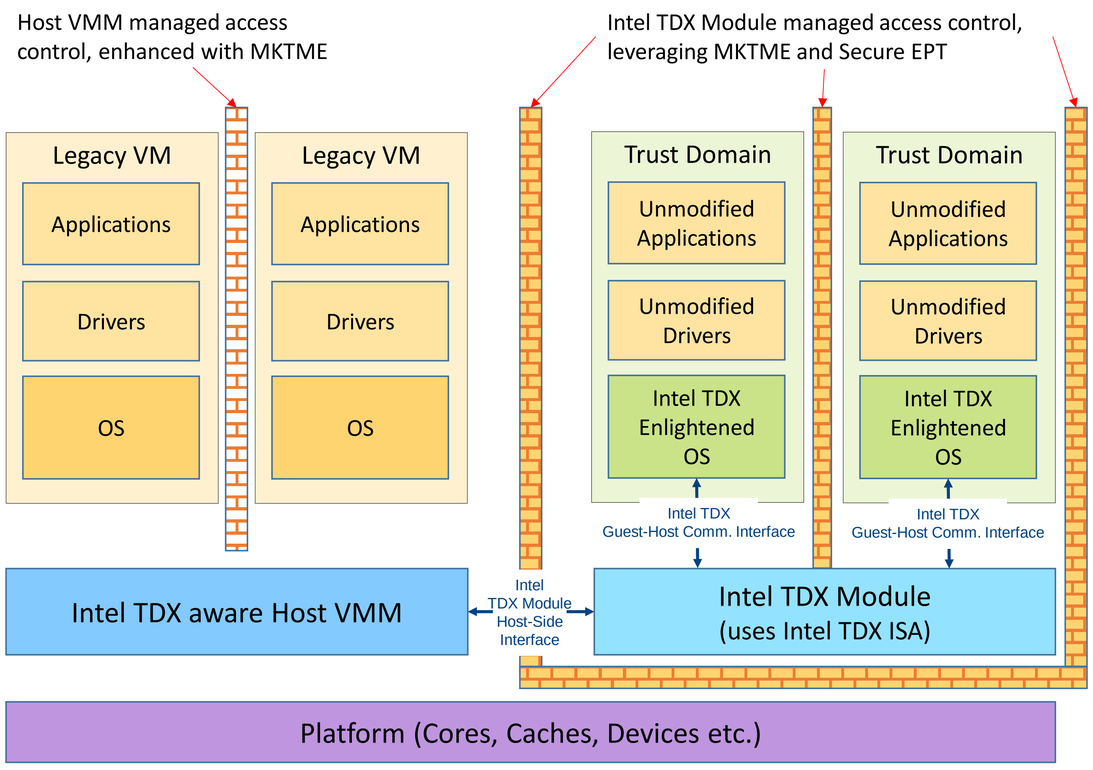
SV-COMP Benchmark: Verifying Intel TDX Module

Dirk Beyer[1], Po-Chun Chien[1], Nian-Ze Lee[2,1], & Thomas Lemberger[1]

[1]LMU Munich [2]National Taiwan University

Intel **T**rust **D**omain **E**xtensions



Source: Fig. 2-1 in Intel TDX Module v1.5 Base Arch. Spec. [2]

ABI Specifications

- TDs and VMM communicate through **A**pplication **B**inary **I**nterfaces
- **Goal**: Verify TDX ABIs (implemented in C + assembly) adhere to the specification under all inputs

Table 5.145: TDH.MNG.CREATE Input Operands Definition

Operand	Description		
RAX	SEAMCALL instruction leaf number and version, see 5.3.1		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the SEAMCALL interface function
	23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
	63:24	Reserved	Must be 0
RCX	The physical address of a page where TDR will be created (HKID bits must be 0)		
RDX	**Bits**	**Name**	**Description**
	15:0	HKID	The TD's ephemeral private HKID
	63:16	Reserved	Reserved: must be 0

Table 5.146: TDH.MNG.CREATE Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code – see 5.3.1
Other	Unmodified

Example: Specification of ABI `TDH.MNG.CREATE` [1]

Defining Verification Tasks

```yaml
- name: tdh_mng_create__requirement__expected
  target:
    filename: formal/harness/tdh_mng_create_harness.c
    method: tdh_mng_create__valid_entry
  before_target:
    - filename: formal/src/initialization.c
      method: init_tdx_general
    - filename: formal/src/initialization.c
      method: init_vmm_dispatcher
    - filename: formal/harness/tdh_mng_create_harness.c
      method: tdh_mng_create__common_precond
  after_target:
    - filename: formal/harness/tdh_mng_create_harness.c
      method: tdh_mng_create__common_postcond
  properties:
    - property_file: unreach-call.prp
      expected_verdict: true
```

Nondet Initialization of `struct`

Havoc memory: by assigning a nondeterministic value to each byte

Havoc object: by nondeterministically initializing each field of the type (if a field is a non-primitive type, recursively initialize it)

Verifier builtin: e.g., `__CPROVER_havoc_object` in CBMC

```c
void _NONDET_struct_tdvps_t(tdvps_t* dest) {
  _NONDET_custom_type(dest, sizeof(tdvps_t));
}
void _NONDET_custom_type(void* base, unsigned int size) {
  for (int i = 0; i < size; i++)
    *((char*)base + i) = _NONDET_uint8t();
}
```

Initialization by havocking memory

```c
void _NONDET_struct_tdvps_s(struct tdvps_s *dest) {
  _NONDET_struct_tdvps_ve_info_s(&((*dest).ve_info));
  _NONDET_array_1D_unsigned_char(&((*dest).reserved_0), 128);
  // ... snipped ...
}
void _NONDET_array_1D_unsigned_char(unsigned char (*dest)[],
                                    int dim0) {
  for (int i = 0; i < dim0; i++)
    (*dest)[i] = _NONDET_uchar();
}
```

Initialization by havocking object

Verification Harnesses

- Initialize global data and assume preconditions
- Mock access to externally defined data and model inline assembly
- Check postconditions

```c
_STATIC_INLINE_ tdx_module_local_t *get_local_data(void) {
#ifdef TDXFV_NO_ASM
  return &local_data_fv;
#else
  uint64_t local_data_addr;
  _ASM_("movq %%gs:%c[local_data], %0\n\t"
        : "=r"(local_data_addr)
        : [local_data] "i"(offsetof(
            tdx_module_local_t, local_data_fast_ref_ptr)));
  return (tdx_module_local_t *)local_data_addr;
#endif
}
```

Contributions

- 290 tasks from 16 host-side ABIs (`TDH`) and 5 guest-side ABIs (`TDG`)
- 🔧 HarnessForge (`gitlab.com/sosy-lab/software/harnessforge`):
  - Assembles single-file verification tasks from real-world C projects
  - Slices off code irrelevant to verification tasks
- Next steps:
  - Experiment with effect of different initialization strategies
  - Develop more tooling to support harness generation (e.g., harness-specific linter)
  - Establish custom annotations for initializing complex types in SV-COMP community

More Information

Intel TDX Module

Verification tasks

HarnessForge

This work is supported by a research gift from Intel.

References

[1] Intel TDX Module v1.5 ABI Specification, `https://www.intel.com/content/www/us/en/content-details/795475/intel-tdx-module-v1-5-abi-specification.html`, accessed: 2024-05-01

[2] Intel Trust Domain Extensions, `https://www.intel.com/content/www/us/en/developer/tools/trust-domain-extensions/documentation.html`, accessed: 2024-05-01