Confidential Compute on RISC-V platforms

Ravi Sahita
Rivos Inc.
Background
Confidential Computing

Confidential Computing is the protection of data-in-use by performing computation in a Hardware-based Trusted Execution Environment (TEE)

- This definition is independent of topological location, which processor does it, and whether encryption or some other isolation technique is used.
- The protection of data in use is against a well-defined adversary.

TEE properties verified via HW-rooted attestation of the Trusted Computing base
Threat Model

- Read of confidential data in memory/CPU register state
- Tamper of confidential data in memory/CPU register state
- Tamper of MMU in-memory structures (address mappings)
- Invalid execution of code handling confidential data
- Shared memory
- Read/Write of confidential data in memory via I/O devices
- Spurious generation, tamper of interrupt delivery
- Timestamp, Debug & Performance monitoring
- Denial of service
- Exposure of data via side channels (arch and u-arch)
- Attestation-oriented (boot, crypto and protocol)
- Operational feature attacks (migration)
- <others>

All current TEE approaches address aspects of this threat model

A Ref. Arch for RISC-V is proposed that has:
- non-ISA interfaces
- ISA extensions
- platform guidelines
- Uses RISC-V H-extension*. Specifies ISA-extensions e.g. Physical Memory Attributes for Confidential memory

- Introduces TEE Security Manager (TSM) - Support different implementations, deployment models and process/VM workloads

- TSM Interfaces TH-ABI and TG-ABI** abstract platform differences

Provide guidelines for non-normative [implementation specific] stuff

---

*https://github.com/kvm-riscv/howto/wiki/KVM-RISCV64-on-QEMU

Deployment Models

- Both deployment models require H extension, and TSM interfaces for:
  - Confidential memory mgmt.
  - vCPU mgmt, context isolation
  - Measurement and Attestation
  - IO assignment
- Platform-specific Implementation aspects remain the same e.g. memory encryption
- Deployment model 2 requires additional HW functions e.g. Confidential Memory PMA
- Additionally - implementations can further reduce M-mode TCB with future HW support
# TH-ABI Plumbing (Interfaces: 1)

<table>
<thead>
<tr>
<th>Task</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get TSM Info e.g. memory pages to set aside per TVM</td>
<td><code>sbi_tee_tsm_get_info</code></td>
</tr>
<tr>
<td>Create/Destroy TVM</td>
<td><code>sbi_tee_create/destroy_tvm</code></td>
</tr>
<tr>
<td>Convert a memory region from Non-Confidential to Confidential</td>
<td><code>sbi_tee_mem_convert_pages</code></td>
</tr>
<tr>
<td>Issue fence for global/TVM TLB invalidations</td>
<td><code>sbi_tee_mem_initiate/local_fence</code></td>
</tr>
<tr>
<td>Assign TVM memory region to confidential, shared or MMIO</td>
<td><code>sbi_tee_tvm_add_shared/conf/mmio_region</code></td>
</tr>
<tr>
<td>Add confidential page mappings</td>
<td><code>sbi_tee_tvm_add_page_table_pages</code></td>
</tr>
<tr>
<td>Add a measured TVM page</td>
<td><code>sbi_tee_tvm_page_add_measured_pages</code></td>
</tr>
<tr>
<td>Add a zero TVM page (lazy add)</td>
<td><code>sbi_tee_tvm_page_add_zero_pages</code></td>
</tr>
<tr>
<td>Blocks page mappings for TVM page(s)</td>
<td><code>sbi_tee_tvm_page_range_block</code></td>
</tr>
<tr>
<td>Unblocks page mappings for TVM page(s)</td>
<td><code>sbi_tee_tvm_page_range_unblock</code></td>
</tr>
<tr>
<td>Relocate a page for an existing mapping for a TVM page</td>
<td><code>sbi_tee_tvm_page_relocate</code></td>
</tr>
<tr>
<td>Promote/Demote a (set of) page mappings for a TVM</td>
<td><code>sbi_tee_tvm_page_promote/demote</code></td>
</tr>
<tr>
<td>Reclaim confidential pages</td>
<td><code>sbi_tee_tvm_page_reclaim</code></td>
</tr>
<tr>
<td>Description</td>
<td>Function</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>Create a TVM vcpu context</td>
<td>sbi_tee_tvm_vcpu_create</td>
</tr>
<tr>
<td>Execute a TVM vcpu context</td>
<td>sbi_tee_tvm_vcpu_run</td>
</tr>
<tr>
<td>Others aspects that will require interfaces:</td>
<td>WIP</td>
</tr>
<tr>
<td>Secure interrupts, TEE IO, TEE workload</td>
<td></td>
</tr>
<tr>
<td>migration etc.</td>
<td></td>
</tr>
</tbody>
</table>
Linux/KVM Mapping

TVM creation requires some additional operations in addition to the ordinary VM creation.

- Get KVM system capability to check if AP-TEE is supported on the RISC-V platform
- New: Get the TSM info

- VM creation (KVM_CREATE_VM)
- New: Set TVM parameters when creating it via modified KVM_INIT_VM.

- VCPU creation (KVM_CREATE_VCPU)
- New: Set TVM-specific VCPU parameters. KVM_TVM_INIT_VCPU.

- Assign (shared) memory to the TVM (e.g. for virtIO)
- New: Assign confidential guest memory and extend the TVM measurement as memory contents are loaded

- New: Finalize TVM so it can be executed

- VCPU RUN (KVM_VCPU_RUN)
- New: Schedules TVM vcpu via TSM TH-ABI
TVM memory mgmt requires some additional operations in addition to the ordinary VM memory mgmt.

Proposal for RISC-V in discussion - **new confidential physical memory attribute enforced by the CPU as part of the MMU (and IOMMU)**

**Post-TVM creation:**
- On-demand assignment of memory
  - New: Only zero memory may be added for not-present page mappings
- Modification of mappings
  - New: Block new TLB mappings via making mapping not-present
  - Modify mapping (relocate, promote, demote etc.)
  - New: Flush old cached mappings (via TSM)
- New: handle new fault conditions during TVM execution
  - Access violation from untrusted host (from MTT)
  - Violation due to code fetch or page walk in non-confidential memory
TG-ABI Plumbing (Interfaces)

<table>
<thead>
<tr>
<th>Function Description</th>
<th>Function Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get the platform attestation capabilities e.g. crypto hash supported</td>
<td>sbi_attestation_get_capabilities</td>
</tr>
<tr>
<td>Get the TCB attestation evidence e.g. X.509 certificate</td>
<td>sbi_attestation_get_evidence</td>
</tr>
<tr>
<td>Extend a measurement register</td>
<td>sbi_attestation_extend_measurement</td>
</tr>
<tr>
<td>Read a measurement register</td>
<td>sbi_attestation_read_measurement</td>
</tr>
<tr>
<td>Invoke untrusted host service e.g. virtIO</td>
<td>sbi_tee_req_vmm_svc</td>
</tr>
</tbody>
</table>

Others to be added for scenarios like IO, Migration etc.

- Work in progress proposal at [https://github.com/sameo/riscv-sbi-doc/pull/1](https://github.com/sameo/riscv-sbi-doc/pull/1)
- PoC at [https://github.com/rivosinc/salus](https://github.com/rivosinc/salus)
Call to Action

● Review and provide feedback on APTEE TH/TG-ABI (discussions on the RVI AP-TEE TG list)
  ○ See discussion of the interfaces here -
    https://docs.google.com/presentation/d/14-rFP23zEdP2t6A9D40J6e3iPYqF7sk0AuqV47OLVEw/edit#slide=id.p

● Join POC efforts for TEE Security Manager (TSM) for RISC-V implementing TH/TG-ABI
  https://github.com/rivosinc/salus

● Extend RISC-V-KVM to interface with proposed TH/TG-ABI -- active task in AP-TEE TG and RVI Hypervisor SIG

● Develop common test cases to evaluate compatibility for Linux/KVM TVM guests across different architectures and scenarios
All deployment models require priv. ISA w/ H extension, and TSM TEEI for:
- Confidential memory mgmt.
- vCPU mgmt, context isolation
- Measurement and Attestation
- IO assignment

Platform-specific implementation aspects remain the same e.g. memory encryption

Deployment model 2 requires additional HW function e.g. Confidential Memory PMA

Deployment models 3 require additional HW/SW function e.g. M-mode TCB reduction