Measured Boot Driver

Refactor and Critical Data measurement design

Manish Badarkhe
02-12-2021

Agenda

- Measured Boot driver Enhancement
  - Previous state of Measured Boot driver
  - Refactored the Measured Boot driver
- Gaps in Measured Boot driver
- Critical data for firmware and its measurement design

Previous state of Measured Boot driver

- Measured Boot functionality was tightly coupled to the Event log driver backend

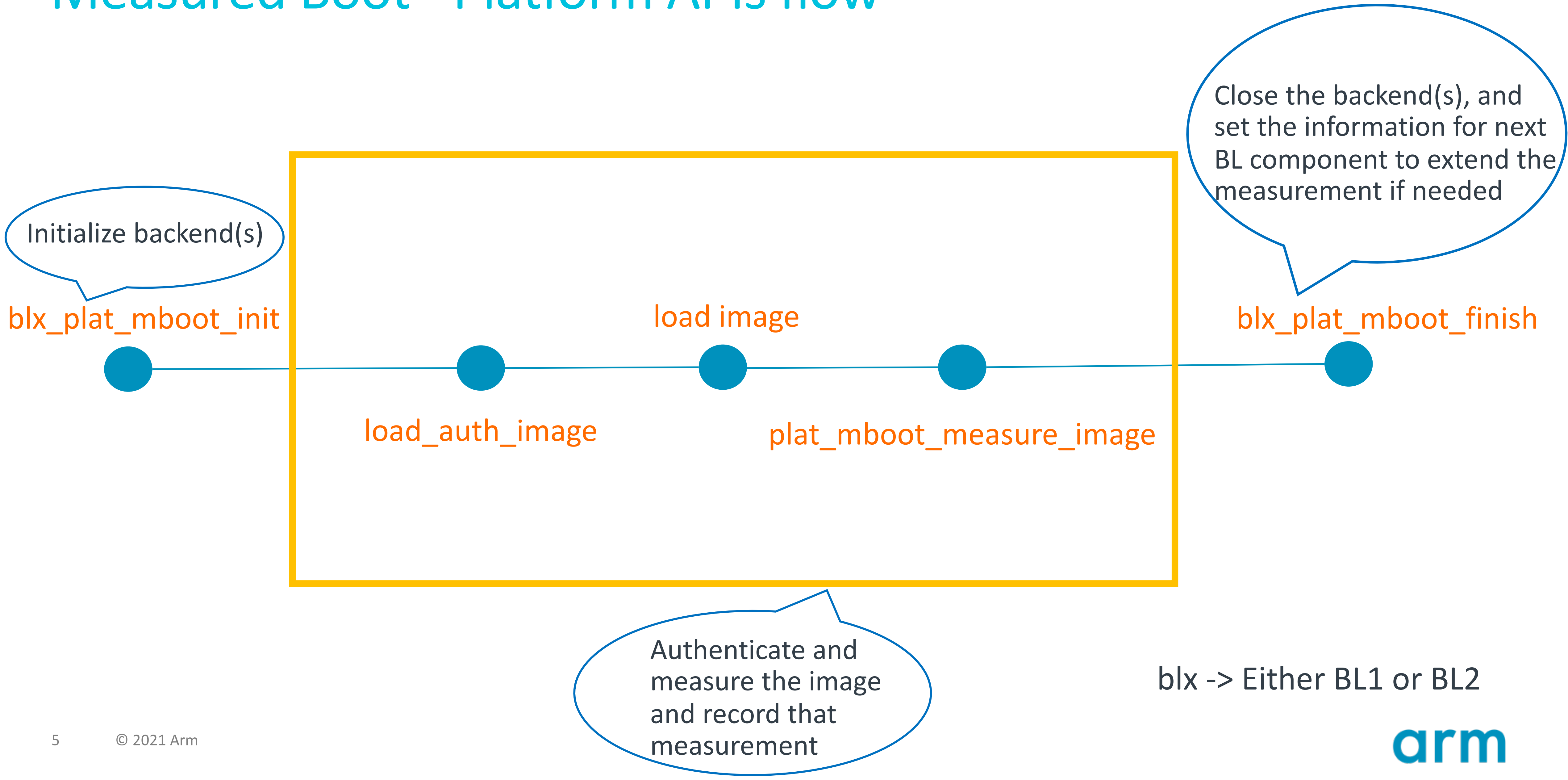
- No capability to record the measurement of images loaded by BL1 in BL1 itself

- Event Log driver was doing work for the platform by calling platform functions

Refactored the Measured Boot driver

- Structured the Measured Boot driver to scale it with multiple backend drivers in the future
- Move image measurement in the generic layer, just after loading and authentication of the image
- Record the measurements of images loaded by BL1 in the BL1 itself
- Pass Event Log buffer information from BL1 to BL2 so that the TCG Event Log buffer initialised by BL1 extended further with the measurements taken by BL2
- Removed platform function calls from Event Log driver to make it self-contained driver
- Patch stack: MeasuredBoot-Refactor and console log EventLog

Measured Boot - Platform APIs flow


Initialize backend(s)
blx_plat_mboot_init
load image
load_auth_image
plat_mboot_measure_image
Authenticate and measure the image and record that measurement
Close the backend(s), and set the information for next BL component to extend the measurement if needed
blx_plat_mboot_finish

blx -> Either BL1 or BL2

Gaps in Measured Boot Driver

- Event Log driver directly uses mbed TLS crypto engine, but we may need a different crypto engine for Measured Boot driver in the future. Hence mbedTLS strong dependency is avoided by this patch:

  https://review.trustedfirmware.org/c/TF- A/trusted-firmware-a/+/11878

- Error handling in the Measured Boot driver
  - Platform defined behavior after refactoring the code
  - Some platforms may decide to panic or allow the system to take alternate available mechanisms to do the measurement.

- Critical Data Measurement

Critical Data Measurement

Critical Data Measurement Design

What is critical data?

- Data that helps to maintain security posture during boot and runtime for example, configuration settings, Security policies, debug settings etc.
- Data that influence the boot flow behavior of the software

Critical Data Measurement Design

Type of critical data

- Static critical data
  - Data that is present in NV-storage and software just consumes it
  - At every boot value of this data remains unchanged
  - Measuring this data at any point in the boot flow is possible

- Dynamic critical data
  - Data that gets updated during the runtime execution of the software
    - Deterministic - it gets updated in a certain valid range or always to some fixed value
    - Non-deterministic - can't expect the exact value of such data, its value depends upon hardware response.

  - This data must get measured after the update and before the consumption For example
    - Consider BL1 does hardware probing and updates critical data in device tree, to be consumed BL2 component, then it makes sense to do the measurement of such data in BL2
    - Consider BL1 updates some global data that to be consumed by drivers present in the BL1, then it makes sense to measure this data after update and before that driver uses it

Critical Data Measurement Design

- Do measure
  - Influence boot flow behaviour parameters
  - Configuration settings, Debug settings, and security policies, these parameters should be in a valid state for a device to maintain its security posture during boot and runtime.
  - Deterministic data before update
  - Data that is being updated by hardware (un-deterministic) put its measurement in different PCR.

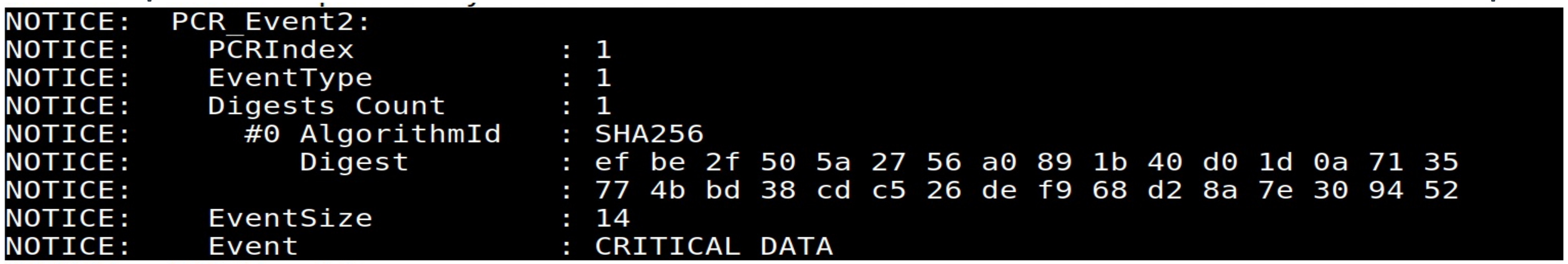
- Do not measure
  - Deterministic data after update
  - Certificates involved in authentication boot flow

Identified Critical Data

Data	Type	Part of	Comments
Accepted flag (trial or regular run)	Dynamic	FWU metadata	Anti rollback counter increments can be avoided if trial-run gets detected instead of regular-run.
Platform NV counter	Static	Platform specific Register/Memory	It may influence boot behaviour as it may start allowing run of malicious/older firmware.
Firmware bank index	Dynamic	FWU metadata	If firmware index gets compromised then it will alter the boot flow (possible miss of some driver and security initialization)
Firmware configurations	Static/Dynamic	Device Tree/Local data structure	Platform needs to decide which static/dynamic properties are critical, and add that particular property as a part of critical data measurement.
Security Policies	Static	Platform specific structures	Set of policies to control IP access by the software affects the security
Debug Settings params	Static	Debug settings	Set debug parameters to enable debug channels affects the security

Critical Data Measurement

- Create a platform defined critical data structure
- Update the members of this structure along with the boot flow (i.e. populate the critical data in global structure).
- Just before jumping to BL31 (Runtime), measure this critical data and record its measurement in the Event Log buffer, as well as any other available backend driver(s) (physical TPM etc)
- Assuming there is no need to support any runtime critical data measurement, that means stop measuring critical data once TF-A BL exits to BL31.
- Patches posted for critical data measurement [Critical Data Patches](), and sample output



```
NOTICE:  PCR_Event2:
NOTICE:    PCRIndex              : 1
NOTICE:    EventType             : 1
NOTICE:    Digests Count         : 1
NOTICE:      #0 AlgorithmId      : SHA256
NOTICE:         Digest           : ef be 2f 50 5a 27 56 a0 89 1b 40 d0 1d 0a 71 35
NOTICE:                          : 77 4b bd 38 cd c5 26 de f9 68 d2 8a 7e 30 94 52
NOTICE:    EventSize             : 14
NOTICE:    Event                 : CRITICAL DATA
```

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