Contribution ID: **112** Type: **not specified**

PCI Component Measurement Architecture / SPDM 1.1

The PCI ECN defining CMA adds the ability (using a DOE mailbox) to establish the identity and verify the component configuration and firmware / executables.

This is done using the protocols defined in the DMTF SPDM 1.1 specification: https://www.dmtf.org/sites/default/files/standards/documen which is also used for the same purpose on other buses such as USB, but we are not aware of any work to support those buses yet. The design is extensible to other buses with an abstracted transport layer (via a single function pointer).

The CMA use of the SPDM 1.1 protocol defines a certificate based public private key authentication mechanism including signed measurements of PCIe component state (firmware and other implementation defined elements) and setup of secure channels for continuing runtime measurement gathering and for other related PCI features such as Integrity and Data Encryption IDE.

An initial implementation will be posted shortly for review, and there are a number of open questions that may benefit from a discussion in this forum:
1) Is there a sufficiently strong case to support CMA natively in the kernel at all?
Some approaches might push this facility into a trusted execution environment. VFs can implement CMA however, to provide this level of authentication and measurement, when in use by a VM. It would be useful to understand other use cases as they motivate the software design and testing.

2) Approach to providing authentication of device certificates? SPDM uses x509 certificates and so relies on a chain of trust. What trust model should we apply? Current code assumes a separate keychain dedicated to CMA and root key insertion from userspace (probably initrd).

3) Method of managing / verifying measurements. The nature of the measurements is implementation defined. In some cases they are not expected to change unless the firmware is flashed, but in others they may change with device configuration. Whilst closely related to the challenges of IMA for files, is it appropriate to reuse that subsystem and tooling?

4) As it's related, is there interest in supporting kernel managed IDE (link encryption)?

5) When do we actually want to make these measurements? (On boot, on driver probe, on reset, on first use of a particular feature, on demand from userspace etc?) Currently they are done on driver probe only.

Other, more detailed questions can be addressed as part of normal discussion on list.

I agree to abide by the anti-harassment policy

I agree

Primary author: CAMERON, Jonathan (Huawei Technologies R&D (UK))

Session Classification: VFIO/IOMMU/PCI MC

Track Classification: VFIO/IOMMU/PCI MC