Running mushroom on Intel TDX

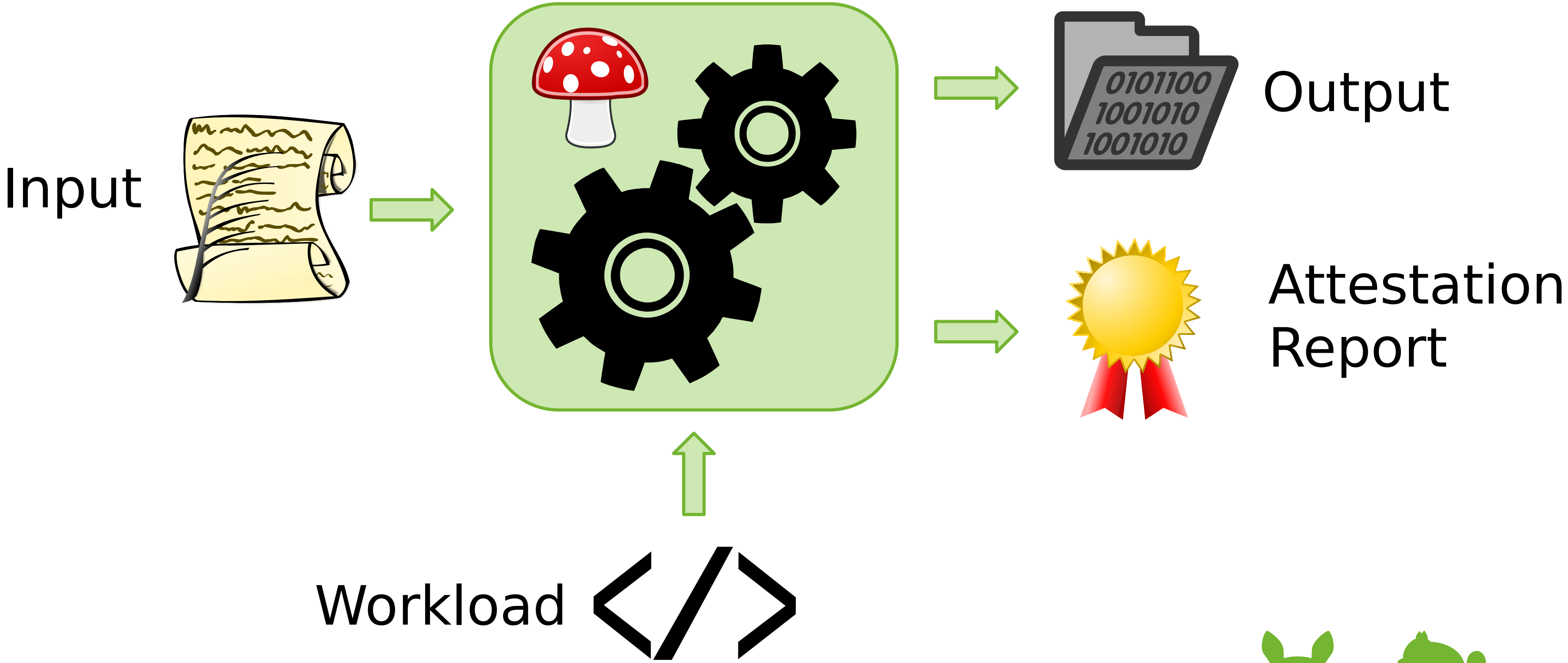
Outline

- Brief introduction to mushroom 🍄
- Mushroom’s “supervisor” architecture
- Intel TDX - TD partitioning
- Supervisor implementation on TDX

whoami

- Tom Dohrmann aka freax13
- Firmware Engineer/Developer/Security Researcher
- ❤ Rust
- ❤ OSDev
- Opinions are my own.

Overview



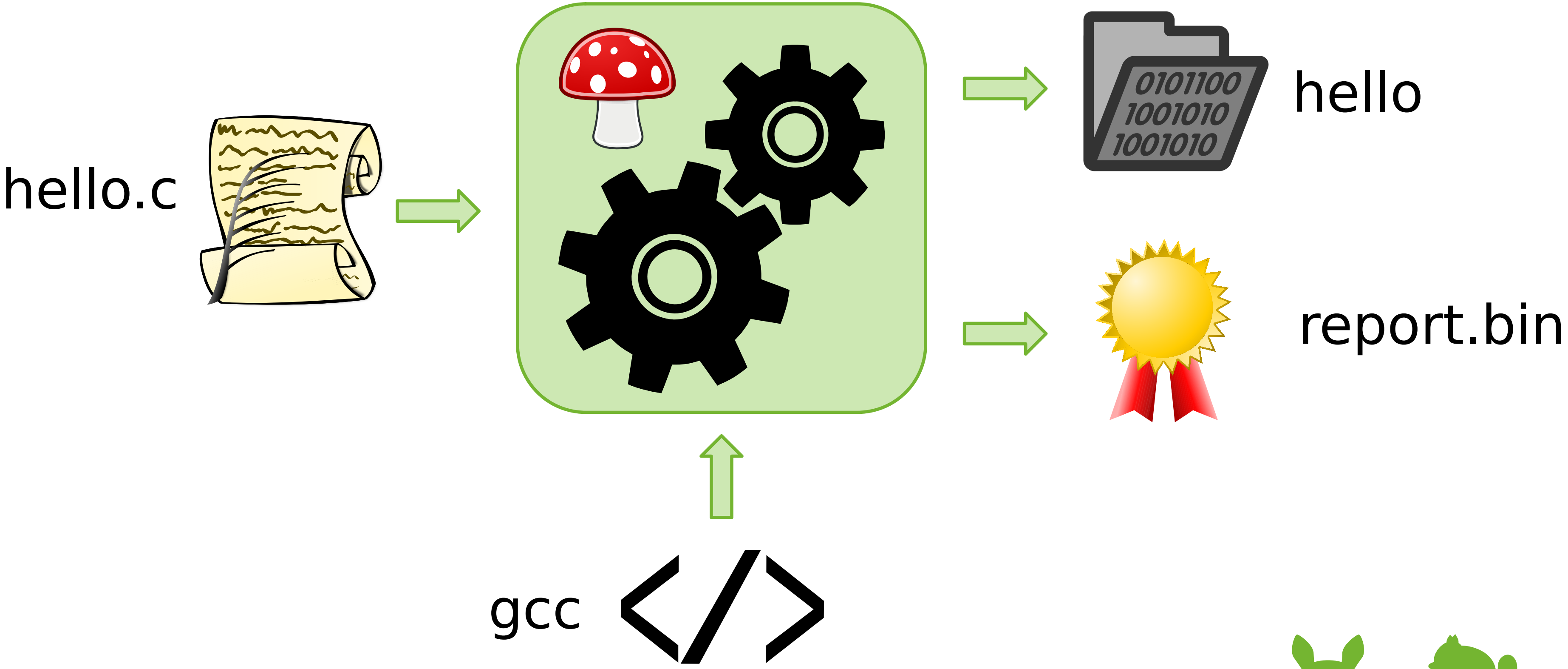
Input

Output

Attestation Report

Workload

Example



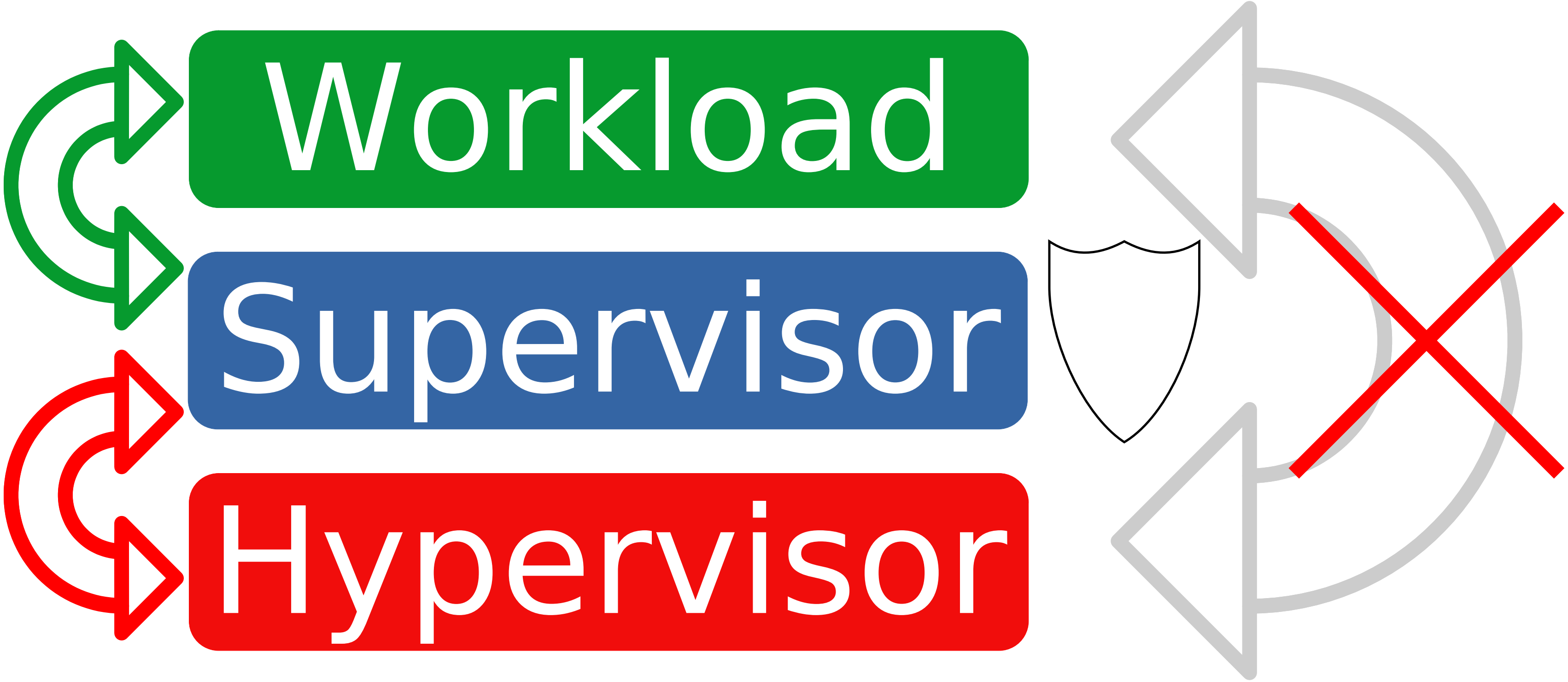
hello.c

hello

report.bin

gcc

Supervisor



Workload

Supervisor

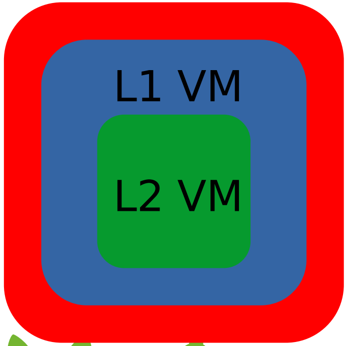
Hypervisor

Supervisor

- Supervisor acts as a “firewall” between the workload and the host by sanitizing all host data.
  - Only the supervisor needs to be audited.
- The supervisor is much smaller (1.8k/2.3k LoC) than the workload kernel (31.4k LoC).
  - Minimal TCB
- The supervisor is hardened against attacks (shadow stacks, CET IBT, no heap, immutable page tables).

Intel TDX - TD Partitioning

- TD partitioning can be used spawn a nested L2 VM.
- The L1 VM (supervisor) acts as a VMM for the L2 VM (workload).
  - It can restrict memory accesses.
    → It can prevent the L2 VM from accessing host memory.
    → L2 VM’s kernel code is immutable.
  - Securely injects interrupts/IPIs.



supervisor control flow

1. Load and verify workload input.
2. Make kernel and init binary memory accessible for L2 VM.
3. Wait for startup signal. vCPU 0 starts immediately.

4. Evaluate interrupt state and inject if needed.
5. Enter L2 vCPU.
6. Handle guest exit or “supervisor call”.
7. Repeat.

8. Create attestation report.

Verifying the Input

- Input file should be part of the attestation report.
  - Neither the supervisor nor the workload kernel should be able to change the identity of the input file in the report.
- Input file shouldn’t be part of the launch measurement.
  - mushroom also supports other TEEs, so we want to avoid tying the input file to architecture specific details.
- MRCONFIGID is initialized with a hash of the input file.
  - Can’t be changed after VM launch.
  - Supervisor verifies that the input file matches the hash.

Updating L2 memory access

- L2_CTLS.ENABLE_SHARED_EPTP = False
- TDG.MEM.PAGE.ATTR.WR updates permissions for a page.
- L2 can access memory only when VALID bit is set.
- Read/Write/Execute Supervisor/Execute User/PW/VPW

- Workload kernel: Valid, Read & Execute Supervisor
- Init binary & input file: Valid & Read
- Hot-plugged memory: Valid, Read, Write & Execute User

Running the Workload

- TDG.VP.ENTER starts running an L2 vCPU and provides information about the guest on exit.

- CPUID emulation
- “Supervisor calls”
  - scheduling, memory hot-plug, output
- IPI emulation & timer interrupts

Attestation

- TDG.MR.REPORT creates a TD report.

- Code → MRTD
- Input → MRCONFIGID
- Output → REPORTDATA

Attestation

- TDG.MR.REPORT creates a TD report.

- Code → MRTD (← Immutable after launch)
- Input → MRCONFIGID (← Immutable after launch)
- Output → REPORTDATA

- QGS converts TD report into TD quote.

Thanks!

Questions?

github.com/freax13/mushroom

FOSDEM 24 - Integrity Protect Workloads with Mushroom

blog.freax13.de