YARN-3557(Support Intel Trusted Execution Technology in YARN scheduler)

High Level Design Doc

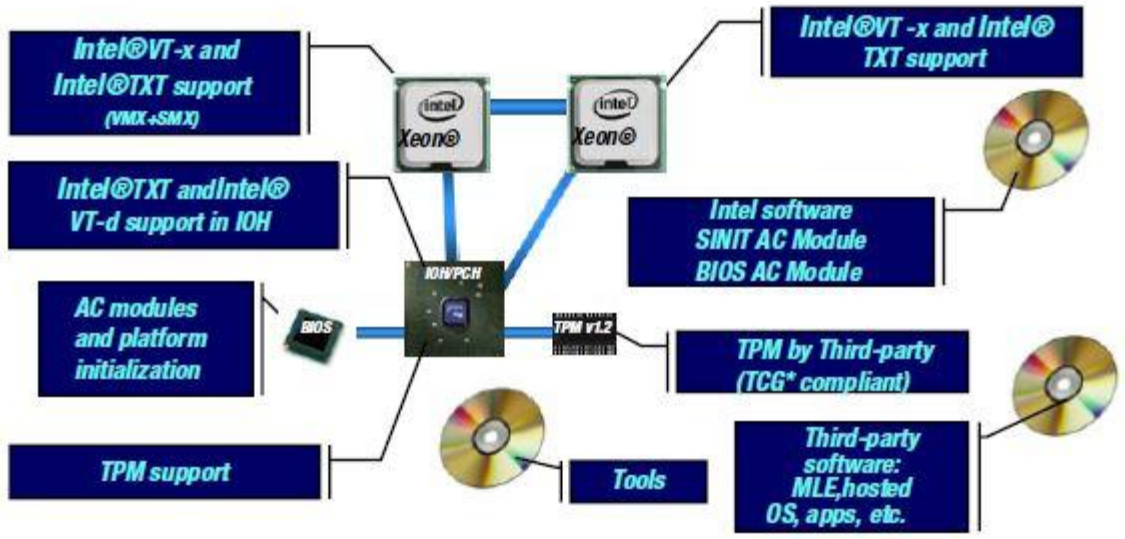
Contents

1. Introduction

1.1. Intel Trusted Execution Technology

Intel Trusted Execution Technology (Intel TXT) defines platform-level enhancements that provide the building blocks for creating trusted platforms. It is a technology that uses enhanced processor architecture, special hardware, and associated firmware that enable certain Intel processors to provide the basis for many new innovations in secure computing. Its primary goal is to establish an environment that is known to be trusted from the very start and further provide system software with the means to provide a more secure system and better protect data integrity. This technology provides discrete integrity measurements that can prove or disprove a software component's integrity. These software components include, but are not limited to, code (such as BIOS, firmware, and operating system), platform and software configuration, events, state, status, and policies.

By providing a hardware-based security foundation rooted in the processor and chipset, Intel TXT provides greater protection for information that is used and stored on servers. A key aspect of that protection is the provision of an isolated execution environment and associated sections of memory where operations can be conducted on sensitive data, isolated from the rest of the system. Likewise, Intel TXT provides for sealed storage where sensitive data such as encryption keys can be securely kept to shield them from being compromised during an attack by malicious code.



Intel TXT platform components

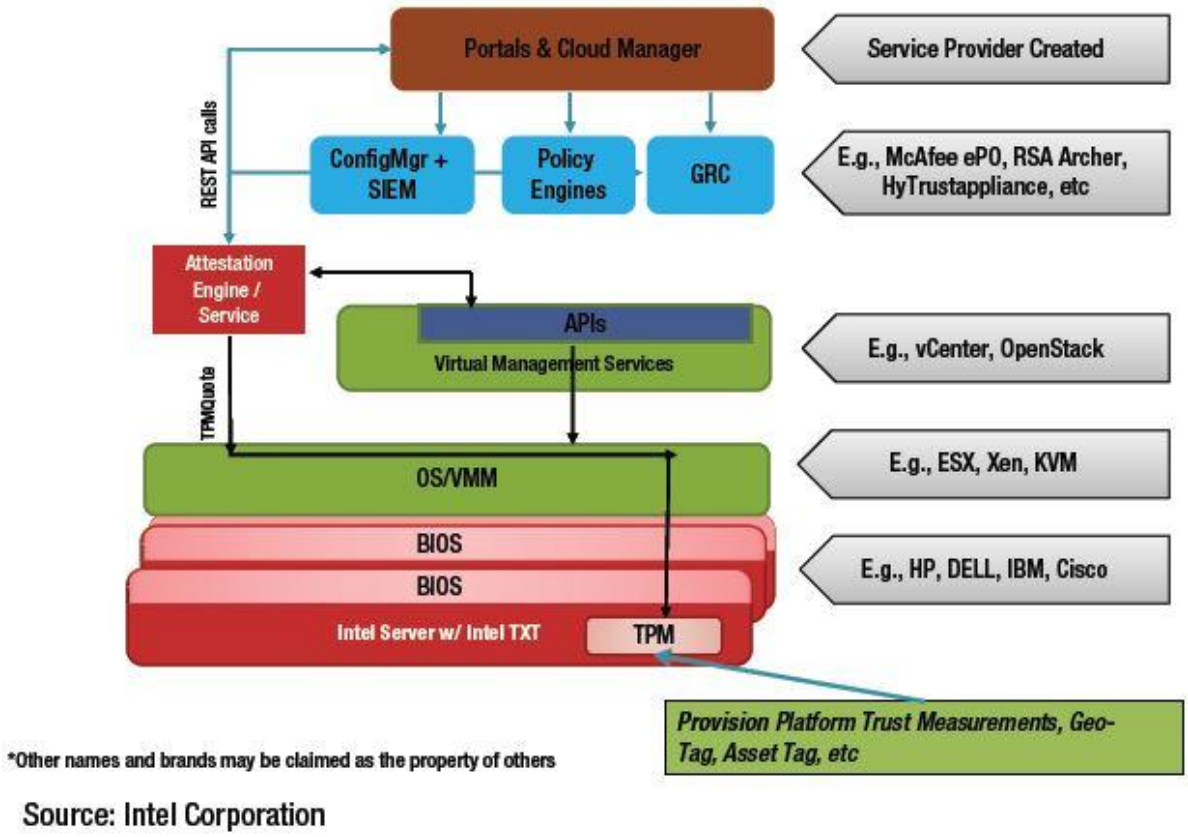
1.2. Remote Attestation

Remote Attestation is a mechanism to provide evidence or proof of some platform operation, value or process. It provides platform trust values to outside entities and drives Intel TXT use models into more scalable, flexible, and operationally valuable assets. Through remote attestation server, a third party can know whether a node is trustworthy.

Currently there are already some remote attestation services available. Open Attestation (OAT) is one of them. It's an Intel-maintained open-source project that is a software development kit (SDK) for managing host integrity verification using TCG-defined remote attestation protocols. The project includes code that was developed by the National Information Assurance Research Lab (NIARL) of the US

National Security Agency—an agency that has a long history of involvement in developing security and trusted computing technologies. The project is available for use and contributions by all; it is hosted on the GitHub repository at https://github.com/OpenAttestation/OpenAttestation.git.

OAT provides RESTful-based Query API and it's already used by OpenStack and oVirt projects. In OpenStack, a component named TrustedFilter was added into Nova scheduler to let trusted hosts via communication communicate with the OAT based attestation service, so a user could create a VM that runs only on trusted hosts.



Attestation Conceptual Architecture

1.3. TXT enablement in Hadoop

The purpose of this JIRA is to enable TXT in Hadoop. An Intel TXT aware YARN scheduler can get the trust status of nodes and schedule the security sensitive jobs (such as Jobs processing sensitive data) on trusted nodes only.

2. Background Knowledge in YARN which can be utilized

2.1 YARN-2492 capability

YARN-2492 provides the capability to restrict YARN applications to run only on cluster nodes that have a specified node label. This mechanism can be utilized for TXT aware YARN scheduler.

Just summarize the capability of YARN-2492 per my understanding as follows:

1) Cluster nodes can be marked with labels which indicate the capacity of the nodes, such as whether a node has GPU available.
2) Queues and Jobs can be marked with labels which indicate the sort of nodes they request.
3) Jobs with labels will only be scheduled to nodes with the required labels

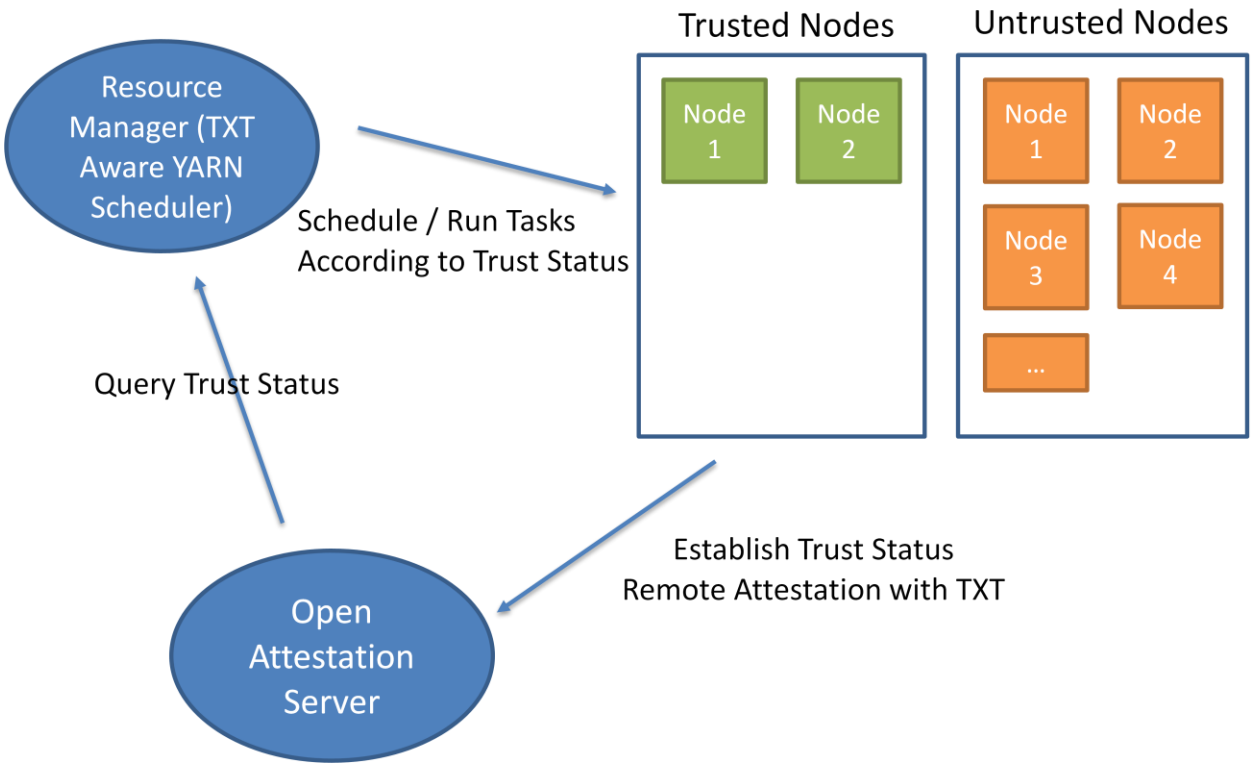
YARN-2492 already provides two mechanisms to configure node labels: centralized node label configuration and distributed node label configuration.

For centralized node label configuration, RM admin can add/remove/remove node labels through CLI.

For distributed node label configuration, YARN-2495 provides a pluggable way to allow admin to specify node labels in each NM. Currently there are two methods available: Configuration based NodeLabelsProvider and Script based NodeLabelsProvider.

3. High Level Design

3.1 Overall Flow



- Nodes establish trust status with Open Attestation Server
- For each cluster node, RM gets its trust status from Open Attestation Server.
- Users submit a job which is marked with a security critical label
- YARN scheduler schedules the job only on trusted nodes if the job is marked as security critical.

3.2 Work need to do

For TXT enablement in Hadoop, we need to mark each NM with a node label according to its trust status.

There are two ways to achieve this:

1. NM retrieve its trust status from OAT and report to RM;
2. RM retrieve the trust status of all cluster nodes from OAT

As labels related to security are very sensitive, it's better to manage these labels through the centralized method. So we choose #2.

To do so, the following gaps need to handle:

1. Currently for centralized node label configuration, it only supports admin configure node label through CLI. Need to provide a mechanism at RM side which can configure node label in the similar way as YARN-2495
2. Currently user can configure centralized node label configuration or distributed node label configuration, but cannot configure both. This feature requires centralized node label configuration. To allow users already choose distributed configuration to use this feature, it's better to allow users to configuration both centralized node label configuration and distributed node label configuration.