CHAPTER 3

Platform Boot Integrity: Foundation for Trusted Compute Pools

In Chapter 2, we introduced the concept of trusted clouds and the key usage models to enable a trusted infrastructure. We provided a brief exposition of the boot integrity usage model, and its applicability across the three infrastructure domains—compute, storage, and network. In this chapter we will take a deeper look into ensuring the boot integrity of a compute platform, which boils down to ensuring the integrity of a number of platform components: the pre-launch and launch components covering firmware, BIOS, and hypervisor. Boot integrity is foundational in embodying the concept of a trusted infrastructure.

This chapter provides an introduction to the concept of roots of trust in a trusted computing platform, the measured boot process, and the attestation that are critical steps for ensuring boot integrity. It also provides an overview of Intel's Trusted Executed Technology (TXT), an example of root of trust technology for asserting platform boot integrity. Complementary to this is the concept of *trusted compute pools*, which is a logical or physical grouping of computing platforms with demonstrated platform boot integrity. Trusted compute pools embody the integrity of the virtual infrastructure, which can then enable granular controls, an essential requirement for virtualized data centers. Here, also, we present a solution reference architecture for building a trusted compute pool in a virtualized data center, and provide a case study of its implementation at the Taiwan Stock Exchange, with a number of typical use cases and the solution components of a successful implementation of trusted compute pools.

The Building blocks for Trusted Clouds

Organizations using or planning to use cloud services are starting to require that cloud service providers offer improved security at the hardware layer and greater transparency of system activities within and below the hypervisor. This means that cloud providers should be able to:

- Give organizations greater visibility into the security states of the hardware platforms running the IaaS for their private clouds.

- Produce automated and standardized reports on the configuration of the physical and virtual infrastructure hosting the customers' virtual machines and data.
- Set policy concerning the physical location of the servers on which the virtual machines are running, and control of the placement and migration of these virtual machines to acceptable locations based on such policy specifications (as some FISMA and DPA requirements dictate).
- Provide measured evidence that their services infrastructure complies with security policies and meets regulated data standards.

What is needed is a set of building blocks for the development of "trustworthy clouds." These building blocks consist of:

- A chain of trust rooted in hardware that extends to the hypervisor.
- A hardening of the virtualization environment using known best methods.
- Provision of visibility for compliance and audit purposes.
- Trust as an integral part of policy management for cloud activity.
- A leveraging of infrastructure and services to address data protection requirements.
- Automation to bring it all together and achieve economies of scale and management efficiency.

Cloud providers and other members of the IT community are carrying out research and development to address this need. A growing ecosystem of technology companies is collaborating to develop a new, interoperable trusted computing infrastructure. The goal is to reduce the risk of attack, such as come from virtual rootkits, by building a hardware-based root of trust founded on the assumption that a hardware-based, bottom-up approach can make this infrastructure more impervious to exploits than does today's mostly software-based approach.

Platform Boot Integrity

As described in the previous chapter, a trusted computing platform is said to have platform boot integrity—or boot integrity, for short—if the key controlling components (namely firmware, BIOS, and hypervisors) have demonstrated integrity. Two steps are needed to assert the integrity of the pre-launch and launch components:

1. A measured boot process.
2. Assurance and enforcement of the executed components as trusted components. This process is called attestation; without this, there is no assurance that the platform is in a trusted state.

Before we describe these two steps, we have to look at roots of trust on a platform, as this is fundamental to a trusted computing platform.

Roots of Trust–RTM, RTR, and RTS in the Intel TXT Platform

Hardware-based roots of trust, when coupled with an enabled operating system, hypervisor, and solutions, lay the foundation for a more secure computing platform. This secure platform ensures hypervisor and VMM integrity at boot from rootkits and other low-level attacks. It establishes the trustworthiness of the server and the host platforms.

There are three roots of trust in a trusted platform:

- Root of trust for measurement (RTM)
- Root of trust for reporting (RTR)
- Root of trust for storage (RTS)

RTM, RTR, and RTS are the system elements that must be trusted, because misbehavior in these normally would not be detectable in the higher layers. In an Intel TXT-enabled platform, the RTM is the Intel microcode, the Core-RTM (CRTM). An RTM is the first component to send integrity-relevant information (measurements) to the RTS. Trust in this component, thus, is the basis for trust in all the other measurements. The RTS contains the component identities (measurements) and other sensitive information. A trusted platform module (TPM) provides the RTS and RTR capabilities in a trusted computing platform.

A trustworthy CRTM reliably measures the integrity of the next piece of code following in the boot sequence. The result of this measurement is extended into the platform configuration register (PCR) in the TPM before the control is transferred to the next program in the sequence. If each component in the sequence in turn measures the next before handing off control, there’s a chain of trust established. If this measurement chain continues throughout the entire boot sequence, the resulting PCR values transitively reflect the measurement of all files used.

In the unlikely event that one of the components in the chain gets compromised, it is re-measured before its execution during the next reboot. Even if the control is transferred to the malicious software, and the malicious software attempts to fake the measurements, it will have to run a cryptographic gauntlet, where the fake measurements extended to PCR would equal the value it would have had after an uncompromised boot. Thus, the cryptographic strength of the SHA-1 hashing algorithm makes it computationally unlikely for the tampered code to calculate an extension value that would “adjust” the PCR values.

Now that we have expplained what RTM and RTS are, let’s look at the measured boot process, which is one of the two steps listed above that are used to assert the integrity of the pre-launch and launch components of a platform.

Measured Boot Process

A *measured boot process*, as shown in the Figure 3-1, is a boot sequence starting at a root of trust for measurement (RTM) initiating a series of measurements consisting of all the relevant trusted compute base (TCB) components into the root of trust for storage (RTS). The measured boot performs no evaluation or verification of any of the component's identities.

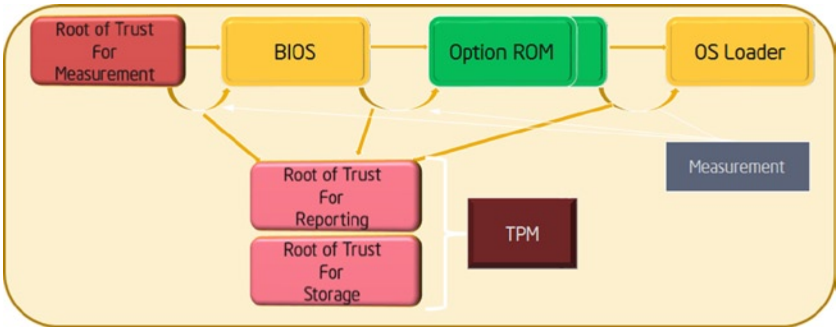

Root of Trust For Measurement
BIOS
Option ROM
OS Loader
Measurement
Root of Trust For Reporting
Root of Trust For Storage
TPM

Figure 3-1. *Measured boot process*

There are two ways defined by the trusted compute group (TCG) to establish this trust during boot:

- Static root of trust (S-RTM)
- Dynamic root of trust (D-RTM)

Figure 3-2 depicts these two boot models and the associated trust chains. As the name *Static Root of Trust for Measurement* (S-RTM) suggests, the entire trust begins with the static, immutable piece of code, which is called the *core root of trust for measurement* (CRTM). On ordinary computing platforms, BIOS is the first component to be executed. Therefore, the trusted platform needs an additional entity to measure the BIOS and act as a CRTM. This entity is a fundamental trusted building block (TBB) that remains unchanged during the lifetime of the platform. The CRTM can be an integrated part of the BIOS itself (e.g., Microsoft Windows 8), like a BIOS boot block. The CRTM can also be a set of CPU instructions that are normally stored within a chip on the motherboard. This latter method can be more resistant to tampering, as exemplified by the Intel TXT.

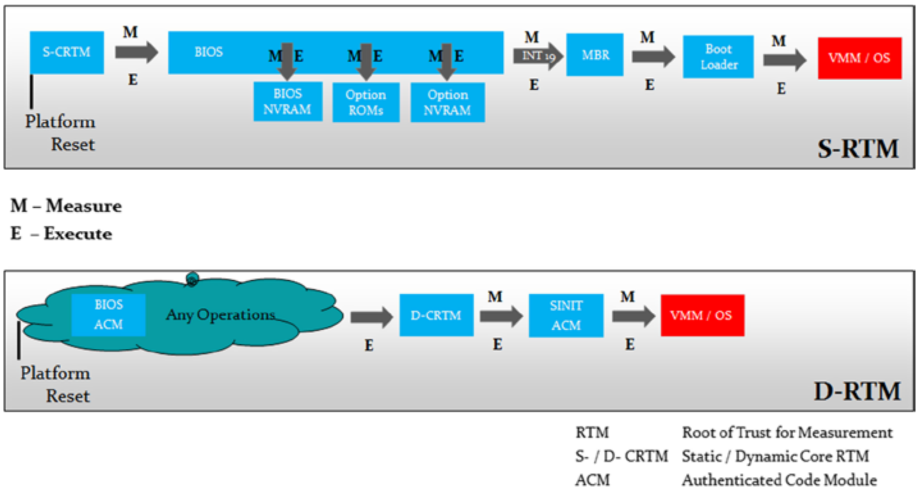


Figure 3-2. *S-RTM and D-RTM trusted chains*

In the static root of trust method, all trust starts with a fixed or immutable piece of trusted code in the BIOS. This trusted piece of code measures the next piece of code to be executed and extends a platform configuration register (PCR) in the TPM based on the measurement before that control is transferred to the next program. If each new program in turn measures the next one before transferring control, there's a chain of trust established. If this measurement chain continues through the entire boot sequence, the resultant PCR values will reflect the measurement of all files used. This "measurement before execution" model therefore leads to a chain of trust that's observable by a remote party wanting to assess the trustworthiness of a system. Hence, S-RTM enables trust on the entire boot chain, including the master boot record, boot loader, kernel, drivers, and all files referenced or executed during boot. These are all parts of a trusted computing base (TCB). In other words, a TCB encompasses the sum of all the components that affect a system's assurance.

However, S-RTM has two shortcomings:

- **Scalability and Inclusivity.** The number of components in a boot chain is large. Each component's trusted computing base (TCB), and hence security, depends on the many layers of code that have been executed earlier in the chain. Windows and Linux have an ill-defined TCB and therefore they require all executable content to be measured, including executables, libraries, and shell scripts. Components determining the chain of trust (including TCB) are subject to frequent patching and updating with their myriad configuration variations. Also, the launch order of elements in the chain may vary, leading to different measurement values in PCRs. Keeping track of the expected values for integrity measurements becomes a nettlesome task.

- **Uncontrolled Scope.** The execution of an S-RTM sequence pulls in code for the evaluation of an OS TCB that's unrelated to the operation of the platform. This forces mostly unnecesary evaluations of software and firmware, including BIOS components loaded and run during the boot process, only to be discarded just to verify the integrity of the TCB.

These shortcomings were identified by the TCG. The newer TCG 1.2 specifications define a new mechanism for an authenticated boot: dynamic root of trust for measurement, or D-RTM.

Dynamic root of trust for measurement (D-RTM) reduces the complexity of the TCB, making the evaluation of the platform state more tractable. With D-RTM, the trust properties of the components are ignored until a secure event, such as an enabled hypervisor launch, triggers and initializes the system, starting the initial root of trust measurement. Components that were staged before the D-RTM secure event are excluded from the TCB and not allowed to execute after the trust properties of the system are established. D-RTM is much more streamlined compared to S-RTM.

The server platforms used in virtualization and cloud data centers present challenging boot scenarios where D-RTM alone won't suffice. The TCB in a true D-RTM implementation will not include the system management modules (SMM), which are needed to support server RAS (reliability, availability, scalability) features. SMM is part of the pre-boot BIOS, and a pure D-RTM implementation excludes these items. Intel TXT provides a hybrid implementation of S-RTM and D-RTM, as described above, to establish trust during the boot process. The book *Intel Trusted Execution Technology for Server Platforms* from Apress has exhaustive coverage of S-RTM and D-RTM.

Attestation

The second step in ensuring boot integrity of a platform is to guarantee that the executed and launched components are trusted components. This process is called *attestation*, and without this step there is no assurance that the platform is in a trusted state. Why is attestation important from a cloud perspective? There are two main considerations for use cases to be instantiated and delivered in a cloud:

- How would the entity needing this information know if a specific platform is Intel TXT enabled, or if a specific server has a defined or compliant BIOS or VMM running on it (i.e., can it be trusted)?
- Why should the entity requesting this information (which, in a cloud environment, could be a resource scheduler or orchestrator trying to schedule a service on a set of available nodes or servers) trust the response from the platform?

An attestation service provides definitive answers to these questions. Chapter 4 covers attestation in detail, including description of a reference attestation platform for Intel-based platforms, code-named Mt. Wilson. But here is a quick summary of the capability.

Attestation ratchets up the notion of roots of trust by making the information from various roots of trust visible and usable by other entities. In a TPM-based implementation of RTS and RTR, it provides a digital signature of platform configuration registers (PCR), with a set of registers in a TPM extended with specific measurements for various launch modules of the software and the requestor validating the signature and the PCR contents. To validate, first the requestor invokes, via an agent on the host or device, the TPM_Quote command, specifying an attestation identity key to perform the digital signature on the set of PCRs to quote, and a cryptographic nonce to ensure freshness of the digital signature. Next, the attestation service validates the signature and determines the trust of the launched server by comparing the measurements from the TPM quote with known-good measurements. It is a critical IT operations challenge to manage the known-good measurement for hypervisors, operating systems, and BIOS software to ensure they are all protected from tampering and spoofing. This capability can be internal to a company, offered by a service provider, or delivered remotely as a service by a trusted third party (TTP). The process is described in detail in Chapter 4.

The measured boot and the attestation thus enable a server/host to demonstrate its boot integrity. Failure of a measured boot process or attestation can initiate a series of remediation steps that are managed and controlled by the policies in the data center. Barring any hardware or configuration issues, then, a failed attestation would mean one of following two conditions:

- Someone or something has tampered with one or more launch components.
- A wrong version (compared to the known-good or whitelist) of BIOS, OS, drivers, and so on has been installed and attempted to launch at the server/host.

Security tools like security information and event management (SIEMs), compliance tools, and configuration checkers would flag these alerts to drive the appropriate remediation actions. In short, having the ability to assert the integrity of a platform is both valuable and necessary. With a set of platforms that have demonstrated integrity, they can be aggregated to do interesting things. This aggregation of platforms is what we refer to as a *trusted compute pool* (TCP).

Trusted Compute Pools

The notion of a trusted compute pool (TCP) relies on the establishment and propagation of a new data center management attribute: *platform trust.* Platform trust derives directly from the boot integrity demonstrated by the server. TCP is a leading approach to aggregate trusted systems and to segregate them from untrusted resources, which results in a split between higher value, more sensitive workloads and commodity application workloads. The principles of TCP operation (see Figure 3-3) are to:

- Create a cloud subsystem that meets the specific and varying security requirements of users.
- Control administrative access to subsystems so that the right workloads get deployed and maintained there.

- Secure, federated, and multi-factored authentication of users and devices accessing the services.
- Continuous monitoring and, on detection of a change in host trust or geolocation, generation of alerts and implementation of configured remediation measures.
- Audits of that segment of the cloud that enables users to verify compliance.

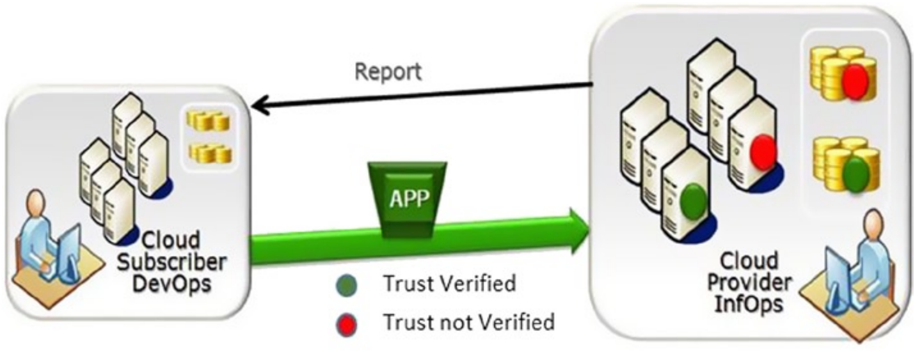


Figure 3-3. *Trusted compute pools*

These trusted pools allow IT to gain the benefits of the dynamic cloud environment while enforcing higher levels of protection for their more critical and security-sensitive workloads. The resources tagged green in Figure 3-3 are trusted, and the resources tagged red are untrusted, as they have not asserted their boot integrity. Critical policies can be defined such that security-sensitive cloud services can be launched only on these resources or migrated only to other trusted platforms within these pools. Also, use of TCPs eliminates the need for air-gapped (i.e., isolated from the rest of the data center) clusters of servers.

TCP Principles of Operation

How is a trusted compute pool created? Platform trust is the primary attribute used by management orchestration and operational software to create a trusted pool. Initially, platform trust is achieved through the use of a trusted platform launch (which, for server platforms, is based on TXT). Once this initial platform trust is established, TCP incorporates additional protections, including visibility of the integrity of the infrastructure and control of the placement and migration of workloads. Figure 3-4 shows a progression of TCP functionality with increasing levels of trust and compliance.

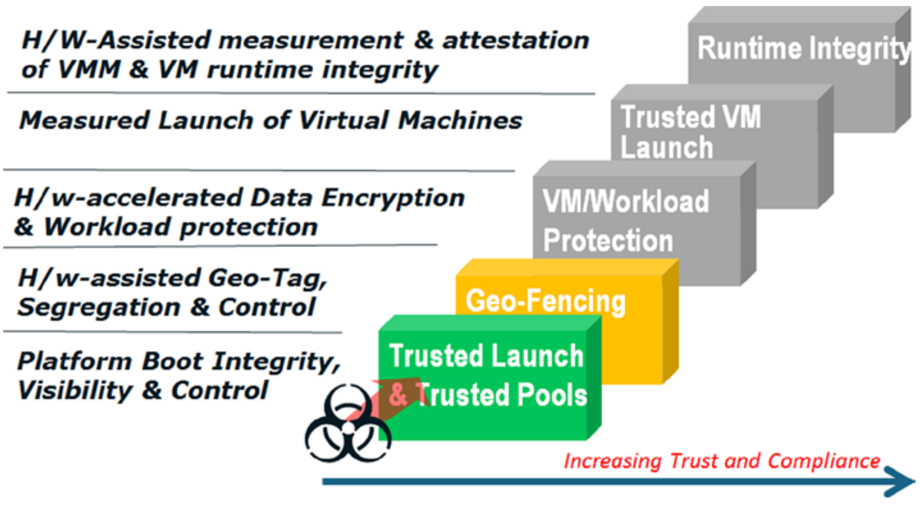

H/W-Assisted measurement & attestation of VMM & VM runtime integrity
Measured Launch of Virtual Machines
H/w-accelerated Data Encryption & Workload protection
H/w-assisted Geo-Tag, Segregation & Control
Platform Boot Integrity, Visibility & Control
Runtime Integrity
Trusted VM Launch
VM/Workload Protection
Geo-Fencing
Trusted Launch & Trusted Pools
Increasing Trust and Compliance

Figure 3-4. *Progression of trusted compute pool usage*

When a trusted pool is created, systems and workloads can be tagged with specific security policies, enabling the monitoring, control, and audits for the placement and migration of workloads into, across, and outside the pool. The most obvious premise behind this is that highly confidential and sensitive applications and workloads must be constrained by policy to run only on systems that have proved to be trusted.

The rest of this section of the chapter describes the flows involved in supporting each of the use cases represented in Figure 3-5.

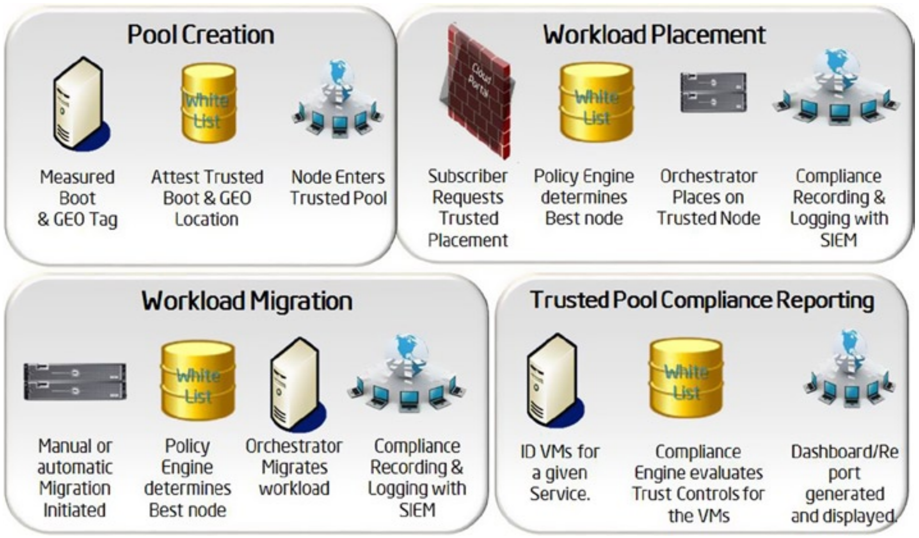

Pool Creation
Measured Boot & GEO Tag
White List
Attest Trusted Boot & GEO Location
Node Enters Trusted Pool
Workload Placement
Cloud Portal
Subscriber Requests Trusted Placement
White List
Policy Engine determines Best node
Orchestrator Places on Trusted Node
Compliance Recording & Logging with SIEM
Workload Migration
Manual or automatic Migration Initiated
White List
Policy Engine determines Best node
Orchestrator Migrates workload
Compliance Recording & Logging with SIEM
Trusted Pool Compliance Reporting
ID VMs for a given Service.
White List
Compliance Engine evaluates Trust Controls for the VMs
Dashboard/Report generated and displayed.

Figure 3-5. *Core use cases for trusted compute pools*

Pool Creation

This is the first step in TCP, involving the creation of a group of platforms with a common level of trust. Pool creation involves the following steps:

1. Virtualization management and orchestration software identifies and enumerates the platforms with demonstrated and attested platform boot integrity.
2. Virtualization management software incorporates the platforms into a trusted pool.

Workload Placement

Once a trusted pool of platforms has been created, workloads can be selected to be placed on that pool based on their security requirements. A typical flow for workload placement would involve the following:

1. A cloud subscriber requests the workload be placed in a trusted pool.
2. Security management tools identify and tag workloads for classification according to certain security properties.
3. Security management tools match platform trust to workload classification according to existing policies.
4. Orchestrator and scheduler software determines the best server to place the workload within the trusted pool, pursuant to existing server selection and security policies. The scheduler requests an attestation of the integrity of the server before the workload is placed on the server, to reaffirm its boot integrity.
5. A compliance record is created to register the launch of the workload in the trusted pool. This record is tied to the hardware root of trust of the server, and can be associated with a set of security controls to meet compliance requirements.

Workload Migration

Infrastructure as a service (IaaS) cloud multi-tenant environments typically use virtualization capability to migrate virtual machines across physical hosts. When it comes to security-sensitive workloads, it is desirable, or perhaps even essential, to meet customer requirements that these migrations occur only between prequalified trusted platforms. A flow representing how to achieve this goal in a TCP environment might occur as follows:

1. A migration of workload is triggered either manually or based on resource orchestrator/scheduler policies.

2. The resource scheduler determines the set of servers that best meets the policy, based on the security standards associated with the workload. (The scheduler requests an attestation of the integrity of the target host.) The first server in the set that meets the integrity requirements is picked.
3. The orchestration software migrates the workload to the new server.
4. A compliance record is created to register the migration of the workload to this new location, including the attestation of integrity at the time of selection.

Compliance Reporting for a Workload/Cloud Service

Being able to prove to an auditing entity that the security requirements of a given workload have been fulfilled is just as important as actually fulfilling those requirements. A flow for compliance reporting might be as follows:

1. The compliance tool enumerates all the virtual machines in the service or workload.
2. The compliance tool evaluates the security controls for each virtual machine; these controls include determining the trail of hosts and the migration of records throughout the virtual machine lifecycle.
3. A report is generated to provide proof that security properties associated with the workload running on TCP have been met. This verifiable proof is linked to the hardware root of trust (provided by TXT) in the participating hosts.

Solution Reference Architecture for the TCP

The Intel TXT-enabled launch is not sufficient to support the TCP uses mentioned in the previous section. Measurement and attestation tell the data center and security management software whether a given host can be trusted, but there is more to it than that. Exposing, transporting, storing, and ultimately consuming platform trust measurements in support of the use cases is an integration challenge across different software and management elements. Successfully doing this requires a well-defined and seamless integration model of multiple security management and data center/cloud management software components; in other words, there needs to be an underlying solution architecture.

Figure 3-6 depicts a reference architecture for these trusted compute pool usages. It prescribes four distinct layers, with each layer serving one of the four functions:

1. Reporting of the source of the platform trust/boot integrity measurements
2. Interface with the boot integrity information via secure protocols
3. Verifification/appraisal of the boot integrity measurements
4. Consumer of the boot integrity verification for policy enforcement, compliance reporting, and remediation

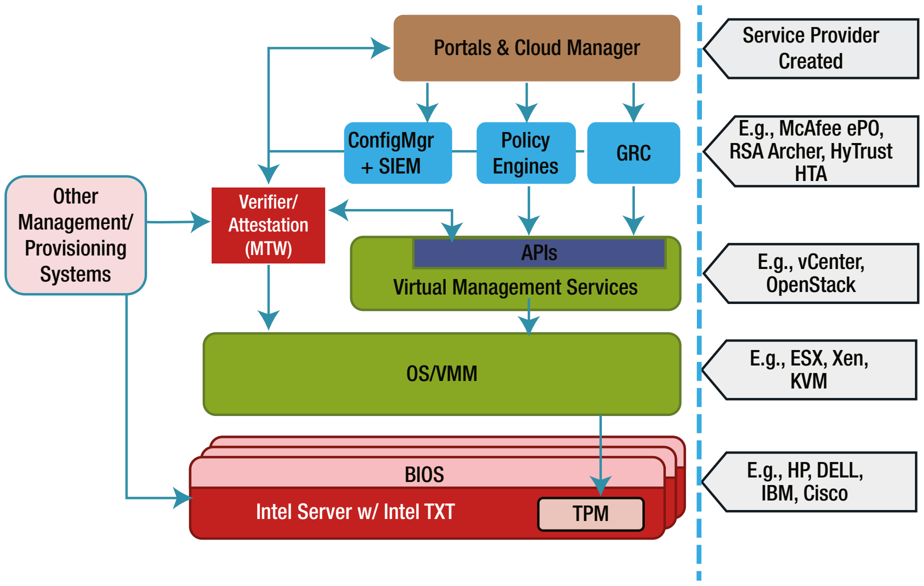

Portals & Cloud Manager
ConfigMgr + SIEM
Policy Engines
GRC
Other Management/ Provisioning Systems
Verifier/ Attestation (MTW)
APIs
Virtual Management Services
OS/VMM
BIOS
Intel Server w/ Intel TXT
TPM
Service Provider Created
E.g., McAfee ePO, RSA Archer, HyTrust HTA
E.g., vCenter, OpenStack
E.g., ESX, Xen, KVM
E.g., HP, DELL, IBM, Cisco

Figure 3-6. *Solution architecture for the trusted compute pools*

Here's a brief overview of the four layers of this software architecture, starting from the base and moving upward.

Hardware Layer

At the base of the architecture we have the physical server hardware. For virtualization and the cloud computing environment, the hardware typically consists of x86 architecture-based servers. These are servers hosting the virtualization and cloud workloads. Intel TXT enables trusted compute pool usages. (See sidebar for a brief introduction to the technology.) In addition to having TXT-enabled CPUs and chipsets, there needs to be Intel Virtualization Technology (VT) and a trusted platform module, or TPM. TXT needs support in the BIOS, as well. By default, TXT and TPM are not enabled in the BIOS in the current generation of servers. Unfortunately, the method for turning on TXT and TPM support varies by vendor; there is no standard for carrying out this operation. There are, however, well-published documents on how to enable TXT and TPM for various OEM vendors, available from Intel and few security management companies supporting Intel TXT.

One of the challenges in scale deployments and enablement of TXT is meeting the need for physical access and the assertion of presence to enable the TPM and TXT on a server platform via the BIOS interface. This limits automation and large-scale enablement. Though each of the OEM provides custom implementation and interfaces for doing this, unless there are architectural solutions such as a physical presence interface (PPI), the provisioning and configuration task won't become any simpler.

Operating System / Hypervisor Layer

Moving up the stack, the second layer is the OS/hypervisor. To participate in a measured launch, an OS or hypervisor has to be enabled for TXT. The changes related to TXT are in the initialization code, and also during termination and shutdown. Additionally, basic enablement means that the operating system or hypervisor can invoke the secure launch process. This entails including a pre-kernel module that can ensure the right SINIT (authenticated code modules from Intel) module is selected and assure the orderly evaluation of the launch components of the software. Intel provides a reference implementation called Trusted Boot (tboot) for the pre-kernel module that can be integrated into OS/hypervisors toward enabling for Intel TXT, and it is the maintainer of this open-source "tboot" project.

Tboot is by far the most widely used mechanism offered by software vendors to enable their OS or hypervisor. SINIT modules on server platforms are generally embedded in the platform BIOS, and are processor- and chipset generation-specific. The tboot components provided by Intel are integrated into the operating systems or hypervisors (by the respective ISVs) and work across multiple generations of platforms. This makes sense, as it allows the most qualified party (in this case, the ISV) to determine which modules are essential for the trusted compute base (TCB) of their software, and therefore which modules to include in the measured launch and in which order.

Tboot technology is included in multiple open-source operating system/hypervisor environments from Linux, to Xen/KVM, to a number of commercial products, such as Red Hat and Citrix XenServer. Other vendors, like VMware, have implemented their own tboot-like functions. It is interesting to note that the percentage of TCB measured by vendors as part of the launch process varies significantly. As of this writing, VMware by far has the most coverage of the TCB. Other OS/VMM vendors have the core kernel and few modules measured. All of these vendors have been actively working toward increasing the amount of TCB that they measure. For detailed coverage of the measured launch environments (MLE) developer guidance, check out the book *Intel Trusted Execution Technology for Server Platforms* from Apress.

With TXT and TPM correctly configured and enabled in hardware, when a TXT-enabled OS/hypervisor is launched, the platform goes through a measured D-RTM launch. Just to refresh your understanding of the TXT launch process, when a TXT launch happens, what you have is a measured launch of the firmware, BIOS, and controlling software like an OS or VMM. These measurements (which are the identities of the various components), as part of the launch process are stored in the various registers in the TPM (RTS and RTR) called PCRs (platform configuration registers) and are verified with an attestation system. TCG PC Spec provides the semantics for where the various measurements are stored in the TPM.

Virtualization/Cloud Management and Verification/Attestation Layer

This is the critical management and orchestration layer in a data center that controls the provisioning, deployment, and lifecycle management of the workloads and virtual machines. This layer serves one of four functions for the trusted compute pool use cases:

1. Provides a secure interface to the measured launch measurements on each of the servers
2. Provides an attestation mechanism to evaluate platform trust and assert its integrity
3. Consumes the trust information, essentially helping to identify which platforms are trusted and which ones are not
4. Makes use of this information to establish an enhanced security capability through policy definition and enforcement linked to platform trust

There are significant differences in terms of interfaces provided to support platform trust. Some, such as Citrix, have developed explicit APIs natively to their hypervisor software (Xen APIs) to provide TCG-compliant access to the launch measurements in the TPM. These APIs are available for any management software to use—for evaluation, attestation, reporting, alerting, and so on—and they maintain the integrity of the measurements. Others, such as VMWare, have tied access to these measurements to their primary virtualization resource management software—vCenter, in this case. VMware provides access to the measurements via run-time vCenter APIs, which when invoked instantiate a TCG-compliant remote attestation protocol to request the measurements for the attesting server/device. None of the virtualization and cloud management software vendors provides verification/attestation software for verification of the measurements. Since attestation is a relatively new concept, it is not yet integral to most of the virtualization and cloud management software. Having attestation services provided through the operating system or hypervisor would establish the function across many enterprise and cloud customers—thereby unlocking the most valuable use models.

Intel in collaboration with security ISVs have developed attestation software for attestation verification. This attestation software is a multi-hypervisor, multi-OS verification/attestation program providing a secure assertion of the hypervisor and platform integrity that is verified against a set of known-good, golden measurements, or whitelist values. Following the key tenet of cloud technology with regard to programmability and automation, the attestation platform exposes all its functionality via well-defined REST & SOAP APIs for querying trust assertions, as well as for provisioning, management, and whitelisting. The attestation process and the Intel attestation platform are covered in detail in Chapter 4. The trust assertion from the attestation/verification software is used by the cloud management software and the security management tools that are in the next layer of the architecture.

Security Management Layer

The security management layer is the top layer, where the platform trust assertion from the previous layer is requested and consumed. This security applications layer includes some classes of traditional security applications focused on event reporting and managing compliance and risk. Because the technologies and trusted compute pools involve platform integrity and trust, workload control, and policy enforcement, it makes perfect sense to have such applications aware of and enabled to detect, report, and act on the trust information available from the Intel TXT–enabled platforms.

In the context of the TCP use model, the security management tools of interest are:

- Workload/VM policy management
- SIEM/configuration management/monitoring
- GRC/compliance

These tools are critical for mainstreaming trust and elements of cloud security into any overall corporate security management systems. This is a crucial requirement, as IT managers do not want a new suite of tools for managing cloud security; they would very much rather see existing tools extended to include the new cloud and virtualized architectures as they adopt them. The primary motivation for these security management tools is to ensure that they have the visibility to platform trust and a set of control functions to management the lifecycle of the VMs/workloads. Though initially the monitoring and enforcement of trust might be periodic, over time we envision that these tools will provide continuous monitoring and enforcement of policies based on trust.

VM/Workload Policy Management

These tools provide the mechanism to specify and define the granular security requirements for the virtual machines and workloads, and to enforce these requirements during the lifetime of those virtual machines. Defining a security policy for a workload runs the gamut from the trivial, such as asserting “I want to run on trusted servers,” to the sophisticated. For an example of the latter, a policy definition could include “Run on servers with trust level X and only on servers that are in geolocation Y, and don’t co-exist with Z type of workloads.” Today, there is no canon for policy definition, nor standards for tagging the workloads. Each of the policy management ISVs carries a particular language of definition and execution environments, with these definitions likely not to be portable or interoperable with other vendor offerings. As these capabilities mature, it is imperative that policy definitions and other matters of semantics become standardized and interoperable across vendors.

Policy tools also provide an interface to feed the following information to other security management tools in this layer of the stack, like the security event management and GRC tools. They provide:

- Auditable information about the policies that have been evaluated
- Evidence considered during policy evaluation
- Whitelists/manifests/known-good measurements considered for decision making
- Reports of decisions made, such as launch or deny workload creation or migration in a certain pool of compute servers

This information is provided in different formats while preserving integrity and maintaining the chain of trust. Hytrust VPA and McAfee ePO are examples of policy management tools for trusted compute pools.

GRC Tools—Compliance in the Cloud

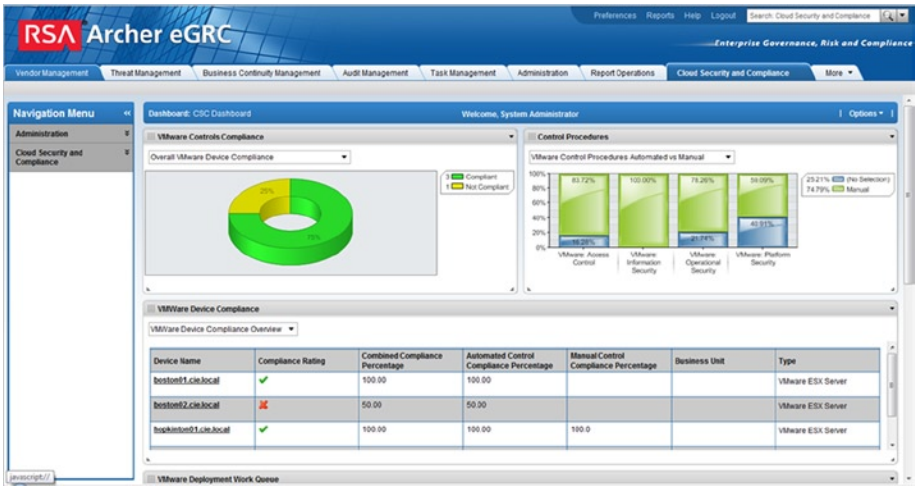
GRC tools set requirements for platform trust and integrity based on workload requirements and security standards, followed by an assessment of the environment to determine security controls in place and to dashboard actual conditions against policy to determine compliance. SIEM tools allow trust events to be captured, reported, logged, and processed for correlation to determined responses or heuristics to indicate whether a larger attack is occurring. Although not every organization will need the high level of security afforded by a trusted computing environment, every organization using cloud services will benefit from the vastly improved control and transparency that a measured chain of trust enables.

Simply being able to verify conditions in the cloud services, down the stack through the hypervisor, brings significant value to users with its visibility into actual states and activities within the cloud and in its improved governance for cloud resources. Internal and private clouds built on a measured chain of trust will:

- Strengthen an organization’s ability to enforce differentiated policies in private clouds
- Enhance monitoring for compliance at all layers within the cloud
- Streamline the auditing process
- Allow for more flexible usage and billing for secure computing resources

Organizations often see the completion of a regulatory audit as the end goal of their compliance efforts. The reality is that compliance is a continuous cycle that starts with technical and operational decisions on how to address control requirements. It's an accomplishment to have auditors give a thumbs-up to your technical and administrative controls. That goal notwithstanding, passing subsequent audits requires continuous maintenance and reporting on those controls. Cloud teams hold the key to making that happen in a scalable, automated manner. The most effective approach to achieving continuous compliance is to define and implement policies, guidelines, standards, and tools that secure the organization's computing systems as a whole, with an eye toward regulatory guidance, standards, and mandates themselves. Ensuring that the corresponding security controls meet or exceed the standards prescribed by the governing body will help ensure a successful audit.

RSA was one of the first ecosystem participants to demonstrate the value of platform trust for GRC uses, with a joint Intel–VMware-RSA demonstration at the RSA Security Show in 2010. Figure 3-7 shows a dashboard view of the status of a security control tied to platform trust. Intel is working with a number of other providers in these market segments to provide customers with ample choice of solutions and capabilities.


RSA Archer eGRC
Enterprise Governance, Risk and Compliance
Preferences Reports Help Logout
Search: Cloud Security and Compliance
Vendor Management
Threat Management
Business Continuity Management
Audit Management
Task Management
Administration
Report Operations
Cloud Security and Compliance
More
Navigation Menu
Administration
Cloud Security and Compliance
Dashboard: CSC Dashboard
Welcome, System Administrator
Options
VMware Controls Compliance
Overall VMware Device Compliance
25%
75%
3 Compliant
1 Not Compliant
Control Procedures
VMware Control Procedures Automated vs Manual
100%
80%
60%
40%
20%
0%
83.72%
16.28%
100.00%
78.26%
21.74%
59.09%
40.91%
25.21% (No Selection)
74.79% Manual
VMware: Access Control
VMware: Information Security
VMware: Operational Security
VMware: Platform Security
VMWare Device Compliance
VMWare Device Compliance Overview
VMware Deployment Work Queue
javascript://

Device Name	Compliance Rating	Combined Compliance Percentage	Automated Control Compliance Percentage	Manual Control Compliance Percentage	Business Unit	Type
boston01.cie.local	✓	100.00	100.00			VMware ESX Server
boston02.cie.local	✗	50.00	50.00			VMware ESX Server
hopkinton01.cie.local	✓	100.00	100.00	100.0		VMware ESX Server

Figure 3-7. *GRC dashboard showing compliance to platform trust*

Now that we have laid out the details of the solution architecture for trusted compute pools, let's focus on a specific example and walk through one solution stack with a reference implementation of the use cases, so as to put these new concepts on a solid footing.

Reference Implementation: The Taiwan Stock Exchange Case Study

The Taiwan Exchange Stock Exchange Corporation (TWSE) is a stock exchange in Taiwan that supports the trading of 758 listed companies. Its primary business drivers are developing new financial products and boosting the number of services it offers. Cloud computing will be part of its ability to do so, but it realizes that strong security controls must first be part of the picture.

A fundamental business and technical requirement for the cloud infrastructure under construction at the TWSE infrastructure is to provide secure systems and trusted compute environments. It has established as crucial the ability to integrate software application solutions that provide TWSE with overall trust and security for its cloud infrastructure and that exploit hardware-based security and include roots of trust and platform attestation. The goals for the proof of concept built for this case study were to enable:

- Greater visibility into the security states of the hardware platforms running the infrastructure as a service (IaaS) for their private clouds.
- Production of automated, standardized reports on the configuration of the physical and virtual infrastructure hosting customer virtual machines and data.
- Controls based on the physical location of the server's virtual machines and control any migration of these virtual machines onto acceptable servers per policy specified.
- Generation of measured evidence that their services infrastructure complies with security policies and with regulated data standards.

To explore the capabilities and challenges of implementing such an infrastructure, TWSE engaged Intel and other key ecosystem partners to develop a multi-phased proof of concept (PoC) implementation of a more secure cloud based on familiar tools, platforms, and software. The basic capabilities under the proof of concept include:

- Measured boot for servers, with platform attestation
- Ability to create trusted compute pools
- Security-controlled workload placement in the trusted compute pools
- Security controlled workload migration into trusted compute pools
- Integration and extension of security and platform trust with McAfee ePolicy Orchestrator* (McAfee ePO)

Solution Architecture for TWSE

For the proof of concept, a number of systems and solutions were selected based on TWSE's current and future business directions and needs. They map directly onto the solution reference architecture layers discussed in the earlier section. As shown in Figure 3-8, these include:

- *Cloud system and infrastructure supported by Cisco*. This includes a Cisco UCS server with Intel Xeon processor E5 family and Intel TXT-enabled, equipped with the optional Cisco TPM part. Three blades were used to establish a mix of trusted and untrusted platforms in the PoC environment.
- *Virtualization solutions supported by VMware.* VMware ESXi 5.1 provides fullly integrated support for Intel TXT and enables remote platform attestation measurements to detect possible malicious changes to BIOS and other critical base-software components of the servers. VMware ESXi 5.1, in conjunction with TXT, measures the critical components of the hypervisor stack when the system boots and it stores these measurements in the platform configuration registers (PCR) of the TPM on the platform.
- *Trust and policy management supported by HyTrust and HyTrust Appliance*. HyTrust Appliance 3.5 provides extensive support for Intel TXT; the HyTrust Appliance verifies the integrity of the physical hardware of the host to ensure that the underlying platform is fully trusted and can implement policies based on this information. It can ensure that specified workloads are permitted to be instantiated only on specific hosts or clusters, the essence of TCP. It also intercepts all administrative access and change requests, determines whether a request is in accordance with the organization's defined policy, and permits or denies the request as appropriate. The HyTrust Appliance is not a physical piece of hardware; it is a VMware vSphere*compatible virtual appliance deployed alongside the rest of the virtual infrastructure. Finally, it provides direct sharing of trust and security information with McAfee ePolicy Orchestrator (McAfee ePO).
- *Security management solution supported by McAfe*e. McAfee ePO unifies security management through an open platform, simplifies risk and compliance management, and provides security intelligence across endpoints, networks, data, and compliance solutions. It helps to manage security, streamline and automate compliance processes, and increase overall visibility across security management activities. McAfee with HyTrust ePO extensions enable communication with the HyTrust Appliance.

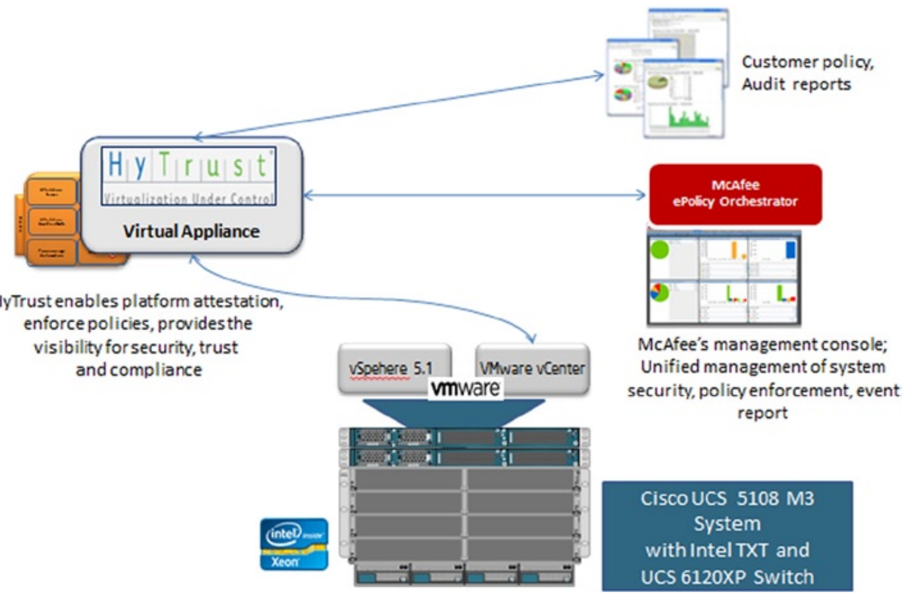
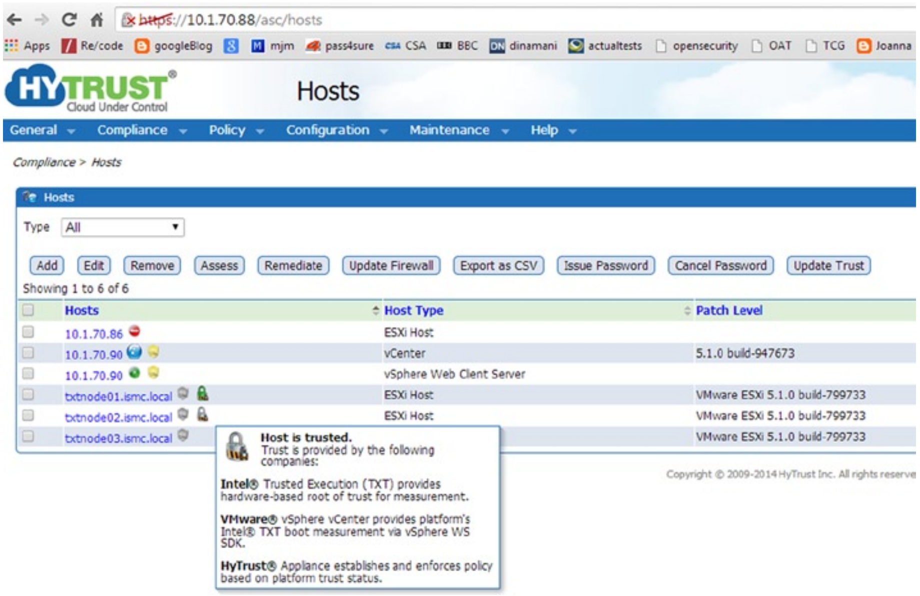

Customer policy, Audit reports
HyTrust
Virtualization Under Control
Virtual Appliance
McAfee ePolicy Orchestrator
HyTrust enables platform attestation, enforce policies, provides the visibility for security, trust and compliance
vSpehere 5.1
VMware vCenter
vmware
McAfee's management console; Unified management of system security, policy enforcement, event report
intel inside Xeon
Cisco UCS 5108 M3 System with Intel TXT and UCS 6120XP Switch

Figure 3-8. *TWSE proof of concept solution components*

Trusted Compute Pool Use Case Instantiation

Although all of the Cisco blades in this PoC were fully Intel TXT-capable, it was important to have a contrast between trusted and untrusted servers so as to differentiate trusted pools and prove the controls and status reporting mechanisms. For this reason, Intel TXT was disabled in the system BIOS configuration settings in one of the Cisco UCS blades to prohibit the system from executing a trusted launch. HyTrust Appliance had full integration of remote attestation capabilities. From VMWare ESXi side, the measured elements included the VMkernel, kernel modules, drivers, native management applications that run on ESXi, and any boot-time configuration options. As shown in Figure 3-9, the trust status dashboard of the HyTrust Appliance shows an unknown BIOS trust status, unknown VMM status, and overall unknown status for the second Cisco UCS blade as a consequence of disabling the Intel TXT support.



Hosts	Host Type	Patch Level
10.1.70.86	ESXi Host	
10.1.70.90	vCenter	5.1.0 build-947673
10.1.70.90	vSphere Web Client Server	
txtnode01.ismc.local	ESXi Host	VMware ESXi 5.1.0 build-799733
txtnode02.ismc.local	ESXi Host	VMware ESXi 5.1.0 build-799733
txtnode03.ismc.local		VMware ESXi 5.1.0 build-799733

Host is trusted.
Trust is provided by the following companies:

Intel® Trusted Execution (TXT) provides hardware-based root of trust for measurement.

VMware® vSphere vCenter provides platform's Intel® TXT boot measurement via vSphere WS SDK.

HyTrust® Appliance establishes and enforces policy based on platform trust status.

Copyright © 2009-2014 HyTrust Inc. All rights reserve

Figure 3-9. *HyTrust trust attestation service dashboard indicating two trusted hosts and one untrusted host*

Remote Attestation with HyTrust

The HyTrust Appliance provides extensive support for Intel TXT, plus policy control functionality for this use case—essentially establishing the parameters and policies for a trusted compute pool. As shown in Figure 3-10, the HyTrust Appliance provides management of critical platform attestation functionality, whitelisting of known-good measurements, and trust operation and report dashboards for trusted compute pools, as well as a broad set of other virtualization security controls for workloads, servers, and administrators. The HyTrust Appliance and these solutions were used to detect, measure, and report the trust of both the server platforms and the hypervisor, and to implement workload controls (VM migration, etc.) based on the required platform trust attributes.

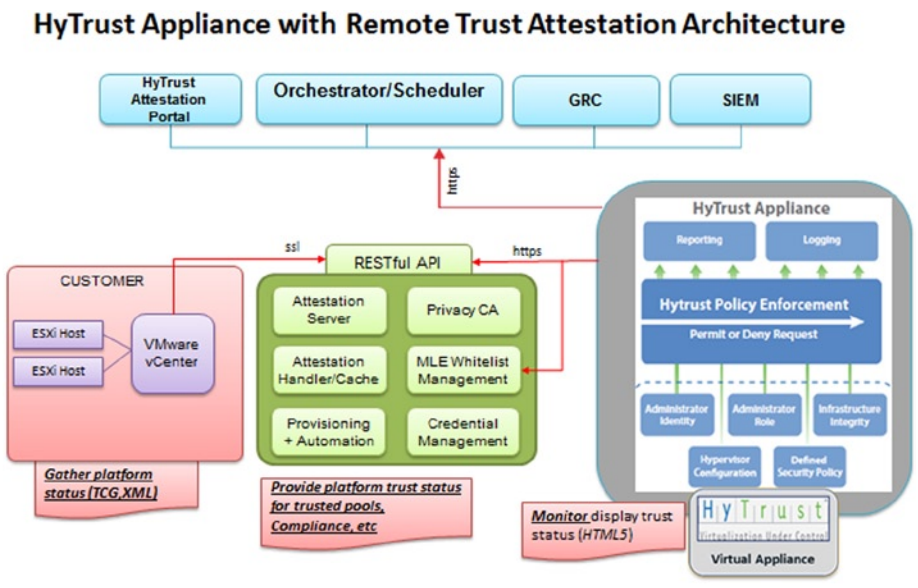
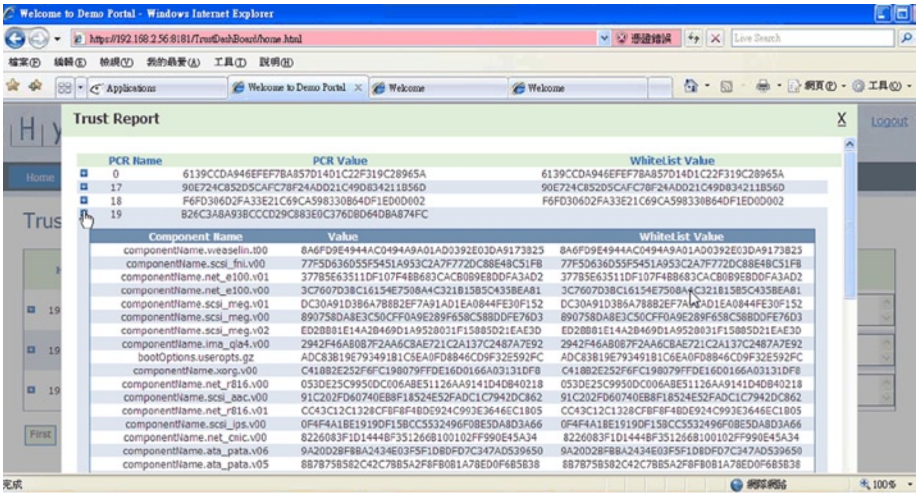


Figure 3-10. *HyTrust Appliance with remote trust attestation architecture*

To summarize, the remote attestation process provides an independent evaluation of the integrity measurements of the firmware, BIOS, and the VMM against known-good (whitelist) program components, and it securely makes that assertion available to the HyTrust Appliance policy enforcement and reporting components. The evaluation of the measurements is comprehensive and covers the core of the BIOS, the BIOS configurations, the VMM kernel, and various VMM modules loaded as part of the VMware ESXi launch. Figure 3-11 shows a snapshot of the actual measurements of an ESXi Server with the known-good or whitelist values.


Welcome to Demo Portal - Windows Internet Explorer
https://192.168.2.56:8181/TrustDashBoard/home.html
Applications
Welcome to Demo Portal
Welcome
Welcome
Trust Report
X
Logout
Home
First

Trust Report

PCR Name	PCR Value	WhiteList Value
0	6139CCDA946EFEF7BA857D14D1C22F319C28965A	6139CCDA946EFEF7BA857D14D1C22F319C28965A
17	90E724C852D5CAFC78F24ADD21C49D834211B56D	90E724C852D5CAFC78F24ADD21C49D834211B56D
18	F6FD306D2FA33E21C69CA598330B64DF1ED0D002	F6FD306D2FA33E21C69CA598330B64DF1ED0D002
19	B26C3A8A93BCCCD29C883E0C376D8D64DBA874FC	

Component Name	Value	WhiteList Value
componentName.weaselin.t00	8A6FD9E4944AC0494A9A01AD0392E03DA9173825	8A6FD9E4944AC0494A9A01AD0392E03DA9173825
componentName.scsi_fni.v00	77F5D636D55F5451A953C2A7F772DC88E48C51FB	77F5D636D55F5451A953C2A7F772DC88E48C51FB
componentName.net_e100.v01	377B5E63511DF107F4BB683CACB0B9E8DDFA3AD2	377B5E63511DF107F4BB683CACB0B9E8DDFA3AD2
componentName.net_e100.v00	3C7607D38C16154E7508A4C321B15B5C435BEA81	3C7607D38C16154E7508A4C321B15B5C435BEA81
componentName.scsi_meg.v01	DC30A91D3B6A7B8B2EF7A91AD1EA0844FE30F152	DC30A91D3B6A788B2EF7A91AD1EA0844FE30F152
componentName.scsi_meg.v00	890758DA8E3C50CFF0A9E289F658C58BDDFE76D3	890758DA8E3C50CFF0A9E289F658C58BDDFE76D3
componentName.scsi_meg.v02	ED2BB81E14A2B469D1A9528031F15885D21EAE3D	ED2BB81E14A2B469D1A9528031F15885D21EAE3D
componentName.ima_qla4.v00	2942F46AB087F2AA6CBAE721C2A137C2487A7E92	2942F46AB087F2AA6CBAE721C2A137C2487A7E92
bootOptions.useropts.gz	ADC83B19E793491B1C6EA0FD8B46CD9F32E592FC	ADC83B19E793491B1C6EA0FD8B46CD9F32E592FC
componentName.xorg.v00	C418B2E252F6FC198079FFDE16D0166A03131DF8	C418B2E252F6FC198079FFDE16D0166A03131DF8
componentName.net_r816.v00	053DE25C9950DC006ABE51126AA9141D4DB40218	053DE25C9950DC006ABE51126AA9141D4DB40218
componentName.scsi_aac.v00	91C202FD60740EB8F18524E52FADC1C7942DC862	91C202FD60740EB8F18524E52FADC1C7942DC862
componentName.net_r816.v01	CC43C12C1328CFBF8F4BDE924C993E3646EC1805	CC43C12C1328CFBF8F4BDE924C993E3646EC1805
componentName.scsi_ips.v00	0F4F4A1BE1919DF15BCC5532496F0BE5DA8D3A66	0F4F4A1BE1919DF15BCC5532496F0BE5DA8D3A66
componentName.net_cnic.v00	8226083F1D1444BF351266B100102FF990E45A34	8226083F1D1444BF351266B100102FF990E45A34
componentName.ata_pata.v06	9A20D2BF8BA2434E03F5F1D8DFD7C347AD539650	9A20D2BF8BA2434E03F5F1D8DFD7C347AD539650
componentName.ata_pata.v05	8B7B75B582C42C7BB5A2F8FB0B1A78ED0F6B5B38	8B7B75B582C42C7BB5A2F8FB0B1A78ED0F6B5B38

Figure 3-11. *Trust attestation service - trust report view*

Use Case Example: Creating Trusted Compute Pools and Workload Migration

Knowing the trust status of both the servers and the hypervisor highlighted the platform trust information to TWSE, as well as defined an appropriate set of operational policies and controls. The reference implementation demonstrated the operational details of the trusted compute pools use cases as follows:

- Creation of trusted compute pools
- Workload placement in the trusted compute pools
- Workload migration into the trusted compute pools
- Dashboard reporting with McAfee ePolicy Orchestrator* (McAfee ePO*)

The HyTrust Appliance enabled the team to intercept all administrative requests for the virtual infrastructure, determine whether the request was in accordance with defined policy, permit or deny that request, and record all administrative access and change requests.

To apply effective end-to-end trust policies for the cloud infrastructure, the team did the following:

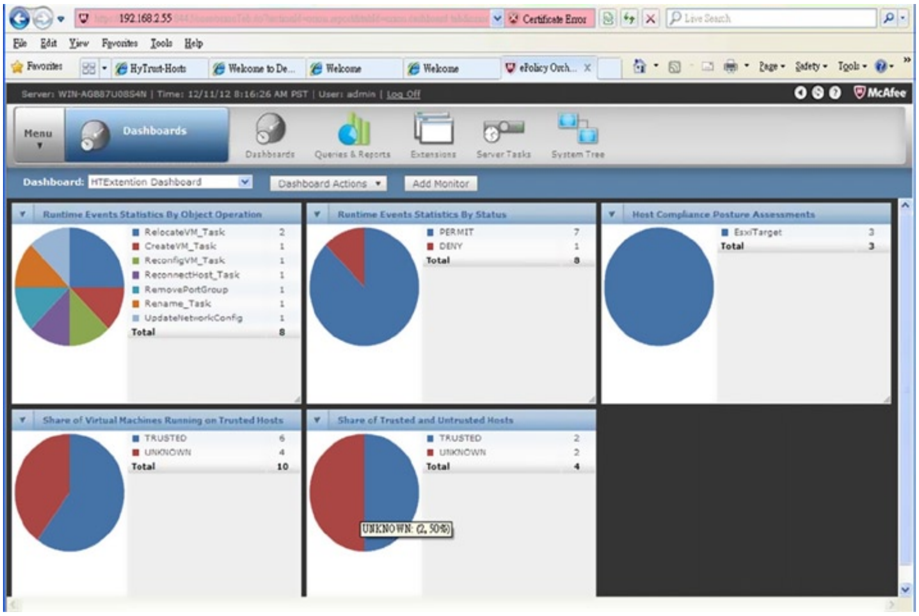
- Created trusted compute pools with Intel TXT
- Identified and labeled the sensitive workloads that required protection
- Configured the trust policies to establish trust requirements

- Assigned and managed workload migration based on defined trust polices
- Enforced trust policies end-to-end
- Recorded all activities, including audit, and compliance; and provided reports

Integrated and Extended Security and Platform Trust with McAfee ePO

A TWSE requirement was the integration and reporting of all security events and enforcement decisions to a SIEM and GRC system. This gave TWSE another common and aggregated management view of its cloud infrastructure. The PoC used the HyTrust Appliance to extend and integrate the trust information for each hypervisor and the virtualized resource functionality to the McAfee ePO console.

The direct integration of the HyTrust Appliance dashboard showed users the Intel TXT trust status of the host on which each VM was running. HyTrust Appliance assessed compliance by comparing a host's current configuration with a hardening configuration template that was customized based on TWSE requirements. It then provided assessment data to the master ePO dashboard for reporting and analysis. HyTrust Appliance gave McAfee ePO a record of all administrative activities, including a unique user ID, and operations attempted by the privileged user, including denied or failed attempts. Figure 3-12 shows the aggregated view of trust within the McAfee ePO dashboard.



Server: WIN-AG887U08S4N | Time: 12/11/12 8:16:26 AM PST | User: admin | Log Off

Dashboard: HTExtention Dashboard

Runtime Events Statistics By Object Operation

🟦 RelocateVM_Task	2
🟥 CreateVM_Task	1
🟩 ReconfigVM_Task	1
🟪 ReconnectHost_Task	1
🟦 RemovePortGroup	1
🟧 Rename_Task	1
🟦 UpdateNetworkConfig	1
Total	**8**

Runtime Events Statistics By Status

🟦 PERMIT	7
🟥 DENY	1
Total	**8**

Host Compliance Posture Assessments

🟦 EsxiTarget	3
Total	**3**

Share of Virtual Machines Running on Trusted Hosts

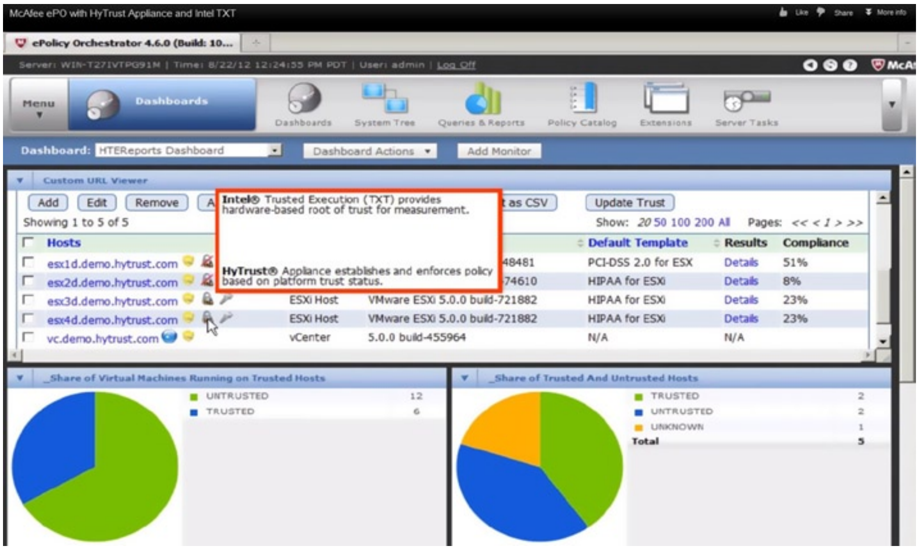
🟦 TRUSTED	6
🟥 UNKNOWN	4
Total	**10**

Share of Trusted and Untrusted Hosts

🟦 TRUSTED	2
🟥 UNKNOWN	2
Total	**4**

Figure 3-12. *McAfee ePO displaying administrator activity and trust status captured by HyTrust Appliance*

Figure 3-13 shows a drilldown view of the trust information in the McAfee ePO system as provided by the seamless integration between the HyTrust Appliance and the McAfee policy orchestrator.


McAfee ePO with HyTrust Appliance and Intel TXT
ePolicy Orchestrator 4.6.0 (Build: 10...
Server: WIN-T27IVTPG91M | Time: 8/22/12 12:24:55 PM PDT | User: admin | Log Off
Menu
Dashboards
Dashboards
System Tree
Queries & Reports
Policy Catalog
Extensions
Server Tasks
Dashboard: HTEReports Dashboard
Dashboard Actions
Add Monitor
Custom URL Viewer
Add
Edit
Remove
as CSV
Update Trust
Showing 1 to 5 of 5
Show: 20 50 100 200 All
Pages: << < 1 > >>
_Share of Virtual Machines Running on Trusted Hosts
_Share of Trusted And Untrusted Hosts

Intel® Trusted Execution (TXT) provides hardware-based root of trust for measurement.

HyTrust® Appliance establishes and enforces policy based on platform trust status.

Hosts			Default Template	Results	Compliance
esx1d.demo.hytrust.com		48481	PCI-DSS 2.0 for ESX	Details	51%
esx2d.demo.hytrust.com		74610	HIPAA for ESXi	Details	8%
esx3d.demo.hytrust.com	ESXi Host	VMware ESXi 5.0.0 build-721882	HIPAA for ESXi	Details	23%
esx4d.demo.hytrust.com	ESXi Host	VMware ESXi 5.0.0 build-721882	HIPAA for ESXi	Details	23%
vc.demo.hytrust.com	vCenter	5.0.0 build-455964	N/A	N/A	

🟩 UNTRUSTED	12
🟦 TRUSTED	6

🟩 TRUSTED	2
🟦 UNTRUSTED	2
🟧 UNKNOWN	1
Total	5

Figure 3-13. *McAfee ePO displaying a drilldown of the server trust status from the HyTrust Appliance*

McAfee ePO's flexible automation capability streamlined the workflows, dramatically reducing the cost and complexity of security and compliance administration.

INTEL TXT ARCHITECTURAL OVERVIEW

Intel TXT is a set of enhanced hardware components designed to protect sensitive information from software-based attacks. Intel TXT features include capabilities in the microprocessor, chipset, I/O subsystems, and other platform components. When coupled with an enabled operating system, hypervisor, and enabled applications, these capabilities provide confidentiality and integrity of data in a time of increasingly hostile environments.

Intel TXT incorporates a number of secure processing innovations (see Figure 3-14), including:

- *Protected execution*. Lets applications run in isolated environments so that no unauthorized software on the platform can observe or tamper with the operational information. Each of these isolated environments executes with the use of dedicated resources managed by the platform.

- *Sealed storage*. Provides the ability to encrypt and store keys, data, and other sensitive information within the hardware. This can be decrypted only by the same environment as encrypted it.
- *Attestation*. Enables a system to provide assurance that the protected environment has been correctly invoked and takes a measurement of the software running in the protected space. The information exchanged during this process is known as the attestation identity key credential, and is used to establish mutual trust between parties.
- *Protected launch*. Provides the controlled launch and registration of critical system software components in a protected execution environment.
- *Trusted extensions integrated into silicon* (processor and chipset). Allow for the orderly quiescence of all activities on the platform such that a tamper-resistant environment is enabled for the measurement and verification processes; and allows for protection of platform secrets in the case of "reset" and other disruptive attacks.
- *Authenticated code modules* (ACM). Authenticate platform-specific code to the chipset and execute in an isolated environment within the processor and the trusted environment (authenticated code mode) enabled by AC Modules to perform secure tasks.

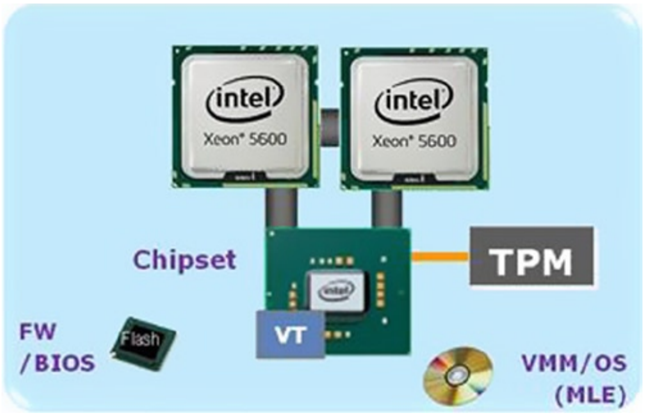


Figure 3-14. *Intel Trusted Execution Technology components*

Intel TXT Principles of Operation

Intel TXT works through the creation of a measured launch environment (MLE) enabling an accurate comparison of all the critical elements of the launch environment against a known-good source. Intel TXT creates a cryptographically unique identifier for each approved launch-enabled component and then provides a hardware-based enforcement mechanism to block the launch of the code that does not match that which is authenticated or, alternatively, indicates when an expected trusted launch has not happened. This hardware-based solution provides the foundation on which IT administrators can build trusted platform solutions to protect against aggressive software-based attacks and to better control their virtualized or cloud environments.

Figure 3-15 illustrates two different scenarios. In the first, the measurements match the expected values, so the launch of the BIOS, firmware, and VMM are allowed. In the second, the system has been compromised by a rootkit hypervisor, which has attempted to install itself below the hypervisor to gain access to the platform. In this case, the Intel TXT-enabled, MLE-calculated hash system measurements differ from the expected value, owing to the insertion of the rootkit. Therefore, the measured environment will not match the expected value and, based on the launch policy, Intel TXT could abort the launch of the hypervisor or report an untrusted launch into the virtualization or cloud management infrastructure for subsequent use.

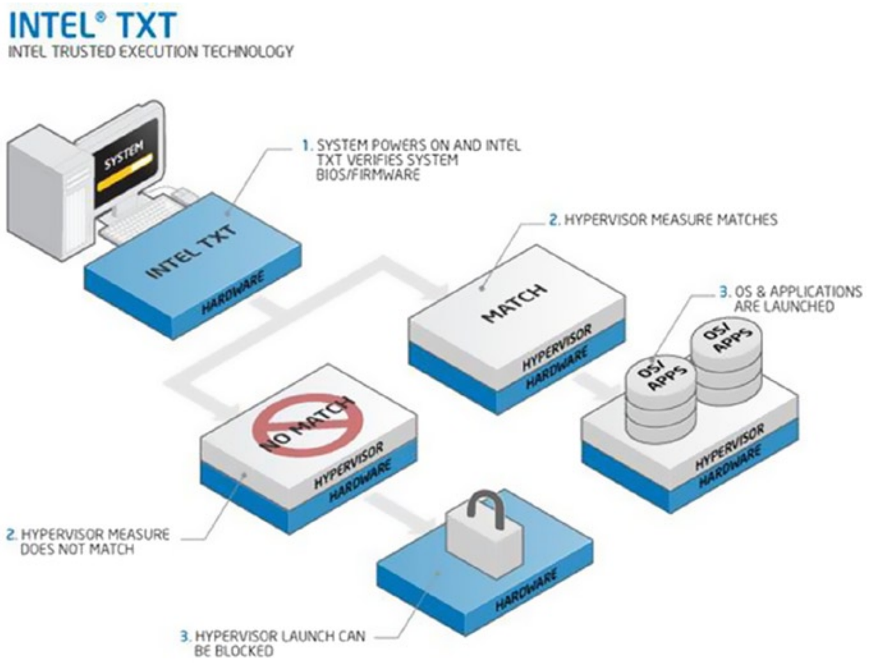

INTEL® TXT
INTEL TRUSTED EXECUTION TECHNOLOGY
SYSTEM
1. SYSTEM POWERS ON AND INTEL TXT VERIFIES SYSTEM BIOS/FIRMWARE
INTEL TXT
HARDWARE
2. HYPERVISOR MEASURE MATCHES
MATCH
HYPERVISOR
HARDWARE
3. OS & APPLICATIONS ARE LAUNCHED
OS/ APPS
OS/ APPS
HYPERVISOR
HARDWARE
NO MATCH
HYPERVISOR
HARDWARE
2. HYPERVISOR MEASURE DOES NOT MATCH
HARDWARE
3. HYPERVISOR LAUNCH CAN BE BLOCKED

Figure 3-15. *How Intel Trusted Execution Technology protects the launch environment*

Summary

In this chapter, we introduced the concept of platform boot integrity and trust. We covered the roots of trust in a trusted compute platform, and the two measured boot models, S-RTM and D-RTM. We introduced the concept of attestation as a critical requirement to assert the boot integrity, and presented the notion of trusted compute pools, including the use cases and the solution reference architecture for enabling trusted compute pools. By reviewing one solution stack and a reference implementation, we reinforced the concept and showed how to enable and use trusted compute pools. Platform trust is the new data center management attribute that can be used to orchestrate and manage the resources of virtualization and cloud data centers so as to meet the corresponding security challenges and requirements.

Looking ahead, Chapter 4 is a deep dive into attestation and view of a commercial implementation of a remote attestation software solution. In addition to platform trust and hardware roots of trust, more and more organizations and service providers are interested in providing visibility of and control to the physical location of the servers where the workloads and data are actually residing and executing. These controls are critical for federal agencies and regulated industries. Chapter 5 will introduce a new concept and control called hardware-assisted asset tag, which can be used to provide isolation, segregation, placement, and migration control of workload execution in multi-tenant cloud environments. Additionally, as a specialization of asset tags, geolocation/geotagging can be enabled to definitively provide visibility of the physical geolocation of the server, which can enable many controls that requirement hardware-based roots of trust to assert the location of the workloads and data. These attributes and the associated controls are dependent on the assertion of the boot integrity of the platform, and hence they become a great adjacency to trusted compute pools and boot integrity.