arm

TF-A Technical Forum

Testing TF-A Measured Boot with a TPM service

Javier Almansa Sobrino

October 2020

Background

Measured Boot needs to be supported by a TPM which is used to securely extend and hold the image measurements.

- The TPM must comply with the TPM 2.0 Specification
- It can be provided as a silicon or a (secure) software service

We need a mechanism to test and validate that the current implementation of Measured Boot in TF-A can interact with a TPM 2.0 compliant device/service

What are we testing

(and what not)

We need to test that the Measured Boot driver can generate a TPM 2.0 compliant event log:

- TF-A must be able to pass the event log to the TPM service securely.
- The TPM service must be able to parse the event log and extract the relevant information from it.
- All the events present on the event log must be successfully extended by the TPM.

We are not validating the TPM records:

- We are not providing a TPM service. We are only interested on test the interoperability between Measured Boot and a TPM 2.0 compliant device.
- Verification of a TPM service is beyond the scope of this test case.
- We do not make assumptions about the correctness of a particular (f)TPM service.

Design considerations

TPM Service implementation

Different approaches were considered on how to implement the TPM Service:

- As an SPCI Secure Partition (as suggested on *Architecture for a Secure Partition TPM Service and TF-A/EDK2 Measured Boot, v0.6 document*).
- As a MM Secure Partition in case SPCI Secure Partition support was not fully supported in time on TF-A.
- As a Trusted Application running on OPTEE.

A decision to implement the TPM service as a Trusted Application running on OPTEE was finally made*:

- OPTEE is widely used by many partners.
- There is already a TPM reference implementation by Microsoft which is TPM 2.0 compliant. The implementation includes a port for ARM32 architecture.

***NOTE**: The scope of the TPM service is just to test the integration of Measured Boot with a real-world TPM service. It is meant to be used only for testing and as a reference implementation.

MS TPM 2.0 Reference Implementation

- Official TCG reference implementation for the TPM 2.0 Specification.
- Available on Github:
  - https://github.com/microsoft/ms-tpm-20-ref
- It consists on a library implementing the TPM 2.0 specification with a few example applications built around that library:
  - A simulator application to run on the host machine showing the TPM functionality.
  - Sample applications with different boards/architectures (Nucleo boards, ARM32).

Requirements

OPTEE (I)

During system boot, BL2 needs to keep an event log in secure memory as the TPM service will not be available yet.

The TPM service then needs to be able to access the event log upon start, so it can extend all its records

- OPTEE cannot pass dynamic configuration/parameters to a Trusted Application, so support for this needed to be added.
- The best approach was to add a new service to OPTEE' system PTA, so a TA can request a copy of the TPM event log provided by TF-A.

```
185  /*
186   * Retrieves a copy of the TPM Event log held in secure memory.
187   *
188   * [out]    memref[0]: Pointer to the buffer where to store the event log.
189   */
190  #define PTA_SYSTEM_GET_TPM_EVENT_LOG    12
```

Requirements

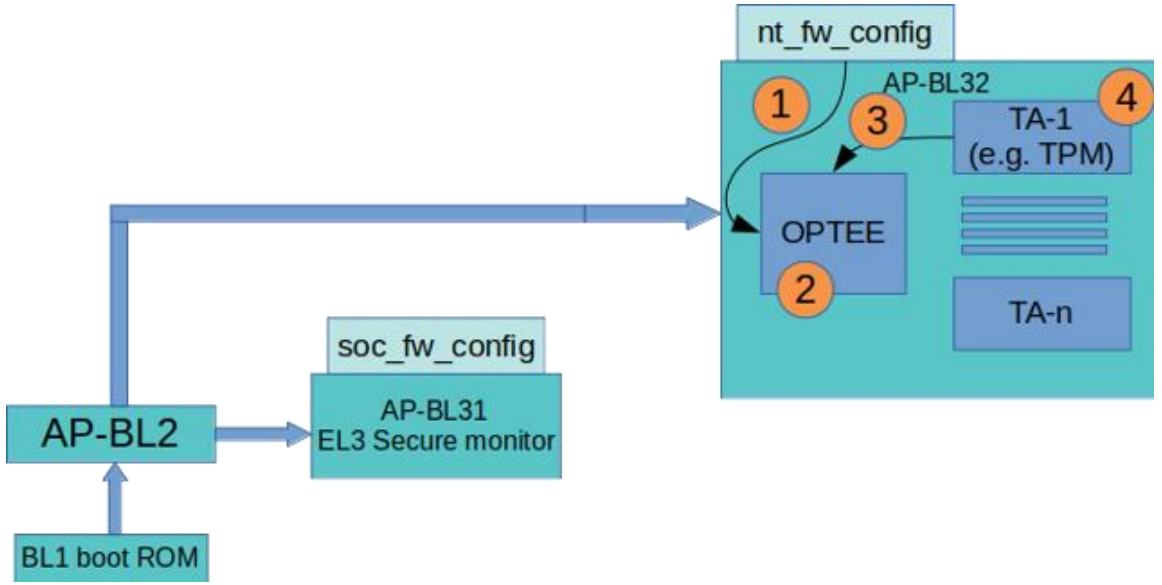
OPTEE (II)

The event log information will be passed from BL2 to OPTEE through a DTB (nt_fw_config) and it will be formed of

- Start address of the TPM Event Log in secure memory.
- Length of the event log.

OPTEE will then map the Event Log internally and it will provide a copy of the whole event log to the TPM service upon request.

NOTE: BL32 should receive the tos_fw_config DTB, however OPTEE does not support the use of DTBs in secure memory, therefore this usecase would pass nt_fw_config instead. OPTEE will then zero the DTB properties with the event log information, to minimize any security risk.



1. During its initialization, OPTEE parses the DTB (nt_fw_config) and extracts the information related to the event log.
2. OPTEE internally maps the secure memory area where the event log is. It then sets to zero the address property in the DTB to avoid leaking it to the NS world.
3. During its initialization, the TPM service request a copy of the event log through the system PTA service. OPTEE will copy the content of the event log inside the buffer passed on the call and it will also return the size of the whole event log.
4. The TPM service will parse the event log and will extend all the events registered on it.

Requirements

fTPM Service

The ARCH32 implementation of the OPTEE TA included on Microsoft's TPM repository is outdated and not aligned with mainline, therefore the build fails.

- We updated the ARCH32 OPTEE TA to catch up with the master branch and now the application builds and runs.*

The fTPM service does not have support to receive and extend an event log.

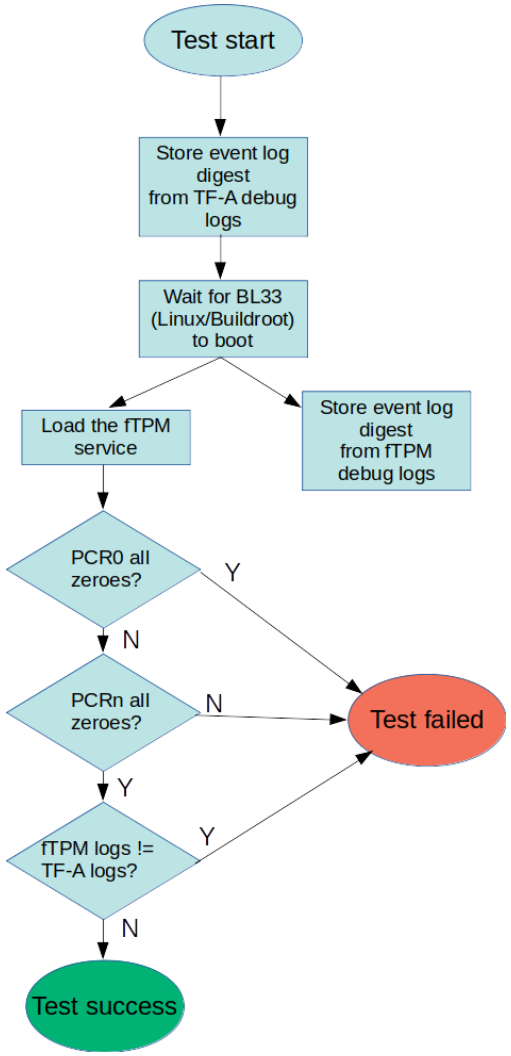
- We added support to the service to receive (via the new PTA_SYSTEM_GET_TPM_EVENT_LOG call on OPTEE) an event log and extend it as part of the fTPM boot process.*

*At the moment, the changes are private and ongoing review prior to be merged to Microsoft's repository.

CI Test pass criteria

CI tests for Measured Boot + fTPM integration verifies that:

- A TPM service can receive the event log as generated by TF-A
  - TF-A and the fTPM print the event logs and the test script verifies that the digests printed by both entities are the same.
- The event log is successfully processed by the fTPM service
  - No errors are generated by the fTPM service when initialized.
  - PCR0 is not all zeroes.
- The event log is well formed
  - PCR0 is not all zeroes.
  - PCR[1,10] are all zeroes.



Test results

fvp-mb-256-optee-romlib:fvp-optee.mb-linux.rootfs+ftpm-romlib-fip.ftpm-aemv8a

```
Welcome to Buildroot, type root or test to login
buildroot login: root
# ftpm
[  290.312394] random: crng init done
count 1 pcrUpdateCounter 30
 digest length 32
 76 b6 bd 85 f8 fe 43 0f 88 db 9e 44 29 07 38 f8
 dc 7d 60 08 12 cb 66 8b ce ac 81 38 20 77 20 84
# pcrread -ha 1
count 1 pcrUpdateCounter 30
 digest length 32
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
pcrread -ha 2
# pcrread -ha 2
count 1 pcrUpdateCounter 30
 digest length 32
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
pcrread -ha 3
```

```
pcrread -ha 9
# pcrread -ha 9
count 1 pcrUpdateCounter 30
 digest length 32
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
pcrread -ha 10
# pcrread -ha 10
count 1 pcrUpdateCounter 30
 digest length 32
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
# spawn diff -s tfa_event_log ftpm_event_log
Files tfa_event_log and ftpm_event_log are identical

Test success!
000: tf-l1-boot-tests-misc/fvp-mb-256-optee-romlib:fvp-optee.mb-linux.rootfs+ftpm-romlib-fip.ftpm-aemv8a: SUCCESS
```

Pending work

1. At the moment, the contributions made to the fTPM TA has not been yet merged to Microsoft repository.
   - We are undergoing internal review of the patches before we can push them externally.
2. The Build system for the FVP images to test the Measured Boot + fTPM is based on the OPTEE Toolkit. We are working to push the changes to OPTEE repository to make them available.
3. After the above has been completed, we plan to update TF-A documentation with instructions to build all the necessary FVP images to manually test the implementation and/or use it as a reference design.

Pending work

(cont)

4. Any event logged on the event log by a NS image (e.g BL33 and beyond) will not be extended by the current implementation of the fTPM.

- The fTPM is not available until an REE (for instance, Linux) is available, so any measurement between BL33 and the REE will be included on the event log in NS Memory.
  - However, the fTPM cannot receive an event log from NS Memory and
  - NS images must not modify the event log on Secure Memory.

Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
ধন্যবাদ
תודה